

Б.С. Гольдштейн, И.М. Ехриель, Р.Д. Перле

ИНТЕЛЛЕКТУАЛЬНЫЕ СЕТИ



МОСКВА «РАДИО И СВЯЗЬ»
2000

УДК 621.395.34

Г63

ББК 32.881

Гольдштейн Б.С., Ехриель И.М., Рерле Р.Д.

Интеллектуальные сети. — М.: Радио и связь, 2000. — 500 с.: ил.

Г63 **ISBN 5-256-01547-8**

Книга посвящена одной из самых интересных телекоммуникационных концепций – интеллектуальным сетям связи. Именно эта концепция оказалась сегодня на острие революционных изменений технологий и услуг связи.

Для инженеров и научных работников, занятых исследованием, разработкой и эксплуатацией телекоммуникационных систем. Книга будет полезна студентам и аспирантам соответствующих специальностей.

Еще одна задача книги – помочь руководителям операторских компаний выбрать стратегически верные решения при введении новых телекоммуникационных услуг.

Научно-техническое издание

ИБ № 2970

ISBN 5-256-01547-8

© Гольдштейн Б.С., Ехриель И.М., Рерле Р.Д., 2000

B.S. Goldstein, I.M. Ekhriel and R.D. Rerle

Intelligent Networks, Moscow, Radio i Sviaz, 2000.

The book is devoted to one of the most interesting telecommunications concepts – the Intelligent Networks. Just that very concept has found itself today on the edge of revolutionary changes in telecommunications technologies and services.

The book is intended for engineers and scientists involved in the research, development, and operation of telecommunications systems. The book will be a valuable reference source for university students and post-graduates studying in these areas.

One more purpose of this book is to assist the managers of operating companies in choosing the strategically right solutions when introducing new telecommunications services.

Scientific and technical edition

Copyright © B.S. Goldstein, I.M. Ekhriel, R.D. Rerle, 2000

Содержание

Предисловие	7
Введение	11
1 Интеллект применительно к услугам и сетям	11
2 Взаимосвязь новых технологий	12
3 Новые услуги, новые стандарты	14
Часть 1 Концепция	17
Глава 1.1 История создания	18
1.1.1 Старый мир и новые услуги	18
1.1.2 Новая архитектура	21
1.1.3 Услуги регионального уровня	23
1.1.4 От IN к AIN	25
1.1.5 Этапы развития AIN	28
1.1.6 Стандартизация назрела	30
Глава 1.2 Введение в IN	33
1.2.1 Эволюция предоставления услуг	33
1.2.2 Элементы сети	34
1.2.3 Модель обслуживания вызова	36
1.2.4 Действующие лица	38
1.2.5 Создание услуг IN	40
Глава 1.3 Концепция и ее модель	43
1.3.1 Архитектурная концепция IN	43
1.3.2 Концептуальная модель IN	44
1.3.3 Связи между плоскостями	48
1.3.4 Архитектура плоскости услуг	50
1.3.5 Архитектура глобальной функциональной плоскости	51
1.3.6 Архитектура распределенной функциональной плоскости	53
1.3.7 Архитектура физической плоскости	58
1.3.8 Сравнение методологий	61
1.3.9 Общие аспекты прикладного протокола INAP	62
Часть 2 Архитектура	71
Глава 2.1 Услуги и атрибуты	73
2.1.1 Услуги	73
2.1.2 Атрибуты услуг	77
2.1.3 Функциональные связи и интерфейсы	81

Глава 2.2 Глобальная функциональная плоскость	87
2.2.1 Блоки SIB, определенные для CS-1	87
2.2.2 Динамические и статические параметры блоков SIB	88
2.2.3 Стадия 1 описания блоков SIB	89
2.2.4 Стадия 1 описания блока BCP	111
2.2.5 Глобальная логика услуг GSL	114
Глава 2.3 Распределенная функциональная плоскость	117
2.3.1 Требования и ограничения	117
2.3.2 Распределенная функциональная модель для CS-1	119
2.3.3 Модель CCF/SSF	121
2.3.4 Модель SRF	136
2.3.5 Модель SCF	137
2.3.6 Модель SDF	140
2.3.7 Стадия 2 описания блоков SIB	141
2.3.8 Стадия 2 описания блока BCP	167
2.3.9 Дополнительные функции	173
2.3.10 Соответствие между POI/POR и DP/PIC	174
Глава 2.4 Физическая плоскость	177
2.4.1 Физические элементы в CS-1	177
2.4.2 Интерфейсы между физическими элементами и протоколы	178
Часть 3 Интерфейсы и протоколы	181
Глава 3.1 Интерфейсы IN	183
3.1.1 Процесс стандартизации INAP	183
3.1.2 Сценарии взаимодействия элементов IN	184
3.1.3 Архитектура и адресация INAP	188
3.1.4 Информационные потоки и операции	190
3.1.5 Абстрактный синтаксис протокола INAP для CS-1	193
Глава 3.2 Процедуры прикладного объекта SSF	203
3.2.1 Функциональная модель и интерфейсы SSF-AE	203
3.2.2 Связь между SSF-FSM и функциями CCF/функциями эксплуатационного управления	204
3.2.3 Машина конечных состояний объекта SSME-FSM	206
Глава 3.3 Процедуры прикладного объекта SCF	221
3.3.1 Функциональная модель и интерфейсы SCF-AE	221
3.3.2 Связь между SCF-FSM и программами SLP/функциями эксплуатационного управления	221
3.3.3 Диаграмма состояний SCME	223
3.3.4 Диаграмма состояний SCSM	224

Глава 3.4 Процедуры прикладного объекта SRF	241
3.4.1 Модель и интерфейсы прикладного объекта SRF-AE	241
3.4.2 Связь между SRF-FSM и функциями эксплуатационного управления/управления связью	242
3.4.3 Диаграмма состояний SRSМ	243
3.4.4 Примеры процедур управления SRF	248
Часть 4 От теории к практике	257
Глава 4.1 Применение концепции IN для спецификации услуг	259
4.1.1 Услуга «FREEPHONE» (Бесплатный вызов)	259
4.1.2 Услуга «Account Card Calling» (Вызов по телефонной карте)	272
4.1.3 Услуга «Premium Rate» (Информационная услуга за дополнительную плату)	285
4.1.4 Услуга «Televoting» (Телеголосование)	286
Глава 4.2 Рынок услуг и оборудования	289
4.2.1 Рыночные аспекты внедрения IN	289
4.2.2 Платформа IN компании Lucent Technologies	292
4.2.3 Платформа IN компании Alcatel	301
Глава 4.3 Аспекты внедрения	307
4.3.1 Варианты доступа к IN	307
4.3.2 Нумерация услуг IN	312
4.3.3 Использование сигнализации OKC-7	313
4.3.4 План действий	315
4.3.5 Добраться до интеллекта	317
4.3.6 Подходы и альтернативы	320
4.3.7 Не все так просто	328
4.3.8 Начисление платы за услуги и взаиморасчеты	331
Глава 4.4 Тестирование протокола INAP	337
4.4.1 Принципы и архитектура аттестационного тестирования ..	337
4.4.2 Архитектура и методы тестирования INAP	340
4.4.3 Особенности тестирования INAP-R	343
Часть 5 Перспективы	351
Глава 5.1 Направления эволюции концепции IN	353
5.1.1 Чего нет в CS-1?	353
5.1.2 IN и современные технологии	354
5.1.3 Гармонизация компьютерных и телекоммуникационных технологий	357
Глава 5.2 Набор возможностей CS-2	361
5.2.1 Нормативная база	361
5.1.2 Сравнительный анализ CS-1 и CS-2	361
5.2.3 Дополнения в INCM для CS-2	363

5.2.4 Особенности базового процесса обслуживания вызова для CS-2	376
5.2.5 Услуги CS-2	380
5.2.6 Аспекты предоставления международных услуг IN	384
Глава 5.3 Поддержка мобильности	387
5.3.1 Подходы к созданию и специфика беспроводных IN	387
5.3.2 Способы перехода к беспроводной IN	390
5.3.3 Стандарт WIN (ANSI TIA)	391
5.3.4 Стандарт CAMEL (ETSI)	394
5.3.5 Система IMT-2000 (FPLMTS)	397
Глава 5.4 IN и Internet	405
5.4.1 Совместить несовместимое	405
5.4.2 Интегрированные IN/Internet-услуги	407
5.4.3 Сигнализация поддержки услуг мультимедиа в сетях IP ...	410
5.4.4 Функциональная архитектура поддержки IP-сетей в IN	416
5.4.5 Услуга «CLICK TO DIAL»	418
5.4.6 Перспективы интегрированных IN/IP-платформ	420
Глава 5.5 Интеграция IN и B-ISDN	423
5.5.1 Особенности архитектуры	423
5.5.2 Модель состояний IN-SSM	427
5.5.3 Услуга «Видео-по-требованию»	431
Глава 5.6 Интеллектуальные сети и TINA	439
5.6.1 Что такое TINA?	439
5.6.2 Общие принципы CORBA	450
5.6.3 IN и TINA	453
5.6.4 Этапы миграции IN к TINA	456
5.6.5 Взаимодействие IN и TINA	458
5.6.6 Интеграция OKS-7 и CORBA	460
5.6.7 Перспективы применения TINA	463
Глава 5.7 Сравнительный анализ способов предоставления услуг	465
5.7.1 Сеть связи как большая система	465
5.7.2 Использование технологии IN для модернизации сети связи	468
Приложения	473
Приложение 1: Действующие рекомендации ITU-T по Интеллектуальной сети	475
Приложение 2: Расширенный код ASN.1 протокола ETSI INAP CS-1	477
Список сокращений	495
Список литературы	499

Предисловие

В начале было слово... Точнее – два слова: **Интеллектуальная сеть**. Трудно предсказать, как сложилась бы судьба этой технологии в России, если бы распространился иной, до сих пор попадающийся в технической литературе перевод английского термина Intelligent Network как «интеллигентная сеть», или какой-нибудь еще перевод, звучащий по-русски столь же мало привлекательно. Принятый же термин «интеллектуальная сеть» с блеском выдержал испытание по строгим критериям современного пиара и быстро завоевал популярность в самых широких кругах связистов. И это – несмотря на то, что речь идет вовсе не о *сети*, а о некоторой сетевой модели, об архитектуре создания и предоставления телекоммуникационных услуг. Да и не столько *интеллектуальной*, сколько программируемой с отделением логики организации услуг от функций коммутации.

К вопросу терминологии мы еще вернемся на страницах этой книги, а пока ограничимся определением понятия Интеллектуальной сети (IN) как некой надстройки над сетью связи, обеспечивающей применение интеллектуальных технологий для обработки запросов пользователями услуг связи из некоторого заранее определенного набора услуг Интеллектуальной сети. Сами эти наборы услуг (как и поддерживающие их функциональные средства и возможности) определяются рекомендациями, которые разрабатывает сектор стандартизации Международного союза электросвязи ITU-T, в частности, рекомендацией Q.1211, регламентирующей «набор CS-1», а также разработанными и разрабатываемыми в настоящее время рекомендациями, регламентирующими «наборы CS-2, CS-3 и CS-4».

Теперь, когда с названием книги стало более или менее ясно, следует, как это принято делать в предисловии, ответить еще на вопросы: *какая это книга, о чем она и для кого она?*

На первый вопрос окончательный ответ может дать только читатель. Но кое в чем следует сразу же признаться. Прежде всего, в том, что авторам не удалось выдержать весь материал книги на едином семантическом уровне: некоторые главы представляют собой довольно обстоятельные и подробные описания, а другие изложены более чем кратко. И не потому, что авторы не захотели или не сумели потрудиться одинаково усердно над всеми частями – просто сегодня далеко не всё одинаково ясно в бурно изменяющемся мире телекоммуникаций. Хорошо специфицированные и проверенные на практике элементы архитектуры рассмотрены в книге достаточно внимательно, а вот относительно возможных

вариантов развития интеллектуальных сетей существуют самые разные мнения, и авторы посчитали себя не в праве не упомянуть, пусть даже скороговоркой, о наиболее важных из них. Более глубоко изучить эти вопросы можно по публикациям, приведенным в списке литературы. Из соображений экономии места ссылки на источники в тексте практически отсутствуют, но обращение к ним может оказаться полезным читателю. Всем заинтересовавшимся этой тематикой рекомендуется также продолжить знакомство с новыми направлениями развития Интеллектуальных сетей по статьям в периодической печати и во Всемирной паутине, в том числе, и по адресу <http://www.loniis.ru>, куда также можно будет сообщить замечания и пожелания, касающиеся этой книги.

Теперь – о чем же здесь будет сказано, а что останется за пределами книги. Как уже отмечалось выше, концепция Интеллектуальной сети представляет собой способ быстро конфигурировать новые телекоммуникационные услуги в соответствии со специфическими для каждой из них требованиями, обеспечивающий одновременную и повсеместную доступность этих услуг абонентам сети связи. Достижению этого способствуют и возможности базовой телефонной сети общего пользования, которые объединяются под единым административным управлением для поддержки широкого спектра услуг. Именно описание концепции Интеллектуальной сети составляет содержание первой из пяти частей данной книги.

Часть 2 посвящена архитектуре IN в рамках набора CS-1 и содержит четыре главы, соответствующие четырем логическим плоскостям в концепции IN: физической плоскости, распределенной функциональной плоскости, глобальной функциональной плоскости и плоскости услуг. Естественным продолжением описания этих плоскостей архитектуры IN является рассмотрение протоколов и интерфейсов Интеллектуальной сети (в первую очередь, – протокола INAP), составившее содержание части 3. Искушенный читатель обратит внимание на то, как просто и кратко удалось изложить основы INAP и процедур прикладного объекта SSF после того, как стали ясными концепция и архитектура Интеллектуальной сети, и на то, с каким трудом решается эта же задача в тех книгах о телекоммуникационных протоколах, где INAP описывается среди других протоколов системы сигнализации ОКС7 без рассмотрения основ IN.

Вопросам практической реализации Интеллектуальных сетей посвящена часть 4. И если вопрос о содержании первой из глав этой части – описании четырех наиболее распространенных услуг IN – не вызывал сомнений, то со второй главой было не так просто. В конце концов, в этой главе было решено оставить описания только двух платформ Интеллектуальной сети: платформы той компании, которая изобрела эту сеть – Lucent Technologies, и той компа-

нии, которая имеет наибольшую емкость установленных в России потенциальных SSP – Alcatel. И это – при том, что в распоряжении авторов имелись материалы об отнюдь не менее интересных решениях: на базе оборудования DMS компании NORTEL, EWSD компании Siemens, AXE компании Ericsson и др. Но для целей книги двух примеров вполне достаточно, а что касается коммерческих советов, то если бы авторы могли позволить себе хоть намек на предпочтительность закупки того или иного оборудования, это был бы намек исключительно на отечественные разработки, ибо экономические законы не указывают никаких иных путей улучшить жизнь населения страны, то есть создать условия, в которых сможет существовать платежеспособный спрос этого населения на услуги Интеллектуальной сети.

Так или иначе, но авторам кажется, что им удалось сохранить беспристрастность вплоть до последней главы части 4, где выбор для рассмотрения отечественного протокол-тестера ОКС-7 был продиктован не экономическими, а сугубо техническими резонами.

Часть 5 посвящена перспективам развития Интеллектуальных сетей. Здесь предпринята попытка осветить все потенциальные возможности развития рассмотренной в книге архитектуры в эпоху конвергенции сетей и услуг. Основными направлениями развития IN, по мнению авторов, являются CS-2, IN и Internet, IN и B-ISDN, IN и TINA, рассмотренные в этой заключительной части книги.

Теперь попытаемся определить, *для кого* написана эта книга. Для всех, кто хочет узнать, что такое Интеллектуальная сеть, как она работает, как создаются новые телекоммуникационные услуги. В первую очередь – для Операторов сетей связи, т.е. для инженеров и менеджеров телекоммуникационных операторских компаний, а также для студентов и аспирантов соответствующих специальностей. На сегодняшнем уровне развития рынка услуг телефонной связи уже очевидно, что для успешного экономического роста операторской компании отнюдь недостаточно предоставлять лишь базовую услугу телефонной связи. Всеобщая тенденция проявляется во впечатляющем темпе роста доходов от предоставления дополнительных услуг связи в сравнении с ростом доходов от традиционных телефонных услуг.

В России возможные доходы от предоставления традиционных телефонных услуг в значительной степени ограничены решениями органов местной власти. В существующих экономических условиях Операторы вынуждены более активно искать новые источники доходов, что теснейшим образом связано с реализацией качественно новой концепции предоставления телекоммуникационных услуг – концепции Интеллектуальной сети.

Книга адресуется также сотрудникам телекоммуникационных компаний, институтов и предприятий, в которых разрабатывается и производится техника связи и которым небезразличны вышеназванные заботы Операторов. Кроме того, менее утилитарный, но стратегически наиболее значимый и существенно расширяющий потенциальную аудиторию ответ на вопрос *для кого* обусловлен еще не до конца осознанной ролью, которую играет Интеллектуальная сеть в условиях конвергенции сетей и услуг связи, являясь неким посредником, «мостиком» между традиционными телефонными сетями с POTS (Plain Old Telephone Services) и перспективными мультисервисными сетями.

И, наконец, *без кого* не появилась бы эта книга. Трудно переоценить помощь коллег и сотрудников авторов по НИО-1 института, как участвовавших в обсуждении самой идеи книги и ее отдельных глав, так и выполнявших конкретные научно-технические разработки по данной тематике. Особо хотелось бы поблагодарить В.А. Соколова, существенно повлиявшего на перевод основных положений книги с бюрократического языка стандартов на технический язык, а затем – с технического языка на русский. Те места, где это удалось не в полной мере, связаны исключительно с упрямством авторов, принимающих на себя также полную ответственность за все недочеты и спорные моменты, которые читатель сможет встретить в этой книге, если, конечно, прочитает ее.

Введение

1 Интеллект применительно к услугам и сетям

Трудно найти другой такой термин как «услуга». Спросите десять специалистов в области телекоммуникаций, что такое «услуга» и вы получите одиннадцать разных ответов. Однако один общий тезис будет отчетливо прослеживаться в их ответах: коммерческий успех и даже просто само существование современных операторов и поставщиков оборудования сетей связи напрямую зависит от того насколько успешно они способны предлагать и внедрять эти самые «услуги», вне зависимости от их сущности, будь то перенос битового потока по выделенной линии или продажа одежды через Internet.

Если же речь заходит об услугах Интеллектуальной сети, то следует ожидать еще больше вариантов ответов, чему дополнительно способствует неоднозначность термина «интеллектуальность». Авторы книги предполагают наличие у читателя сформировавшегося мнения о том, что такое услуги связи, и акцентируют внимание лишь на том, что такое услуги Интеллектуальной сети и что такое Интеллектуальные сети.

В процессе своего развития в направлении предоставления услуг связи коммутационные системы с программным управлением превратились в большие компьютерные системы с возможностью программирования их работы. Программное управление позволило реализовать в коммутационных системах такие услуги как переадресация, ограничение потока вызовов, телефонные карты, что может рассматриваться как введение в них своего рода «интеллектуальности».

В той или иной форме «интеллектуальность» существовала в сетях связи в течении всей истории телекоммуникаций. В ранний период развития телефонии «интеллектуальность» была, большей частью, возложена на человека (оператора, телефониста), способного, если надо, применить свой интеллект при обслуживании вызова.

Стремительное распространение программно управляемых коммутационных узлов и возможность объединения их в сети с помощью общеканальной сигнализации ОКС-7 позволило отделить от рутинных функций управления соединением те функции, которые непосредственно связаны с логикой услуг, реализовать их в отдельном оборудовании и обеспечить удаленный доступ к нему с целью совместного использования коммутационными узлами всей сети связи. Именно последнее обстоятельство и дало основание называть сети, построенные по такому принципу, «интеллектуальными сетями», подразумевая под этим то, что «интеллектуальность» в сетях обычной

архитектуры, сконцентрированная в коммутационных узлах и доступная только их абонентам, в Интеллектуальных сетях (IN – Intelligent Network) становится общедоступной.

Строго говоря, нельзя сказать, что термин «интеллектуальная сеть» удачен и точно отражает суть. Правильнее было бы назвать такие сети «программируемыми», так как речь идет о дополнительном использовании существующих в них компьютерных ресурсов. Важно также понимать, что Интеллектуальная сеть не имеет ничего общего с известным термином «искусственный интеллект».

Несмотря на постоянно сопровождающие его применение дискуссии, название IN принято во всем мире (Advanced IN – AIN в США) и используется специалистами в области связи при упоминании о стандартизированной в мировом масштабе современной концепции распределения функций создания и предоставления услуг и специализированной компьютерной сети, организованной по сформулированным в этой концепции правилам.

Введенная международным союзом электросвязи ITU-T (International Telecommunication Union – Telecommunication standardization sector) концепция IN, определяет архитектуру аппаратных и программных средств, предусматривающую обмен вызовами специальных процедур между коммутационной системой и сетью во время организации связи. Исполнение этих процедур может управлять процессами коммутации и иными сетевыми ресурсами с целью выполнения функций «интеллектуальной» маршрутизации, тарификации, взаимодействия с пользователем. Первые стандарты IN, известные как CS-1, позволяют реализовать достаточно ограниченные возможности внешнего управления коммутационными ресурсами. Выпущенные в развитие концепции стандарты CS-2, а также находящиеся в стадии подготовки в ITU-T стандарты по CS-3 и CS-4 должны обеспечить гораздо большие возможности.

Заложенный в концепции потенциал еще только начинает раскрываться, следуя стремительному развитию информационных технологий. Однако уже сейчас ясно, что принципы архитектуры IN будут играть ключевую роль в интеграции стационарных сетей и сетей подвижной связи, телефонных сетей и сети Internet, и применимы к широкополосным сетям АТМ.

2 Взаимосвязь новых технологий

Конкуренция на рынке услуг связи в России сегодня уже не новость. Все чаще абоненты и поставщики услуг имеют возможность выбора из имеющихся операторов того, который использует наиболее современные сетевые технологии и способы предоставления услуг. Операторы начинают уделять особое внимание изучению спро-

са потребителей на услуги связи и поиску оптимального технического решения, обеспечивающего их введение.

Длительность проходившей на страницах российской печати дискуссии о том, что означает термин Intelligent Network (IN) и как лучше перевести его с английского языка на русский, не повлияла на быстроту, с которой специалисты в области связи (да и далекие от нее) осознали, что IN – это всего лишь архитектурная концепция организации сетей связи, наиболее предпочтительная с точки зрения введения в эти сети новых услуг.

В конце 90-х годов, когда IN уже получила русское название «интеллектуальная сеть», дискуссия перешла в практическую плоскость. Это было связано с тем, что необходимость разработать конкретные проекты IN для сетей разных операторов выявила немало проблем, требующих системно-технической проработки. Авторы книги попытались проанализировать некоторые из таких проблем и наметить пути их решения, сопроводив все это подробным описанием концепции.

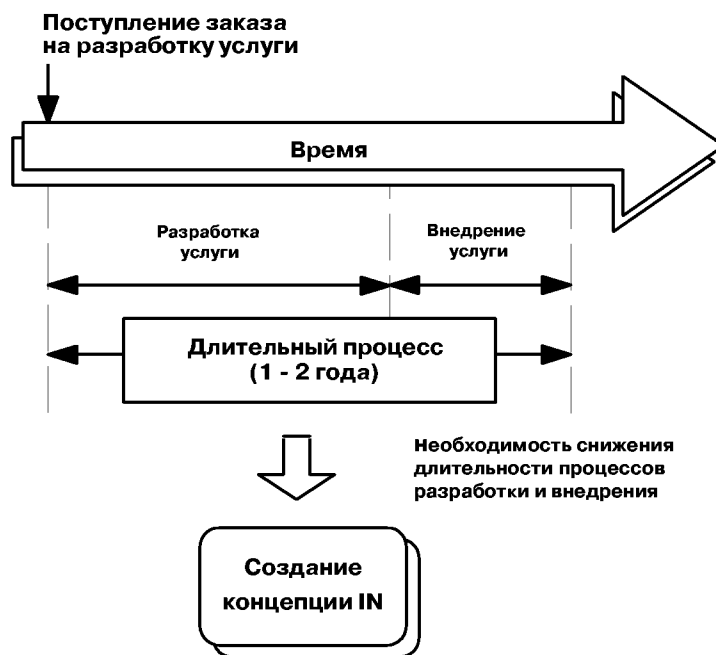


Рис. 1 Основная причина создания концепции IN

К идеям Intelligent Network человечество пришло далеко не сразу. Лишь в конце 80-х годов мировая индустрия телекоммуникации стала свидетелем введения нескольких новых технологий: узкополосной ISDN, общеканальной сигнализации ОКС-7 и Интеллектуальных сетей IN. Каждая технология была нацелена на решение определен-

ной группы задач, связанных с развитием сетей связи. Однако заложенные в этих технологиях фундаментальные свойства, такие как спецификации протоколов в соответствии с семиуровневой моделью взаимодействия открытых систем (Open System Interconnection, OSI), позволили взаимно дополнять и обогащать каждую из них.

ISDN позволила довести до конечного пользователя два цифровых канала 64 Кбит/с, обеспечив возможность одновременной передачи речи и данных по обычной абонентской линии телефонной сети. Система ОКС–7 обеспечила общеканальную сигнализацию для управления телефонными соединениями и для предоставления услуг, не связанных с этими соединениями. Концепция IN создала новую архитектуру предоставления услуг.

Новая архитектура, основная идея которой состоит в том, чтобы отделить процессы традиционной коммутации от процедур предоставления новых услуг, появилась после того, как стало очевидно, что в условиях жесткой конкурентной борьбы оператор сети связи должен уметь предоставлять услуги, ориентированные на группы пользователей с сильно различающимися потребностями, и иметь возможность быстро создавать и развертывать эти новые услуги (см. рис. 1).

3 Новые услуги, новые стандарты

С появлением дополнительных услуг рынок телекоммуникаций подвергся серьезным изменениям. Если выделить в нем сектор, связанный с традиционными услугами предоставления соединений, и сектор, связанный с дополнительными услугами, то можно видеть, что первый испытывает сильнейшее ценовое давление со стороны второго. На рисунке 2 показана тенденция изменения удельного веса различных дополнительных услуг в доходах телекоммуникационных компаний.

На первых порах у каждого производителя был свой перечень дополнительных услуг, который определялся возможностями выпускаемых им коммутационных станций, а услуги из такого перечня могли быть предоставлены только абонентам, линии которых включены в АТС этого производителя.

Попытка стандартизировать перечень услуг и распространить возможность их предоставления на всю сеть связи была предпринята при создании концепции ISDN. Извне ISDN представляется как одна коммутационная станция, оказывающая своим абонентам ряд стандартизованных основных и дополнительных услуг. Это оказалось возможным благодаря введению на абонентском участке системы сигнализации DSS1 (Digital Subscriber Signalling #1), а на междолинном участке – системы сигнализации ОКС–7 с подсистемой ISUP (ISDN User Part). Однако основная проблема, связанная с необходи-

мостью замены программного обеспечения на всех станциях сети при введении любой новой услуги, в ISDN осталась нерешенной. Замена обходится недешево и требует от оператора умения предвидеть, какого рода услуги будут иметь коммерческий успех. В противном случае не только окажутся «выброшенными на ветер» деньги, потраченные на введение новой услуги, но и будет упущена та часть потенциальных клиентов, которые перейдут к оператору, угадавшему, какую новую услугу целесообразно ввести.

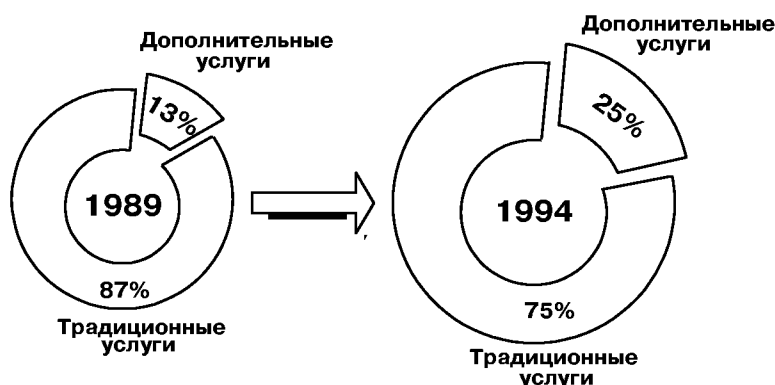


Рис. 2 Удельный вес дополнительных услуг

Концепция IN предусматривает подход к функциональному распределению процедур поддержки услуг, специфицированный в виде международных стандартов и позволяющий операторам сетей связи оперативно развертывать новые услуги, максимально эффективно используя существующую инфраструктуру своих сетей. Разработка стандартов была мотивирована интересом производителей оборудования к унификации возможностей быстрой и экономически эффективной реализации услуг. Сильное влияние оказало и давление со стороны операторов, которые столкнулись с большими трудностями, обусловленными наличием в их сетях коммутационного оборудования разных производителей, каждый из которых предлагал (как в отношении перечня услуг, так и в отношении путей их реализации) свои решения, несовместимые с решениями других производителей. На рисунке 3 приведены основные предпосылки создания концепции IN.

Следует отметить, что во многих странах с развитой телекоммуникационной инфраструктурой практика предоставления дополнительных услуг в пределах всей сети общего пользования существовала задолго до введения услуг IN. Для таких стран переход к Интеллектуальной сети оказался лишь переходом к другому, более эффективному, способу предоставления услуг, характеризующемуся значительной степенью стандартизации. В некоторых из этих стран и до сих пор примерно одинаковые (с точки зрения абонента) услуги предоставляются одновременно разными способами – традиционным и посредством IN.

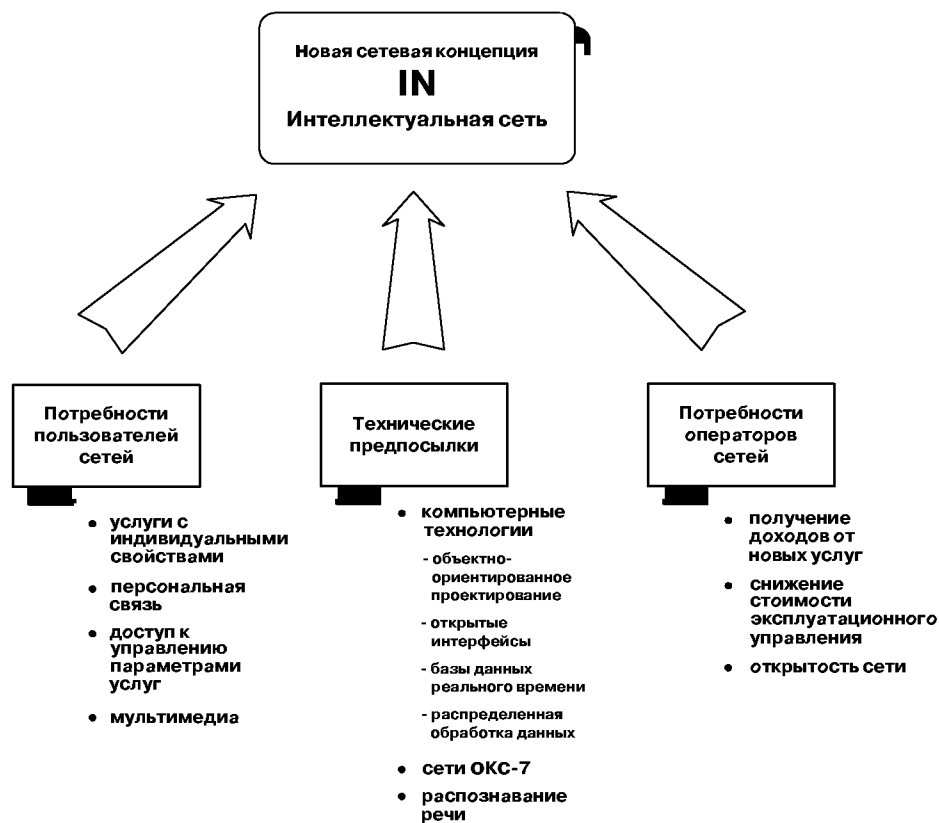


Рис. 3 Предпосылки создания концепции IN

В отличие от традиционного подхода, архитектурная концепция IN предполагает четкое разделение всех функций создания, модификации и предоставления услуг, а также эксплуатационного управления ими, на небольшое количество программных модулей, взаимодействие между которыми обеспечивают стандартные интерфейсы, а перечень функций каждого из которых строго определен.

Коммутационные станции, оснащенные необходимыми функциональными модулями, и обособленные специализированные программно-аппаратные комплексы с другими функциональными модулями, оказывающие таким станциям содействие в предоставлении новых услуг, называются узлами IN. Стандартные интерфейсы между узлами IN поддерживаются системой сигнализации ОКС-7 с использованием INAP – прикладного протокола Интеллектуальной сети.

Часть 1

Концепция

Глава 1.1

История создания

1.1.1 Старый мир и новые услуги

Долгое время телефонные компании, владевшие сетями общего пользования, предоставляли своим клиентам лишь традиционный набор услуг, обеспечивающих установление соединений. К середине XX-го века в развитых странах спрос на такие услуги был полностью удовлетворен. Операторы столкнулись с тем, что темпы роста доходов от предоставления традиционных услуг значительно снизились. К тому же, обычные телефонные соединения двух абонентов больше не могли удовлетворить потребности клиентов делового сектора, и уже со второй половины 60-х годов операторы сетей связи стали предлагать дополнительные услуги обычным абонентам и услуги, которые позднее назовут услугами Centrex, – корпоративным клиентам.

То, что мы сегодня обозначаем аббревиатурой IN, берет свое начало с идей инженеров Bell Labs, предложивших в середине 70-х годов ряд способов предоставления дополнительных услуг в сети общего пользования. Эти идеи содержали архитектурные решения, предусматривавшие нестандартное использование тех ограниченных возможностей, которыми обладали тогда коммутационные станции.

К началу внедрения этих решений в ряде местных телефонных сетей США уже появились первые коммутационные станции с управлением по записанной программе. Эта система управления позволила предоставить абонентам АТС ряд новых услуг, таких как «вызов на ожидании» (абонент, ведущий разговор, получал уведомление о новом входящем вызове) и «переадресация вызова». Такого рода услуги получили название дополнительных.

Междугородная сеть Северной Америки к концу 60-х годов была построена на коммутационном оборудовании No.4ACrossbar координатной системы, позволявшем производить несложные действия с номерной информацией (пересчет, удаление и добавление цифр номера блоками по 3 или по 6 цифр) для поддержки процедур маршрутизации вызовов в сети. Те же возможности были функционально повторены и несколько расширены при создании электронных междугородных АТС системы 4ESS, начало эксплуатации которых относится к 1976 году.

Это оборудование позволило компании AT&T, которая была также и монопольным оператором телефонной сети США, в 70-х годах реализовать ряд дополнительных услуг, доступных абоненту любой АТС сети общего пользования. В отличие от обычных «внутристанционных» дополнительных услуг, они получили название «сетевых услуг» (network services). Одна из таких услуг, известная как услуга INWATS (Inward wide area telecommunications service), предоставлялась с 1967 года и была ориентирована на клиентов делового сектора. Услуга позволяла абоненту любой местной АТС осуществлять бесплатные междугородные вызовы телефонных номеров крупных компаний. Компании заключали с оператором телефонной сети соглашение, которое обязывало их оплачивать (нередко со значительной скидкой) определенное количество входящих междугородных вызовов, поступающих к их сотрудникам. Таким линиям присваивались 10-значные номера, начинающиеся с цифр 800 (по этой причине дальнейшие модификации этой услуги стали называть «услугой 800»). В североамериканском плане нумерации цифры 800 не являлись кодом междугородной зоны, и потому было предложено использовать их в качестве префикса для «бесплатных» междугородных вызовов.

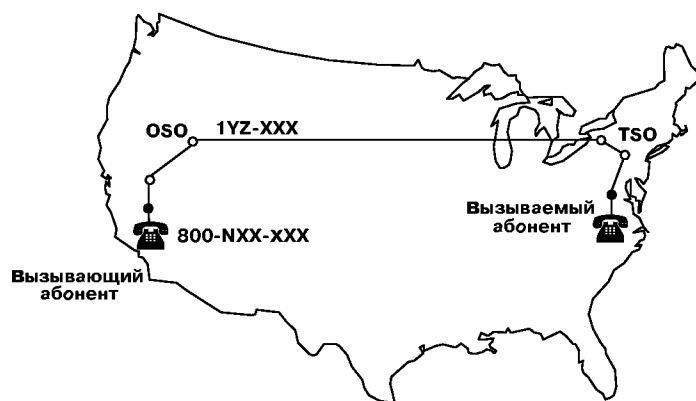


Рис. 1.1.1 Схема предоставления услуги INWATS

Популярность INWATS стремительно росла в конце 60-х и начале 70-х годов. Однако реализация этой услуги в телефонной сети, построенной на координатных АТС No.4ACrossbar, возможности которых, как уже говорилось, были ограниченными, требовала введения в сеть дополнительных средств маршрутизации вызовов. Для этой цели были созданы специальные сетевые узлы, названныеOSO (Originating screening office) иTSO (Terminating screening office), которые проверяли правомочность вызовов INWATS и обеспечивали маршрутизацию их в направлении зоны, где установлена АТС с абонентскими линиями компании-клиента. Схема предоставления услуги INWATS приведена на рисунке 1.1.1.

1.1.2 Новая архитектура

Очень скоро ключевая роль в обеспечении маршрутизации между ОСО и ТСО была возложена на принципиально новую технологию сигнализации – систему межстанционной сигнализации по общему каналу CCIS (Common channel interoffice signalling). Система CCIS (прообраз системы ОКС–7) была введена в 1976 году с целью повысить эффективность использования ресурсов телефонной сети и предназначалась первоначально для скоростной передачи цифровым способом адресных сигналов и информации линейной сигнализации.

Сеть CCIS, с самого начала представлявшая собой высоконадежную сеть пакетной коммутации и соединявшая компьютеры управляющих устройств цифровых АТС, рассматривалась как основа для организации перспективных услуг телефонной сети общего пользования. Быстро доказав свои преимущества, общеканальная система сигнализации была введена не только на электронных АТС системы 4ESS, но и на некоторых координатных АТС No.4ACrossbar, система управления которых имела специализированный электронный блок пересчета номерной информации.

В 1979 году была предложена идея ввести в CCIS новый сетевой элемент, получивший название сетевой базы данных. При такой архитектуре, используя соответствующую адресацию, любая АТС, оборудованная средствами общеканальной сигнализации, могла обратиться через сеть CCIS к этой базе данных и получить от нее необходимую информацию.

Первоначально эта идея рассматривалась как средство более эффективной реализации услуги INWATS. Вместо рутинной маршрутизации вызова через сеть связи с заменой адресной информации в каждом транзитном узле, ближайший исходящий узел ОСО приостанавливал обработку запроса INWATS и передавал набранный абонентом номер, содержащий этот запрос, через сеть CCIS к базе данных. Компьютер базы данных производил пересчет 10-значного номера INWATS в обычный «физический» номер, который через сеть CCIS возвращался к ОСО, и тот производил маршрутизацию обычным образом в соответствии с полученным физическим номером. Это было не только намного эффективнее ранее использовавшегося решения, но и позволяло осуществить другую, до того невиданную вещь: предоставить абонентам услуги INWATS, единый «логический» номер для доступа к нескольким филиалам своей компании, находящимся в разных регионах страны.

Сетевая база данных логических номеров рассматривались как централизованный ресурс, хотя фактически, из-за ограничений, накладываемых производительностью компьютеров, она представляла собой распределенный между несколькими компьютерами общий ресурс, строго администрируемый оператором сети.

Архитектурные решения, предложенные для реализации услуги CCIS INWATS (т.е. INWATS на основе сети CCIS), впервые были использованы при автоматизации обслуживания вызовов по телефонным кредитным картам (Calling card). Это произошло в 1980 году, начало же предоставления CCIS INWATS было положено в 1981 году. В апреле 1982 года Федеральная комиссия США по связи утвердила тарифы на услугу Expanded 800 Service («расширенная услуга 800»), предоставляющую ряд новых возможностей, что удалось сделать благодаря оригинальной сетевой архитектуре.

Еще во время работы над новой архитектурой стало ясно, что можно получить и более впечатляющие результаты, если использовать ее в качестве многоцелевой платформы, предназначенной для создания широкого спектра сетевых услуг. Это послужило толчком к началу работ, направленных на то, чтобы специфицировать набор элементарных операций, которые должны выполнять, взаимодействуя друг с другом, оконечная АТС и сетевая база данных. Комбинируя последовательность выполнения единожды определенных элементарных операций, оказалось возможным реализовывать и гибко модифицировать новые услуги с совершенно разными свойствами.

Кульминацией развития описываемого подхода стала архитектура DSDC (Direct services dialling capabilities). Основными элементами архитектуры были сетевая база данных, получившая название NCP (Network control point), коммутационные станции, оснащенные функциями доступа к базе данных и названные ACP (Action point), и ряд элементарных команд (с соответствующими параметрами), независимых от услуг и используемых при взаимодействии между NCP и ACP.

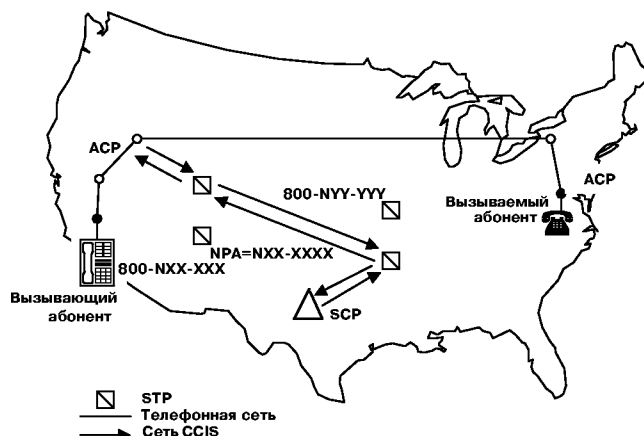


Рис. 1.1.2 Схема предоставления "продвинутой услуги 800"

В процессе создания архитектуры DSDC специалистами Bell Labs был предложен еще один важный сетевой компонент – интеллектуальная периферия IP (Intelligent peripheral), первая реализация кото-

рой носила название NSCX. IP обеспечила автоматизацию диалога вызывающего абонента с сетью при установлении соединения, передавая ему заранее записанные речевые уведомления и принимая его ответы, передаваемые с телефонного аппарата многочастотным способом (DTMF).

Первой услугой, реализованной в соответствии с архитектурой DSDC, стала Advanced 800 Service («продвинутая услуга 800»), послужившая прототипом самой известной из услуг IN – «бесплатный вызов» (FRH – FreePhone). Схема предоставления услуги приведена на рисунке 1.1.2. Услуга предусматривала такие возможности, как распределение вызовов по нескольким направлениям в соответствии с заданной клиентом пропорцией и изменение маршрутизации по заказу клиента. Услуга SDN (Software defined network) стала основой для спецификации другой ключевой услуги IN – VPN («виртуальная частная сеть»).

Архитектура DSDC постоянно развивается и до сих пор служит платформой для многих популярных услуг, предоставляемых компанией AT&T корпоративным и частным абонентам на базе телефонной сети Северной Америки. За это время количество NCP и IP разных типов вышло далеко за пределы первоначальных прогнозов, и в типичный рабочий день в 1995 году почти половина из 200 млн. вызовов, обслуженных сетью компании, использовала ресурсы архитектуры DSDC. Более того, уже после демонополизации междугородной телефонной сети США, новые операторы, такие как MCI и SPRINT, создали свои варианты физической реализации этой архитектуры и стали предоставлять своим клиентам функционально аналогичные услуги.

1.1.3 Услуги регионального уровня

В 1984 году решением суда о демонополизации компания Bell System была разделена на несколько частей. Это существенно повлияло на дальнейшее развитие концепции DSDC. Во-первых, требование о разделении предоставления услуг междугородной и местной связи, дало толчок к распространению общеканальной сигнализации CCIS и архитектуры DSDC на уровень местных сетей. Во-вторых, группа системных инженеров, стоявших у истоков создания архитектуры DSDC, также разделилась: часть осталась в Bell Labs, другие вошли в новую лабораторию Bell Core, созданную, чтобы выполнять научно-исследовательские работы для региональных операторов (RBOC), получивших после разделения Bell System полную независимость.

Такая ситуация вызвала сильнейшую конкурентную борьбу за право владения ресурсами сетевых баз данных, в результате чего ре-

гиональным компаниям, все же, удалось разделить патентные права на архитектуру DSDC и на соответствующие технологии, разработанные Bell System. Однако RBOC были вынуждены развертывать собственные базы данных для предоставления сетевых услуг на местном уровне.

Вследствие этого лаборатория Bell Core направила свои усилия на создание требований к предоставлению сетевых услуг местного уровня в рамках архитектуры DSDC. К этому же времени при упоминании о реализованных таким образом услугах начал широко использоваться термин Intelligent Network (IN).

Стремление региональных телефонных компаний не зависеть от поставщиков оборудования побудило их предъявить к концепции ряд новых требований, собранных воедино в лабораториях Bell Core и оказавших влияние на разработку стандартов для IN. Одним из этого ряда было требование обеспечить совместную работу компонентов архитектуры IN, поставляемых разными производителями. Компания Nortel (ранее Northern Telecom) поставила региональным операторам значительное количество коммутационных станций для их сетей еще до разделения Bell System; впоследствии в этих сетях появились АТС и других производителей. Новые стандарты должны были не только гарантировать совместную работу существующего разнородного оборудования, но и содействовать привлечению новых поставщиков, способных поддержать архитектуру IN.

Немаловажное значение имела и та часть решения суда о разделении, которая запрещала региональным компаниям и их лабораториям производить собственное оборудование, включая АТС. Инженеры Bell Core, работавшие над созданием требований к услугам IN, имели опыт совместной с инженерами компании Western Electric работы по переводу спецификаций услуг в технические требования к коммутационному оборудованию. Однако они предвидели большие трудности в том, чтобы распространить хорошо отлаженный процесс взаимодействия на отношения с другими производителями.

Оказалось, напротив, что концепция IN способна значительно упростить процесс взаимодействия благодаря переносу большей части функций, обеспечивающих предоставление сетевых услуг, из оконечной АТС в базу данных, которая в терминах IN получила название SCP. Теперь процесс предоставления и модификации услуг выполнялся программными средствами SCP (Service control point), разработку которых Bell Core смогла оставить за собой, поскольку это не противоречило решению суда о разделении.

В середине 1980-х годов Bell Core разработала спецификации для продвинутой версии IN, которая была названа IN/2 в отличие от ранних спецификаций IN/1. Продвинутый вариант предлагал много новых возможностей, таких как приостановка обслуживания вызова

в середине процесса установления соединения и манипуляции конфигурацией соединения. Однако эти требования остались нереализованными, так как производители оборудования сочли их невыполнимыми в полном объеме.

В то же время, некоторые региональные операторы разработали, в рамках архитектуры IN, собственные коммерческие и технические решения и реализовали их в своих сетях. Впоследствии эти решения также были учтены Bell Core при работе над спецификацией услуг. Основным результатом этой фазы развития IN Bell Core стали спецификации Advanced IN (AIN) Release 1.0. Но, к сожалению, их постигла та же участь: соглашение между производителями о создании такого оборудования не было достигнуто. Тогда Bell Core было предложено вводить AIN менее крупными шагами, спецификации которых были названы AIN 0.0, AIN 0.1, AIN 0.2 и т.д. Этот подход сохранился до настоящего времени.

Таким образом, концепция, которая первоначально создавалась для гибкой маршрутизации междугородных вызовов и начисления платы за них, была значительно расширена услугами, ориентированными на потребности местных сетей (с использованием потенциала дополнительных услуг, имеющегося в оконечных АТС), и требованиями обеспечения функционирования в среде разных производителей.

1.1.4 От IN к AIN

Как уже было сказано, в середине 80-х годов лаборатория Bell Core разработала первый вариант концепции Интеллектуальной сети, получивший название Intelligent Network 1 (IN/1), суть которой отражена на рисунке 1.1.3. Именно в концепции IN/1 логика, обеспечивающая предоставление услуг IN, впервые была перенесена из коммутационных станций во внешние базы данных, названные SCP.

По технологии IN/1 были развернуты две услуги – «услуга 800» и «услуга телефонных кредитных карт». Архитектура IN/1 еще не была независимой от услуг, поэтому для той и для другой услуги требовались разные SCP. Для взаимодействия с базами данных было разработано и введено в коммутационные станции дополнительное программное обеспечение. Новые функции позволили станциям распознавать ситуации (точки приостановки обслуживания вызова), в которых нужно было взаимодействовать с SCP. Взаимодействие осуществлялось через сеть ОКС. Коммутационные станции, способные приостанавливать обслуживание вызова и обмениваться данными с SCP, получили название SSP (Service switching points).

Для поддержки процессов создания, тестирования и развертывания услуг с введением новой архитектуры потребовались новые средства эксплуатационного управления. Так как IN/1 не была неза-

висимой от услуг, для каждой из них требовалась своя система. Это приводило к тому, что точки приостановки обслуживания вызова в программном обеспечении коммутационных систем были специфическими для каждой услуги. Для реализации «услуги 800» требовались не только свои точки приостановки обслуживания, но и отдельные SCP, а также соответствующие системы загрузки в SCP данных, необходимых именно для «услуги 800».

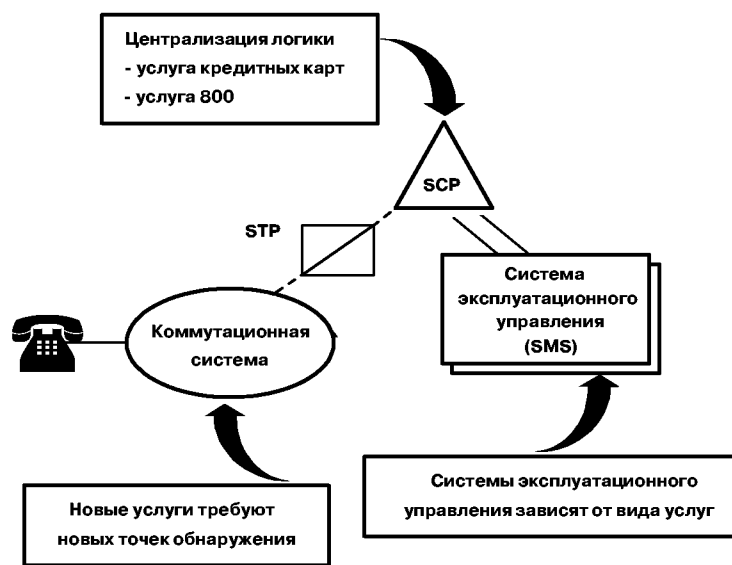


Рис. 1.1.3 Архитектура IN/1

В таком ориентированном на конкретную услугу окружении невозможно было использовать имеющиеся программно-аппаратные средства, поддерживающие одну услугу, для поддержки другой услуги. Таким образом, в архитектуре IN/1 логика, обеспечивающая предоставление услуг, уже была отделена от коммутационных систем, однако весь набор средств, нужных для реализации той или иной услуги, все еще оставался зависимым от того, какая это услуга.

Структура, приведенная на рисунке 1.1.4, выглядит очень похожей на предыдущую, однако имеет одно существенное отличие. Система эксплуатационного управления теперь уже независима от услуг. Точки приостановки обслуживания и платформа поддержки логики в SCP могут в этой архитектуре использоваться совместно разными услугами. Такая независимая от услуг архитектура была названа AIN – Advanced IN (прямой перевод – продвинутая Интеллектуальная сеть – звучит не очень красиво, но отражает суть).

Первая версия AIN, архитектура которой приведена на рисунке 1.1.5, получила естественное название AIN/1. Кроме традиционных (и уже знакомых нам по IN/1) элементов появился один новый –

вспомогательный пункт управления AD (Adjunct). Он выполняет функции, аналогичные функциям SCP, но подключен к SSP не через сеть ОКС, а непосредственно, через высокоскоростной интерфейс. Структура сообщений прикладного уровня, используемых при взаимодействии SSP с AD, идентична структуре сообщений, используемых при связи с SCP.

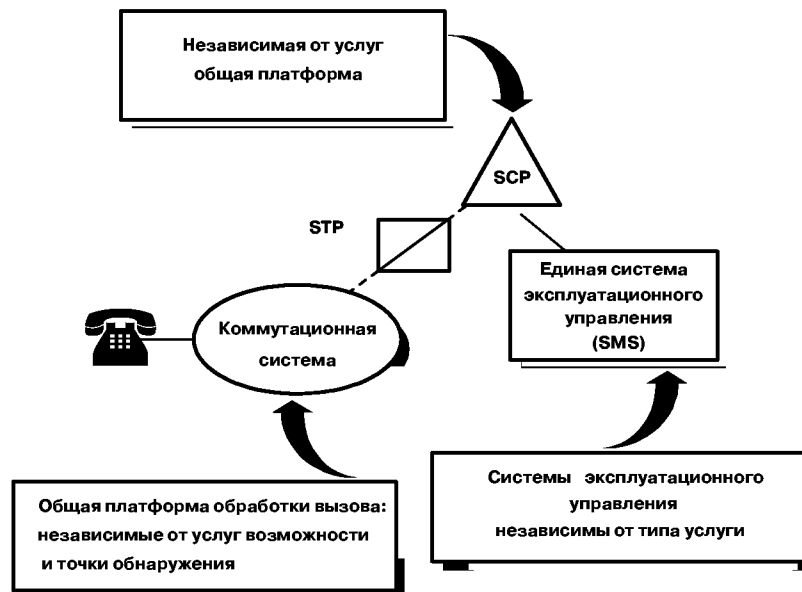


Рис. 1.1.4 Архитектура AIN

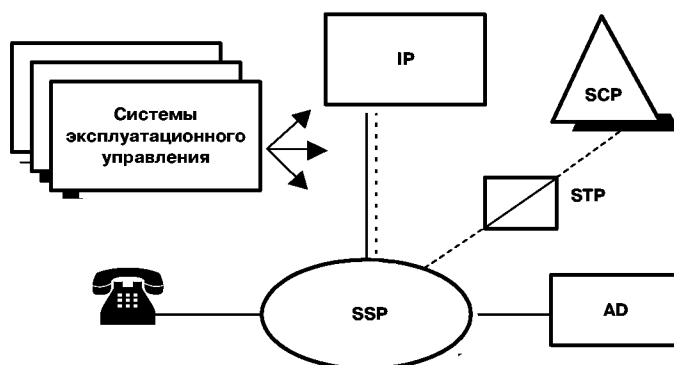


Рис. 1.1.5 Архитектура AIN/1

Архитектура AIN/1 предусматривает, наряду с реализуемыми сегодня, ряд таких требований, реализовать которые пока не позволяют технологические возможности производителей оборудования.

Однако потребность операторов в услугах Интеллектуальной сети оказалась настолько велика, что они выступили инициаторами немедленного начала поэтапной реализации возможностей, декларированных в AIN.

В качестве стартовой позиции производители оборудования связи, работающие на рынке Северной Америки, согласовали характеристики первого этапа реализации, получившего название AIN/0.1, хотя и у этого этапа был предшественник – AIN/0.0

1.1.5 Этапы развития AIN

1.1.5.1 Архитектура AIN/0.0

В спецификации AIN/0.0 определены три состояния процесса обслуживания и, соответственно, три точки приостановки обслуживания вызова, называемые также точками обнаружения обращений к услугам AIN или *триггерными точками*. В каждой точке можно задать от одного до трех критериев для приостановки обслуживания. Например, для триггерной точки «трубка снята» предусмотрено два критерия – немедленная реакция и реакция после приема определенного количества цифр. Приостановка возможна также в точках «прием и анализ цифр» и «маршрутизация».

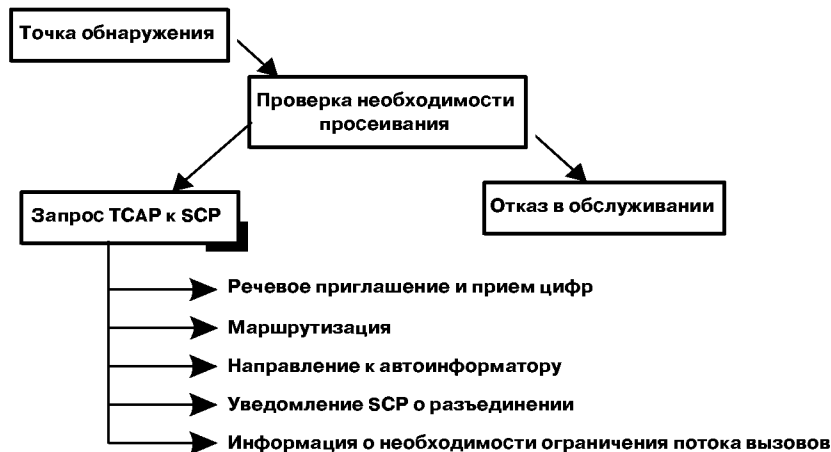


Рис. 1.1.6 Алгоритм обслуживания вызова для AIN/0.0

Перед тем как сделать запрос SCP, осуществляется проверка, не перегружены ли необходимые для его обработки ресурсы. Процедура просеивания вызовов позволяет SCP уведомить SSP о запрете передачи запросов с определенными номерами. Во время действия запрета заданная часть или все абоненты, набирающие такие номера, получают отказ, а остальные вызовы обрабатываются в соответствии с нормальным алгоритмом (см. рис. 1.1.6).

В спецификациях AIN/0.0 предусмотрено до 75 речевых уведомлений. Эти спецификации базируются на ANSI-варианте подсистемы TCAP (Transaction capabilities application part) версии 1 и предусматривают единственное сообщение TCAP для всех трех триггерных точек.

1.1.5.2 Архитектура AIN/0.1

AIN/0.1 имеет два основных отличия от AIN/0.0. Первое отличие – формальная модель процесса обслуживания вызова, второе – набор сообщений между коммутационной станцией и SCP. Модель процессав AIN/0.1 разделена на две части – исходящую и входящую, в отличие от AIN0.0, где исходящая и входящая стороны не различались.

Максимальное количество речевых уведомлений увеличено до 254. Модель процесса обслуживания вызова на исходящей стороне содержит не три, а четыре триггерные точки: «исходящий запрос связи», «информация получена», «информация проанализирована» и «сетевая занятость». Модель процесса обслуживания вызова на входящей стороне имеет пока только одну триггерную точку – «входящий запрос связи».

Увеличение числа триггерных точек расширяет возможности воздействия на процесс обслуживания вызова со стороны логики услуг, находящейся в SCP. В качестве дополнительной функции SCP получил возможность активизировать и деактивизировать триггерные точки в коммутационной системе, а также следить за состоянием ее ресурсов. Кроме передачи инструкций для маршрутизации, SCP может поручить SSP следить за состоянием определенной абонентской линии (свободна она или занята) и сообщать ему об изменении ее состояния. Спецификация AIN/0.1 поддерживает все стандартные возможности ISDN (конечно, определенные ANSI).

Разделение модели обслуживания вызова на две части означает, что теперь триггерные точки позволяют логике услуг воздействовать на обслуживание как исходящего вызова, так и входящего.

Спецификации AIN/0.1 основаны на ANSI-варианте TCAP версии 2, в которой набор сообщений отличается от набора сообщений предыдущей версии. Если первой версией предусматривалось одно сообщение от SSP к SCP для всех триггерных точек, то во второй версии TCAP определены отдельные сообщения для каждой из четырех триггерных точек процесса обслуживания на исходящей стороне и одно – для триггерной точки процесса на входящей стороне.

1.1.5.3 Архитектура AIN/0.2

Несмотря на то, что основным стимулом разработки спецификации AIN/0.2 было желание поддержать услугу персональной свя-

зи PCS (Personal communication service) и возможность приема от абонента цифр номера речевым способом VAD (Voice activated dialling), все требования к архитектуре остались независимыми от вида услуг.

Основными особенностями второй фазы спецификаций стали интерфейс между коммутационной станцией и внешним устройством, например, IP, и новые триггерные точки – «занятость абонента» и «отсутствие ответа». Еще одно нововведение – обработка списка событий. Для этого, кроме триггерных точек, были определены *точки обнаружения событий* (EDP – Event detection points). Теперь SCP мог передать в SSP список тех событий, которые тот должен отслеживать в процессе организации связи, и, в случае их возникновения, уведомлять об этом SCP. Примерами таких событий могут быть занятость вызываемого абонента, отсутствие ответа, доступность ресурса и т.д.

AIN/0.2 предусматривает также, для определенных ситуаций, маршрутизацию вызовов по умолчанию. Это означает, что в случае сбойной ситуации вызов, вместо обычного отказа в обслуживании, может быть автоматически направлен на заданный заранее номер, или же абоненту может быть передано речевое извещение о причине отказа. Архитектура AIN/0.1, как и AIN/0.0, предполагала, что речевые извещения обеспечиваются ресурсами коммутационной станции, т.е. SSP. С введением AIN/0.2 появилась возможность размещать такие извещения в отдельно стоящем внешнем оборудовании, а именно в интеллектуальной периферии IP.

1.1.6 Стандартизация назрела

В течение первого десятилетия своего существования IN была чисто американским феноменом, что легко объясняется ранним возникновением конкуренции на рынке услуг связи США, уникальной ситуацией с разделением компании AT&T, быстрым введением сетей ОКС и использованием решений, основанных на применении сетевых баз данных.

Однако в конце 80-х годов, после того как ITU-T принял международные стандарты для системы сигнализации ОКС-7, и когда появились первые признаки конкуренции в отрасли связи в Старом Свете, некоторые европейские страны ввели в своих сетях новые услуги на базе технологии, аналогичной той, которая существовала на ранней стадии развития IN в США. Американские производители вышли на европейский рынок и создали в отдельных странах (Англия – 1988; Италия и Испания – 1991) сети IN в соответствии с Североамериканскими стандартами. В Японии компания NTT ввела архитектуру с сетевой базой данных в 1985 году.

Некоторые из новых услуг (VPN, Calling Card) по своей природе требовали международной стандартизации. Следуя требованиям времени, ИТУ–Т в 1989 году начал разработку серии рекомендаций для Интеллектуальной сети. Ставилась задача создать концепцию, которая могла бы стать основой для единой, в мировом масштабе, архитектуры IN. Такая архитектура должна была определить сетевые элементы, функциональные объекты, модели программных процессов, машины состояний и протоколы, которые могли бы быть сравнительно просто интегрированы в существующие телефонные сети.

Аналогично тому, как это было при стандартизации протоколов сигнализации, процесс стандартизации IN предполагает поэтапную разработку спецификаций серии Q.1200. На каждом этапе разрабатывается пакет рекомендаций, специфицирующий некоторый набор возможностей создания и предоставления услуг с определенными атрибутами. Специфицировать сами услуги в ИТУ–Т, как правило, не предполагается, чтобы оставить на рынке услуг место для конкуренции производителей и операторов. К настоящему времени ИТУ–Т выпустил рекомендации для наборов возможностей, предусматриваемых на двух первых этапах развития IN, – наборов CS–1 и CS–2. Готовится выпуск рекомендаций для CS–3, а с 2000 года началась работа над CS–4.

Рекомендации, относящиеся к CS–1, имеют номера Q.121X. Работа над ними велась с 1989 года. Первая версия спецификаций была одобрена в 1992 году, после чего они были пересмотрены в 1995 году. Услуги, которые должен поддерживать набор CS–1, ограничены пересчетом номерной информации, гибкой тарификацией и ведением диалога с пользователем.

Работа над спецификацией CS–2 была начата в 1992 году и завершена в 1997 году. Рекомендации, относящиеся к CS–2, имеют номера Q.122X. К возможностям, введенным в CS–1, добавлены возможности, позволяющие создавать услуги, которые допускают управление несколькими участниками связи, услуги, при предоставлении которых возможно взаимодействие разных сетей, услуги, связанные с мобильностью пользователя.

Работа над спецификацией CS–3 начата в 1996 году и должна быть закончена в 2000 году. Первоначально предполагалось, что в рамках архитектуры CS–3 будут полностью поддерживаться системы мобильной и персональной связи, как для стационарных, так и для мобильных сетей, в соответствии с требованиями Универсальной системы мобильной связи (UMTS) и Международной подвижной связи (IMT–2000), а также интеграция с сетями эксплуатационного управления (TMN) и с широкополосной сетью интегрального обслуживания (B–ISDN).

Однако стало ясно, что выполнить эти амбициозные планы не удастся. Большая часть задач была перенесена на период реализации CS-4 по той причине, что модель процесса обслуживания вызова в широкополосной сети значительно отличается от модели, которая использована в CS-1 и CS-2 и основана на модели обслуживания вызова в узкополосных сетях, и в связи с незавершенностью рекомендаций по UMTS и IMT-2000.

Глава 1.2

Введение в IN

1.2.1 Эволюция предоставления услуг

До начала 60-х годов логика предоставления услуг была неотделима от аппаратных средств коммутационной станции. Для решения вопроса об услугах, которые оператор сети хотел бы предложить своим клиентам, он встречался с производителем оборудования и согласовывал с ним характеристики будущей услуги и дату ее реализации. После этого оператор планировал развертывание услуги в своей сети, сводившееся к модификации аппаратных средств во всех коммутационных станциях.

И без того непростой, процесс этот осложнялся тем, что оператору, как правило, приходилось иметь дело с несколькими разными производителями оборудования. И иногда случалось так, что услуги в зоне обслуживания этого оператора оказывались не полностью идентичными, различаясь, например, в разных частях города или страны. Кроме того, после введения услуги в эксплуатацию модифицировать ее с учетом требований новых групп клиентов тоже было очень непросто. Зачастую для этого приходилось согласовывать с поставщиком оборудования дополнительные изменения аппаратных средств. Как следствие, оператору требовались годы, чтобы спланировать и реализовать в своей сети новую услугу.

Введение новой услуги требовало детального анализа передаваемой абонентом номерной информации, нередко приводя к необходимости предоставления отдельных пучков каналов для маршрутизации вызовов в зависимости от того, какая услуга была затребована.

В середине 60-х годов появились коммутационные станции с программным управлением. Это был существенный шаг вперед, позволивший сделать логику предоставления услуг программируемой, что значительно облегчило реализацию процесса предоставления услуг. Однако концепция предоставления услуг не была модульной. Добавить новую услугу к уже предоставляемым становилось все труднее, по мере возрастания сложности отдельных услуг и зависимости между ними. Оператор не мог сам использовать логику, поддерживающую одну услугу, для поддержки другой услуги.

Другой аспект традиционного предоставления услуг – способ передачи той сигнальной информации, которая нужна коммутационной станции, чтобы установить соединение. Для сигнализации и для передачи речи на всем пути от исходящей до входящей АТС

использовались одни и те же каналы. Если вызываемый абонент был занят, эти каналы использовались непроизводительно.

С введением в середине 70-х годов общеканальной сигнализации (ОКС) сети связи претерпели кардинальное изменение. ОКС использует сеть пакетной передачи данных, логически отделенную от сети разговорных каналов. Протокол, по которому работает сегодня сеть ОКС, – это система сигнализации №7 (ОКС–7). Сеть ОКС содержит конечные пункты сигнализации SP (Signalling points), обычно совмещенные с коммутационными станциями, транзитные пункты сигнализации STP (Signalling transfer points) и соединяющие их сигнальные звенья SL (Signalling links). Эти звенья организуются в тех же системах передачи, которые используются для разговорных каналов. Одного звена сети ОКС–7 достаточно, чтобы обеспечить сигнализацию для сотен разговорных каналов при значительном увеличении скорости установления соединений.

Помимо снижения продолжительности установления соединения и повышения надежности сигнализации, сети ОКС–7 обеспечили возможность предоставления абонентам сетей связи новых услуг, таких как «идентификация номера вызывающего абонента», «замкнутая группа пользователей», и некоторых других, широкое распространение которых связано с сетями ISDN. Спроектированная для широкого применения и на долгую перспективу, система ОКС–7 имеет в своем составе средства (специализированные протоколы), способные обеспечить связь между коммутационными станциями и территориально удаленными базами данных.

1.2.2 Элементы сети

На рисунке 1.2.1 приведена архитектура так называемой платформы IN, которую составляют узлы Интеллектуальной сети и связи между ними.

Узел коммутации услуг SSP (Service switching point) представляет собой обычную коммутационную станцию, в которой сохраняются все функции управления процессом предоставления основных услуг связи, оснащенную дополнительными программными средствами. SSP обеспечивает доступ абонентов сети связи к услугам IN и поддерживает протоколы взаимодействия с другими элементами IN. SSP определяет, что принятый им от абонента вызов требует обращения к услугам IN, и направляет соответствующий запрос в узел управления услугами SCP. Запрос может содержать номер вызывающего абонента, набранные им цифры номера, код требуемой услуги и другие параметры. После оснащения коммутационного оборудования функциями SSP услуги IN могут вводиться и удаляться путем соответствующих изменений конфигурации SSP, производимых техниче-

ским персоналом через обычный интерфейс оператора. Никаких изменений системного прикладного программного обеспечения (версии ПО) при этом не требуется.

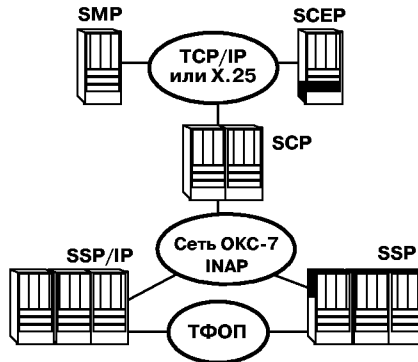


Рис. 1.2.1 Платформа Интеллектуальной Сети

Интеллектуальная периферия IP (Intelligent peripheral) выполняет вспомогательные функции, поддерживающие диалог с абонентом, такие как передача приглашения к набору дополнительных цифр, прием цифр, передаваемых абонентом многочастотным способом (DTMF), распознавание речи и некоторые другие. Интеллектуальная периферия может либо быть встроена в SSP, либо реализована в обособленном оборудовании. IP управляется со стороны SCP по протоколу INAP. Для подключения IP к SSP используются соединительные линии с сигнализацией, поддерживаемой подсистемой ISUP системы ОКС-7, или линии первичного доступа ISDN с цифровой абонентской сигнализацией DSS1.

Узел управления услугами SCP (Service control point) содержит программы, централизованно реализующие логику услуг, программные средства, поддерживающие протоколы взаимодействия с другими элементами сети, системное программное обеспечение, а также базу данных реального времени. SCP принимает запрос от SSP и возвращает ему инструкции для дальнейшей обработки вызова в соответствии с логикой затребованной услуги. До приема от SCP нужных инструкций обслуживание вызова в SSP приостанавливается. SCP отвечает за обслуживание вызова до тех пор, пока управление соединением не будет передано обратно в SSP. В течение времени, пока за обслуживание вызова отвечает SCP, SSP может передавать ему сообщения о результатах выполнения требуемых операций.

Система эксплуатационного управления и среда создания услуг SMP/SCEP (Service management point/Service creation environment point) предоставляют оператору сети возможности контроля и управления параметрами и конфигурацией услуг IN. Среда создания услуг

содержит средства конструирования, модификации и тестирования услуг до начала коммерческой эксплуатации и средства загрузки соответствующих программ в SMP. SMP обеспечивает эксплуатационное управление действующими услугами, а также управление подготовкой новых услуг и их введением. Протоколы взаимодействия между SMP, SCEP и SCP Международным союзом электросвязи пока не определены, однако стандартными протоколами де-факто стали X.25 и TCP/IP.

1.2.3 Модель обслуживания вызова

Для понимания процессов, происходящих в SSP при установлении соединения и при наблюдении за ним вплоть до разъединения, удобно использовать *модель базового процесса обслуживания вызова*. Модель содержит последовательность точек, отображающих состояния этого процесса (PIC – Point in call), между которыми могут присутствовать точки обнаружения (DP – Detection point) обращений к услугам IN или событий, которые представляют интерес с точки зрения логики услуг IN.

Точки PIC являются представлениями обычных действий, выполняемых коммутационной станцией во время установления соединения, и состояний, через которые проходит процесс обслуживания вызова с момента, когда абонент снял трубку, до окончания связи. Например, нулевое состояние – это состояние, в котором SSP следит за свободной абонентской линией. В качестве других состояний (или точек PIC) можно назвать состояние вызова абонентом станции («трубка снята»), состояние, когда станция принимает набираемые абонентом цифры номера («накопление информации»), «анализ информации», «маршрутизация», «оповещение» и т.д.

Через подобные состояния проходит процесс обслуживания вызова в любой станции (с функциями SSP или без них). Однако рассматриваемая ниже формальная модель процесса обслуживания вызова, требующего услуг IN, используется только в концепции IN, а потому любая коммутационная станция с функциями SSP должна соответствовать этой модели. Эта модель, содержащая в себе модель базового процесса обслуживания вызова *во взаимодействии с логикой услуг IN*, приведена на рисунке 1.2.2.

Точки обнаружения обращений к услугам IN (по-английски они называются TDP – trigger detection points, в связи с чем их удобно, для краткости, называть *триггерными точками*), отмечают приостановку базового процесса обслуживания вызова для обращения к логике услуг IN, происходящую в соответствии с заранее назначенным критерием. Таким критерием могут быть определенное сочетание цифр в набранном абонентом номере, префикс, категория вы-

зывающей абонентской линии и т.д. Важно отметить, что эксплуатационный персонал SSP может самостоятельно определять триггерные точки (т.е. делать их обнаруживаемыми) и назначать критерии для обращения к IN.

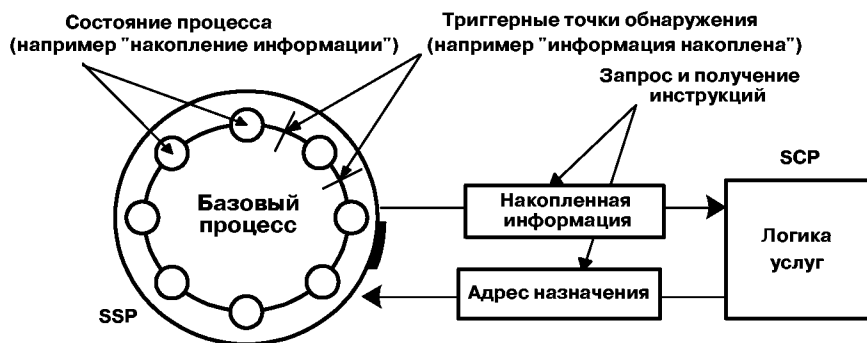


Рис. 1.2.2 Модель процесса обслуживания вызова

Кроме триггерных точек, назначаемых статически для каждого набора CS, определены также назначаемые динамически со стороны SCP *точки обнаружения событий* (EDP – Event detection point), интересных с точки зрения логики услуг IN. Такими событиями могут быть, например, занятость вызываемого абонента, ответ, отбой абонента и т.д. Переданная в SCP информация о том, какое именно событие наступило, используется сервисной логикой для того, чтобы принять решение о дальнейших инструкциях, которые нужно направить к SSP.

Если в процессе обслуживания вызова обнаруживается активная триггерная точка, процесс приостанавливается до тех пор, пока SSP и SCP не закончат обмен информацией, в результате которого определяются параметры следующего состояния базового процесса.

Рассмотрим пример работы модели. Предположим, что базовый процесс обслуживания вызова вышел из нулевого состояния, прошел состояние «трубка снята» и находится в состоянии «накопление информации». Если накопленная информация отвечает заданному критерию, процесс приостанавливается и «срабатывает» триггерная точка «информация накоплена». SSP формирует сообщение с необходимыми данными и направляет его через сеть ОКС к SCP. После приема от SCP ответного сообщения, в котором содержатся инструкции для маршрутизации вызова, SSP переходит в следующее состояние «анализ информации». Далее процесс обслуживания вызова происходит обычным образом вплоть до разъединения.

Так, в самых общих чертах, можно описать работу модели, которая является очень важным элементом концепции Интеллектуальной сети. Модель принципиально отличается от ранее существовавших моделей, в которых обработка вызова коммутационной станцией проходила от начального до конечного состояния без остановки.

1.2.4 Действующие лица

Разберемся с действующими лицами, участвующими в предоставлении и в использовании услуг IN.

Оператор сети IN – это организация, владеющая программно-аппаратными ресурсами Интеллектуальной сети. Если у оператора имеется SCEP, он может самостоятельно конструировать новые услуги для продажи прав их предоставления другим лицам или организациям.

Поставщик услуги – это организация, которая занимается коммерческим предоставлением услуги. Нередко оператор сети является одновременно и поставщиком большинства услуг IN.

Производитель оборудования IN зачастую поставляет не только оборудование для создания необходимой инфраструктуры, но и готовый пакет услуг, пригодных для их предоставления на текущем этапе развития сети.

Абоненты услуг IN могут быть двух типов:

- 1-я группа – это абоненты, желающие иметь возможность обращаться к услуге IN, которая предлагается другими действующими лицами для широкого пользования;
- 2-я группа – это физические или юридические лица, абонирующие услугу IN у поставщика для собственного пользования и/или для предложения ее на рынке телекоммуникационных услуг.

Структура рынка услуг IN может быть представлена следующим образом:

- поставщики услуг IN и операторы сетей, используемых для предоставления этих услуг;
- производители оборудования платформы IN;
- пользователи услугами.

Связь между различными элементами и участниками рынка, каждый из которых способен выполнять не одну функцию, приведена на рисунке 1.2.3. К примеру, крупный телекоммуникационный оператор может быть поставщиком услуг конечным пользователям и одновременно предоставлять ресурсы платформы IN другим фирмам-поставщикам услуг. С другой стороны, абонент телефонной сети общего поль-

зования может быть и конечным пользователем услуг IN, например, абонент телефонной сети из делового сектора становится одновременно подписчиком услуг FPN и VPN.



Рис. 1.2.3 Структура рынка услуг IN

К первой из указанных категорий участников рынка относятся:

- сетевые операторы: телекоммуникационные фирмы, организующие и эксплуатирующие сети передачи и распределения информации. Они создают инфраструктуру IN и осуществляют ее обслуживание. Использовать эту инфраструктуру для предоставления услуг IN, а также ресурсов сети, могут поставщики услуг, к которым относятся как сетевые операторы, так и другие фирмы-поставщики услуг;
- поставщики услуг: телекоммуникационные фирмы, предоставляющие услуги IN конечным пользователям и абонентам на базе либо собственных платформ IN, либо платформ других операторов. Примером может служить операторская фирма, арендующая часть емкости сети у более крупного оператора;
- независимые поставщики услуг: телекоммуникационные фирмы, не являющиеся собственниками сетей. Они поставляют услуги, используя платформу IN какой-либо операторской фирмы. В качестве примеров приведем банки, предлагающие консультации по возможным инвестиционным проектам, или перепродавцов, предлагающих услугу персонального номера абонентам сетей подвижной связи.

В категорию поставщиков элементов IN входят:

- поставщики коммутационных систем для сетей связи;
- изготовители средств вычислительной техники, обеспечивающие перенос части возможностей коммутационной системы в распределенные вычислительные платформы;
- поставщики программного обеспечения для создания услуг IN и управления ими.

Наконец, к категории пользователей услугами принадлежат:

- абоненты услуг: частное лицо или организация, подписавшаяся на услугу IN, предлагаемую поставщиками. Абонент оплачивает счет за пользование услугой, иногда принимает участие в определении ее параметров. В ряде случаев он может предлагать услугу конечному пользователю. Примером могут служить компании, ставшие абонентами услуги FPH;
- конечные пользователи: частное лицо или организация, имеющая доступ к услуге IN. Конечный пользователь, например, пользователь услугой FPH, не обязательно оплачивает ее.

1.2.5 Создание услуг IN

Предположим, что сеть связи оснащена в соответствии с архитектурой IN. Коммутационные станции снабжены функциями SSP, обслуживание вызовов в них происходит согласно введенной для концепции IN формальной модели. SSP соединены с одним (или для повышения надежности со связанной парой) SCP, который способен обрабатывать запросы, поступающие от всех SSP сети, используя для этой цели программы логики услуг. Как же происходит процесс создания конкретной услуги, которая должна предоставляться в этом окружении?

Программируемые сервисные возможности SCP подконтрольны поставщику услуги через SMP/SCEP. SCP содержит также специфические для каждой услуги данные, некоторые из которых могут модифицировать через SMP и поставщик, и абоненты услуг. Таким образом, абоненты могут конфигурировать параметры своих услуг (разумеется, только с разрешения поставщика услуг).

Теперь, когда логика предоставления услуг находится под контролем их поставщика, ему гораздо проще и дешевле создавать и эксплуатировать услуги. Оператор сети IN может достаточно оперативно предоставить поставщику возможность проведения опытной эксплуатации услуги, определив соответствующие триггерные точки в одном или нескольких своих SSP и загрузив в SCP логику предоставления этой услуги. На основе анализа результатов опытной эксплуатации поставщиком услуги может быть принято решение о целесообразности ее предоставления и о том, какую территорию она будет охватывать.

SCP обычно представляет собой высокопроизводительный компьютер, имеющий повышенную надежность. Программное обеспечение SCP состоит из двух частей. Первая часть – базовая платформа – предоставляется производителем SCP и построена таким образом, что может использоваться одновременно несколькими прикладными про-

граммами, реализующими логику предоставления разных услуг. Эти прикладные программы составляют вторую часть программного обеспечения и содержат ресурсы, необходимые для работы сервисной логики. Во время инсталляции (то есть загрузки) новой услуги в SCP эти ресурсы должным образом модифицируются.

Средства, обеспечивающие создание услуг и позволяющие задать для каждого абонента фактически индивидуальные характеристики нужных ему услуг, содержатся в SCEP. Многие производители оборудования IN предусматривают средства создания услуг с применением пользовательского интерфейса на основе графического редактора, управляемого мышью. При наличии такого интерфейса у проектировщика услуг отсутствует необходимость в использовании традиционных программных средств. Вместо этого используется программные средства, управляемые через систему меню, а услуги конструируются путем ввода в программы нужных параметров.

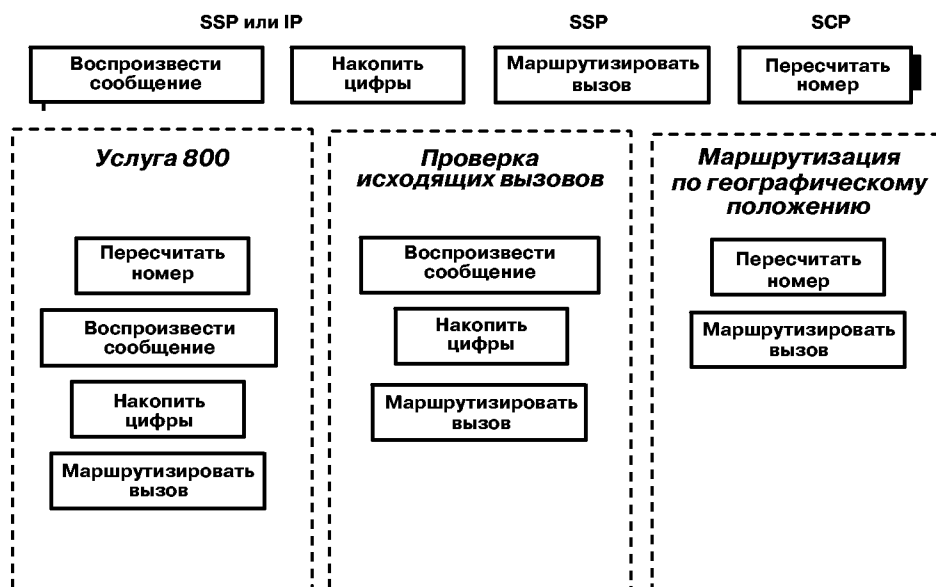


Рис. 1.2.4 Иллюстрация подхода к созданию услуг

На рисунке 1.2.4. приведен пример, разъясняющий использование при создании услуг независимых конструктивных модулей. Имеются следующие программные модули, доступные проектировщику через SCEP: «воспроизвести сообщение», «накопить цифры», «маршрутизировать вызов» и «пересчитать номер». SSP умеет воспроизводить сообщения и накапливать цифры, то же самое может делать IP. Маршрутизировать вызов может только SSP, а производить пересчет номера – только SCP.

Собирая имеющиеся программные модули в различные сочетания, можно создавать услуги, обладающие индивидуальными особенностями, например, такими как единый логический номер с интерактивным диалогом во время пользования услугой, проверка правомочности исходящих вызовов или маршрутизация по географическому положению.

Средства SCEP позволяют моделировать предоставление услуги в различных сетевых конфигурациях. Имеется также возможность тестирования разных аспектов, связанных с предоставлением услуг, таких как взаимодействие между услугами, взаимодействие между их составляющими, взаимодействие между сетями.

Созданная и оттестированная в SCEP новая услуга (т.е. программные средства ее реализации) через систему эксплуатационного управления SMP одновременно загружается во все SCP сети оператора. Как мы уже поняли, в SSP для поддержки услуги не требуется никакого нового программного обеспечения, нужно лишь назначить соответствующие триггерные точки.

Глава 1.3

Концепция и ее модель

1.3.1 Архитектурная концепция IN

Дабы не вызывать недоумения читателей непривычным для традиционной телефонии словосочетанием и не давать повода для произвольных его толкований, приведем определение термина «Интеллектуальная сеть», данное ITU-T. В рекомендациях Q.1201 и Q.1290 сказано, что термин IN (Intelligent Network, то есть Интеллектуальная сеть) обозначает архитектурную концепцию, которая применима к сетям электросвязи и предусматривает строго определенный набор гибко используемых средств, способствующих созданию и введению в сети связи новых услуг, в том числе, услуг, управляемых пользователем. Концепция IN устанавливает набор правил, основная идея которых состоит в том, что правила эти не зависят от создаваемой услуги и от структуры сети, в которой услуга будет предоставляться. Если услуга реализована в соответствии с названными правилами, ее называют услугой IN или услугой Интеллектуальной сети.

Следует отметить, что любая услуга может быть реализована и другими способами. Более того, до создания концепции IN услуги, подобные услугам IN, реализовывались разными производителями коммутационного оборудования по-разному. Конечный пользователь сегодня зачастую может не знать, каким способом предоставляется услуга, которой он пользуется. По этой причине было бы неверно говорить (как это часто делается) о каком-то особом наборе «интеллектуальных услуг», подразумевая, что эти услуги могут быть созданы исключительно средствами Интеллектуальной сети. Правильно называть услугами Интеллектуальной сети те услуги, которые реализуются в соответствии с концепцией IN, в отличие от таких же услуг, реализуемых какими-либо другими способами.

В технической литературе встречается также использование термина «Интеллектуальная сеть» и при упоминании об организации связи физических элементов, определенных в рамках концепции и необходимых для ее реализации, то есть термин этот применяется как название специализированной информационной сети, предусматриваемой в соответствии с концепцией IN. Дело в том, что большая часть логики, бывшая ранее частью программного обеспечения каждой АТС, при реализации Интеллектуальной сети переносится в небольшое количество специализированных компьютеров. Услуги IN поддерживаются путем информационного обмена между коммутационными станциями, вышеупомянутыми компьютерами и некото-

рыми другими специализированными устройствами. Назначение каждого элемента IN будет подробно рассмотрено ниже. Пока только заметим, что применять определение «интеллектуальная» к сетям, связывающим такого рода оборудование, следует, лишь отчетливо понимая, что роль этих сетей заключается в том, чтобы сделать «интеллектуальность» распределенной.

Концепция IN базируется на трех основополагающих свойствах, коими должны обладать средства реализации и предоставления услуг IN: *независимость от вида услуг, от структуры сети и от производителя оборудования*. Основная часть содержания данной книги посвящена детальному рассмотрению названных свойств и описанию требований к тому, как их можно реализовать.

Независимость от вида услуг подразумевает, что сигнализация между элементами IN не изменяется при введении новой услуги. Обеспечить это непросто, так как разработка распределенно выполняемых программ – дорогое удовольствие и занимает много времени. IN должна «скрывать» от разработчика услуги тот факт, что реализующие эту услугу программы выполняются распределенно. Чтобы это можно было обеспечить, специфицирован ряд элементарных функций, представляющих собой «конструкционные блоки», которые многократно используются проектировщиком при создании самых разных услуг. Независимость конструкционных блоков от вида услуг радикально облегчает процессы их спецификации и создания. Здесь напрашивается аналогия с блочным строительством зданий, когда архитектор, имея ограниченный набор строительных блоков, может, тем не менее, проектировать дома, обладающие индивидуальными свойствами.

Под независимостью от структуры сети понимается возможность гибкого размещения сетевых функций и ресурсов IN и эффективного управления ими, не зависящая от особенностей оборудования, на базе которого развивается сеть связи.

1.3.2 Концептуальная модель IN

После того как ITU-T принял решение о необходимости стандартизации аспектов, связанных с сетевыми услугами, встал вопрос о том, решение какого производителя или оператора взять за основу для спецификации новой архитектуры. При общности подхода, как это часто бывает, имелись значительные различия в деталях конкретной реализации. Отсутствие общего мнения в отношении деталей физической реализации протоколов нижнего уровня не давало возможности сдвинуться с места.

Был найден следующий выход: начать процесс стандартизации «сверху вниз», то есть от общего к частному, и продвигаться вперед по мере достижения компромисса. В качестве конечной цели была

определена, в самых общих чертах, так называемая долговременная архитектура *IN*, при использовании которой теоретически смогут быть полностью реализуемы все три упомянутых выше основополагающих свойства *IN*. Так как при сегодняшнем уровне технологии это невозможно, продвижение к целевой архитектуре *IN* должно идти поэтапно, с тем, чтобы на каждом этапе стандартизации функциональные возможности *IN* и требования к их реализации соответствовали текущему уровню развития коммутационной техники и техники обработки информации. Проще говоря, на момент стандартизации каких-либо функциональных средств *IN* должны существовать средства их реализации.

Для каждого этапа ITU-T определяет те услуги, которые можно реализовать на базе имеющейся технологии. Перечни услуг (и их составных элементов), определяемые для разных этапов, получили название Наборов возможностей (Capability sets – CSs). Для каждого такого набора предусмотрен отдельный пакет рекомендаций ITU-T, причем все возможности, определенные для предыдущего набора, обязательно входят в следующий набор.

Концепция Интеллектуальной сети описывается в рекомендациях ITU-T серии Q.12XX, причем группа рекомендаций Q.120X описывает общие аспекты концепции, группа Q.121X – аспекты реализации услуг, соответствующих первому набору возможностей (CS-1), Q.122X – аспекты, соответствующие второму набору (CS-2) и т.д. К настоящему времени ITU-T выпустил рекомендации для CS-1 и CS-2 и продолжает работать над рекомендациями для CS-3 (см. рис. 1.3.1.).

В качестве общей для всех CS модели описания архитектуры *IN* была предложена *концептуальная модель IN* (INCM – *IN conceptual model*), которая отражает эту архитектуру в разных плоскостях, дающих разную степень детализации. Модель (см. рис. 1.3.2.) содержит четыре расположенные одна над другой плоскости, каждая из которых является абстрактным представлением (со своей степенью детализации) тех возможностей, которыми обладает сеть, построенная в соответствии с концепцией *IN*. Следует отметить, что модель эта не имеет никакого отношения к известной семиуровневой модели взаимодействия открытых систем, каждый уровень которой служит для описания разных объектов (протоколов). INCM же предусматривает разные формы представления одного и того же объекта, а именно архитектуры *IN*.

Связанные с той или иной услугой функции, возможность и/или необходимость распределенного функционирования, а также связи между элементами архитектуры, абстрактно представляются в INCM посредством определенных на каждой плоскости функциональных объектов, причем функциональные объекты соседних плоскостей заданным образом соотносятся друг с другом.

При помощи INCM можно проектировать услуги и моделировать процесс их предоставления для сетей IN, имеющих разную структуру и разные принципы организации. Рассмотрим более подробно, чем различаются и зачем нужны разные плоскости модели.

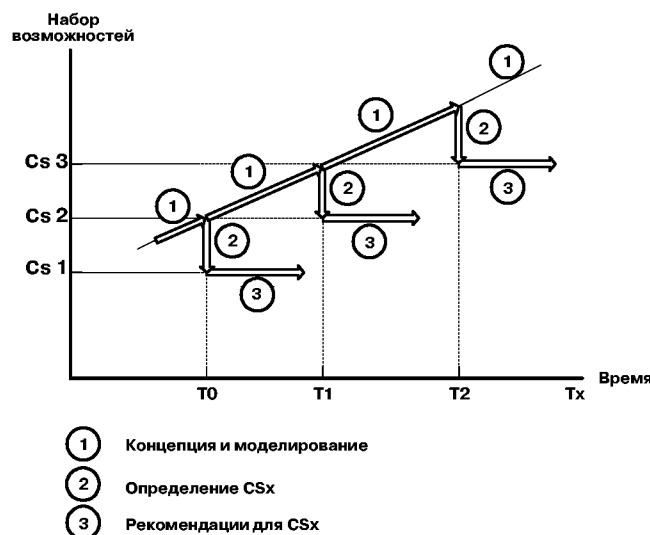


Рис. 1.3.1 Процесс стандартизации IN

Верхняя плоскость модели – *плоскость услуг* – представляет услуги так, как они «видны» конечному пользователю. Такое представление не содержит информации, относящейся к способу и деталям реализации услуги в сети. То, что услуга реализована в рамках IN, при представлении ее на плоскости услуг невидимо. Зато на этой плоскости видно, что услуги (*services*) компонуются из одной или из нескольких разных стандартизованных составляющих, каждую из которых пользователь воспринимает как одно из характерных свойств или, что то же самое, как один из *атрибутов* услуги (*service features*). Для каждого этапа стандартизации определяются совокупность таких составляющих и правила их использования.

На *глобальной функциональной плоскости* «появляется» сеть IN в виде единого функционального объекта. На этой плоскости представлены независимые от услуг *конструктивные блоки* (SIB – Service independent building block), одним из которых является SIB, реализующий *базовый процесс обслуживания вызова* (BCP – Basic call process), а также точка обращения BCP к другим SIB, называемая *иницилирующей точкой* (POI – Point of initiation) и *точки возврата* в BCP (POR – Point of return).

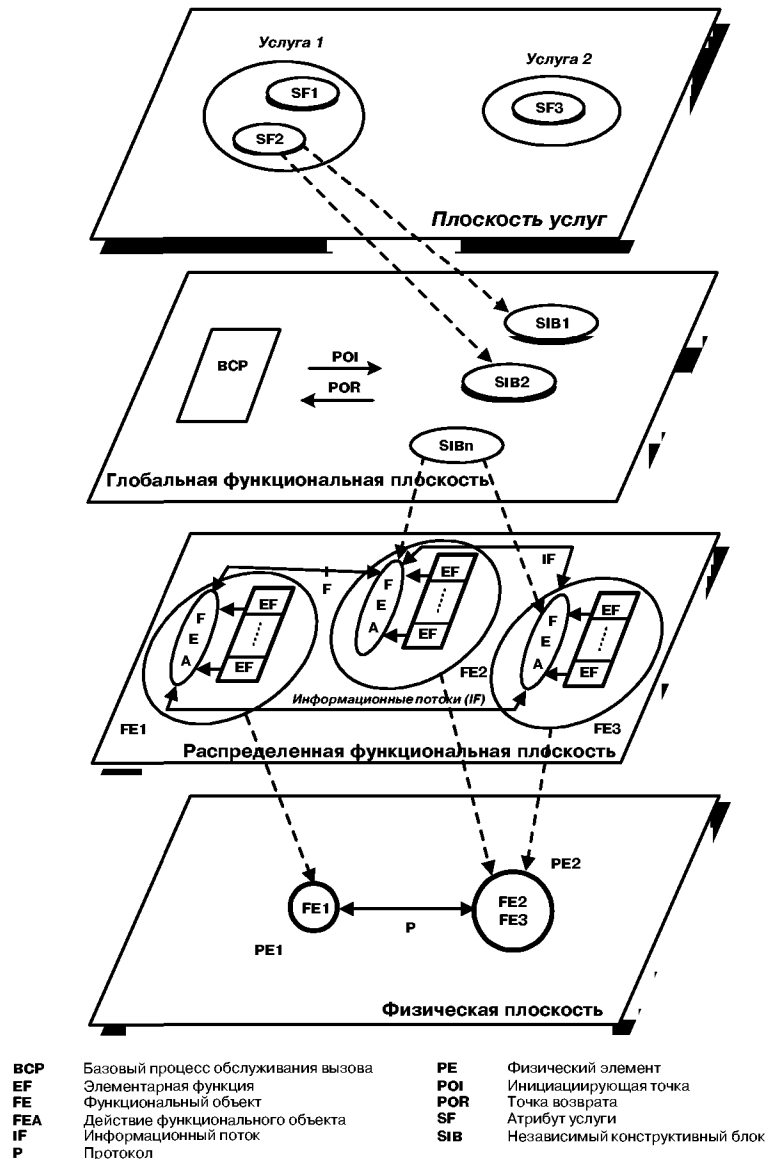


Рис. 1.3.2 Концептуальная модель IN

BCP выполняет традиционные для обычной коммутационной станции функции, такие как установление соединения, разъединение, хранение оперативных данных, необходимых для дальнейшей обработки, и имеет возможность (при обнаружении запроса услуги IN) обращаться к другим процессам. Точка POI представляет собой функциональный интерфейс между логикой процесса BCP и логикой другого процесса, который обеспечивает предоставление услуги (или

одной из составляющих услуги) IN. После завершения этого другого процесса происходит возврат через другой функциональный интерфейс (через точку POR) в процесс BCP, который продолжает работу, используя данные, полученные при возврате. Необходимость в спецификации точек POI и POR вызвана тем, что одна и та же «цепочка» SIB может представлять совершенно разные услуги (или составляющие услуг), смотря по тому, в каких точках процесса BCP она начинается и/или заканчивает свои действия.

На *распределенной функциональной плоскости* виден тот факт, что реализация услуги в среде IN производится программными средствами распределенным образом. Каждый *функциональный объект* (FE – Functional entity), определенный на этой плоскости, может выполнять целый ряд определенных для него *действий* (FEAs – Functional entity actions). Одно и то же FEA может быть определено для нескольких разных FE, однако любое FEA выполняется всякий раз только каким-то одним FE.

Рассмотренные выше независимые конструктивные блоки SIB представляются на *распределенной функциональной плоскости* в виде последовательностей действий, выполняемых функциональными объектами. Некоторые такие действия связаны с обменом информацией между объектами, что отображено на этой плоскости в виде информационных потоков.

Физическая плоскость представляет физические элементы (PE – physical entities) сети, в которой реализована концепция IN. Этими PE могут быть коммутационные станции, специализированные компьютеры или базы данных. На физической плоскости показано также, в каких PE размещаются те или иные FE и какие протоколы поддерживают информационный обмен между разными PE.

Логика процессов, обеспечивающих предоставление услуг IN, представляется на разных плоскостях по-разному. На глобальной функциональной плоскости видна лишь *глобальная логика услуг* (GSL – Global service logic), содержащая сервисные модули, каждый из которых поддерживает одну из составляющих услуг (один из атрибутов). На распределенной функциональной плоскости та же логика видна как набор распределенно выполняемых программ. Наконец, на физической плоскости видно, что программы, реализующие логику услуг, размещаются в предназначенных для этого программно-аппаратных комплексах – узлах Интеллектуальной сети.

1.3.3 Связи между плоскостями

Рассмотрим, каким образом объекты, принадлежащие соседним плоскостям INCM, соотносятся между собой. Концепцией IN задается строго определенное взаимное соответствие между архитектур-

ными элементами плоскостей модели. Рисунок 1.3.3 иллюстрирует основные принципы этого соответствия.

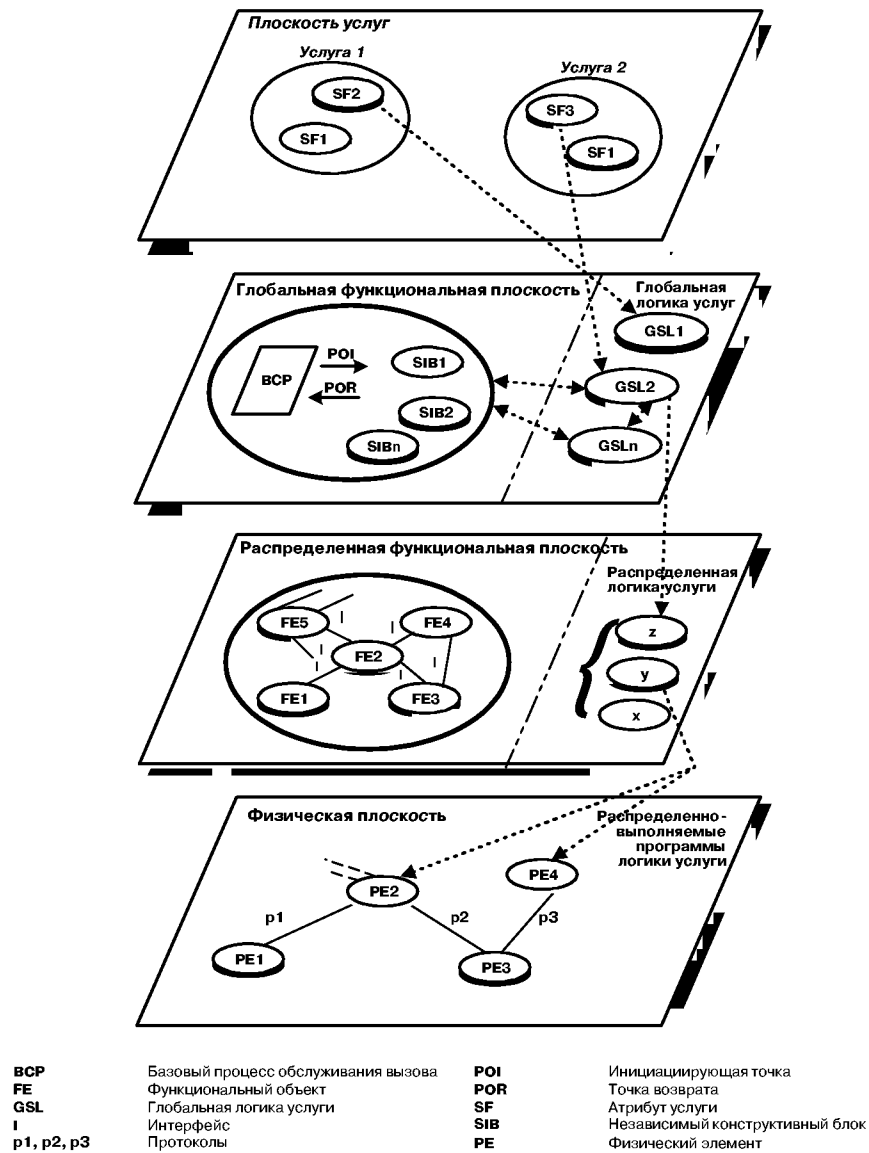


Рис. 1.3.3 Представление логики услуг в концептуальной модели IN

Атрибуты услуг, введенные на плоскости услуг, реализуются на глобальной функциональной плоскости соответствующими сочетаниями конструктивных блоков SIB, в число которых всегда входит BCP,

и сервисных модулей глобальной логики услуг. Для каждого атрибута соответствие устанавливается на этапе создания услуги.

Каждый SIB, введенный на глобальной функциональной плоскости, должен быть представлен хотя бы в одном FE распределенной функциональной плоскости, но может быть реализован и в нескольких FE. В последнем случае могут понадобиться совместные действия разных FE.

FE распределенной функциональной плоскости определяют, какими свойствами должны обладать физические элементы PE, в которых они размещены. Каждый FE размещается только в одном PE, но в одном PE может размещаться более одного FE. Связи между разными FE, видные на распределенной функциональной плоскости, специфицируются на физической плоскости в виде протоколов.

Поскольку любая услуга может предоставляться как в обычной сети, так и в сети, реализованной в соответствии с концепцией IN, на плоскости услуг должен быть предусмотрен определенный набор правил для того, чтобы одинаковые услуги могли работать совместно. Очевидно, что необходимость взаимодействия между услугами, реализованными по-разному, оказывает влияние и на другие плоскости модели.

Модель INCM дает также возможность описывать взаимодействие нескольких Интеллектуальных сетей. Распределенная функциональная плоскость делится на несколько частей, каждая из которых представляет одну IN. Затем определяются попарные связи в виде информационных потоков между FE, принадлежащими разным сетям. Семантическое значение и содержание каждого информационного потока устанавливается с учетом функциональных возможностей каждой из сетей.

Физическая плоскость также делится на несколько частей, представляющих каждая одну сеть, а FE распределенной функциональной плоскости соотносятся с взаимодействующими физическими элементами. Для обеспечения связи между PE, размещенными в разных физических сетях, определяются и стандартизируются протоколы взаимодействия.

1.3.4 Архитектура плоскости услуг

Под *услугой*, применительно к IN, понимается отдельно предлагаемый продукт, который образован из одной или нескольких обязательных составляющих и может быть модифицирован путем введения в него других составляющих. *Составляющая услуги* – это про-

дукт, предлагаемый либо как обязательная часть одной или нескольких услуг, либо как дополнительная часть, позволяющая модифицировать услугу (услуги). И в том, и в другом случае каждая составляющая отображает на плоскости услуг одно из свойств (*атрибутов*) той услуги, в состав которой она входит. Это дает оператору сети возможность продемонстрировать потенциальному клиенту то, в каком виде сеть может предоставлять ему ту или иную услугу.

Как это происходит? Оператор IN предлагает потенциальному клиенту перечень тех атрибутов, которыми может обладать каждая из услуг, предоставляемых его сетью. Клиент выбирает атрибуты, которые ему нужны, и формулирует словесное описание того, как он представляет себе услугу, обладающую этими атрибутами. К описанию добавляются сведения о логических номерах, списочных телефонных номерах, кодах аутентификации и др., и получается спецификация услуги для конкретного клиента.

Далее проектировщик услуг моделирует эту услугу на каждой из плоскостей INCM. Моделирование на плоскости услуг достаточно тривиально и сводится к использованию атрибутов услуги в качестве компоновочных модулей для ее спецификации и создания. Важным моментом является то, что составляющая услуги, образующая атрибут, является наименьшей неделимой ее частью, которую может затребовать пользователь при конструировании услуги под его требования.

Пользуясь возможностями IN, оператору гораздо проще обеспечить и быстрее организовать предоставление услуги даже узкой группе клиентов на ограниченный срок. А именно этого требует современный рынок услуг связи: умения работать с большим количеством сравнительно небольших групп пользователей, каждая из которых имеет собственные (и, часто, – динамически меняющиеся) требования.

1.3.5 Архитектура глобальной функциональной плоскости

Функции, представленные на этой плоскости, не являются специфическими ни для конкретных услуг, ни для их составляющих. Эти функции, как уже было сказано, называются независимыми от услуг конструктивными блоками (SIB). Для услуги, представленной на плоскости услуг в виде совокупности своих составляющих, на глобальной функциональной плоскости задается соответствие между каждой составляющей и одним или несколькими SIB.

В реализации всех услуг (как по правилам IN, так и по иным правилам) всегда участвует базовый процесс обслуживания вызова BCP.

Если при организации или во время связи пользователей требуется услуга IN, то при обращении ВСП к другим SIB управление передается глобальной логике услуг (см. рис. 1.3.4).

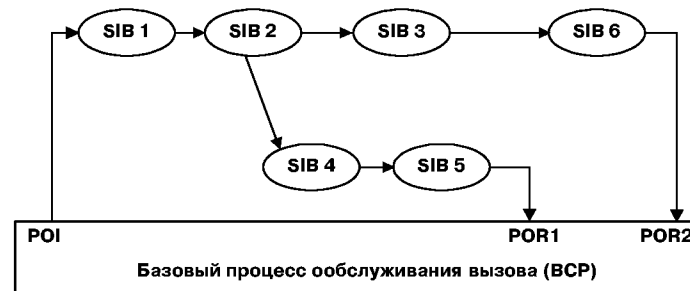


Рис. 1.3.4 Моделирование на глобальной функциональной плоскости

По своей сути и определению, никакой SIB, включая ВСП, не зависит от услуги и, как следствие, не может содержать сведения о последующем развитии событий. Единственным элементом на глобальной функциональной плоскости, зависящим от услуг и содержащим информацию о последовательности выполнения блоков SIB, является глобальная логика услуг.

Глобальная логика хранит сведения, необходимые для образования цепочки блоков SIB, сведения о последовательности их соединения и вариантах решения, а также данные, требующиеся для работы каждого из них. В конце цепи SIB она задает точку возврата в процесс ВСП. Для каждой услуги требуется по крайней мере одна точка POI, но, в зависимости от логики, нужной для данной услуги, могут быть определены несколько точек POR.

Главная особенность глобальной функциональной плоскости – это возможность многократного использования блоков SIB при создании новых услуг. Детальная реализация SIB на этой плоскости невидима и он представляется неделимым блоком с определенным интерфейсом. Реализуются SIB действиями одного или нескольких функциональных объектов распределенной функциональной плоскости. Однако вся прелесть конструирования услуг IN состоит в том, что этот факт скрыт от того, кто создает новую услугу.

Любая составляющая услуги может быть описана с помощью конечной цепочки SIB. И если сами по себе SIB от услуг не зависят, то при создании цепочки SIB, формирующей ту или иную составляющую услуги, каждое звено такой цепочки приобретает некоторые элементы зависимости от этой составляющей. Указанная зависимость выражается посредством параметров, специфицируемых для каждого SIB и задаваемых глобальной логикой услуг.

Существуют параметры двух видов – динамические и статические. Динамические параметры имеют разные значения для разных вызовов. Статические параметры задаются применительно к каждой составляющей услуги.

В качестве динамических параметров могут выступать номер вызывающего абонента, номер вызываемого абонента и т.п. Эти данные могут поступать от BCP (номер вызывающего абонента), генерироваться каким-либо SIB (пересчитанный номер) или приниматься непосредственно от абонента (например, PIN-код). По отношению к SIB динамические параметры могут быть входными и выходными.

Статические параметры каждого SIB в цепочке фиксированы. В качестве примера статического параметра можно привести индикатор файла, в котором хранится номерная информация для SIB, производящего пересчет номера.

В концепции IN базовый процесс BCP рассматривается как SIB, выполняющий традиционные для обычной коммутационной станции функции создания, поддержки и нарушения связи между абонентами. Если в процессе BCP одна и та же цепь SIB начинается и заканчивается в разных иницирующих точках и в разных точках возврата, она вызывает выполнение разных услуг.

Глобальная логика определяет порядок, в котором должны соединяться SIB для их последовательного исполнения при предоставлении определенной услуги. Эту логику можно сравнить с клеем, соединяющим конструкционные блоки в нужном порядке. Она уникальна для каждого вызова, но использует такие общие для всех обращений к определенной услуге элементы как точки POI и POR, логические связи между разными SIB, входные и выходные параметры (статические и динамические), определенные для каждого SIB.

1.3.6 Архитектура распределенной функциональной плоскости

Распределенная функциональная плоскость позволяет увидеть информационные потоки (связи) между функциональными объектами глобальной функциональной плоскости, возникающие в процессе предоставления услуги. Приводимая вкратце в данном параграфе методология функционального описания IN базируется на требованиях к спецификации услуг связи, подробно изложенных в рекомендации ITU-T Q.65.

Архитектура распределенной функциональной плоскости, общая для всех наборов CS, приведена на рис 1.3.5. Овалами показаны размещенные на этой плоскости функциональные объекты (FE), каждый

из которых представляет собой уникальную группу функций, сосредоточенную в одном физическом элементе IN и являющуюся подмножеством полного множества функций, необходимых для реализации услуги. Немного забегаая вперед, отметим, что время показало необходимость введения новых объектов в эту предложенную на ранних этапах архитектуру.

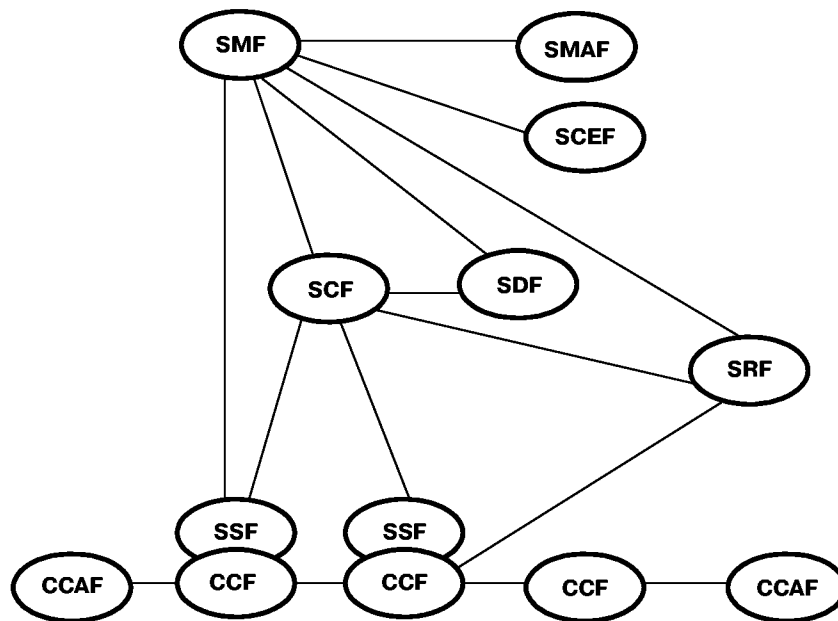


Рис. 1.3.5 Архитектура распределенной функциональной плоскости

Определены следующие правила:

- в одном физическом элементе может размещаться один или несколько FE;
- один FE не может быть разделен между двумя физическими элементами;
- FE одного типа могут быть размещены в разных физических элементах, однако дублирование одинаковых FE в одном физическом элементе не допускается.

Взаимодействие между функциональными объектами в модели носит название информационного потока. Связь двух любых FE отображается в модели группой информационных потоков, которые обозначаются соединяющими эти FE линиями. Связь может иметь идентификатор типа, уникально определяющий группу информационных потоков в модели (например, r1, r2 и т.д.), причем возможно существование в ней нескольких связей одного и того же типа. Если же линии между FE нет, это означает, что связи, требующей стандартиза-

ции в рамках модели IN, между этими FE не существует. При наличии связи между двумя FE, размещенными в разных физических элементах, характеристики этой связи определяют требования к протоколу информационного обмена между этими элементами. Как мы помним, спецификация протокола – это вопрос, относящийся к физической плоскости, а на распределенной функциональной плоскости информация, передаваемая через интерфейс между объектами, определена безотносительно к протоколу ее передачи.

Требование, запрещающее разделение FE на части, которые размещаются в разных физических элементах, введено в дополнение к требованиям рекомендации Q.65. Этим достигается ограничение количества разрешенных интерфейсов и, как следствие, телекоммуникационных протоколов. А ведь именно они, как мы помним, являлись камнем преткновения на пути создания международных спецификаций IN.

Применительно к распределенной функциональной плоскости можно оперировать такими понятиями как *функции коммутации услуг* и *функции управления услугами*; разделение процесса предоставления услуг на эти группы функций и составляет квинтэссенцию концепции Интеллектуальной сети. Однако на распределенной функциональной плоскости определены и другие функциональные объекты. Все объекты разделены на следующие группы.

Группа 1: функции, относящиеся к управлению связью пользователя:

- Функции коммутации услуг (SSF – Service switching function) обеспечивают взаимодействие между функциями управления связью пользователя и функциями управления услугами;
- Функции специализированных ресурсов (SRF – Specialized resource function) – предоставляют специализированные ресурсы, требующиеся для предоставления услуг IN;
- Функции управления связью пользователя (CCF – Call control function) – относятся к запросу связи и к управлению соединением в классическом смысле;
- Функции поддержки доступа (CCAF – Call control agent function) – обеспечивают доступ пользователя к функциям CCF.

Группа 2. Функции, относящиеся к услугам:

- Функции управления услугами (SCF – Service control function) – содержат логику услуг и управляют действиями SSF, CCF, SRF и SDF при предоставлении услуг;
- Функции предоставления данных для услуг (SDF – Service data function) – обеспечивают доступ в реальном времени со стороны SCF к данным, требующимся в процессе предоставления услуги IN, и их проверку.

Группа 3. Функции, относящиеся к созданию услуг и к эксплуатационному управлению услугами:

- Функции среды создания услуг (SCEF – Service creation environment function) – обеспечивают конструирование и тестирование услуг, подготавливают их введение;
- Функции рабочей станции (SMAF – Service management access function) – обеспечивают доступ техперсонала к функциям SMF;
- Функции эксплуатационного управления услугами (SMF – Service management function) – обеспечивают введение услуг IN и эксплуатационное управление ими.

Рассмотрим коротко роль каждой группы функций. Пользователь получает доступ к функциям CCF/SSF через CCAF. Функциональный объект CCAF принимает от пользователя запрос организации связи и передает его к CCF/SSF независимо от типа абонентской линии и используемой в ней сигнализации. Во время обработки этого запроса CCF/SSF может обнаружить необходимость обращения к услуге IN, что, в свою очередь, приведет к активизации логики услуг IN, которая реализуется специализированными программами SCF. Чтобы обеспечить предоставление услуги IN конечному пользователю, SCF инструктирует CCF/SSF о нужных действиях по управлению соединением, а также, если это требуется, о необходимости привлечения функций SRF и SDF. Заметим, что вся логика услуг, реализуемых традиционным способом, полностью обеспечивается функциями CCF.

Для каждого набора CS определяются группа необходимых функциональных объектов и разрешенные связи между ними. При этом, конечно же, сохраняется преемственность между последующими и предыдущими наборами CS. Так, в CS–2 к имеющимся в CS–1 объектам и связям добавлены некоторые другие, необходимые, например, для поддержки мобильности пользователей.

Связи между функциональными объектами организуются по принципу клиент–сервер. Связь может быть активизирована только объектом, выступающим в роли клиента, однако при разных обстоятельствах клиентом может выступать любой из функциональных объектов. Прекратить связь могут как клиент, так и сервер.

С точки зрения решаемых ими задач, области применения и накладываемых ограничений функциональные объекты (SSF, SCF и др.) описываются отдельно для каждого CS. Однако для облегчения работы с объектами введено несколько общих для всех CS полезных средств моделирования, таких как модель состояний базового процесса обслуживания вызова (BCSM – Basic call state model) и сегментная модель обслуживания вызова (CSM – Call segment model).

Вспомним, что любой SIB представляется на распределенной плоскости в виде одного или нескольких действий (FEA), выполняе-

мых функциональными объектами. Базовый процесс обслуживания вызова BCP моделируется на этой плоскости в виде двух BCSM – модели исходящей стороны и модели входящей стороны.

Модель BCSM представляет собой описание в виде машины конечных состояний тех действий CCF, которые требуются для организации и поддержания связи между пользователями. Модель идентифицирует перечень действий CCF при управлении соединением и показывает, как эти действия выполняются совместно.

Многие аспекты BCSM извне невидимы. Стандартизации подлежат только те из них, которые влияют на SSF и видимы для логики услуг, а также степень абстракции и детализации этой «видимости». BCSM идентифицирует такие состояния базового процесса обслуживания вызова, в которых допускается взаимодействие между CCF и логикой услуг. Модель позволяет выделить те события в этом процессе, которые могут привести к активизации логики услуг, или те события, о которых уже активизированная логика должна быть уведомлена, а также специфицировать те состояния процесса, в которых такие события могут наступать.

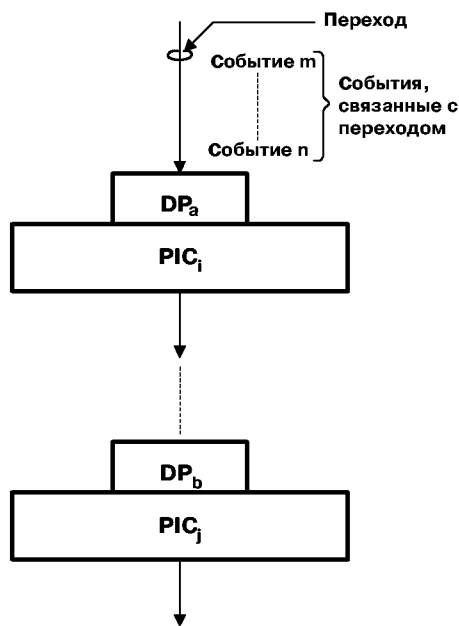


Рис. 1.3.6 Компоненты BCSM

На рисунке 1.3.6. показаны компоненты, используемые при описании BCSM. Точки *PIC* (Points in call) отображают в модели состояния процесса, связанные с действиями CCF и представляющие интерес с точки зрения логики услуг. Точки *DP* (Detection points) ото-

бражают в модели такие состояния процесса BCP, в которых возможна передача управления от CCF к SCF. *Переходы* происходят под воздействием *событий* и описывают нормальное продвижение процесса от одной PIC к другой. Используя компоненты BSCM, можно определить различные подмножества PIC, DP, событий и переходов для каждого из наборов CS.

Чтобы было понятно, как CCF/SSF «видят» связь, устанавливаемую между пользователями, и как управляют ею, полезно ввести модель, дающую общее представление этой связи. Такое представление дает сегментная модель CSM (Call segment model). Компонентами модели являются *сегмент доступа*, *базовый сегмент*, *сегмент составляющих услуги* и *звенья* (см. рис. 1.3.7).

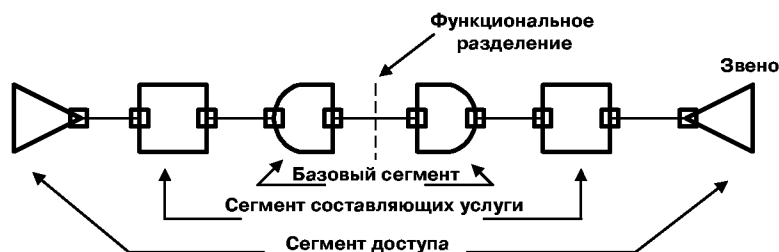


Рис. 1.3.7 Компоненты модели CSM

Сегменты доступа представляет внешние точки доступа в/из CCF/SSF, такие как абонентские и соединительные линии. Базовые сегменты представляют связь между сегментами доступа, поддерживаемую средствами коммутации и передачи и описываемую моделью BSCM. Так как базовый исходящий и базовый входящий сегменты функционально разделены, они изолируют процессы управления связью и услугами одного конечного пользователя от соответствующих процессов другого пользователя. Сегмент составляющих услуги соответствует логике услуги, затребованной одним из пользователей. Звенья представляют средства передачи и сигнализации между сегментами. В процессе управления связью пользователей сегменты динамически соединяются и удаляются.

1.3.7 Архитектура физической плоскости

Физическая плоскость представляет физические элементы (объекты) и интерфейсы между ними. Всем объектам распределенной функциональной плоскости должны быть поставлены в соответствие элементы физической плоскости, причем один физический элемент может соответствовать одному или нескольким функциональным объектам. Для физических элементов определяется стандартный интерфейс и протокол передачи информации.

Элемент физической плоскости, в котором содержится один или несколько объектов распределенной функциональной плоскости, мы будем называть *узлом Интеллектуальной сети*. Узел, содержащий функции SSF, называется SSP, узел с функциями SCF – SCP и т.д. В качестве примера распределения функций по узлам на рисунке 1.3.8 показана физическая архитектура для набора CS-1. Дадим определения узлам, приведенным на этом рисунке.

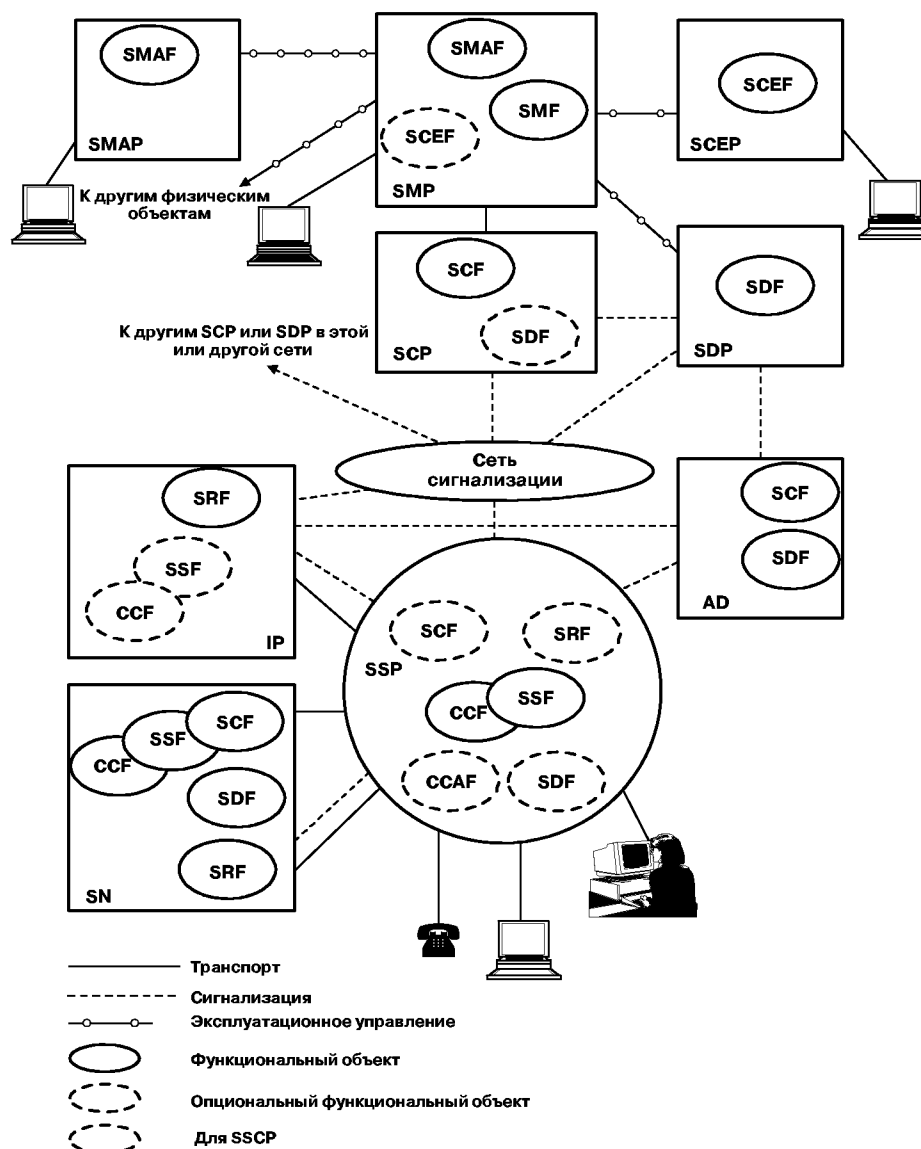


Рис. 1.3.8 Архитектура физической плоскости

Узел коммутации услуг SSP. В дополнение к функциям обеспечения доступа пользователей к сети (если SSP является оконечной ATC) и функциям традиционной коммутации, SSP обеспечивает доступ к услугам IN. SSP содержит средства обнаружения вызовов, требующих услуг IN, и средства взаимодействия с другими узлами. В нем размещаются функции CCF и SSF (а также CCAF, если это – оконечная ATC). Как опция, SSP может дополнительно содержать SCF и/или SRF и SDF.

Узел управления услугами SCP. SCP содержит программы, реализующие логику услуг IN, и, как опция, – данные об абонентах услуги. С целью повышения надежности и разделения нагрузки в сети несколько SCP могут выполнять одинаковые программы логики услуг и хранить одни и те же данные. SCP содержит SCF (и, как опция, – SDF). SCP получает информацию из SDP непосредственно или через сеть сигнализации, причем SDP может находиться как в той же самой IN, так и в другой. SCP соединяется с SSP (а при необходимости – и с IP) через сеть сигнализации. SCP может также взаимодействовать с IP через SSP.

Узел хранения данных для услуг SDP. SDP содержит данные, которые используются программами, реализующими логику услуг. В SDP размещается единственный функциональный объект – SDF. SDP может быть доступен узлу SCP и/или SMP непосредственно или через сеть сигнализации. В свою очередь, он может иметь доступ к другому SDP в своей или в другой IN.

Интеллектуальная периферия IP. IP содержит специальные ресурсы для взаимодействия сети с пользователем. Коммутационная матрица, предназначенная для подключения пользователя к ресурсам, может быть доступна внешним программам реализации логики услуг. В качестве примеров специализированных ресурсов можно назвать средства управления речевыми извещениями, средства синтеза и распознавания речи, средства приема цифр, передаваемых многочастотным кодом DTMF, мосты конференцсвязи, конвертеры протоколов сигнализации и т.п. IP содержит функции SRF (как опция, – также и SSF/CCF). Функции SSF/CCF используются для обеспечения внешнего доступа к внутренним ресурсам. Интеллектуальная периферия подсоединяется к одному или нескольким SSP и/или к сети сигнализации. SCP может запросить SSP подключить своего абонента к ресурсам IP, подсоединенной как непосредственно к этому, так и к другому SSP.

Вспомогательный узел управления AD (Adjunkt). Вспомогательный пункт управления функционально аналогичен SCP (т.е. содержит те же функциональные объекты), но подключен к SSP непосредственно. Связь между ними обеспечивается через высокоскоростной интерфейс, однако структура сообщений прикладного протокола сохраняется той же, что и при использовании сети ОКС.

Узел услуг SN. SN может управлять услугами IN и обеспечивать взаимодействие с пользователями. Узел услуг может взаимодействовать непосредственно с одним или несколькими SSP, при этом как сигнальные, так и транспортные связи между ними организуются по принципу «точка–точка». Функционально SN содержит SCF, SDF, SRF и SSF/CCF. SSF/CCF связаны с SCF в SN и недоступны со стороны внешних SCF.

Узел коммутации и управления услугами SSCP. SSCP представляет собой комбинацию SCP и SSP в одном узле. Функционально содержит CCAF, CCF, SSF и SCF. Связь между SCF/SDF и CCAF/CCF/SSF может обеспечиваться по нестандартному для сетей IN протоколу, обеспечивающему, однако, все необходимые для поддержки услуг возможности. SSCP может, как опция, содержать также и функции SRF. Интерфейс между SSCP и другими узлами – такой же, как у SSP.

Узел эксплуатационного управления SMP. SMP обеспечивает эксплуатационное управление услугами, а также управление подготовкой новых услуг и их введением. Система выполняет такие функции как администрирование баз данных, тестирование услуг, управление трафиком, хранение сетевых данных. SMP обязательно содержит SMF и может содержать также SMAF и SCEF. Система SMP имеет доступ ко всем имеющимся в сети узлам.

Узел среды создания услуг SCEP. SCEP используется для определения, конструирования и тестирования услуг IN и последующей их загрузки в SMP. Функционально содержит SCEF, взаимодействует прямо с SMP.

Узел доступа к системе эксплуатационного управления услугами SMAP. SMAP обеспечивает пользователям (таким как менеджеры услуг и некоторые клиенты) доступ к SMP. Одно из потенциальных применений SMAP – обеспечение единой точки доступа пользователя к нескольким SMP. Функционально SMAP содержит SMAF и взаимодействует непосредственно с SMP.

1.3.8 Сравнение методологий

Методология создания услуг, применявшаяся до недавнего времени, например, для спецификации услуг и протоколов ISDN, использует трехступенчатый подход. На первом этапе услуга описывается с точки зрения пользователя. На втором этапе определяются сетевые функции, необходимые для того, чтобы услуга работала. Выходным результатом этого этапа являются декомпозиция соответствующих сетевых ресурсов в функциональные объекты, спецификация действий этих объектов и информационных потоков между ними. На третьем этапе специфицируются протоколы.

Можно заметить, что эти три этапа соответствуют трем из четырех плоскостей концептуальной модели IN: плоскости услуг, распределенной функциональной плоскости и физической плоскости. При создании концепции IN использовали именно эту трехэтапную схему, включая связанную с ней, прямо скажем, непростую для понимания терминологию. Но, что очень важно, глобальная функциональная плоскость не имеет аналога в «старой» терминологии и представляет принципиально новую концепцию. Действительно, так как услуга на этой плоскости может конструироваться из независимого от услуг набора SIB, для взаимосвязи которых существует стандартная спецификация протокола, то отпадает необходимость разрабатывать отдельный протокол нижнего уровня для каждой новой услуги.

Суммируя все сказанное, можно сказать, что концептуальная модель IN представляет собой средство для четкого разграничения этапов проектирования услуг и последовательности действий на каждом из них. Моделируя процедуры управления связями пользователей на распределенной функциональной и на физической плоскостях этой модели, можно производить оценку возможных вариантов архитектуры IN и сравнение этих вариантов с точки зрения их экономической целесообразности и эффективности функционирования.

1.3.9 Общие аспекты прикладного протокола INAP

1.3.9.1 Окружение INAP

Принципы создания и предоставления услуг в рамках архитектурной концепции Интеллектуальной сети определяются концептуальной моделью, содержащей четыре плоскости, которые были подробно рассмотрены в предыдущих разделах. На распределенной функциональной плоскости модели действия, выполняемые разными блоками SIB, объединяются в группы, называемые функциональными объектами. При внедрении услуг IN эти функциональные объекты могут гибко распределяться по физическим элементам сети – узлам IN. В процессе предоставления услуг IN функциональные объекты из разных физических элементов взаимодействуют друг с другом, причем взаимодействие происходит в форме диалога: один функциональный объект запрашивает выполнение операции, а другой выполняет ее и возвращает первому результат.

Все необходимые для этого связи между физическими элементами сети осуществляются через стандартизованные интерфейсы. Специально для поддержки информационных потоков между узлами сети IN специфицирован прикладной протокол Интеллектуальной сети INAP (Intelligent Network application protocol), который определяет синтаксис и семантику вызываемых операций, назначение

и порядок их обработки. Протокол INAP будут рассматриваться достаточно подробно в главе 3.1, здесь лишь отметим, что на сегодня INAP может быть поддержан системой сигнализации OKC-7 и цифровой абонентской системой сигнализации DSS1.

Протокол INAP представляет собой прикладной протокол, т.е. протокол 7-го уровня модели Взаимодействия открытых систем (модели OSI). INAP предоставляет услуги для поддержки взаимодействия между прикладными процессами (AP – Application process), происходящими в узлах IN (например, в SSP, SCP, IP). Прикладной процесс является самым верхним уровнем абстрактного представления в INAP и описывает обработку запроса услуги в узле сети. Один прикладной процесс может использовать несколько прикладных объектов (AE – Application entity), каждый из которых поддерживает специфический набор функций (например, SSF AE, SRF AE, SCF AE), обеспечивающих взаимодействие с удаленными прикладными процессами.

AE представляет собой абстрактное описание функций, которые могут быть востребованы прикладным процессом AP для взаимодействия с удаленным AP. AE содержит определение каждой функции и правила использования этих функций. Базовым компонентом объекта AE является прикладной сервисный элемент (ASE – Application service element).

ASE объединяет в себе группу логически связанных функций, которые, в соответствии с рекомендацией ITU-T Q.775, могут быть использованы более чем одним AE. Применительно к IN, ASE представляют собой набор спецификаций процедур обслуживания вызова, известных читателю как операции. Для примера скажем, что при взаимодействии функций SCF и SSF прикладной сервисный элемент обращения к услуге SCF – *SCF activation ASE* – содержит описание операции запроса логики услуги – *InitialDP*, а прикладной сервисный элемент обработки событий – *Event handling ASE* – описывает операцию *EventReportBCSM*, которая сообщает SCF о таком событии в базовом процессе обслуживания вызова (например, о занятости абонента), уведомление о котором было запрошено заранее с помощью операции *RequestReportBCSM*, также описываемой этим ASE.

Прикладной процесс (например, в SSP) устанавливает логическую связь (так называемую *ассоциацию*), пользуясь которой, он будет взаимодействовать с другим прикладным процессом (например, в SCP). После этого начинается собственно выполнение операций. Существуют определенные правила, в соответствии с которыми устанавливается порядок выполнения операций. За последовательность операций в ASE отвечает специальная функция. Если существует всего одна ассоциация, это – функция управления одиночной (отдельной) ассоциацией SACF (Single association control

function). Если одновременно имеется несколько ассоциаций, необходима синхронизация взаимодействия во всех установленных ассоциациях, которую обеспечивает общая для всех SACF функция управления множеством ассоциаций (MACF – Multiple association control function).

Все средства (ассоциация, относящиеся к ней ASE, функции SACF), которые поддерживают диалог между двумя функциональными объектами, размещенными в разных узлах IN (например, диалог между SSF и SCF), образуют объект одиночной логической связи SAO (Single association object). Для наглядности на рисунке 1.3.9 приведена структура прикладного объекта AE.

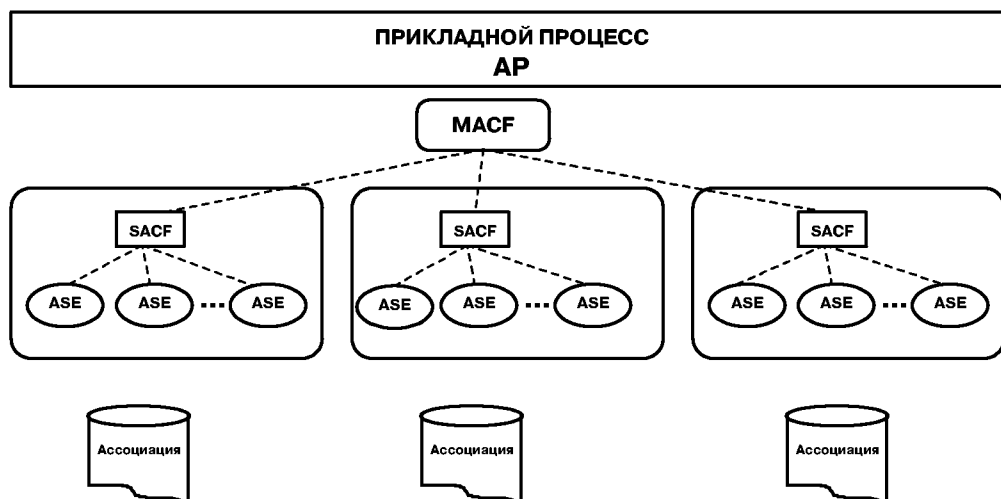


Рис. 1.3.9 Структура прикладного объекта AE

Чтобы вышеизложенное стало более понятным, рассмотрим простую аналогию. Предположим, что какой-либо абонент хочет получить обычную телефонную связь с другим абонентом. Будем рассматривать процесс организации этой связи как прикладной процесс (AP). При этом телефонный аппарат будет прикладным объектом (AE), который содержит следующие прикладные сервисные элементы (ASE): рычаг аппарата – «ASE-Рычаг», клавиши для набора цифр – «ASE-Цифры», клавиши для набора специальных символов – «ASE-*, #» и т.п. Все эти ASE участвуют в установлении соединения через телефонную сеть, иными словами, в создании ассоциации. Функции управления одиночной ассоциацией – SACF – должны в этом случае содержать, например, правило, говорящее о том, что *перед набором номера трубка должна быть снята с рычага*. Если телефонный аппарат поддерживает соединения по двум линиям, то нужны еще и функции управления множеством ассоциаций MACF, которые со-

держат правила переключения с одной линии на другую, а также правила объединения или разделения линий.

Протокол INAP является пользователем протокола ROSE (Remote operations service element – сервисный элемент удаленных операций), определенного в рекомендациях ITU–T X.219 и X.229, в том смысле, что INAP использует для переноса своей информации блоки данных протокола ROSE. Протокол ROSE содержится внутри подуровня компонентов TCAP системы сигнализации OKC–7 (ITU–T, Q.771–775) и DSS1 (ITU–T, Q.932). Сам по себе, ROSE является стандартизованным прикладным сервисным элементом. Поскольку ROSE предоставляет услуги вызова удаленных процедур, он используется во многих приложениях с распределенной обработкой. Для него определены четыре типа блоков данных протокола (PDU – Protocol data unit):

- Invoke – обращение;
- Return Result – возврат результата;
- Return Error – возврат ошибки;
- Reject – отказ.

Чтобы понять модель протокола ROSE, рассмотрим два прикладных объекта, взаимодействующих друг с другом. Один из них является клиентом, а другой – сервером. Клиент подготавливает запрос выполнения сервером некоторой операции (семантика операции является прозрачной для ROSE) посредством процедуры вызова блока данных протокола ROSE – *Invoke PDU*. Данная процедура в своих параметрах содержит информацию о требуемой операции. Запрошенная клиентом операция упаковывается в блок данных *Invoke PDU* и передается к серверу, который пытается ее выполнить. В зависимости от результата сервер может ответить одним из следующих способов:

- в случае успешного выполнения операции результат передается клиенту в виде блока данных Return Result PDU;
- в случае неуспеха клиенту передается сообщение об ошибке в блоке данных Return Error PDU;
- в случае, когда сервер не распознал запроса, он может послать блок данных с отказом Reject PDU.

Каждый запрос выполнения операции в точности определяет, ожидается ли ответ, и, если ожидается, то в каком виде он должен быть отправлен. Предусмотрены операции четырех классов:

- операции класса 1 требуют отчета как об успешном, так и о неуспешном результате выполнения операции;
- операции класса 2 требуют отчета только об успешном результате выполнения операции;

- операции класса 3 требуют отчета только о неуспешном результате выполнения операции;
- операции класса 4 не требуют никакого отчета.

Заметим, что OSI ROSE имеет пять классов операций, но подсистема TCAP ОКС-7 использует только четыре из них, а именно со 2-го по 5-ый. Первые три класса операций подразумевают ожидание пользователем ответа, поэтому необходимо наличие таймеров на обеих сторонах, чтобы быть уверенным, что процесс не «повис».

Нам осталось ввести последнее понятие, относящееся к определению прикладного протокола, в частности, протокола INAP. Этим понятием является *прикладной контекст* (AC – Application context). Как и ранее, обратимся к примеру, где абонент – *прикладной процесс* (AP) – сидит перед телефонным аппаратом – *прикладным объектом* (AE), намереваясь начать разговор. Будем рассматривать слова, и даже целые предложения, как *операции*, которые объединяются в смысловые (контекстные) *прикладные сервисные элементы* (ASE), например, «ASE – пицца», «ASE – музыка» и т.д. В начале разговора оба собеседника согласовывают предмет обсуждения. Пусть таким предметом у них будет обед из пяти блюд, т.е. базой разговора будет «ASE – пицца», а обсуждение должно вестись согласно правилам SACS, заключающимся в том, что *порядок обсуждения должен соответствовать порядку употребления блюд, а именно: закуска, суп, мясо или рыба, сыры и десерт*. Все, что подлежало согласованию в данном примере, и является прикладным контекстом.

Формально прикладной контекст может быть определен как набор ASE и правил, которые должны соблюдаться при взаимодействии прикладных процессов друг с другом. Прикладной процесс, который инициировал взаимодействие, предлагает один или более контекстов в блоке данных (PDU) и получает ответ, в котором возможность использования контекста либо подтверждается, либо отвергается, либо предлагается другой контекст. В последнем случае текущая ассоциация должна быть закрыта, и открыта новая для представления нового набора прикладных контекстов.

1.3.9.2 Особенности протокола INAP

Услуги, предоставляемые протоколом INAP. Семантика услуг, предоставляемых протоколом INAP, определена на распределенной функциональной плоскости концептуальной модели IN. Основной задачей протокола INAP является перенос информации, которой обмениваются функциональные объекты (FE) и которая определена в информационных потоках (IF) и в соответствующих информационных элементах (IE). В этом месте возникает законный вопрос: почему протокол отвечает за обмен информацией между функциональными объектами (FE), а не физическими объектами – узлами сети IN

(РЕ)? Ответ содержится в разделе 3 рекомендации ITU–Т Q.1208, где излагается ключевой принцип архитектурной концепции IN, а именно то, что «*протоколы должны быть определены таким образом, чтобы функциональные объекты можно было размещать по физическим элементам любым способом по желанию операторов и производителей оборудования*».

Словарь INAP. Словарь протокола INAP состоит из операций, поддерживаемых ROSE, и их параметров, которые, в свою очередь, соответствуют представленным на распределенной функциональной плоскости информационным потокам и информационным элементам.

Теперь читателем достигнута достаточная степень понимания того, как, в принципе, информационные потоки и информационные элементы распределенной функциональной плоскости могут быть использованы в окружении, описанном выше. Рассмотрение детальных проекций информационных потоков и информационных элементов в соответствующие операции и параметры отнесено в часть 3 данной книги, посвященную подробному описанию интерфейсов IN для набора CS–1.

Кодирование INAP. Рекомендация ITU–Т Q.1208 предписывает использовать для кодирования протокола INAP язык абстрактных описаний – ASN.1. Язык ASN.1 подобен языку Pascal и предназначен для независимого от кодирования определения блоков данных PDU прикладного уровня, которые, сами по себе, являются структурами данных. Язык ASN.1 содержит набор элементарных типов данных и способов создания структурированных типов данных из элементарных типов данных.

Примерами элементарных типов данных могут служить типы: *INTEGER (ЦЕЛЫЙ)*, *REAL (ДЕЙСТВИТЕЛЬНЫЙ)*, *BOOLEAN (БУЛЕВСКИЙ)*, *ENUMERATED (ПЕРЕЧИСЛЕННЫЙ)*, *BIT STRING (БИТОВАЯ СТРОКА)*, *OCTET STRING (СТРОКА ОКТЕТОВ)*.

Примерами структурированных типов данных, используемых в INAP, являются:

- конструкция *SET* – означает структуру и соответствует конструкции *RECORD* в языке Pascal;
- конструкция *SEQUENCE* – это структура (*SET*) с упорядоченными компонентами, так что они должны посылаться строго в том порядке, какой определен в спецификации;
- для определения массивов данных любых типов могут быть использованы конструкции *SIZE* и *OF*. Например, *SEQUENCE SIZE [1...10] OF <тип>*;
- Конструкция *CHOICE* соответствует конструкции *CASE* в языке Pascal и используется для объявления альтернатив;

- Конструкция **ANY** означает, что может быть использован любой из определенных типов. Такая конструкция особенно удобна для спецификации общих конструкций. Чтобы избежать очевидной двусмысленности, конструкция **ANY** должна сопровождаться выражением **DEFINED BY**, которое указывает определение типа.

Например, параметр **FilteringTimeOut**, соответствующий информационному элементу **Filtering Timeout IE**, должен определять либо длительность периода просеивания, либо момент времени, когда просеивание должно быть прекращено. Совокупность этих требований приводит к следующему виду спецификации:

```
FilteringTimeOut ::= CHOICE {  
    duration [0] Duration,  
    stoptime [1] DateAndTime  
}
```

Первое и четвертое выражения в фигурных скобках, т.е. **duration** и **stoptime**, являются идентификаторами, за ними следуют их тэги в квадратных скобках – **[0]** и **[1]**, соответственно. Третье и шестое выражения – **Duration** и **DateAndTime**, обозначают типы идентификаторов.

Преимущества спецификаций **ASN.1** перед детальным кодированием подобны преимуществам использования языков программирования высокого уровня перед программированием в двоичных кодах. Тем не менее, **ASN.1** не всегда использовался в промышленности по той причине, что до недавнего времени имелся только один стандартный метод преобразования спецификаций **ASN.1** в конечный код. Этот метод задан базовыми правилами кодирования (**BER** – Basic encoding rules) и имеет тот недостаток, что при его использовании получаются недостаточно компактные сообщения. Здесь важно заметить, что в будущих рекомендациях по **IN** будет разрешен другой метод, соответствующий правилам пакетного кодирования (**PER** – Packet encoding rules).

Процедуры INAP. Процедуры протокола **INAP** выполняют функции синхронизации действий, относящихся как к приему, так и к передаче сообщений между взаимодействующими объектами. Именно процедуры и вызывают основные затруднения в процессе распределенной обработки. В то время как ошибки в синтаксисе протокола могут быть легко обнаружены и откорректированы человеком, нарушения в синхронизации являются настолько сложными, что их фактически невозможно выявить на стадии проектирования. Более того, процедурные ошибки часто приводят к такому «странному» поведению системы, что определить его причину бывает крайне трудно. И, что еще хуже, инциденты могут иметь «блуждающий» характер, вследствие чего нормальную ситуацию бывает невозможно восстановить.

В рекомендациях **ITU-T** процедуры протокола обычно специфицируются двумя методами: стрелочными диаграммами и описанием на

языке SDL. Стрелочные диаграммы наглядно показывают общую картину обмена сообщениями между взаимодействующими объектами и служат для иллюстрации основной идеи протокола. Но с их помощью невозможно отразить все многообразие сочетаний сообщений, учитывающее все возможные ошибочные случаи. Описания на языке SDL охватывают все возможные ситуации; более того, существуют специальные отладочные средства, позволяющие проверить правильность разработанных SDL-описаний. Отмеченные достоинства, разумеется, сказываются на объеме SDL-описаний и их обзримости. Данные обстоятельства наглядно иллюстрирует приложение к обновленной редакции Q.1218, в котором содержится полный набор SDL-описаний всех процедур, относящихся к набору CS-1.

Часть 2

Архитектура

Глава 2.1

Услуги и атрибуты

2.1.1 Услуги

Набор возможностей 1 (CS-1) – это первый этап стандартизации IN. В рекомендации ITU-T Q.1211, которая является введением в CS-1, представлен общий обзор понятий, относящихся к CS-1, даны соответствующие определения и описаны основные характеристики и возможности этого набора.

CS-1 представляет собой начальный набор возможностей, который является подмножеством целевой долговременной архитектуры IN и предполагает наличие сетевых функциональных средств для поддержки услуг IN, как уже определенных, так и находящихся в стадии определения ITU-T.

Хотя IN и представляет собой не зависящую от услуг архитектуру, услуги и атрибуты услуг, которые должен поддерживать CS-1, существенны с точки зрения используемых в рамках CS-1 блоков SIB, модели процесса обслуживания вызова и принципов управления услугами.

CS-1 ориентирован на поддержку так называемых услуг и атрибутов услуг *типа А*, которые характеризуются следующим:

- услуга (атрибут услуги) соотносится всякий раз только с одним участником связи и независима от услуг (атрибутов), соотношенных с другим (другими) участником (участниками) этой связи;
- в любой момент времени на одни и те же аспекты связи между пользователями может воздействовать только одна группа функций управления услугами SCF.

Все остальные услуги (атрибуты услуг) относятся к *типу В* и набором CS-1 не поддерживаются. В рекомендации ITU-T Q.1211 для первой фазы IN определено 25 услуг и 38 атрибутов услуг. Ниже приведен полный перечень услуг, которые должен поддерживать набор возможностей CS-1, и дано краткое описание каждой из этих услуг.

AAB – автоматическое альтернативное предъявление счета (Automatic alternative billing). Услуга позволяет обслуживаемому абоненту произвести вызов с любого телефонного аппарата. Плата за связь относится на счет абонента, который определен для этой услуги и не принадлежит ни к вызывающей, ни к вызываемой сторонам. Для обращения к услуге пользователь набирает код доступа. Затем, по требованию интеллектуальной периферии, он набирает код владельца счета и свой личный идентификационный номер (PIN-код). По этим

данным проверяется кредитоспособность владельца счета и правомочность набранного PIN-кода.

ABD – сокращенный набор (Abbreviated dialling). Услуга позволяет абоненту виртуальной частной сети использовать для вызова другого абонента этой сети его сокращенный номер, набирая, например, только 4 цифры, даже если линии этих двух абонентов включены в разные коммутационные станции.

ACC – вызов с отнесением платы на счет, указанный в карте (Account card calling). Эта услуга позволяет абоненту осуществлять вызов с любого таксофона, оборудованного устройством чтения карт. Плата за связь автоматически относится на личный или на служебный банковский счет, который записан в карте. Абоненту присваивается код доступа и PIN-код. Он обращается к услуге путем набора кода доступа, по запросу вводит свой PIN-код и вставляет карту в устройство чтения. Система проверяет данные и информирует абонента о возможности доступа, после чего он может набирать номер нужного ему абонента.

CCC – вызов по кредитной карте (Credit-card calling). Услуга позволяет автоматически относить плату за любой исходящий вызов на личный счет абонента в банке. Абонент должен набрать номер своей карты и PIN код, а затем – номер вызываемого абонента.

CON – конференцсвязь (Conference calling).* Услуга позволяет соединить несколько абонентов для организации телефонной конференцсвязи. Число участников, соединяемых одновременно, может быть разным, в зависимости от характеристик «моста», объединяющего речевые сигналы, и от требований к качеству обслуживания.

CD – распределение входящих вызовов (Call distribution). Услуга позволяет абоненту направлять входящие к нему вызовы на разные терминалы в соответствии с правилом распределения, которое может изменяться самим абонентом. Возможны три варианта:

- циклическое распределение, при котором вызовы распределяются по всем терминалам равномерно;
- процентное распределение, когда к каждому терминалу направляется заранее установленная для него доля вызовов;
- иерархическое распределение, при котором выбор терминала, к которому должен быть направлен очередной вызов, определяется приоритетным списком.

CF – переадресация вызовов (Call forwarding). Все вызовы, поступающие к абоненту услуги, направляются на другой номер.

CRD – распределенная ремаршрутизация вызовов (Call rerouting distribution). Когда абонент пользуется этой услугой, те входящие к нему вызовы, которые сталкиваются с определенными ситуациями

(абонент занят, абонент не отвечает в течение определенного количества посылок вызова, переполнена очередь или действует ограничение количества принимаемых вызовов), ремаршрутизируются в соответствии с предварительно заданным правилом: либо направляются на другой номер (включая пейджер или речевую почту), либо подключаются к средствам воспроизведения речевого сообщения (стандартного или записанного самим абонентом), либо устанавливаются на ожидание.

*CCBS** – завершение вызова, встретившего занятость вызываемого абонента (*Call completion to busy subscriber*). Услуга позволяет информировать вызывавшего абонента об освобождении абонента, бывшего занятым, не требуя нового набора номера.

DCR – маршрутизация вызовов по условию (*Destination call routing*). Услуга позволяет заказавшему ее абоненту задавать маршрутизацию направленных к нему вызовов к разным терминалам в соответствии с:

- временем дня, днем недели и др.;
- географическим положением вызывающего абонента;
- приоритетом (например, зависящим от PIN);
- стоимостными коэффициентами;
- коэффициентами распределения нагрузки, устанавливаемыми абонентом.

FMD – переадресация «вслед за собой» (*Follow-me diversion*). Услуга позволяет абоненту при изменении своего местонахождения назначать телефонный номер для переадресации на этот номер вызовов, поступающих на его обычный номер. Вызовы будут переадресовываться на номер, который абонент определил последним. Абонент может изменять этот номер из любой точки сети.

FPN – бесплатный вызов (*Freephone*). Услуга позволяет относить плату за связь на счет абонентов (заказчиков) услуги (например, компании по продаже авиабилетов) и предоставляет вызывающему абоненту возможность доступа по единому номеру, например, к ближайшему от него пункту заказа авиабилетов.

MAS – массовые вызовы – обработка входящих вызовов, связанных с объявлениями или играми (*Mass calling*). Используя эту услугу, оператор сети может временно присвоить заказавшему ее абоненту единственный номер. Когда этот номер набирает абонент телефонной сети, ему передается речевое сообщение с предложением набрать дополнительно цифру, соответствующую выбранному им варианту, которая фиксируется в счетчике. Тарифные коэффициенты для начисления платы за вызовы, направленные к этому номеру, могут изменяться.

MCI – идентификация злонамеренных вызовов (Malicious call identification). Услуга позволяет абоненту регистрировать злонамеренные вызовы. Регистрируется следующая информация: номер вызываемой стороны, номер вызывающей стороны, время и дата вызова. К услуге можно обратиться в течение или после активной фазы связи, но до того, как вызванный абонент даст отбой.

OCS – просеивание исходящих вызовов (Originating-call screening). Услуга позволяет абоненту назначить список номеров, набор которых с его аппарата запрещен. Запрет можно обходить, набирая всякий раз надлежащий код.

PRM – услуга с дополнительной оплатой (Premium rate). Абоненту начисляется плата как за полученную связь, так и за дополнительную информацию, предоставляемую ее владельцем – заказчиком услуги. Доход распределяется между оператором сети связи и владельцем информации.

SCF – избирательная переадресация вызовов при занятости/отсутствии ответа (Selective call forwarding on busy/no answer). Услуга дает вызываемому абоненту возможность переадресовывать на другой номер часть вызовов, поступивших к нему, когда он занят или не отвечает в течение Y секунд или X посылок вызова.

SEC – защитное просеивание (Security screening). Услуга предусматривает проверку прав абонента перед предоставлением ему доступа к сети, к системе или к приложению. Проверка производится на основе набираемого абонентом PIN-кода. Предусматривается фиксация данных о попытках получить доступ, набирая неверный PIN-код (сколько было попыток, в течение какого периода времени, кем и откуда они производились), что повышает уровень защиты.

SPL – разделение оплаты (Split charging). Услуга предусматривает раздельное начисление платы вызываемой и вызывающей сторонам в заданной пропорции.

TCS – просеивание входящих вызовов (Terminating call screening). Услуга запрещает входящие вызовы в соответствии со списком запретов и, как опция, в зависимости от времени суток

UAN – универсальный номер доступа (Universal access number). Услуга обеспечивает доступ к терминалам абонента, размещенным в разных районах или зонах, путем набора одного и того же универсального номера. Маршрутизация входящего вызова может осуществляться в зависимости от географического положения вызывающего абонента.

UDR – маршрутизация, определяемая абонентом (User defined routing). Услуга позволяет абоненту задать несколько разных маршрутов для исходящих от него вызовов (через частную сеть, через сеть общего пользования, через виртуальную или смешанную сеть)

с тем, чтобы выбор одного из нескольких возможных маршрутов производился всякий раз с учетом списка их предпочтительности.

UPT – универсальная персональная связь (Universal personal telecommunications). UPT – услуга мобильности, позволяющая вызывать абонента путем набора его персонального номера (PN), в какой бы из множества сетей он ни находился и каким бы сетевым доступом ни пользовался. Набранный PN пересчитывается в физический номер, обеспечивающий маршрутизацию вызова туда, где абонент находится в данный момент.

VOT – телеголосование (Televoting). Услуга предоставляет своему заказчику возможность провести опрос общественного мнения с использованием телефонной сети. Для разных вариантов ответа назначаются разные номера. Каждый участник голосования набирает тот номер, который соответствует его мнению. Возможно использование единственного телефонного номера, набрав который, участник голосования получает речевую подсказку и сообщает свое мнение либо путем дополнительного набора, либо в ходе речевого диалога.

VPN – виртуальная частная сеть (Virtual private network). Услуга позволяет, используя ресурс сети общего пользования, построить виртуальную частную сеть, содержащую в себе функции нескольких УПАТС и функции СЕНТРЕХ. Каждому абоненту VPN может быть приписан либо класс разрешенных услуг, либо перечень его прав и привилегий.

Примечание: Услуги, отмеченные знаком (*), могут быть поддержаны в CS-1 лишь частично, поскольку они требуют наличия, кроме атрибутов услуг типа А, также и атрибутов услуг типа В.

2.1.2 Атрибуты услуг

Ниже приведен полный перечень атрибутов услуг набора CS-1 и дано краткое описание каждого атрибута.

ABD – сокращенный набор (Abbreviated dialling). Атрибут позволяет задавать сокращенные номера для виртуальной частной сети VPN, что дает абоненту VPN возможность вызвать другого абонента своей компании, набрав присвоенный тому сокращенный номер, даже если линии этих двух абонентов включены в разные коммутационные станции сети общего пользования.

ATT – оператор (Attendant). Атрибут позволяет абонентам VPN соединяться с рабочим местом оператора VPN для получения сведений об услуге VPN (например, информации о номерах абонентов) путем набора специального кода доступа.

AUTC – проверка прав (Authentication). Атрибут позволяет удостовериться в том, что абонент имеет право пользоваться теми функциями сети, которые он запросил.

AUTZ – разрешающий код (Authorization code). Атрибут дает возможность абоненту VPN обойти ограничения прав связи, наложенные на тот аппарат, которым он в данный момент пользуется.

ACB – автоматический обратный вызов (Automatic call back).* Атрибут позволяет вызываемому абоненту автоматически вызвать последнего из вызывавших его абонентов.

CD – распределение вызовов (Call distribution). Атрибут позволяет абоненту задать правило распределения входящих к нему вызовов между двумя или несколькими терминалами.

CF – переадресация (Call forwarding). Атрибут дает вызываемому абоненту возможность переадресовывать входящие к нему вызовы на другой номер независимо от статуса линии этого абонента.

CFC – переадресация при занятости или при отсутствии ответа (Call forwarding on busy/no answer). Атрибут дает вызываемому абоненту возможность переадресовывать на другой номер вызовы, поступающие к нему, когда он занят или если он не дает ответа в течение определенного числа посылок вызывного сигнала.

GAP – прореживание потока вызовов (Call gapping). Атрибут позволяет поставщику услуг автоматически ограничить интенсивность потока вызовов, направленных к определенному абоненту.

CHA – удержание соединения (Call hold with announcement).* Атрибут позволяет абоненту перевести установленное соединение в режим удержания с передачей удерживаемому абоненту музыки или речевого уведомления.

LIM – ограничение количества вызовов (Call limiter). Атрибут позволяет абоненту, имеющему в своем распоряжении несколько терминалов, ограничить количество вызовов, одновременно требующих связи с этими терминалами, назначив максимально допустимое число таких вызовов (если все терминалы заняты, вызовы могут переадресовываться на другой номер).

LOG – регистрация вызовов (Call logging). Атрибут позволяет производить запись о каждом вызове, требующем связи с определенным номером.

QUE – установка вызова на ожидание (Call queueing). Атрибут позволяет вызываемому абоненту в случае занятости вызываемого абонента ожидать его освобождения и получать требуемую связь сразу, как только тот освободится. После установки на ожидание вызываемому абоненту передается уведомление о том, что на его вызов будет дан ответ, когда нужная линия станет доступна.

*TRA** – *переключение связи (Call transfer)*. Атрибут позволяет абоненту перевести активное соединение в режим удержания, соединиться с другим абонентом и переключить на него удерживаемую связь.

*CW** – *уведомление об ожидающем вызове (Call waiting)*. Атрибут предусматривает передачу занятому вызываемому абоненту уведомления о том, что к нему поступил (и установлен на ожидание) новый вызов.

CUG – *замкнутая группа пользователей (Closed user group)*. Атрибут позволяет абоненту стать членом замкнутой группы пользователей VPN, как правило, имеющих возможность исходящей и/или входящей связи только внутри этой группы. Абонент может быть членом более чем одной группы CUG. В этом случае CUG определяется таким образом, что отдельные ее члены могут вызывать абонентов вне этой CUG и/или принимать вызовы извне.

*COC** – *соединение для справки (Consultation calling)*. Атрибут позволяет абоненту перевести активное соединение в режим удержания и соединиться с другим абонентом, чтобы получить от него нужную справку.

CPM – *управление профилем обслуживания (Customer profile management)*. Атрибут позволяет абоненту управлять в реальном времени профилем услуг, которыми он пользуется (переопределять пункты назначения, изменять содержание речевых сообщений, модифицировать правила распределения вызовов и т.д.)

CRA – *воспроизведение записанного сообщения (Customised recorder announcement)*. Атрибут позволяет обслуживаемому абоненту передать в ответ на неудачный входящий вызов заранее записанное сообщение. Абонент может записать несколько разных сообщений, информирующих о той или иной причине неудачи (например, «все линии заняты», «сейчас нерабочее время» и т.д.).

CRG – *назначение типов вызывного сигнала (Customised ringing)*. Атрибут позволяет абоненту назначить разные типы вызывного сигнала для вызовов, поступающих к нему от разных вызывающих абонентов.

DUP – *подсказка вызываемому абоненту (Destinating user prompter)*. Атрибут позволяет передать вызываемому абоненту речевое сообщение, предлагающее ему набрать дополнительные цифры, которые могут быть использованы логикой услуги для продолжения процесса обслуживания вызова.

FMD – *переадресация «вслед за собой» (Follow-me-diversion)*. Атрибут позволяет абоненту VPN назначать номер, на который следует переадресовывать входящие к нему вызовы. Этот номер может быть номером внутри VPN или номером телефонной сети общего пользо-

вания, причем один и тот же номер могут использовать для переадресации одновременно несколько абонентов. Если абонент назначает для этой цели другой номер, прежнее назначение отменяется.

MAS – массовые вызовы (Mass calling). Атрибут позволяет обрабатывать большое количество входящих вызовов, связанных с объявлениями или играми.

MMC – конференц-связь (Meet-me-conference).* Атрибут позволяет абоненту зарезервировать ресурс, поддерживающий многостороннюю связь, и указать время начала, дату и продолжительность конференции. В назначенный день и час каждый из предполагаемых участников многосторонней связи должен набрать номер, присвоенный этому ресурсу, чтобы получить доступ к конференции.

MWC – многосторонняя связь (Multiway calling).* Атрибут позволяет абоненту установить соединения одновременно с несколькими абонентами.

OFA – доступ извне (Off-net access). Атрибут позволяет абоненту VPN получить доступ в свою виртуальную частную сеть с любого терминала телефонной сети общего пользования, набирая определенный PIN-код. Разным PIN-кодам могут присваиваться разные приоритеты или привилегии. Одним и тем же PIN-кодом могут пользоваться несколько абонентов.

ONC – внесетевой вызов (Off-net calling). Атрибут позволяет абоненту VPN вызвать абонента, не являющегося пользователем этой VPN. Вызов пользователя другой VPN также считаются внесетевым.

ONE – единый номер (One number). Атрибут позволяет абоненту, владеющему двумя и более терминалами в разных географических точках, иметь единственный телефонный номер для всех этих терминалов. Абонент может назначить, к какому из терминалов должны направляться вызовы из той или иной географической зоны.

ODR – маршрутизация в зависимости от местоположения вызывающего пользователя (Origin dependent routing). Атрибут дает абоненту возможность допускать или не допускать к своим терминалам входящие вызовы. Если вызов допущен, он направляется к тому или иному терминалу в зависимости от того, в какой географической зоне находится вызывающий пользователь.

OCS – просеивание вызовов по признаку места их возникновения (Originating call screening). Атрибут позволяет абоненту запретить направлять к нему вызовы из определенных мест на основе кода района, от которого поступил вызов.

OUP – подсказка вызывающему абоненту (Originating user prompt-er). Атрибут позволяет передать вызывающему пользователю сообщение, предлагающее ему набрать цифру или серию цифр многочастотным кодом DTMF. Эти цифры могут служить для дальнейшей маршрутизации вызова или для проверки прав связи.

PN – персональный номер (Personal numbering). Атрибут поддерживает услугу UPT, обеспечивая абонента этой услуги уникальным номером, который служит для получения с ним связи. Абонент может иметь несколько номеров UPT, используемых для разных приложений (например, бизнес-UPT-номер – для деловой связи, частный-UPT-номер – для неделовой связи).

PRMC – дополнительная плата (Premium charging). Атрибут позволяет установить размер дополнительной платы и затем разделить ее между оператором и вызванным абонентом в том случае, когда второй из них рассматривается как поставщик услуги.

PNP – частный план нумерации (Private numbering plan). Атрибут позволяет управлять планом нумерации внутри частной сети, который может отличаться от плана нумерации сети общего пользования.

REVC – начисление платы вызываемому абоненту (Reverse charging). Атрибут предоставляет возможность заказчику услуги (например, услуги Freephone) принимать вызовы с начислением платы за каждый входящий вызов на свой счет.

SPLC – раздельная оплата (Split charging). Атрибут позволяет разделить оплату связи между вызываемой и вызывающей сторонами.

TCS – просеивание входящих вызовов (Terminating call screening). Атрибут позволяет абоненту запрещать или разрешать входящие к его терминалам вызовы в зависимости от того, какой номер был набран.

TDR – маршрутизация в зависимости от времени (Time dependent routing). Атрибут дает абоненту возможность допускать или не допускать к своим терминалам входящие вызовы. Если вызов допущен, он направляется к тому или иному терминалу в зависимости от времени суток, дня недели, дня года и т.д.

Примечание: Атрибуты, отмеченные знаком (*), могут быть поддержаны в CS-1 лишь частично, поскольку они требуют наличия возможностей предоставления, кроме услуг типа А, также и услуг типа В.

2.1.3 Функциональные связи и интерфейсы

На рисунке 2.1.1 представлены, в виде опорных точек, 13 функциональных связей.

A. CCAF-CCF;	H. SMF-SDF;
B. CCF-CCF;	I. SMF-SRF;
C. CCF-SRF;	J. SMF-SMAF;
D. SSF-SCF;	K. SMF-SCEF;
E. SCF-SRF;	L. SMF-SSF/CCF;
F. SCF-SDF;	M. SSF-CCF.
G. SMF-SCF;	

Для CS-1 определены только три из этих связей, а именно D, E и F.

Возможности управления требуются только для первых шести из приведенного списка функциональных связей (т.е. для связей A, B, C, D, E и F). Для них определены четыре группы средств управления, называемые классами управления:

- **класс 1:** Средства управления соединением;
- **класс 2:** Средства управления обслуживанием вызова;
- **класс 3:** Средства управления услугой IN;
- **класс 4:** Средства эксплуатационного управления.

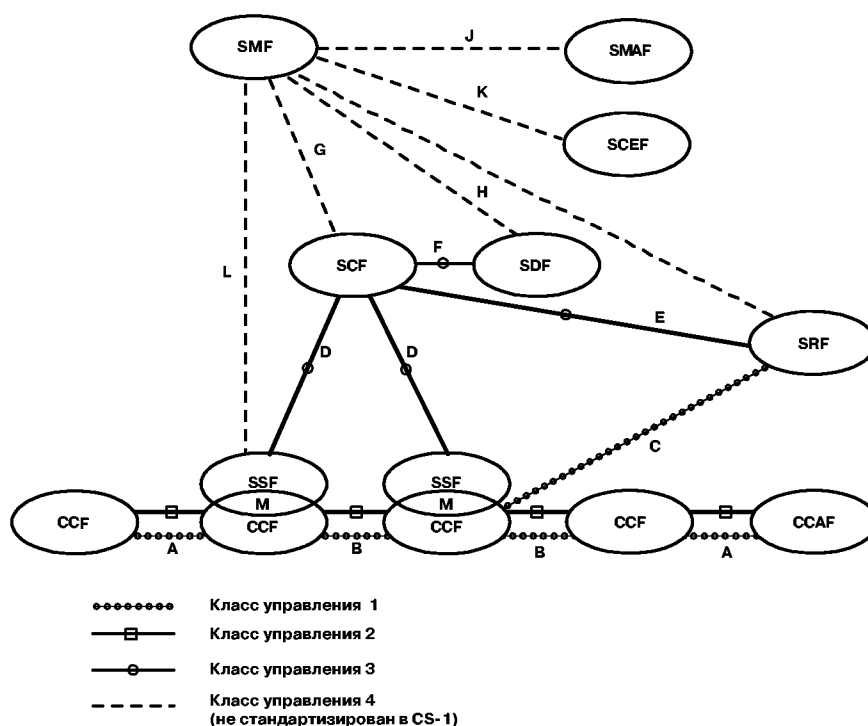


Рис. 2.1.1 Виды связей и опорных точек для CS-1

Функциональная связь в некоторой опорной точке может предусматривать один или несколько классов управления. Любое сочетание функциональной связи и класса управления называется управляющей связью. Управляющая связь обозначается строкой вида <буква>.<цифра>, где <буква> обозначает функциональную связь, а <цифра> – класс управления. Например, D.3 означает управляющую связь между функциональными элементами SSF и SCF для класса управления 3 (IN service control capabilities). В таблице 2.1.1 при-

ведены интерфейсы, поддерживающие различные управляющие связи, а в таблице 2.1.2 показаны все управляющие связи, поддерживаемые набором CS-1.

Таблица 2.1.1

Управляющая связь	Интерфейс	Примечания
A.1 A.2	DSS1/Q.931; DSS1/Q.932.	Для сети, структурированной согласно принципам IN CS-1, управляющие связи могут быть поддержаны существующими стандартными интерфейсами.
B.1 и B.2	OKC-7/ISUP.	
C.1	DSS1/Q.931, OKC-7/ISUP.	
D.3	OKC-№7/TCAP, DSS1/Q.932;	В CS-1 специфицированы три новые функциональные связи в опорных точках D, E и F. Физические аспекты реализации этих связей не предполагают прямого физического интерфейса между участвующими сетевыми функциями. Например, сообщения, относящиеся к опорной точке E, могут проходить через отдельный физический элемент, содержащий SSF.
E.3	OKC-7/TCAP, DSS1/Q.932;	
F.3	OKC-7/TCAP, DSS1/Q.932.	
M		В наборе CS-1 не стандартизована.

Таблица 2.1.2

CCAF									
CCF	A1,2	B1,2							
SSF		(M3)*							
SRF		C1							
SDF					(Q3)				
SCF			D3	E3	F3	P3	(O3)		
SCEF									
SMAF									
SMF			(L3)	(I3)	(H3)	(G3)	(K3)	(J3)	(R3)
	CCAF	CCF	SSF	SRF	SDF	SCF	SCEF	SMAF	SMF
(...) Для CS-1 не стандартизовано * Определяется производителем Используется при взаимодействии сетей Пустое пространство - отсутствие связи									

Процесс, в котором несколько сетей кооперируются с целью предоставления услуги, называется взаимодействием сетей. Необходимость такого взаимодействия возникает, если абоненту нужна услуга, которую одна сеть самостоятельно предоставить не может. Типичный пример – ситуация, когда данные, необходимые для услуги

(например, UPT или VPN), размещаются не в той сети, где возник вызов. Несмотря на то, что взаимодействующие сети могут иметь разные типы доступа (например, ТФОП, ISDN и др.) и разные уровни структуры IN, услуги IN должны предоставляться абонентам обычным образом, независимо от этих различий.

Подобно рисунку 2.1.1, описывающему функциональные связи между функциями IN и соответствующие опорные точки, рисунок 2.1.2 описывает возможные функциональные связи между функциями IN, размещенными в двух разных сетях.

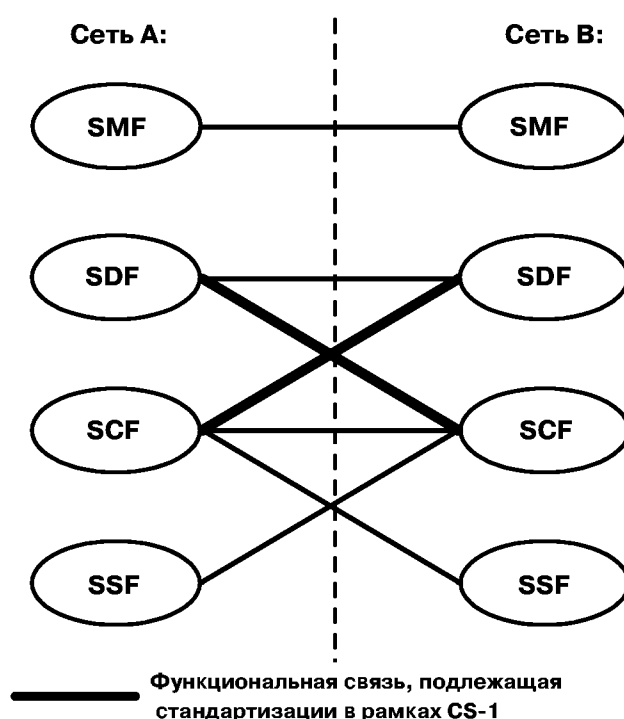


Рис. 2.1.2 Функциональные связи при взаимодействии сетей

В отношении CS-1 необходимо отметить следующее:

- поскольку взаимодействие вида «SSF в одной сети – SCF в другой сети» в рамках CS-1 не требуется, то соответствующая функциональная связь не является предметом спецификаций;
- поскольку CS-1 поддерживает только услуги типа А (с управлением из одной точки), то в нем не предусматривается управляющая связь между двумя SCF, а поэтому функциональная связь не является предметом спецификаций;
- поскольку при предоставлении некоторых услуг IN (например, UPT, VPN) SCF выполняет функции пересчета номера и проверки прав

пользования посредством обмена информацией с SDF, то такая функциональная связь является предметом спецификаций;

- поскольку связь SDF – SDF нужна, чтобы скрыть от SCF распределенность базы данных IN, а это не относится к управлению услугами IN, то данная функциональная связь не является предметом спецификаций;
- поскольку связь SMF – SMF относится к эксплуатационному управлению услугами (т.е. к сети TMN), то эта функциональная связь не является предметом спецификаций.

Глава 2.2

Глобальная функциональная плоскость

2.2.1 Блоки SIB, определенные для CS-1

Характеристики глобальной функциональной плоскости (GFP – Global functional plane), применительно к CS-1, специфицированы в рекомендации ITU-T Q.1213, где описаны модель глобальной функциональной плоскости, набор из 14 независимых от услуг блоков SIB и специализированный SIB базового процесса обслуживания вызова BCP. Кроме того, в рекомендации Q.1213 определены 9 иницирующих точек POI для интерфейса с глобальной логикой услуг (GSL) и 6 точек возврата POR, а также специфицированы, применительно к CS-1, связи между плоскостью услуг и глобальной функциональной плоскостью.

В рекомендации ITU-T Q.1213 определены, как необходимые для набора CS-1, следующие блоки SIB:

- ALGORITHM (АЛГОРИТМ);
- AUTHENTICATE (АУТЕНТИФИКАЦИЯ);
- CHARGE (НАЧИСЛЕНИЕ ПЛАТЫ);
- COMPARE (СРАВНЕНИЕ);
- DISTRIBUTION (РАСПРЕДЕЛЕНИЕ);
- LIMIT (ОГРАНИЧЕНИЕ);
- LOG CALL INFORMATION (ЗАПИСЬ ИНФОРМАЦИИ О ВЫЗОВЕ);
- QUEUE (ОЧЕРЕДЬ);
- SCREEN (ПРОСМОТР СПИСКА);
- SERVICE DATA MANAGEMENT (ЭКСПЛУАТАЦИОННОЕ УПРАВЛЕНИЕ ДАННЫМИ ДЛЯ УСЛУГ);
- STATUS NOTIFICATION (ИЗВЕЩЕНИЕ О СТАТУСЕ);
- TRANSLATE (ПЕРЕСЧЕТ);
- USER INTERACTION (ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ);
- VERIFY (ПРОВЕРКА).

Дополнительно к этому набору из 14 блоков SIB Европейским Институтом Стандартов (ETSI) были определены еще 7 блоков:

- CONNECT (СОЕДИНИТЬ);
- CONTINUE (ПРОДОЛЖИТЬ);
- DISCONNECT RESOURCE (ОСВОБОДИТЬ РЕСУРС);
- EDP INFO (ИНФОРМАЦИЯ EDP);

- EDP REQUEST (ЗАПРОС EDP);
- INITIATE CALL (ИНИЦИИРОВАТЬ ВЫЗОВ);
- RELEASE CALL (РАЗЪЕДИНИТЬ).

Блоки SIB, по определению, независимы от того, какие услуги/атрибуты услуг они реализуют. Любой SIB не содержит информации ни о предыдущем, ни о последующем SIB, используемом в описании услуги/атрибута услуги.

2.2.2 Динамические и статические параметры блоков SIB

Чтобы описать конкретную услугу или конкретный атрибут услуги с помощью базового набора блоков SIB, необходимо всякий раз вводить в эти блоки некоторые элементы зависимости от услуг. Для этого служат параметры, значения которых определяются данными, позволяющими настраивать тот или иной SIB на выполнение требуемых функций. Параметры специфицируются независимо для каждого SIB и доступны ему через глобальную логику услуги.

Для каждого блока SIB требуются данные двух типов. Это – данные, относящиеся к конкретному вызову (Call instance data – CID), и данные, связанные с поддержкой конкретной услуги (Service support data – SSD). Данные CID специфицируют характеристики абонентов и пользователей услуги (например, информацию о вызывающей или о вызываемой линии) и определяют *динамические параметры*, значения которых изменяются с каждым новым вызовом. Эти данные могут быть получены из блока базового процесса обслуживания вызова BCP, могут генерироваться каким-то другим блоком или вводиться самим абонентом. Данные SSD определяют *статические параметры*, значения которых от вызова к вызову не изменяются, а зависят от услуги или атрибута услуги. В состав SSD входят *фиксированные параметры* и *указатели полей данных CID* (CIDFP). Значения фиксированных параметров определяются описанием услуги (или атрибута услуги) и одинаковы для всех вызовов, обслуживание которых связано с обращением к этой услуге или атрибуту. Указатели полей определяют, какие CID нужны блоку для данной услуги, и несут информацию о логическом размещении этих CID. С каждым значением динамического параметра связан свой указатель CIDFP. Если какому-либо блоку SIB для выполнения его функций требуются CID, то через параметры SSD назначается соответствующий CIDFP. Указатели полей обозначаются как «CIDFP-xxxx», где «xxxx» – названия (имена) требуемых данных. Например, если блоку TRANSLATE нужны для пересчета данные CID с информацией о вызываемой стороне, то соответствующее поле данных SSD блока TRANSLATE определяет, что нужная информация может быть найдена при помощи CIDFP – Значение Фильтра.

В примере с блоком TRANSLATE одному атрибуту услуги может потребоваться пересчет номера вызывающей линии, а другому атрибуту – пересчет номера вызываемой линии. В обоих случаях данные, необходимые этому блоку, специфицируются полем идентификатора вызывающей линии, однако значение указателя CIDFP-Значение Фильтра изменяется. В первом случае значение CIDFP-Значение Фильтра устанавливается равным номеру вызывающей стороны, тогда как во втором случае CIDFP-Значение Фильтра устанавливается равным номеру вызываемой стороны.

Указатель CIDFP, специфицированный для услуги или атрибута, фиксирован и неизменен при всех обращениях к этой услуге или атрибуту, а значение динамического параметра, к которому он адресует, изменяется с каждым новым обращением. При любом обращении к услуге или атрибуту соответствующий CIDFP доступен всем блокам SIB, из которых составляется цепь, формирующая эту услугу или атрибут.

2.2.3 Стадия 1 описания блоков SIB

Каждый блок SIB описывается стандартно по определенному шаблону, который включает в себя следующие поля:

ОПРЕДЕЛЕНИЕ:	Описание блока SIB с точки зрения создания услуги.
ВЫПОЛНЯЕМЫЕ ДЕЙСТВИЯ:	Описание действий, выполняемых блоком SIB.
ВОЗМОЖНОЕ ПРИМЕНЕНИЕ:	Примеры услуг, где данный SIB может быть использован.
ВХОД:	Вход в каждый SIB специфицируется тремя элементами: - один логический старт; - данные SSD; - данные CID.
ВЫХОД:	Выход из каждого SIB специфицируется двумя элементами: - одно или несколько логических завершений; - данные CID, которые определяют полученные в результате выполнения этого SIB параметры, специфические для данного вызова и нужные другим SIB или BCP, чтобы завершить предоставление услуги.
ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ:	Графическое представление, используемое для описания входа, работы и выхода блока SIB, иллюстрирует рис. 2.2.1. Каждый SIB имеет один логический старт и одно или несколько логических завершений. Соответствующие логические потоки изображены слева и справа сплошными линиями со стрелками; надписи рядом со линиями идентифицируют каждый из этих потоков. Параметры SSD изображены сверху пунктирными линиями со стрелками и сопровождаются надписями сбоку от линий. Аналогичным образом внизу показаны параметры CID, причем входные и выходные CID изображены отдельно друг от друга.
SDL - ДИАГРАММА:	Эта диаграмма дает графическое представление стадии 1 описания SIB с использованием SDL-макро в соответствии с правилами, определенными в рекомендации ITU-T Z.100.

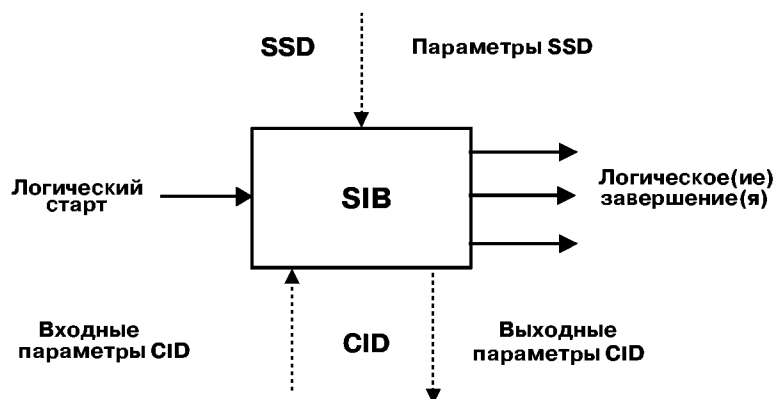


Рис. 2.2.1 Графическое представление блока SIB

2.2.3.1 Блоки, определенные ITU-T

БЛОК ALGORITHM. Первоначально этот блок предназначался для выполнения самых разных математических операций. Однако опыт внедрения продиктовал необходимость сужения математических функций блока и, соответственно, области его применения. В результате, алгоритм, выполняемый данным блоком, сведен к увеличению или к уменьшению целочисленного значения CID на заданное целое число (инкремент/декремент). Блок ALGORITHM может применяться в услугах Mass calling и Televoting.

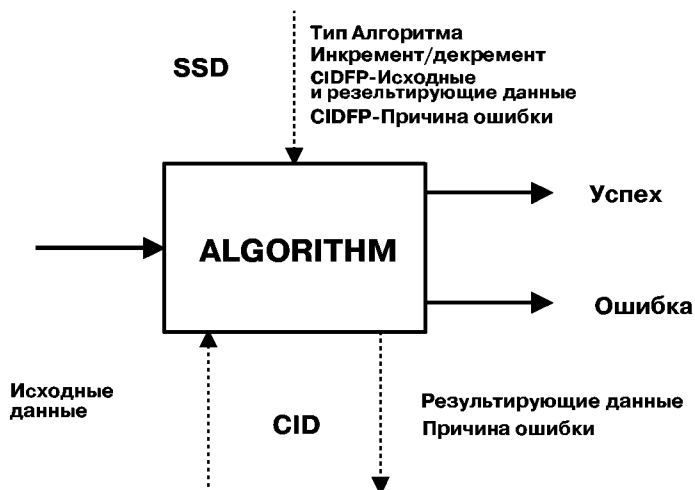


Рис. 2.2.2 Блок ALGORITHM

Данные SSD блока содержат тип алгоритма математической обработки (увеличить/уменьшить), целочисленное значение (инкремент или декремент), на которое исходные данные будут увели-

ваться или уменьшаться, указатель поля входных данных CID, куда помещаются исходные данные и, затем, – результирующие данные, а также указатель поля выходных данных CID, куда будет помещена возможная причина ошибочного выполнения блока. Входные данные CID блока содержат параметр, который хранит данные, подлежащие математической обработке, а выходные данные CID содержат результат выполнения соответствующего математического алгоритма и возможную причину ошибочного выполнения блока. Графическое представление блока приведено на рисунке 2.2.2.

БЛОК AUTHENTICATE. Этот блок присутствует только в версии ITU-T и, в соответствии со своим названием, выполняет функции проверки подлинности, иными словами, проверяет, является ли потенциальный пользователь услуги тем, за кого он себя выдает. Блок позволяет установить три опции механизма проверки подлинности:

- «простой механизм», предусматривающий проверку только имени пользователя и пароля;
- «внешний механизм», предусматривающий привлечение внешних средств проверки посредством указателя на соответствующую программу;
- отсутствие проверки.

Блок AUTHENTICATE может использоваться во всех услугах. Параметры SSD блока содержат несколько полей:

- *Имя аутентификации* является необязательным полем и присутствует, только когда установлен «простой механизм» аутентификации; это поле используется для обнаружения местонахождения объекта, содержащего пароль аутентификации, для установления с ним разрешенной связи; значение данного поля, равное нулю, указывает, что имя аутентификации может изменяться и поэтому должно предоставляться входными данными CID;
- *Пароль аутентификации* также является необязательным полем и присутствует, только когда установлен «простой механизм» аутентификации; это поле предоставляет данные, которые должны соответствовать имени аутентификации (условие, разрешающее связь между объектами); значение данного поля, равное нулю, указывает, что имя аутентификации может изменяться и поэтому должно предоставляться входными данными CID;
- *Идентификатор механизма аутентификации* определяет, какой из вышеуказанных механизмов будет применен при установлении разрешенной связи;
- *Указатель поля данных CIDFP-Причина ошибки* определяет, в какое поле выходных данных CID будет помещена информация о причине ошибочного выполнения блока;

- *Указатель поля данных CIDFP-Идентификатор разрешенной связи* определяет, в какое поле выходных данных CID будет помещено значение идентификатора разрешенной связи;
- *Указатель поля данных CIDFP-Имя аутентификации* определяет, какие входные данные CID должны использоваться в качестве имени аутентификации;
- *Указатель поля данных CIDFP-Пароль аутентификации* определяет, какие входные данные CID должны использоваться в качестве пароля аутентификации.

Входные данные CID содержат два необязательных поля: *Имя аутентификации* и *Пароль аутентификации*, которые применяются только при простом механизме аутентификации и нулевых значениях одноименных полей данных SSD.

Выходные данные CID содержат два поля: *Причина ошибки*, указывающая условия, при которых произошло ошибочное выполнение блока, и *Идентификатор разрешенной связи*, посредством которого будут передаваться соответствующие операции.

Блок имеет два типа логических завершений: успешное и ошибочное. Графическое представление блока приведено на рисунке 2.2.3.

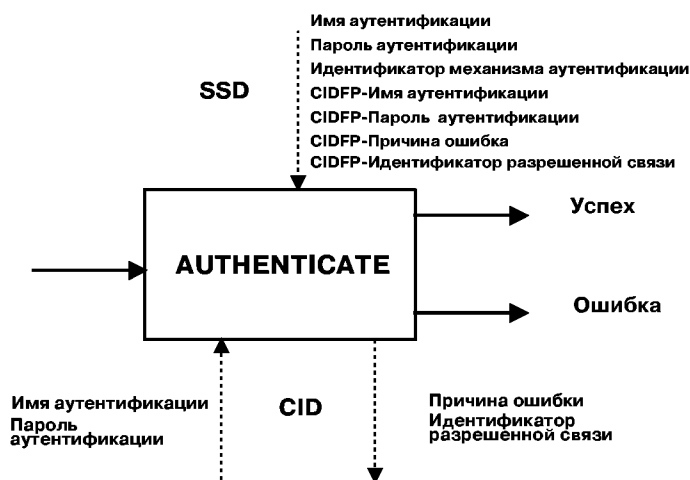


Рис. 2.2.3 Блок AUTHENTICATE

БЛОК CHARGE. Данный блок выполняет одну из самых важных функций, обеспечивающую получение доходов от предоставления услуг. По этой причине функции данного блока многократно обсуждались разными организациями, связанными со стандартизацией. Функции начисления платы, выполняемые данным блоком, дополня-

ют аналогичные функции базового процесса обслуживания вызова (BCP). В рекомендации ITU-T Q.1213 перечислены некоторые ограничения, касающиеся функций блока CHARGE, а именно то, что стандартом не определяется выходной формат данных, что не обязательно определяется вся информация, которая может понадобиться для начисления платы, и что процесс выставления счетов абоненту находится вне рамок стандарта.

Блок CHARGE предназначен для начисления платы за пользование ресурсами Интеллектуальной сети и может быть использован несколько раз одной и той же услугой/атрибутом услуги. Плата за одновременно или последовательно возникающие вызовы может начисляться на один и тот же счет. В Интеллектуальной сети плата может начисляться за использование каналов, специализированных ресурсов IP (фразы автоинформатора, хранение речевых сообщений), вычислительных ресурсов SCP. Информация о начисленной плате может быть направлена на банковский счет (вызывающего абонента, вызванного абонента или указанный в телефонной карте) или в таксофон. Данные SSD блока CHARGE содержат следующие поля:

- *Количество счетов* для начисления платы;
- *Счет*, который специфицируется двумя параметрами: номер счета и процентное распределение платы по счетам, при этом суммарное значение должно равняться 100%;
- *Тип ресурса*, за использование которого будет начисляться плата;
- *Единицы* – определяют дополнительную плату за использование ресурса специфицированного типа;
- *Идентификатор услуги/атрибута*, за пользование которой(ым) начисляется плата;
- *Указатель поля данных CIDFP-Импульсы*, который определяет, какие данные CID задают параметры тарифных импульсов;
- *Указатель поля данных CIDFP-Причина ошибки*, который определяет место в выходных данных CID, куда будет помещена информация о причине ошибочного выполнения блока.

Входные данные CID содержат информацию о номере линии, к которой относится начисляемая плата. Это может быть вызывающая линия, специально набранный номер или адрес пункта назначения. Кроме того, данные CID содержат номер счета для начисления платы, которым может быть номер, введенный в процессе обслуживания вызова (номер телефонной или кредитной карты), а также информацию, которая указывает, что для начисления платы используются тарифные импульсы. Блок имеет два типа логических заверше-

ний: Успех и Ошибка. Сопровождающие выходные данные CID содержат сведения о причине ошибки, т.е. определяют те условия, из-за которых произошла ошибка во время выполнения блока. Графическое представление блока приведено на рисунке 2.2.4.

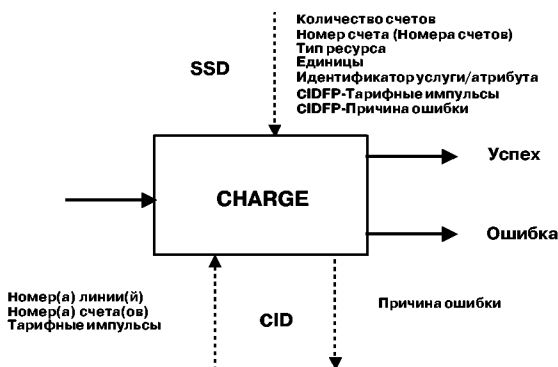


Рис. 2.2.4 Блок CHARGE

БЛОК COMPARE. Данный блок производит сравнение выбранного из входных данных значения с эталонной величиной; при этом возможны три варианта результата:

- выбранное значение больше, чем эталонная величина;
- выбранное значение меньше, чем эталонная величина;
- выбранное значение равно эталонной величине.

Блок COMPARE может быть использован, например, чтобы установить, не превышает ли текущее количество вызовов максимально разрешенную величину. Кроме того, этот блок позволяет определить, как соотносится текущее время со временем, указанным пользователем, чтобы можно было принять решение, зависящее от времени. Сравнение может производиться по следующим критериям: время суток, день недели или день года. Эталонной величиной при этом являются заданные пользователем значения. Блок COMPARE может использоваться в услугах, где требуется маршрутизация в зависимости от времени, а также в услуге Call completion to busy subscriber. Данные SSD блока COMPARE содержат следующие поля:

- *Тип сравнения* (значение идентификатора или время);
- *Указатель поля CIDFP-Данные*, который определяет, куда помещены данные для сравнения;
- *Эталонная величина* – специфицирует величину, с которой будет производиться сравнение;

- *Указатель поля CIDFP-Причина ошибки*, который указывает то место в выходных данных CID, куда будут помещены сведения о причине ошибки выполнения блока.

Блок имеет четыре типа логических завершений: *Больше, чем эталонная величина*; *Меньше, чем эталонная величина*; *Равно эталонной величине* и *Ошибка*. Сопровождающие выходные данные CID содержат сведения о причине ошибки, т.е определяют те условия, из-за которых произошла ошибка во время выполнения блока. Графическое представление блока приведено на рисунке 2.2.5.

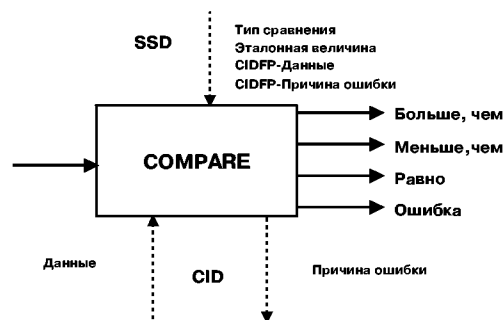


Рис. 2.2.5 Блок COMPARE

БЛОК DISTRIBUTION. Этот блок является обобщением блока COMPARE. Данный SIB сначала выполняет некий указанный пользователем алгоритм, чтобы определить, по какому из имеющихся логических завершений будет продолжена обработка вызова логикой услуги. Количество логических завершений зависит от типа применяемого алгоритма. Предусмотрены четыре типа алгоритмов: распределение в соответствии с установленным процентным соотношением, циклическое распределение, распределение в зависимости от времени суток и распределение в зависимости от дня недели. Блок DISTRIBUTION может использоваться в услугах Mass calling, Televoting и Freephone.

Данные SSD блока содержат в качестве полей: *Тип алгоритма*, *Число логических завершений*, *Параметры алгоритма* и *Указатель поля CIDFP-Причина ошибки*. Данный блок не содержит входных данных CID. Блок имеет $(n+1)$ логических завершений, где n берется из соответствующего параметра SSD. Логическое завершение *Ошибка* является $(n+1)$ -ым. Сопровождающие выходные данные CID содержат сведения о причине ошибки, т.е определяют те условия, из-за которых произошел сбой во время выполнения блока. Графическое представление блока приведено на рисунке 2.2.6.

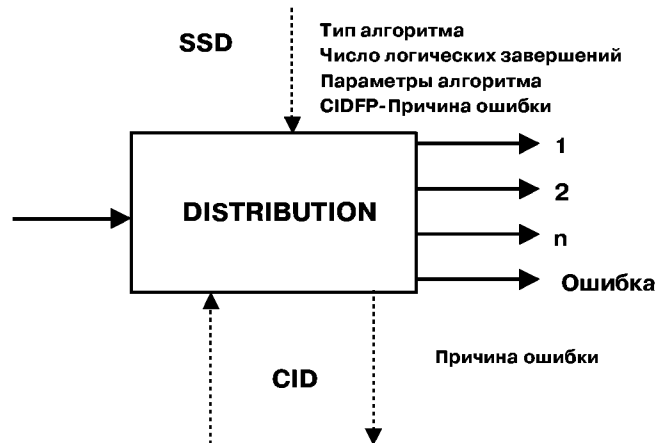


Рис. 2.2.6 Блок DISTRIBUTION

БЛОК LIMIT. Некоторые атрибуты услуг требуют, чтобы Интеллектуальной сетью обрабатывалась только определенная часть из всех вызовов, связанных с обращением к услуге IN. Эти функции реализуются блоком LIMIT с целью предотвратить перегрузку ресурсов сети. Конкретный алгоритм выполнения данной процедуры определяется данными SSD, которые содержат поля, определяющие *Тип алгоритма* ограничения, *Указатель поля входных данных CIDFP-Файл*, где находится текущее содержимое счетчика вызовов, а также *Указатель поля выходных данных CIDFP-Причина ошибки*, куда будет помещена информация о причине возможной ошибки выполнения блока. Обычно используется один из двух типов алгоритмов:

- внутри каждого интервала времени продолжительностью в Q секунд обработка вызовов разрешается в течение S секунд (интенсивность потока вызовов, требующих услуги IN, ожидается такой, что число вызовов, поступивших в течение S секунд, займет ресурсы процессоров на время $Q > S$ секунд);
- из P поступающих вызовов разрешается обработка только N вызовов ($N < P$) (например, за секунду поступает P вызовов, требующих услуги IN, а процессоры могут обработать за секунду только $N < P$ вызовов).

Блок имеет три типа логических завершений: *Пропустить*, *Запретить* и *Ошибка*.

Блок LIMIT в связке с другими блоками, например, с блоком COMPARE, может обеспечить реализацию процедуры ограничения числа вызовов в зависимости от времени. Блок LIMIT может использоваться в услугах Mass calling, Televoting и Freephone. Графическое представление блока приведено на рисунке 2.2.7.

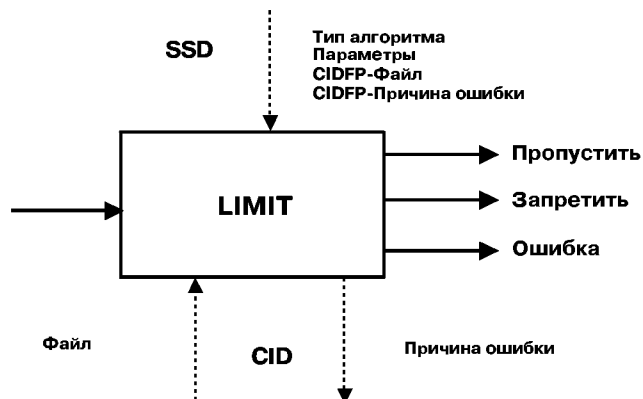


Рис. 2.2.7 Блок LIMIT

БЛОК LOG CALL INFORMATION. Этот SIB, применяемый во всех услугах, производит запись информации о вызове в файл с целью ее дальнейшего использования системой технического обслуживания сети. Тип информации, подлежащей записи, определяется данными SSD, в которых также содержится имя файла, куда нужно записывать информацию, и список указателей полей необходимых данных CID. Информация о вызове может включать в себя разные типы данных CID, например, момент возникновения вызова, момент отбоя, момент установления соединения, набранные цифры номера, номер пункта назначения; дополнительно набранные цифры (например, номер телефонной карты), номер вызывающей линии, длительность ожидания, характеристики средств доставки информации, причины ошибок и др. Данные SSD блока содержат следующие поля:

- *Указатель поля данных CIDFP-Данные*, который определяет, какие именно данные CID будут использоваться при записи информации о вызове;
- *Имя файла для записи*;
- *Указатель поля данных CIDFP-Идентификатор разрешенной связи*, который определяет, какие именно данные CID должны использоваться в качестве идентификатора разрешенной связи;
- *Указатель поля данных CIDFP-Причина ошибки*.

Входные данные CID блока содержат два поля:

- *Данные*, которые определяют, какая именно информация о вызове должна будет записываться в файл;
- *Идентификатор разрешенной связи* предоставляет характеристики установленной разрешенной связи, посредством которой будет происходить обмен операциями при выполнении блока.

Выходные данные CID содержат единственное поле – *Причина ошибки*, определяющее условие, из-за которого произошел сбой при выполнении блока. Блок имеет два типа логических завершений: *Успех* и *Ошибка*. Графическое представление блока приведено на рисунке 2.2.8.

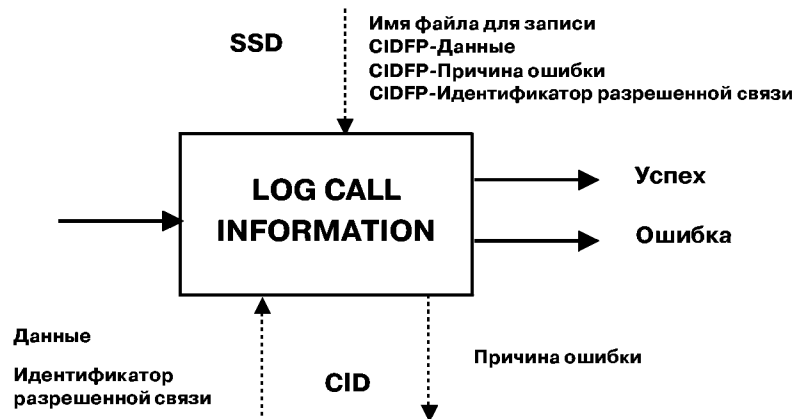


Рис. 2.2.8 Блок LOG CALL INFORMATION

БЛОК QUEUE. Блок QUEUE предусматривает установку вызовов в очередь для ожидания освобождения занятых (недоступных) ресурсов и выбор этих вызовов из очереди по мере освобождения ресурсов. Этот блок выполняет все необходимые функции, связанные с возможностью ожидания, а именно:

- разрешает немедленное обслуживание вызова, если соответствующие ресурсы доступны;
- помещает вызов в очередь, если ресурсы недоступны;
- (опция) передает ожидающему абоненту речевую или музыкальную фразу;
- разрешает продолжить обслуживание очередного вызова при освобождении ресурсов.

В связи со сложностью реализации блока QUEUE в Рекомендации ITU-T Q.1213 приводится SDL-диаграмма, описывающая алгоритм его функционирования. Процесс, выполняемый этим SIB, проходит следующие шаги.

1. Отмечает вызов, как установленный в очередь.
2. Запускает таймер, ограничивающий время пребывания в очереди, поскольку недопустимо, чтобы вызов находился в очереди бесконечно долго.
3. Увеличивает на единицу «показание» счетчика занятых мест в очереди.

4. Как опция, запускает процедуру передачи вызывающему пользователю речевой фразы.
5. Запрашивает у исходящей станции информацию о состоянии абонентской линии, т.е. не повешена ли трубка. В случае, если трубка повешена, исходящая станция должна послать обратное сообщение, чтобы сеть смогла освободить ресурсы.
6. Ожидание. В этой точке происходит приостановка процесса обслуживания вызова до тех пор, пока не произойдет одно из трех следующих событий:
 - Сработал таймер. В этом случае, все ресурсы, которые связаны с обслуживанием вызова, освобождаются.
 - Отказ от ожидания (отбой ожидающего пользователя). В этом случае, как и в предыдущем, все ресурсы, связанные с обслуживанием вызова, освобождаются.
 - Ресурс доступен. В этом случае вызов выводится из очереди, а исходящая АТС получает адрес освободившегося ресурса.

Заметим, что весь обмен сообщениями, о котором шла речь, происходит только между исходящей станцией и функцией управления услугой. Входящая сторона в обработке вызова, требующего услугу IN, не участвует.

Данные SSD блока QUEUE определяют доступную длину очереди, максимальное число вызовов одновременно обслуживаемых ресурсом, и максимально разрешенное время пребывания вызова в очереди. Входные данные CID блока QUEUE содержат метку соединения, устанавливаемого в процессе обслуживания вызова. Блок имеет пять типов логических завершений:

- *Ресурс доступен;*
- *Отказ от ожидания;*
- *Сработал таймер, лимитирующий время ожидания;*
- *Очередь заполнена;*
- *Ошибка.*

Блок QUEUE может применяться во всех услугах, использующих атрибут ожидания обслуживания. Графическое представление блока приведено на рисунке 2.2.9.

БЛОК SCREEN. Блок SCREEN производит сравнение идентификатора (например, PIN-кода, введенного пользователем, номеров вызывающей и вызываемой сторон) с заданным списком. Этот SIB может использоваться для проверки идентификации пользователя или его PIN-кода с последующим разрешением или запретом доступа к запрашиваемым функциям, а также для просеивания исходящих

или входящих вызовов по признаку сетевых адресов. Многократное использование блока SCREEN в соединении с другими блоками, такими как TRANSLATE и COMPARE, может привести к более сложным алгоритмам просмотра. Блок может быть использован в услугах переадресации (по условию занятости или отсутствия ответа вызываемого абонента), просмотра списка исходящих и/или входящих вызовов, а также в услугах с отнесением платы на счет или с оплатой по кредитной карте.

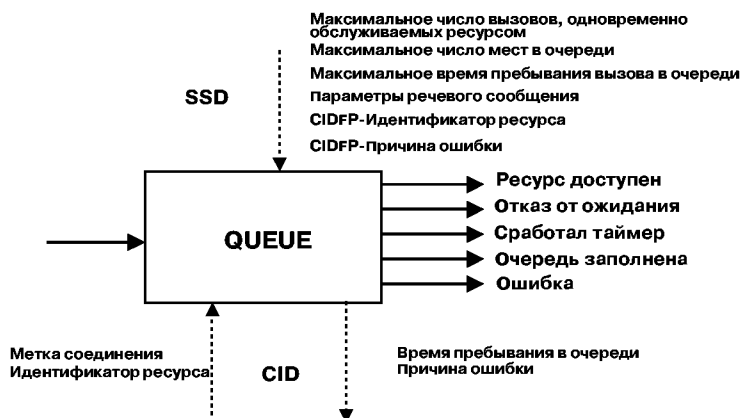


Рис. 2.2.9 Блок QUEUE

Блок имеет три типа логических завершений: *Соответствие*, *Несоответствие* и *Ошибка*. Сопровождающие выходные данные CID содержат сведения о причине ошибки, т.е. определяют те условия, из-за которых произошел сбой во время выполнения блока. Графическое представление блока приведено на рисунке 2.2.10.

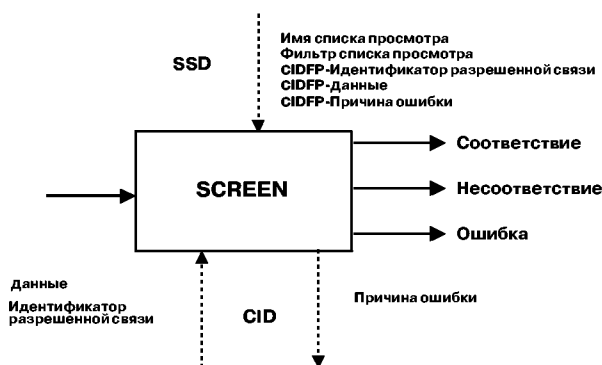


Рис. 2.2.10 Блок SCREEN

БЛОК SERVICE DATA MANAGEMENT (SDM). Этот блок предоставляет возможности запроса, замены и изменения специфических данных об абоненте услуги, хранящихся в сетевой базе данных, и выполняет следующий набор действий:

- добавление объектов в базу данных и удаление объектов из базы данных;
- отыскание и изменение значений атрибутов объектов в базе данных;
- возврат к значениям атрибутов, принятым «по умолчанию»;
- увеличение или уменьшение значений атрибутов на заданную величину.

Данные SSD определяют файл абонентских данных, элемент данных в файле, а также тип действия, которое будет выполняться над этими данными, и содержат указатели полей данных CID на объекты абонентской базы данных:

- *Имя объекта*, определяющее в файле абонентских данных объект, который будет использован при выполнении блока;
- *Действие*, определяющее операции, которые будут производиться над абонентскими данными. Разрешенными являются следующие типы операций: замена существующего элемента данных абонентского файла новым элементом, запрос (поиск) элемента данных абонентского файла и копирование его в соответствующее поле выходных данных CID, увеличение (инкремент) или уменьшение (декремент) значения элемента данных в абонентском файле, присвоение текущему значению атрибута абонентских данных значения, принятого по умолчанию, добавление в базу данных нового объекта и удаление объекта из базы данных;
- *Индикатор атрибута* указывающий на тот объект в файле абонентских данных, по отношению к которому будут выполнены действия;
- *Значение инкремента/декремента*;
- *Указатель поля данных CIDFP-Информация*, определяющий, какое именно поле данных CID должно быть использовано для размещения информации;
- *Указатель поля данных CIDFP-Атрибут*, определяющий, какие именно данные CID должны быть использованы в качестве индикатора атрибута абонентских данных;
- *Указатель поля данных CIDFP-Запрошенные данные*, определяющий, в какое поле выходных данных CID будут помещены запрошенные данные;
- *Указатель поля данных CIDFP-Идентификатор разрешенной связи*;
- *Указатель поля данных CIDFP-Причина ошибки*.

Входные данные CID содержат новое значение элемента абонентских данных для занесения его в абонентскую базу данных, а также

определяют элемент базы данных, над которым будут выполняться действия.

SIB имеет два логических завершения: *Успех* и *Ошибка*. Здесь следует отметить, что если данные, полученные от пользователя (с помощью блока USER INTERACTION), являются некорректными, то ветвь логического завершения *Ошибка* может привести к новому сеансу взаимодействия с пользователем. Пользователь получает сведения о причине ошибки и имеет возможность ввести другие данные.

Данный SIB может использоваться в услугах Call forwarding, UPT и других, где требуется управление данными пользователя. Графическое представление блока приведено на рисунке 2.2.11.

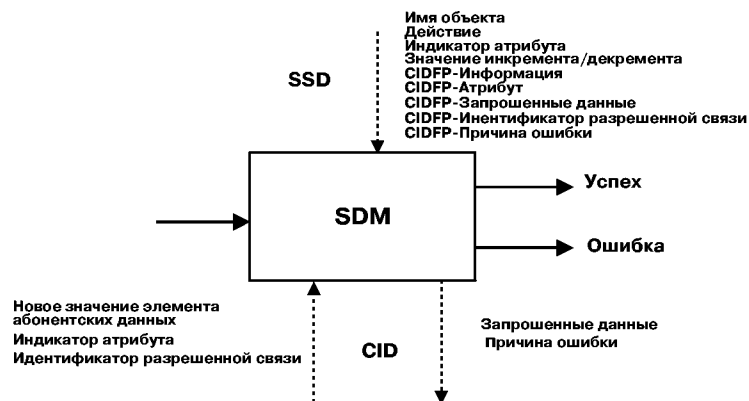


Рис. 2.2.11 Блок SERVICE DATA MANAGEMENT (SDM)

БЛОК STATUS NOTIFICATION. Этот блок предоставляет возможность запросить информацию о текущем состоянии или об изменении состояния ресурсов сети. Под ресурсами понимаются абонентские или соединительные линии, состояние которых может быть «свободны» или «заняты». Данный блок может использоваться в услугах/атрибутах услуг CCBS, Freephone, Call transfer, Call distribution. Возможны четыре типа действий, связанных с получением информации о статусе:

1. Запрос сведений о статусе ресурса. Ответом является извещение о текущем статусе этого ресурса.
2. Ожидание нужного статуса ресурса. Ответ поступает тогда, когда этот ресурс перейдет в нужное состояние (например, из состояния «занято» в состояние «свободно»), или немедленно, если ресурс уже находится в нужном состоянии. До перехода ресурса в нужное состояние процесс, в контексте которого выполняется данный SIB, приостанавливается.

3. Непрерывный мониторинг – отслеживание и запись изменений статуса. Ответ содержит сведения обо всех изменениях статуса ресурса.
4. Отмена непрерывного мониторинга.

Данные SSD включают в себя спецификацию одного из вышеперечисленных типов действий, спецификацию ресурса (абонентская линия, соединительная линия), данные для настройки таймера, ограничивающего продолжительность мониторинга, индикатор файла для записи текущего статуса ресурса, а также указатели полей выходных данных CID, куда будет помещена результирующая информация о статусе ресурса и о причине ошибки выполнения блока. Входных данных CID блок не имеет. Блок имеет три типа логических завершений: *Успех*, *Сработал Таймер* и *Ошибка*. Графическое представление блока приведено на рисунке 2.2.12.

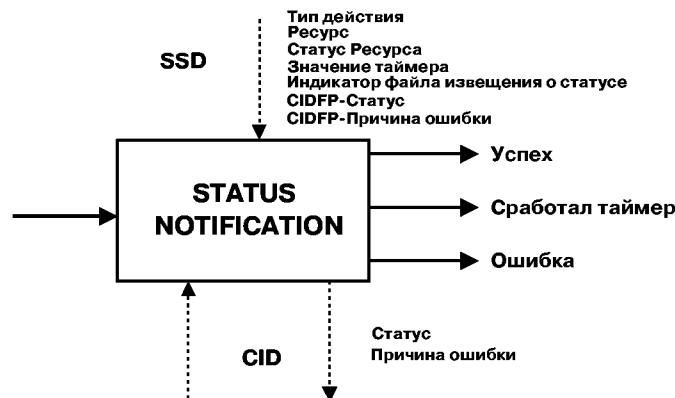


Рис. 2.2.12 Блок STATUS NOTIFICATION

БЛОК TRANSLATE. Блок TRANSLATE выполняет основную функцию, поддерживающую возможности из набора CS-1, которая требуется для всех ключевых услуг: Freephone, User defined routing, VPN, UPT, Call forwarding, Call transfer. Входная информация пересчитывается (либо непосредственно, либо с использованием других данных CID, например, идентификации вызывающей линии) в выходную информацию, которая представляет собой физический адрес, необходимый для маршрутизации. В совокупности с другими блоками, например, с таким как COMPARE, блок TRANSLATE может обеспечить маршрутизацию в зависимости от времени суток. Данные SSD блока TRANSLATE содержат следующие поля:

- *Имя объекта* определяет, где размещаются данные для пересчета;
- *Фильтр пересчета* определяет элементы объекта данных и фильтры, которые должны применяться к элементам при пересчете. Значения элементов хранятся в указателе поля данных CIDFP-Значение(я) фильтра;

- *Пересчитанный элемент* определяет какой(ие) элемент(ы) объекта данных должны возвращаться блоком в качестве значений пересчитанных данных;
- *Указатель(и) поля CIDFP-Значение(я) фильтра* определяет, какие именно данные CID должны использоваться в качестве информации для пересчета;
- *Указатель поля CIDFP-Пересчитанная информация* определяет, где в выходных данных CID будет размещена пересчитанная информация;
- *Указатель поля CIDFP – Идентификатор разрешенной связи*;
- *Указатель поля CIDFP-Причина ошибки*.

Входные данные CID содержат поля *Значение(я) фильтра(ов)* и *Идентификатор разрешенной связи*. Выходные данные CID содержат поле *Пересчитанная информация* и поле *Причина ошибки*. Блок имеет два типа логических завершений – *Успех* и *Ошибка*. Завершение *Ошибка* может содержать сведения о причине, по которой пересчет оказался невозможен (например, проблемы во внешней базе данных). Графическое представление блока приведено на рисунке 2.2.13.

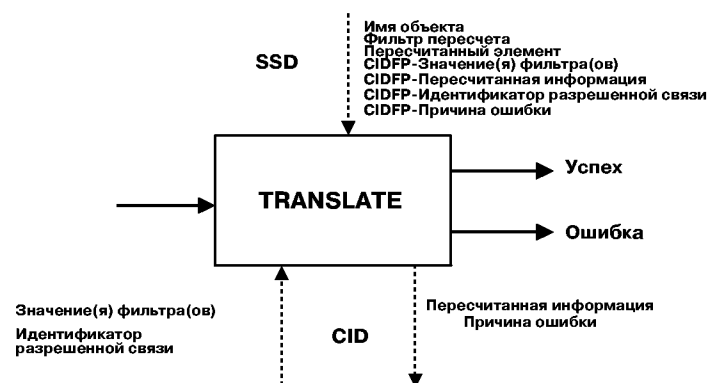


Рис. 2.2.13 Блок TRANSLATE

БЛОК USER INTERACTION. Данный блок обеспечивает возможность обмена информацией между сетью и конечным пользователем (либо вызывающим, либо вызываемым). Сеть может передавать пользователю информацию в виде DTMF сигналов, речевых фраз или акустических сигналов типа «Ответ станции» или «Занято», а также принимать от пользователя необходимую информацию, например, в виде DTMF сигналов. Многократное взаимодействие сети с пользователем происходит при предоставлении большинства услуг. В некоторых случаях услуга не предполагает никаких иных действий,

кроме взаимодействия с конечным пользователем. Для примера рассмотрим услугу, предоставляемую некой компанией, которая продает товары, перечисленные в каталоге. Когда пользователь вызывает услугу, ему со стороны сети предлагается ввести номер товара согласно каталогу и номер кредитной карты. Вся принимаемая информация, включая телефонный номер этого пользователя, записывается сетью. Затем компания доставляет товар пользователю по адресу, определенному на основании его телефонного номера.

Данные SSD блока USER INTERACTION содержат все параметры диалога с пользователем, которые определяют:

- тип сообщения, передаваемого от сети пользователю;
- необходимость повторения этого сообщения; если сообщение необходимо повторять, то определяется длительность (в секундах) интервала между повторениями, максимальное число повторений и максимальное время, отведенное на передачу сообщения, включая его повторения;
- условия приема информации от пользователя (формат данных, может ли пользователь начинать передачу своей информации до прекращения передачи сообщения со стороны сети).

SIB имеет два типа логических завершений: Успех и Ошибка. Входные данные CID блока USER INTERACTION идентифицируют пользователя. Выходные данные CID содержат информацию, полученную от пользователя или информацию о причине ошибки (некорректные параметры, несоответствующее состояние процесса обслуживания вызова, преждевременный отказ от обслуживания, истечение времени, отведенного на прием информации, некорректное число введенных цифр и др.).

Блок USER INTERACTION может использоваться в большинстве услуг из набора CS-1. Графическое представление блока приведено на рисунке 2.2.14.

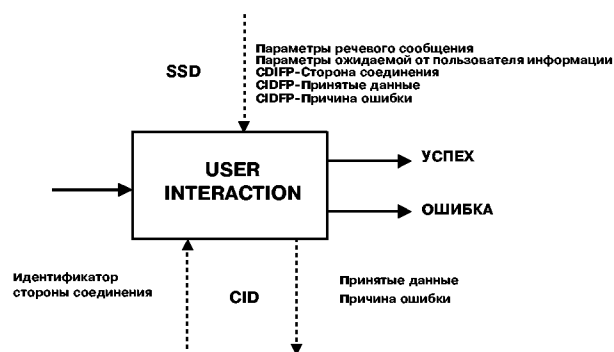


Рис. 2.2.14 Блок USER INTERACTION

БЛОК VERIFY. Блок VERIFY в цепочке блоков SIB обычно следует за блоком USER INTERACTION, когда информация от вызывающей стороны уже получена. Поэтому данный SIB используется в тех же услугах, что и блок USER INTERACTION. Роль этого SIB заключается в «грамматическом разборе» входной информации на предмет оценки ее синтаксической корректности. Теоретически, некоторые грамматические правила могут ультимативно определяться данными SSD, однако в настоящее время эти правила ограничиваются описанием алфавитно-цифровых строк, имеющих ограниченную длину и достаточно простой формат, с использованием символов-разделителей типа # или *.

Блок VERIFY имеет три логических завершения: *Соответствие*, указывающее на положительный результат проверки, *Несоответствие* и *Ошибка*. В последнем случае сопровождающие выходные данные CID идентифицируют причину ошибки (например, некорректные значения параметров). Графическое представление блока приведено на рисунке 2.2.15.

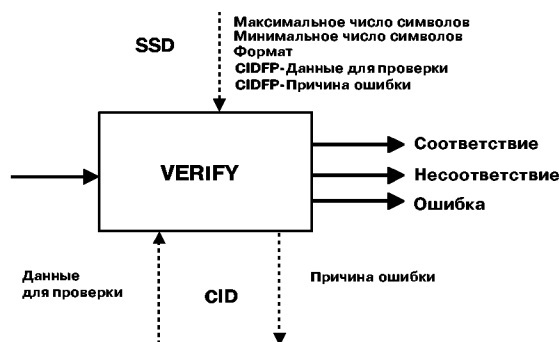


Рис. 2.2.15 Блок VERIFY

2.2.3.2 Блоки, определенные ETSI

БЛОК CONNECT. Блок CONNECT привлекается в контексте обслуживаемого вызова и служит для установления соединения между вызывающей стороной и стороной, адрес которой указан в параметрах CID блока CONNECT. После выполнения действий данным блоком, блок BCP возобновляет процесс установления соединения с новыми данными. Блок CONNECT может использоваться во всех услугах CS-1.

Данные SSD блока содержат указатели полей CID, где размещаются, например, такие данные как адрес пункта назначения, идентификатор (номер) вызывающей стороны, информация о категории вызывающей стороны и т.д. Входные данные CID содержат сетевой адрес пункта назначения, сетевой адрес вызывающей стороны, характеристики средств доставки информации и категорию вызываю-

щей стороны. Блок имеет два типа логических завершений: Успех и Ошибка. В последнем случае сопровождающие выходные данные CID идентифицируют причину ошибки: неправильный адрес пункта назначения или неправильный идентификатор вызывающей стороны. Графическое представление блока приведено на рисунке 2.2.16.

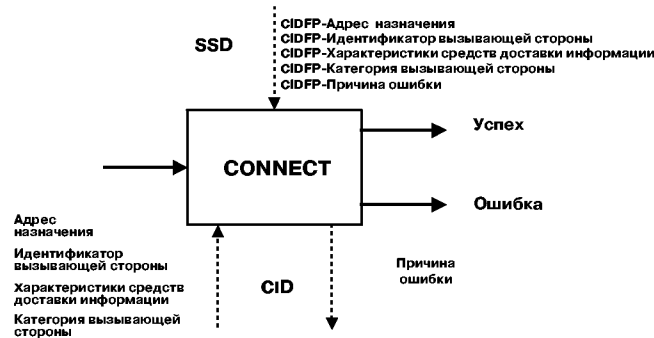


Рис. 2.2.16 Блок CONNECT

БЛОК CONTINUE. Основной функцией этого блока является продолжение базового процесса обслуживания вызова, начиная от точки, где он был приостановлен. Блок CONTINUE указывает ВСР, что процесс следует продолжить без изменения данных. Блок CONTINUE может использоваться во всех услугах CS-1.

Данный блок не содержит данных SSD, а также входных и выходных данных CID. Блок имеет один тип логического завершения – Успех, который означает, что указание продолжить процесс обслуживания вызова передано ВСР. Графическое представление блока приведено на рисунке 2.2.17.

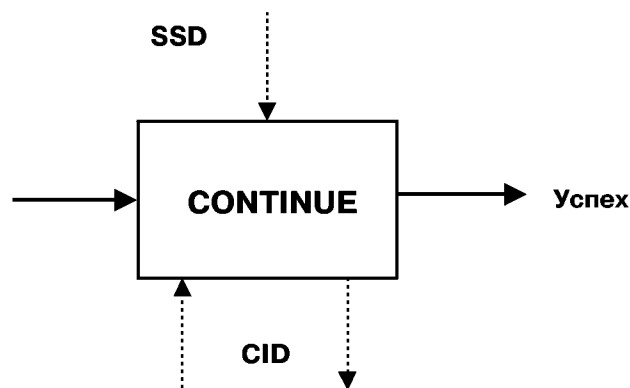


Рис. 2.2.17 Блок CONTINUE

БЛОК DISCONNECT RESOURCE. Блок указывает ВСР на необходимость освободить все задействованные для обслуживания данно-

го вызова специализированные ресурсы. Использование данного блока требуется в большинстве услуг CS-1.

Блок не содержит данных SSD, а также входных и выходных данных CID. Блок имеет один тип логического завершения – Успех, который означает, что запрос освобождения ресурсов передан ВСП. Графическое представление блока приведено на рисунке 2.2.18.

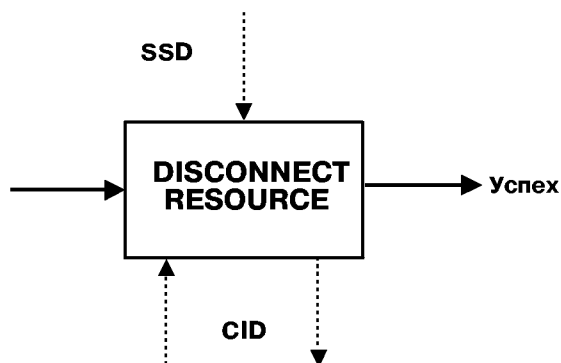


Рис. 2.2.18 Блок DISCONNECT RESOURCE

БЛОК EDP INFO. Функцией блока является предоставление информации от точки обнаружения события (EDP) при возникновении самого события. Блок используется в большинстве услуг CS-1.

Данные SSD блока EDP INFO содержат указатели полей выходных данных CID, которые специфицируют, какие именно поля CID содержат специфическую для данного события информацию (*CIDFP-Специфическая информация*), идентификатор участника связи (*CIDFP-Идентификатор участника связи*), а также данные о типе сообщения (*CIDFP-Тип сообщения*), который, в свою очередь, определяет тип точки EDP: запрос или уведомление. Блок не имеет входных данных CID. Для блока определен единственный тип логического завершения *Успех*, который означает, что информация от точки обнаружения события предоставлена, а сама точка деактивизирована. Графическое представление блока приведено на рисунке 2.2.19.

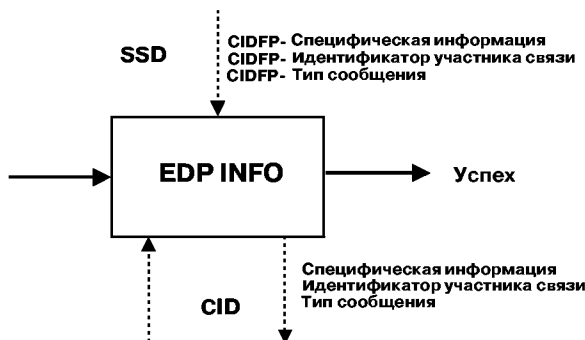


Рис. 2.2.19 Блок EDP INFO

БЛОК EDP REQUEST. Функцией этого блока является активизация точек обнаружения событий в базовом процессе BCP. Запрошенное событие возникает после возврата в BCP в точке POR. Блок EDP REQUEST может быть использован в большинстве услуг CS-1. Данные SSD блока EDP REQUEST содержат два поля:

- Типы EDP, которые нужно активизировать;
- Указатель поля выходных данных *CIDFP-Причина ошибки*, куда будут помещены сведения о причине ошибочного выполнения блока.

Если типы указанных в параметре точек представляют собой *O_No_Answer* (исходящая часть, отсутствие ответа) или *T_No_Answer* (входящая часть, отсутствие ответа), то в параметре указывается значение таймера, срабатывание которого означает отсутствие ответа. Если же тип указанной точки – *CollectedInfo* (информация накоплена), то в параметре дополнительно указывается число цифр, которое должно быть накоплено.

В блоке отсутствуют входные данные CID и определены два типа логических завершений: *Успех*, который означает, что запрос активизации точки выполнен, и *Ошибка*. В выходных данных CID содержатся сведения о причине ошибочного выполнения блока. Графическое представление блока приведено на рисунке 2.2.20.

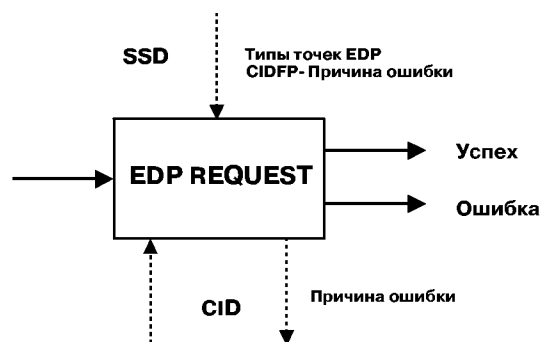


Рис. 2.2.20 Блок EDP REQUEST

БЛОК INITIATE CALL. Данный блок используется для организации соединения с определенным пунктом назначения по инициативе логики услуги. На момент привлечения блока связь с BCP отсутствует (имеется только управляющая связь). BCP привлекается после выполнения блока INITIATE CALL. Блок может использоваться во всех услугах CS-1.

Данные SSD блока INITIATE CALL содержат сетевой адрес, куда должно быть установлено соединение, сетевой адрес исходящей стороны, характеристики средств доставки информации, затребованные исходящей стороной, а также указатели полей выходных данных CID, куда будет помещена информация о номере пункта назначения, об идентификаторе вызывающей стороны, о характеристиках средств доставки информации, о категории вызывающей стороны и о причине ошибочного выполнения блока.

В блоке отсутствуют входные данные CID. Блок имеет два типа логических завершений: *Успех* и *Ошибка*. Графическое представление блока приведено на рисунке 2.2.21.

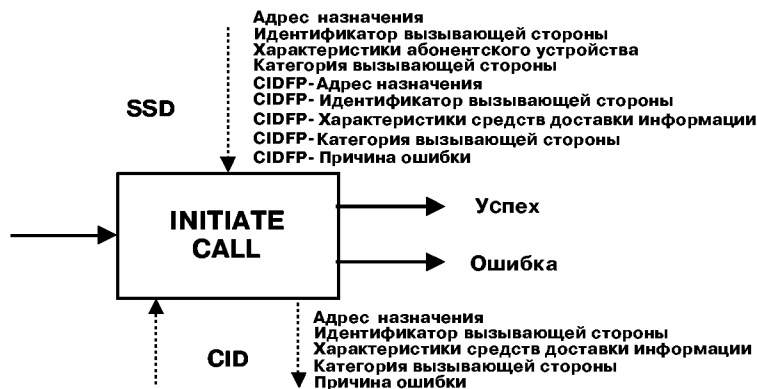


Рис. 2.2.21 Блок INITIATE CALL

БЛОК RELEASE CALL. Данный блок предназначен для разъединения на любом этапе процесса обслуживания вызова. Блок может быть использован во всех услугах CS-1. В данных SSD блока содержатся сведения о причине разъединения. В блоке отсутствуют как входные, так и выходные данные CID. Блок имеет один тип логического завершения – *Успех*, который означает, что информация о разъединении передана в BCP. Графическое представление блока приведено на рисунке 2.2.22.



Рис. 2.2.22 Блок RELEASE CALL

2.2.4 Стадия 1 описания блока BCP

Блок BASIC CALL PROCESS (BCP) определен в рекомендации I.329/Q.1203 как специализированный SIB, выполняющий основные (базовые) функции обслуживания вызова, в том числе:

- установление соединения;
- сохранение данных CID для дополнительных функций обслуживания вызова;
- разъединение.

На глобальной функциональной плоскости определены независимые от услуг блоки SIB, в том числе, – BCP, а также иницирующие точки (POI) и точки возврата (POR), представляющие собой функциональные интерфейсы между блоком BCP и цепочками других SIB.

В иницирующей точке POI процесс BCP приостанавливается, и осуществляется переход к последовательному выполнению цепочки действий, реализующих логику нужной услуги IN. Выполнение любой цепочки завершается возвратом в BCP через точку возврата POR. BCP продолжает свою работу на основе данных, являющихся результатом выполнения логики услуги IN и полученных им через точку возврата. В CS-1 определены следующие точки POI.

CALL ORIGINATED (ВЫЗОВ ИНИЦИИРОВАН)	Эта POI указывает, что пользователь инициировал вызов, но еще не приступил к набору номера.
ADDRESS COLLECTED (АДРЕС ПРИНЯТ)	Эта POI указывает, что адресная информация, набранная пользователем, принята.
ADDRESS ANALYSED (АДРЕС ПРОАНАЛИЗОВАН)	Эта POI указывает, что произведен анализ и определен смысл адреса, например, что это номер услуги Бесплатный вызов.
PREPARED TO COMPLETE CALL (ГОТОВНОСТЬ ОРГАНИЗОВАТЬ СВЯЗЬ)	Эта POI указывает, что сеть готова попытаться установить связь с вызываемой стороной.
BUSY (ЗАНЯТОСТЬ)	Эта POI указывает, что требуемый пользователь в данный момент занят.
NO ANSWER (НЕТ ОТВЕТА)	Эта POI указывает, что требуемый пользователь не отвечает на вызов.
CALL ACCEPTANCE (ВЫЗОВ ПРИНЯТ)	Эта POI указывает, что вызов принят входящей станцией, но соединение между вызывающей и вызываемой сторонами не установлено.
ACTIVE STATE (АКТИВНОЕ СОСТОЯНИЕ)	Эта POI указывает, что соединение установлено и связь вызывающей и вызываемой сторон активна.
END OF CALL (ОКОНЧАНИЕ СВЯЗИ)	Эта POI указывает, что один из участников связи отключился.

В CS-1 определены следующие точки возврата.

CONTINUE WITH EXISTING DATA (ПРОДОЛЖИТЬ РАБОТУ С ИМЕЮЩИМИСЯ ДАННЫМИ)	Эта POR указывает, что BCP должен продолжить обслуживание вызова без модификации данных. Используется только в версии ITU-T.
PROCEED WITH NEW DATA (ПРОДОЛЖИТЬ РАБОТУ С НОВЫМИ ДАННЫМИ)	Эта POR указывает, что BCP должен продолжить обслуживание вызова, используя при этом только модифицированные данные. Используется только в версии ITU-T.
HANDLE AS TRANSIT (ОБРАБАТЫВАТЬ КАК ТРАНЗИТНЫЙ)	Эта POR указывает, что BCP должен работать с вызовом так, как если бы он только что поступил.
CLEAR CALL (ПРЕКРАТИТЬ ОБСЛУЖИВАНИЕ ВЫЗОВА)	Эта POR указывает, что BCP должен прекратить обслуживание вызова и освободить занятые для его обслуживания ресурсы. Используется только в версии ITU-T.
INITIATE CALL (ИНИЦИИРОВАТЬ ВЫЗОВ)	Эта POR указывает, что нужно инициировать запрос связи, которая может быть независимой от существующей связи или быть в ее контексте. Используется только в версии ITU-T.

Ниже приводятся основные характеристики BCP в соответствии со стадией 1 стандартного описания блоков SIB.

ОПРЕДЕЛЕНИЕ:	Этот специализированный SIB поддерживает доступ к услугам/атрибутам услуг IN, представленным цепочками других SIB и глобальной логикой услуг. Интерфейсы между BCP и глобальной логикой услуг описываются иницилирующими точками (POI) и точками возврата (POR).
ВЫПОЛНЯЕМЫЕ ДЕЙСТВИЯ:	BCP содержит набор иницилирующих точек POI, и если при обслуживании вызова встречается такая точка, глобальная логика услуг организует выполнение соответствующей цепочки блоков SIB. По завершении выполнения цепочки SIB обслуживание вызова продолжается согласно алгоритму, который определяется соответствующей точкой возврата POR. <i>Примечание: Вызов, не требующий обращения к услугам IN, обрабатывается блоком BCP без участия глобальной логики услуг.</i>
ВОЗМОЖНОЕ ПРИМЕНЕНИЕ:	Применяется во всех услугах CS-1.

ВЫХОД:	Определяет иницирующую точку и параметры данных, передаваемых глобальной логике услуг.
Логический выход	Иницирует глобальную логику услуг из соответствующей точки POI.
Данные SSD:	
- набор точек POI	Определяет те точки в BCP, где для данной услуги может потребоваться логика услуг IN.
- CIDFP-CLI	Специфицирует, какие данные CID должны использоваться в качестве идентификатора вызывающей линии CLI.
- CIDFP-Category	Специфицирует, какие данные CID должны использоваться в качестве данных о категории линии, определяемой идентификатором вызывающей линии CLI.
- CIDFP-Dialled	Специфицирует, какие данные CID должны использоваться в качестве набранного номера.
- CIDFP-Destination	Специфицирует, какие данные CID должны использоваться в качестве номера пункта назначения.
- CIDFP-Call reference	Специфицирует, какие данные CID определяют метку соединения.
- CIDFP-Bearer	Специфицирует, какие данные CID должны использоваться в качестве характеристик средств доставки информации, запрошенных вызывающей линией.
Данные CID:	
- Calling line identity	Идентифицирует сетевой адрес, где возник данный вызов.
- Calling line category	Определяет характеристики вызывающей линии (например, таксофон, оператор и т.п.).
- Dialled number	Указывает номер, набранный вызывающей стороной.
- Destination number	Указывает номер, набранный вызывающей стороной. Хотя номер пункта назначения изначально совпадает с набранным номером, однако он может быть модифицирован при предоставлении услуги IN.
- Call reference	Отмечает соединение, устанавливаемое в процессе обслуживания конкретного вызова.
- Bearer capabilities	Специфицирует характеристики средств доставки информации ISDN (рек. ITU-T Q.931), которые запрошены вызывающей стороной.
ВХОД:	
Логический вход	Возобновляет действия блока BCP в точке POR, которую определила глобальная логика услуг.
Данные CID:	
- Destination number	Определяет сетевой адрес пункта назначения, с которым должно быть установлено соединение, предоставляемое обслуживаемому вызову. В частности, цепочка блоков SIB может создать адрес пункта назначения, отличающийся от набранного номера.
ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ:	Приведено на рисунке 2.2.23.

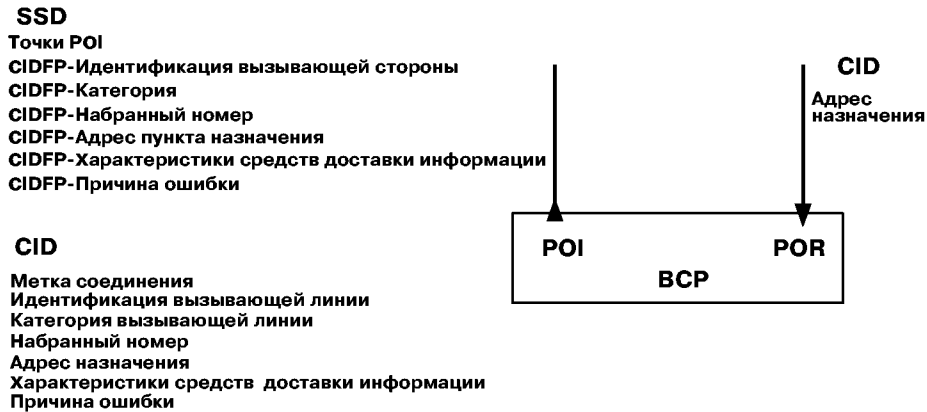


Рис. 2.2.23 Блок BCP

2.2.5 Глобальная логика услуг GSL

Глобальная логика услуг (GSL) описывает порядок, в котором из блоков SIB могут составляться цепочки, реализующие услуги/атрибуты услуг. Для конкретной услуги (атрибута услуг) CS-1 глобальная логика услуг описывает:

- точку POI, определяющую момент функционального перехода из процесса BCP к цепочке блоков SIB;
- точки POR, в которых может происходить возврат из цепочки SIB в процесс BCP;
- перечень SIB, объединяемых в цепочку, и порядок их объединения. Цепочка начинается в точке POI, определенной в 1), и заканчивается несколькими точками POR, определенными в 2);
- статические и динамические параметры (SSD, CID) для каждого SIB в цепочке.

Глобальная логика услуг GSL на глобальной функциональной плоскости «видит» процесс BCP как единый ресурс. С учетом этого определены необходимые виды взаимодействия GSL с процессом BCP.

От BCP к GSL:

- 1) *Логический старт цепочек SIB* – представлен точками POI.
- 2) *Данные* – представлены данными CID, которые необходимы цепочкам блоков SIB для реализации атрибутов услуг IN. Примерами данных CID, за которые несет ответственность процесс BCP, являются идентификация вызывающей линии и набранный номер.

От GSL к BCP:

- 1) *Логическое завершение цепочек блоков SIB* – представлено точками POR.
- 2) *Данные* – представлены данными CID, которые были определены одним или более SIB в цепочке. Примером таких CID может служить адрес пункта назначения.

Вызовы, не требующие обращения к услугам IN, обслуживаются только BCP. Когда необходимо обращение к услуге (атрибуту услуги) IN, BCP формирует соответствующий запрос. Для реализации требуемого атрибута глобальная логика услуг должна иметь «шаблон» для цепочки блоков SIB, соответствующей данному атрибуту (см. рис. 2.2.24). При создании новых атрибутов услуг в сервисную логику должны вводиться новые «шаблоны».

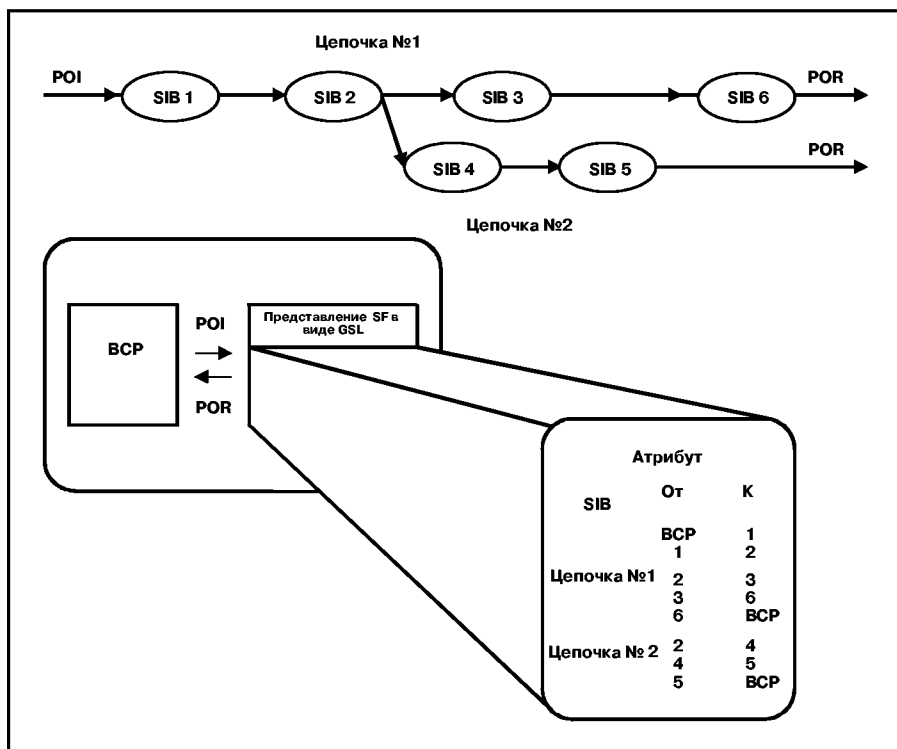


Рис. 2.2.24 Проекция плоскости услуг на глобальную функциональную плоскость

Глава 2.3

Распределенная функциональная плоскость

2.3.1 Требования и ограничения

Архитектура распределенной функциональной плоскости для набора возможностей CS-1, с одной стороны, определяется услугами, которые должны быть поддержаны, и, с другой стороны, ограничивается сетевыми технологиями, имевшимися на момент стандартизации CS-1. Функциональные средства, которые поддерживают возможности CS-1, обеспечивают:

- доступ пользователя к процессам предоставления соединений и/или услуг;
- обращение к услугам и управление их предоставлением;
- взаимодействие пользователя с функциями управления услугами;
- эксплуатационное управление услугами.

Рассмотрим перечисленные тезисы более подробно. Что касается доступа конечного пользователя к процессам предоставления соединений и/или услуг, то в CS-1 он обеспечивается как для обычных аналоговых линий и линий ISDN (BRI и PRI) на абонентском участке, так и для соединительных линий (каналов) с традиционными системами сигнализации и системой ОКС-7 на межстанционном участке.

Процессы предоставления соединений и услуг IN в наборе CS-1 основаны на существующих возможностях современных коммутационных систем. Общая модель обработки вызова при установлении соединения между двумя абонентами дополнена функциями коммутации услуг, которые служат для обращения к логике услуг IN. После обращения к ней логика услуг работает под контролем функций управления услугой и функций, ответственных за работу с данными об услуге.

При спецификации CS-1 использован предусматриваемый архитектурой IN распределенный подход к реализации процесса предоставления соединения/услуги, однако основная ответственность за сохранность соединения и за управление ресурсами, используемыми для его предоставления, оставлена за имеющимися сегодня в коммутационных станциях функциями управления коммутацией. Это обстоятельство определяет большую часть перечисляемых ниже основных ограничений и функциональных требований, относящихся к CS-1.

- Функции управления связью пользователей (CCF) и функции коммутации услуг (SSF) настолько тесно связаны между собой, что связь между ними не стандартизуется.
- Соединение может быть установлено либо между двумя или несколькими пользователями, либо между пользователем и самой сетью.
- Вызов может быть инициирован или конечным пользователем, или функциями управления услугами (SCF) от имени пользователя.
- Соединение может проходить через несколько коммутационных станций, при этом не обязательно, чтобы все они имели функции SSF. Логика IN потенциально может быть затребована на любой из тех станций, в которых имеются функции SSF, однако каждый раз она работает независимо (то есть предыстория соединения логике услуг в промежуточных SSP неизвестна).
- С точки зрения координации процессов предоставления соединений коммутационная станция рассматривается как имеющая две функционально разделенные группы ресурсов, одна из которых привлекается к исходящей, а другая – ко входящей части соединения. Такое разделение необходимо для того, чтобы в АТС с функциями SSF логика услуг, привлекаемая к работе с исходящей частью соединения (от имени вызывающего абонента), могла работать независимо от логики услуг, привлекаемой к работе со входящей частью соединения (от имени вызываемого абонента).
- Одному пользователю могут быть доступны одновременно несколько услуг IN. Более того, этому же пользователю могут быть доступны и другие дополнительные услуги (не IN). Следовательно, должен быть предусмотрен механизм взаимодействия услуг и их атрибутов, обеспечивающий, ко всему прочему, предоставление по каждому запросу одной и только одной услуги. Логика этой услуги может быть сколь угодно сложной, но она должна быть единственной, к которой происходит обращение при данном запросе.
- Распределенный подход и обусловленная им дополнительная сложность процесса предоставления соединений/услуг требуют адекватных средств обнаружения и устранения повреждений и сбоев, обеспечивающих во всех случаях корректное завершение процесса с соответствующим речевым уведомлением пользователя о причине неудачи.

Функции эксплуатационного управления услугами, необходимые для поддержки управляющих ресурсов (SCF, SDF, SRF), и соответствующие интерфейсы для CS-1 не стандартизованы. Однако пользователю может быть разрешен доступ к информации эксплуатационного управления для изменения некоторых параметров, связанных с предоставляемой ему услугой.

2.3.2 Распределенная функциональная модель для CS-1

На рисунке 2.3.1 приведена распределенная функциональная модель CS-1. Эта модель является подмножеством общей модели, рассмотренной в главе 1.3, где были определены функциональные объекты и связи между ними.

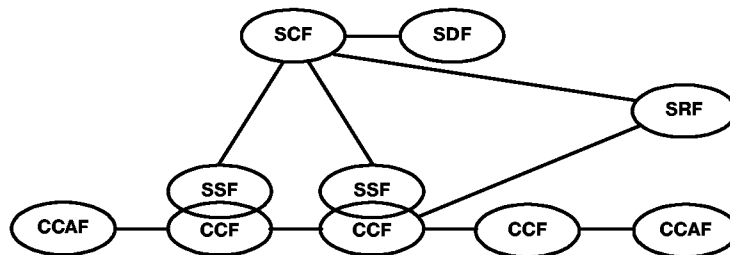


Рис. 2.3.1 Модель распределенной функциональной плоскости для CS-1

В чем же отличие архитектуры на рисунке 2.3.1 от архитектуры на рисунке 1.3.6.? Для CS-1 определены только функциональные объекты CCAF, CCF, SSF, SCF, SDF и SRF и соответствующие связи. Кроме того, для CS-1 не специфицированы связи вида SCF-SCF и SDF-SDF. Аспекты, касающиеся реализации эксплуатационного управления услугами, оставлены на усмотрение оператора сети.

Функции поддержки доступа CCAF представляют собой интерфейс между пользователем и сетевыми функциями CCF управления связью пользователя. Проще говоря, функции CCAF выполняют совместно абонентский терминал (телефон с импульсным набором номера, телефон с многочастотным набором, терминал ISDN), абонентская линия (в совокупности с абонентской сигнализацией) и, возможно, опорная АТС, не содержащая функций IN. Эти средства обеспечивают прием от пользователя и передачу к функциям CCF информации, необходимой для доступа к услуге IN, а также прием от CCF индикаций прохождения соединения и передачу к CCF запроса разъединения.

Функции управления связью пользователя (CCF) относятся к запросу связи и к управлению соединением в классическом смысле. Важное свойство функций CCF состоит в обеспечении триггерного механизма для обнаружения вызовов, требующих доступа к ресурсам IN.

Функции коммутации услуг (SSF) представляют собой интерфейс между функциями управления связью пользователя и функциями управления услугами. Они распространяют логику CCF, предусматривая обнаружение вызовов, требующих доступ к конкретным услугам,

и возможность взаимодействия с SCF. По информации, полученной от пользователя (через CCF), SSF может определить, какая именно нужна услуга, и передать имеющуюся на станции и необходимую для логики этой услуги информацию к функциям SCF через стандартизованный интерфейс. В соответствии с полученными от SCF ответными инструкциями функции SSF модифицируют функции CCF.

Функции управления услугами (SCF) принимают от SSF информацию о том, какая именно услуга запрашивается, и информацию, специфическую для данного вызова (номер вызывающего абонента, номер, набранный этим абонентом и т.п.). К обработке информации привлекается логика запрошенной услуги. В процессе обработки могут запрашиваться дополнительные сведения от функций хранения данных для услуг (SDF), например, данные о соответствии между физическим и логическим адресами, текущий размер кредита, тариф и др. Основным результатом обработки информации является физический адрес, который SCF передают обратно в SSF.

Функции хранения данных для услуг (SDF) содержат данные, относящиеся к услуге и к абонентам. Эти данные доступны SCF в реальном времени и используются ими по мере надобности при подготовке информации, нужной для предоставления услуги IN.

Функции специализированных ресурсов (SRF) реализуют такие вспомогательные возможности как прием сигналов многочастотного набора номера, передача речевых извещений, предоставление мостов конференцсвязи и т.п.

Как видно из рисунка 2.3.1, в соответствии с распределенной моделью CS-1 процесс предоставления услуги IN заключается в установлении соединения объектами CCF/SSF, в выполнении логики услуги в SCF, а также в использовании вспомогательных ресурсов и данных (в объектах SRF и SDF). В рекомендации Q.1214 для CS-1 даны модели каждого функционального объекта распределенной функциональной плоскости в виде машины конечных состояний.

Как мы уже знаем, предусматриваемые концепцией IN средства моделирования обслуживания вызовов функциями CCF/SSF используют абстрактное представление процессов обслуживания вызовов и установления соединений, не зависящее от реализации оборудования и от его производителя. Такое представление делает обозримыми для SCF действия CCF/SSF и их ресурсы, что обеспечивает возможность взаимодействия SCF с SSF при выполнении логики услуг.

При моделировании средств поддержки логики услуг используется абстрактное представление действий и ресурсов SCF, необходимых для выполнения логики услуг, а также абстрактное представление действий и ресурсов SRF и SDF, доступных SCF.

Моделирование с использованием абстрактных представлений не предполагает, что производители оборудования обязаны соблюдать точное соответствие между компонентами модели и оборудованием, в котором они размещаются. Однако, учитывая тот факт, что в подавляющем большинстве случаев приведенные в рекомендациях ITU абстрактные модели реализуются производителями практически один к одному, далее мы рассмотрим модели CCF/SSF, SCF, SRF и SDF для CS-1 более подробно.

2.3.3 Модель CCF/SSF

2.3.3.1 Основные компоненты

Модель внутренних ресурсов CCF/SSF представлена на рисунках 2.3.2 и 2.3.3. Она ориентирована на услуги (атрибуты услуг) типа A, то есть на такое обслуживание вызовов, когда услуги IN предоставляются независимо вызывающей и вызываемой стороне соединения. На рис. 2.3.2 приведена модель для исходящей и входящей частей соединения в одной и той же ATC с функциями SSF, на рисунке 2.3.3 – для исходящей и входящей частей одного соединения между разными ATC с функциями SSF.

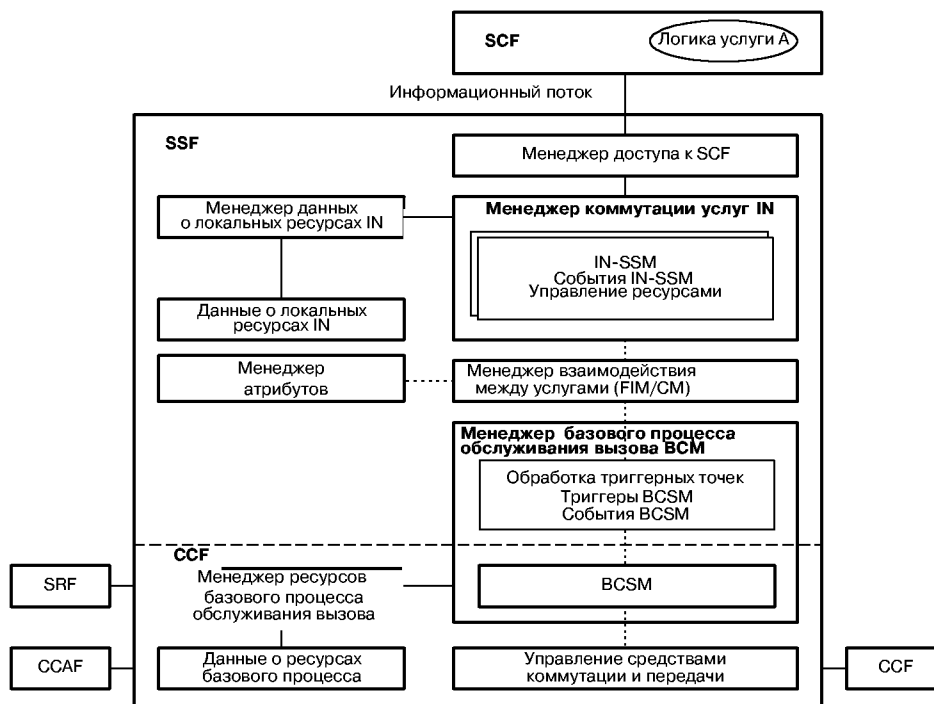


Рис. 2.3.2 Модель внутренних ресурсов CCF/SSF для одной ATC

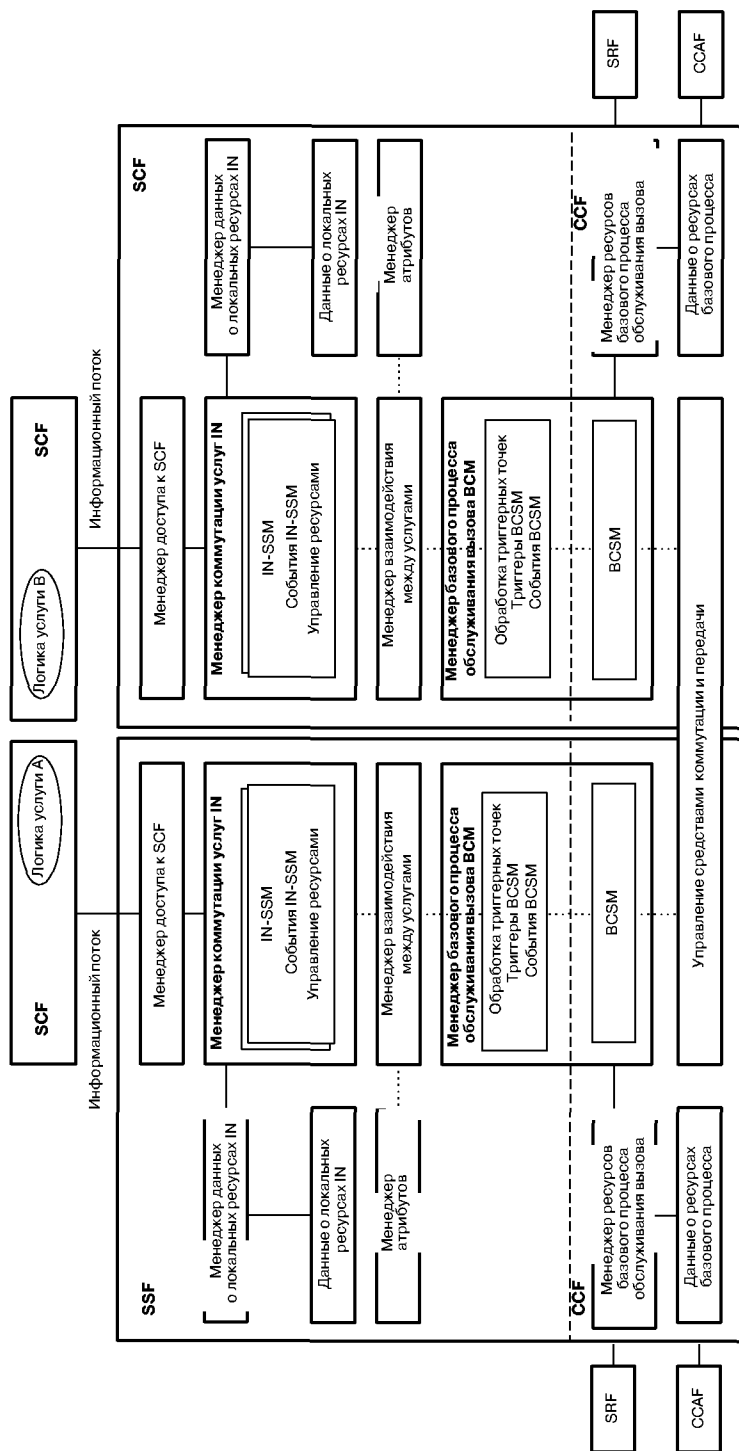


Рис. 2.3.3 Модель внутренних ресурсов CCF/SSF в исходящей и входящей АТС

С точки зрения функций IN модель CCF/SSF содержит следующие основные блоки: BCM – менеджер базового процесса обслуживания вызова, IN-SM – менеджер коммутации услуг IN, FIM/CM – менеджер взаимодействия между услугами.

BCM является абстрактным представлением той части коммутационной станции, в которой реализованы базовые функции управления связью пользователя и установлением соединений между пользователями. Он отслеживает происходящие в процессе управления события, о которых необходимо известить SCF. Кроме того, в BCM реализована модель состояний базового процесса обслуживания вызова BCSM и функции обработки точек обнаружения DP.

IN-SM служит интерфейсом, который делает видимыми для SCF события, происходящие в CCF/SSF, и обеспечивает доступ SCF к ресурсам CCF/SSF.

FIM/CM предусматривает механизм, обеспечивающий поддержку нескольких одновременных обращений к логике услуг (как IN, так и обычных) при обслуживании одного вызова. В частности, он может предотвращать одновременное обращение к логике услуг IN и не IN. В любом случае FIM-CM предоставляет функциям SSF унифицированную информацию о процессе обслуживания вызова.

BCSM представляет собой модель действий CCF, необходимых для установления и поддержания связи между пользователями. Она идентифицирует действия, относящиеся к базовому процессу обслуживания вызова, и показывает то, как эти действия объединяются. Не все аспекты BCSM видны логике услуг и подлежат стандартизации. BCSM есть средство для представления и анализа того, какие именно детали поведения функций CCF должны быть видны извне и какой уровень детализации нужен для их адекватного отображения.

BCSM идентифицирует точки базового процесса, в которых логика услуг может с ним взаимодействовать. В частности, модель содержит средства для описания событий, ведущих к запросу логики услуг, точек, в которых эти события должны отслеживаться, и точек, в которых возможна передача управления функциям SCF.

Применительно к CS-1 модель BCSM отражает функциональное разделение между процессами установления исходящей и входящей частей соединения, что показано на рисунках 2.3.2 и 2.3.3. Исходящая и входящая части управляются каждая своим базовым процессом обслуживания вызова. Для того, чтобы различать состояния и точки обнаружения этих двух процессов, в названиях состояний и точек присутствует буква «О» (Originating) и «Т» (Terminating), соответственно, для исходящей и входящей частей модели.

Состояние 1: PIC1: O_Null & Authorize Origination Attempt (исходящая сторона, исходное состояние, проверка правомочности запроса исходящей связи)	
<i>Стартовое событие:</i>	Освобождение ресурсов, использовавшихся в предыдущем соединении (переходы из DP9 - O_Disconnect или DP10 - O_Abandon), или после завершения обработки по умолчанию ошибочных ситуаций.
<i>Функции:</i>	Освобождение линий и каналов. Проверка соответствия запроса правам связи пользователя.
<i>Доступная информация:</i>	Характеристики средств доставки информации. Номер вызывающего абонента. Категория вызывающего абонента. Доступность ресурсов SRF. Профиль услуг. Тип доступа (ISDN или обычный). Номер вызываемого абонента (для линий ISDN). Префикс и/или коды доступа (для линий ISDN). Иные данные, получаемые при запросе услуг ISDN в сообщении SETUP системы сигнализации DSS1 или в сообщении IAM подсистемы ISUP OKC-7.
<i>Выходные события:</i>	Индикация запроса исходящей связи (например, снятие трубки, сообщение SETUP или IAM) и подтверждение права связи (переход к DP1 - Origination_Attempt_Authorized).
<i>События выхода по исключению:</i>	Индикации отказа от продолжения установления соединения.

Состояние 2: PIC2: Collect Information (прием информации)	
<i>Стартовое событие:</i>	Индикация запроса исходящей связи (например, снятие трубки, сообщение SETUP или IAM) и подтверждение права связи (переход из DP1 - Origination_Attempt_Authorized).
<i>Функции:</i>	Прием адресной информации, анализ адресной информации на предмет ее полноты (если нужно, - прием переменного числа цифр).
<i>Доступная информация:</i>	Номер для начисления платы. Номер вызывающего абонента. Полученная от абонента или от встречной АТС адресная информация, содержащая: - код доступа, - префикс, - индикатор плана нумерации (для линий ISDN), - накопленные цифры.
<i>Выходные события:</i>	Наличие полной начальной адресной информации от исходящей стороны (переход к DP2 - Collected_Info).
<i>События выхода по исключению:</i>	Окончание выдержки времени, отведенной на прием цифр. Отсутствие системных ресурсов для приема цифр. Прием некорректной адресной информации. Отказ вызывающей стороны от установления соединения.

Состояние 3: PIC3: Analyse_Information (анализ информации)	
Стартовое событие:	Наличие полной начальной адресной информации от исходящей стороны (переход от DP2 - Collected_Info) или переход из PIC4 при занятости выбранного маршрута.
Функции:	Анализ принятой адресной информации и/или ее пересчет в соответствии с планом нумерации для определения физического адреса назначения и вида связи (местная, междугородная, международная).
Доступная информация:	Номер для начисления платы. Номер вызывающего абонента. Результат анализа полученной от абонента или от встречной АТС информации, содержащей: - номер вызываемой стороны, - индикатор плана нумерации, - вид связи (местная, междугородная, ...).
Выходные события:	Доступен физический адрес назначения и вид связи (переход к DP3- Analysed_Info).
События выхода по исключению:	Отказ от продолжения установления соединения. Получение некорректной адресной информации.

Состояние 4: PIC4: Routing and Alerting (маршрутизация и оповещение)	
Стартовое событие:	Доступен физический адрес назначения и вид связи (переход к DP3- Analysed_Info).
Функции:	Интерпретация адреса назначения и вида связи, включая пересчет списочного номера в адрес физического порта. Проверка права вызывающего абонента на данную связь (с учетом принадлежности к замкнутой группе, ограничений на междугородные вызовы и т.п.). Передача акустического сигнала о вызове второго абонента и ожидание уведомления о его ответе.
Доступная информация:	Номер счета для начисления платы. Номер вызывающего абонента. Номер вызываемой стороны. Индикатор плана нумерации. Вид связи. Информация о маршруте.
Выходные события:	Получение от T_BCSM индикации ответа вызываемой стороны (переход к DP7 - Answer). Занятость маршрута (переход к DP3 - Analyse Information).
События выхода по исключению:	Отсутствие ресурсов для выбранного маршрута. Занятость вызываемой стороны. Отсутствие ответа вызываемой стороны. Отказ вызывающей стороны от ожидания ответа. Отсутствие прав на связь данного вида.

Состояние 5: PIC5: O_Active (исходящая сторона, активное состояние)	
Стартовое событие:	От T_BCSM получена индикация ответа вызываемой стороны (переход к DP7 - O_Answer).
Функции:	Установление соединения между исходящей и входящей сторонами, начисление платы за связь и наблюдение за состоянием связи.
Доступная информация:	Та же, что и для PIC4.
Выходные события:	Обнаружение запроса дополнительной услуги в виде, например, кратковременного нажатия на рычаг телефонного аппарата, многочастотного сигнала DTMF, сообщения DSS1 (переход к DP8 - O_Mid_Call). Отказ вызывающей или вызванной стороны от продолжения связи (переход к DP9 - O_Disconnect).
События выхода по исключению:	Сбой соединения.

Состояние 6: PIC6: O_Exception (исходящая сторона, выход по исключению)	
Стартовое событие:	Возникновение ситуации, предполагающей выход по исключению из любой точки PIC.
Функции:	Стандартная (принятая по умолчанию) обработка исключительных ситуаций, предусматривающая освобождение задействованных ресурсов.
Доступная информация:	Та, которая доступна в точке, где возникла исключительная ситуация.
Выходные события:	Завершение обработки исключительной ситуации функциями CCF/SSF (переход к PIC1 - O_Null & Authorize_Originating_Attempt).
События выхода по исключению:	Отсутствуют.

2.3.3.3 Входящая BCSM

Входящая половина BCSM (T_BSCM) соответствует той части ресурсов CCF, которые несут ответственность за установление соединения к вызываемому абоненту. Модель T_BSCM для CS-1 приведена на рисунке 2.3.5. Рассмотрим точки PIC, связанные с T_BSCM.

Состояние 7: PIC7: T_Null & Authorize_Terminating_Attempt (входящая сторона, исходное состояние и проверка правомочности запроса входящей связи)	
Стартовое событие:	Освобождение ресурсов, занятых в предыдущем соединении (переход от DP17 - T_Disconnect или от DP18 - T_Abandon). Окончание обработки исключительной ситуации.
Функции:	Освобождение линий и каналов. Контроль исходного состояния. Проверка правомочности входящего вызова.
Доступная информация:	Номер счета для начисления платы. Номер вызывающей стороны. Категория вызывающей стороны. Номер вызываемой стороны. Дополнительная информация, доставленная входящей системой сигнализации.
Выходные события:	Индикация приема входящего вызова и разрешение направить его к адресату.
События выхода по исключению:	Индикация отказа со стороны исходящей части или отрицательный результат проверки права входящей связи.

Состояние 8: PIC8: Select Facility & Present Call (выбор ресурса и извещение о входящем вызове)	
Стартовое событие:	Индикация приема входящего вызова и разрешение направить его к адресату (переход от DP12 - Term_Attempt_Authorized).
Функции:	Выбор ресурса для обслуживания вызова. Подача извещения о вызове к вызываемому терминальному оборудованию (сообщения SETUP в случае ISDN или вызывного сигнала в случае аналоговой абонентской линии).
Доступная информация:	Та же, что для PIC7.
Выходные события:	Входящая сторона извещается о вызове (переход к PIC9 - T_Alerting). Получен ответ вызываемой стороны (переход к DP15 - T_Answer).
События выхода по исключению:	Вызываемая сторона занята или недоступна (переход к DP13 - T_Busy). Получена индикация отказа вызывающей стороны от связи (переход к DP18 - T_Abandon).

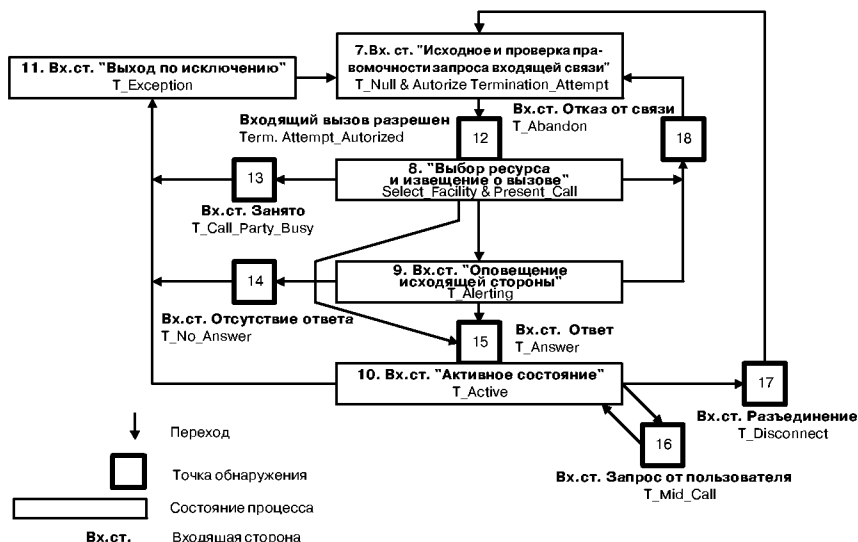


Рис. 2.3.5 Модель входящей части BCSM для CS-1

Состояние 9: PIC9: T_Alerting (входящая сторона передает оповещение)	
Стартовое событие:	Входящая сторона извещается о вызове.
Функции:	Передача индикации оповещения к O_BSCM и ожидание ответа вызываемой стороны.
Доступная информация:	Та же, что для PIC8.
Выходные события:	Ответ вызываемой стороны (переход к DP15 - T_Answer).
События выхода по исключению:	Отсутствие ответа (переход к PIC14- No_Answer). Получена индикация отказа вызывающей стороны от связи (переход к DP18 - T_Abandon).

Состояние 10: PIC10: T_Active (входящая сторона, активное состояние)	
Стартовое событие:	Ответ вызываемой стороны.
Функции:	Передача индикации ответа вызываемой стороны к O_BCSM. Установление соединения между исходящей и входящей сторонами, наблюдение за состоянием связи.
Доступная информация:	Та же, что для PIC9.
Выходные события:	Прием от вызванной стороны запроса услуги (атрибута услуги), например, кратковременное нажатие на рычаг телефонного аппарата, сигнал DTMF, сообщение DSS1 (переход к DP16 - T_Mid_Call). Запрос разъединения вызванной стороной или от O_BCSM (переход к DP17 - T_Disconnect).
События выхода по исключению:	Сбой соединения.

Состояние 11: PIC11: O_Exception (входящая сторона, выход по исключению)	
Стартовое событие:	Возникновение условий, предполагающих выход по исключению из любой описанной выше точки PIC
Функции:	Индикация в сторону O_BCSM возникновения нештатной ситуации; стандартная обработка исключительных ситуаций, предполагающая освобождение задействованных ресурсов.
Доступная информация:	Та, которая имеется в точке, где возникла исключительная ситуация.
Выходные события:	Завершение обработки исключительной ситуации функциями CCF/SSF (переход к PIC7 - T_Null & Authorize_Originating_Attempt).

2.3.3.4 Точки обнаружения

Точки обнаружения DP представляют собой такие точки в базовом процессе обслуживания вызова, в которых могут быть обнаружены события, представляющие интерес для логики услуг IN. В случае необходимости информация о таких событиях передается к функциям SCF. Для того, чтобы это было возможно, соответствующая DP должна быть активизирована. Только в этом случае программы логики услуг, находящиеся в SCF, смогут влиять на последующее обслуживание вызова. Если DP не активизирована, то CCF/SSF продолжает работать с вызовом без обращения к SCF. Точки обнаружения характеризуются следующими атрибутами.

Механизмом активизации. Точки обнаружения могут быть активизированы статически или динамически. Статическая активизация производится функциями SMF. Такие точки остаются в активизированном состоянии до момента их деактивизации со стороны SMF. Динамическая активизация производится SCF в контексте управляющей связи между SSF и SCF при обслуживании конкретного вызова, причем DP остается активизированной до окончания этой управляющей связи.

Критерием. Под критерием понимаются условия, которые должны быть удовлетворены, чтобы к SCF было передано уведомление о том, что встретилась активизированная DP.

Логической связью. Если встречена активизированная DP и удовлетворен соответствующий ей критерий, функции SSF могут обмениваться информационными потоками с SCF, используя некую абстрактную среду, носящую название «логическая связь». Логическая связь может быть *управляющей* (используя ее, SCF влияет на процесс обслуживания вызова) и *контрольной* (используя ее, SCF может лишь вести мониторинг процесса, не оказывая на него никакого воздействия).

Необходимостью приостановки базового процесса обслуживания вызова. При условии, что встретилась активизированная DP, удовлетворяется соответствующий ей критерий и установлена управляющая связь, SSF может приостановить процесс обслуживания вызова для того, чтобы дать возможность функциям SCF влиять на дальнейший ход этого процесса. Если необходимость приостанавливать процесс отсутствует, функции SCF уведомляются о том, что встретилась определенная DP, но их ответная реакция не ожидается. Этот атрибут точки обнаружения назначается таким же образом, каким осуществляется ее активизация.

В соответствии с рассмотренными атрибутами для CS-1 определены четыре типа точек обнаружения:

- триггерная точка обнаружения, запрос (TDP-R);
- триггерная точка обнаружения, уведомление (TDP-N);
- точка обнаружения события, запрос (EDP-R);
- точка обнаружения события, уведомление (EDP-N).

Атрибуты перечисленных типов точек обнаружения приведены в таблице 2.3.1.

Для точек обнаружения могут быть назначены как индивидуальные (для каждой линии или канала), так и групповые критерии, или даже единый критерий для всей АТС. Среди основных критериев, применимых к точкам обнаружения CS-1, можно назвать назначенный триггер, класс обслуживания, идентификатор В-канала, последовательность цифр, префикс, код доступа, номер вызывающего абонента, номер вызываемого абонента, тип адреса, характеристики средств доставки информации.

Обработка точек обнаружения включает в себя следующее. Выполняются действия по управлению трафиком (прореживание потока вызовов и просеивание). Производится проверка того, удовлетворяется ли назначенный для DP критерий. Запускается механизм, обеспечивающий корректность взаимодействия программ логики услуг IN и не IN. Формируется информационный поток в сторону SCF.

Кроме того, одна и та же точка обнаружения может быть активизирована несколько раз в качестве TDP-R с разными критериями, приоритеты которых устанавливаются административной процедурой. Каждый следующий критерий анализируется только в случае, если не удовлетворяется предыдущий, или если после отработки предыдущего удовлетворенного критерия процесс обслуживания вызова возвратился к той же DP (при условии, что управляющая связь, обеспечивавшая отработку предыдущего критерия, завершена или заменена контрольной). Критерии, связанные с TDP-N, обрабатываются независимо от наличия или отсутствия управляющей связи.

Управляющая связь сохраняется до тех пор, пока есть активизированные для данной части соединения EDP-R, и завершается, когда таковых больше нет или когда происходит разъединение. Во время существования управляющей связи точки EDP-R могут динамически деактивизироваться со стороны SCF. Динамическая деактивизация EDP-R со стороны SSF производится после того, как они встретились и об этом был извещен SCF, или же после разъединения.

Управляющая связь заменяется контрольной в случае, если больше нет активизированных EDP-R, но остались активизированные EDP-N. Когда не остается и активизированных EDP-N или когда произведено разъединение, контрольная связь тоже завершается. Динамическая деактивизация точек EDP-N производится так же, как и точек EDP-R.

Соблюдение приведенных правил гарантирует поддержку множества таких комбинаций обработки TDP/EDP, которые обеспечивают соблюдение принципа управления услугой из одной точки (single point of control) – основополагающего для набора возможностей CS-1. Дотошный читатель может найти полный перечень этих комбинаций в рекомендации Q.1214.

2.3.3.5 Менеджер коммутации услуг IN-SM

Основную часть IN-SM (IN-switching manager) составляет модель состояний процесса коммутации услуг IN-SSM (IN-Switching state model), представляющая процесс обслуживания вызова IN функциями SSF/CCF в терминах состояний соединения. При этом используются средства объектно-ориентированного описания.

Модель IN-SSM определяет группу объектов внутри CCF/SSF, которые видимы (а значит и управляемы) со стороны SCF. Можно выделить несколько типов объектов внутри самой модели IN-SSM, однако в рекомендации Q.1214 внимание акцентируется только на основном из них, который, в свою очередь, состоит из ряда объектов, отображающих ресурсы коммутации и передачи, и носит название «управление соединением».

IN-SSM создается при каждом обращении к логике услуг IN, требующем управления соединением. Создание IN-SSM либо является следствием того, что в BCSM встречается TDP, либо инициируется со стороны SCF независимо от наличия TDP. Разрушается IN-SSM после того, как со стороны SCF получена информация о завершении работы логики услуги.

Входящий в IN-SSM объект «управление соединением» дает функциям SCF абстрактное представление одной изолированной части устанавливаемого соединения, управляемой либо исходящей, либо входящей частью BCSM. Такая изолированная часть называется «сегментом соединения». Концепция сегментов соединения может быть использована, чтобы более ясно определить два фундаментальных ограничения, касающиеся услуг и атрибутов услуг *типа А* (напомним, что услуги только этого типа поддерживаются в CS-1). Ограничение, состоящее в том, что услуга (атрибут услуги) *типа А* соотносится всякий раз только с одним участником связи и не зависит от услуг (атрибутов), соотнесенных с другим участником этой связи (см. Главу 2.1), можно интерпретировать как односегментность услуги. Ограничение, состоящее в том, что при предоставлении услуг *типа А* на одни и те же аспекты связи между пользователями в любой момент времени может воздействовать только одна группа функций управления услугами SCF (то есть для услуг этого типа возможно управление только из одной точки), можно интерпретировать как односегментность управляющей связи между SCF и BCSM (см. рис. 2.3.7).

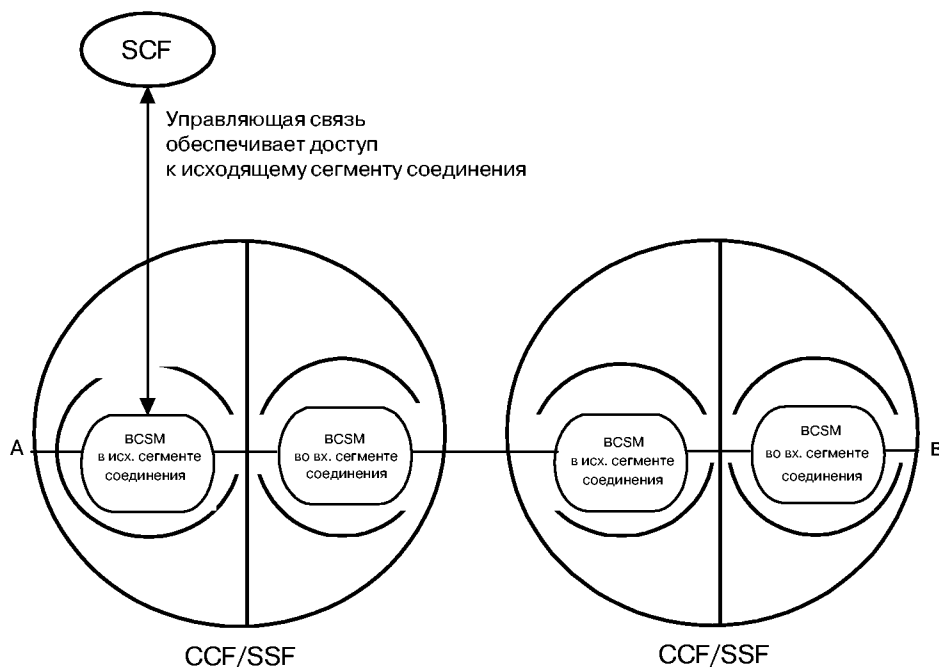


Рис. 2.3.7 Управление из одной точки (односегментная управляющая связь SCF - O_BCSM)

Управляющее воздействие на изолированный сегмент со стороны сразу нескольких SCF недопустимо. Однако информационные потоки, содержащие отчеты о точках обнаружения одновременно несколькими SCF, допускаются. Кроме того, допускается завершить управляющую связь с одним SCF и инициировать управляющую связь с другим SCF.

2.3.3.6 Менеджер взаимодействия услуг FIM/CM

Функции SCF могут управлять несколькими трактами и соединениями при поддержке нескольких одновременно активных BCSM. В связи с этим, в числе прочего, необходима координация действий, обусловленных одновременно возникающими в разных BCSM событиями, и действий по приостановке/возобновлению процессов обслуживания, происходящих в разных BCSM, но относящихся к одному IN-SSM.

Менеджер взаимодействия услуг FIM (Feature interaction manager) должен определять, логика какой услуги должна быть затребована для данной точки обнаружения. FIM должен содержать механизм выбора необходимой логики для предоставления соответствующему пользователю нужной ему услуги, будь то услуга IN или не IN.

Этот же механизм, в случае необходимости, должен запрещать одновременное предоставление таких услуг IN и таких услуг не IN, взаимодействие между которыми может привести к неадекватному действию хотя бы одной из них. Для реализации этого ограничения может использоваться как статический, так и динамический механизм. Статический механизм вводится функциями SMF на этапе развертывания услуг, динамический механизм требует усложнения FIM. Для CS-1 предполагается использование простейшего статического механизма.

2.3.3.7 Функционирование модели

Попробуем описать последовательность действий, выполняемых объектами модели CCF/SSF, для иллюстрации роли основных компонентов этой модели и связей между ними.

Пользователь взаимодействует с CCF/SCF через CCAF с целью запросить связь. Менеджер BCM создает BCSM, которая представляет основные функции управления соединением, необходимые для организации и поддержки этой связи. В процессе управления соединением в BCSM отслеживаются события, связанные с обслуживанием вызова.

BCM обрабатывает события, происходящие в точках обнаружения в BCSM. В случае, если в активизированной точке обнаружения удовлетворяется соответствующий ей критерий, BCM информирует FIM/CM о состоянии BCSM и об обнаруженном событии. Если BCM

нужны инструкции, работа BCSM приостанавливается в данной DP до их получения. В противном случае BCSM продолжает работать.

Используя полученную от BCM информацию, FIM/CM (Feature interaction manager/call manager) определяет, нужна ли для обработки события логика услуг IN или логика обычных услуг. Кроме того, принимается решение, нужно ли активизировать новую логику услуги, или событие может быть обработано уже активизированной к данному моменту логикой.

Если для обработки события необходимо новое обращение к логике услуг IN, FIM/CM информирует об этом IN-SM и снабжает его информацией о событии и о состоянии BCSM. Если для обработки события необходим запрос обычной услуги (не-IN), FIM/CM информирует об этом поп-IN-SM (менеджера услуг не-IN), который несет ответственность за дальнейшее обслуживание вызова.

IN-SM принимает и обрабатывает информацию о событиях, связанных с услугами IN. Если необходимо новое обращение к логике услуг, IN-SM создает новую IN-SSM, которая представляет состояния соединения в виде, понятном программам логики услуг в SCF. Затем менеджер коммутации услуг формирует и направляет в сторону SCF информационный поток, содержащий сведения о текущем состоянии IN-SSM.

SCF принимает и обрабатывает информационный поток от SSF и активизирует логику затребованной услуги, после чего направляет к SSF ответный информационный поток, содержащий требование к IN-FM изменить состояние IN-SSM таким образом, чтобы был реализован нужный атрибут услуги. SCF может также потребовать от SSF информировать его об определенной группе событий внутри BCSM, то есть указать группу точек EDP, которые должны быть активизированы.

IN-SM принимает и обрабатывает информационный поток от SCF с целью изменить должным образом состояние IN-SSM. При этом IN-SM передает соответствующий запрос к FIM/CM, а также следит за изменением состояния IN-SSM с целью обнаружить события, о которых необходимо информировать SCF.

FIM/CM принимает запрос от IN-SM и проверяет его правомерность с учетом того, логика каких услуг к данному моменту активизирована. После этого FIM/CM передает к BCM указание, какие функции должны быть выполнены, и требование отслеживать события в BCSM.

Выполняя полученное указание, BCM манипулирует состояниями одной или нескольких BCSM. В процессе работы с BCSM он выполняет соответствующие функции управления ресурсами, а также следит за событиями в BCSM. Обнаружив в BCSM событие, BCM информирует об этом FIM/CM.

FIM/CM определяет, как следует обрабатывать это событие, после чего сообщает IN-SM, что событие связано с активной в данный момент логикой услуги IN.

IN-SM обрабатывает информацию о событии следующим образом. При условии, что событие связано с активной в данный момент логикой услуги IN, IN-SM обновляет текущее состояние IN-SSM с тем, чтобы отразить состояние соединения пользователя и передать в информационном потоке от SSF к SCF информацию о событии и о состоянии IN-SSM.

В рассматриваемом нами случае SCF обрабатывает информационный поток следующим образом. При условии, что событие связано с активной логикой услуги IN, содержание информационного потока передается соответствующей программе. Затем формируется ответный информационный поток к SSF, содержащий требование, чтобы IN-SM изменил состояние IN-SSM.

Обмен информационными потоками между SSF и SCF продолжается, пока логика услуги не достигнет завершения (не останется никаких EDP или обслуживание вызова ресурсами SSF/CCF перейдет в область, где, в соответствии с логикой данной услуги, возникновение новых EDP не ожидается).

2.3.4 Модель SRF

Модель функционального объекта, содержащего специализированные ресурсы, приведена на рисунке 2.3.8. Отличительной особенностью SRF является то, что внутренние ресурсы всегда управляются по запросу, получаемому извне от других функциональных объектов.

Так, в целях поддержки процесса обслуживания вызова, SRF имеет логические связи с CCF/SSF и SCF. SCF управляет соединением между SSF/CCF и SRF, а также направляет к SRF запросы, содержащие необходимые инструкции. В процессе формирования ответа на запрос SSF функциям SCF может понадобиться войти в диалог с пользователем (вызывающим или вызываемым).

Функции SCF в CS-1 передают SRF инструкции о необходимости начать диалог только после того, как будет установлено соединение между SSF/CCF и SRF. В ходе диалога между пользователем и функциональным блоком SRF последний может передавать пользователю необходимые речевые подсказки и принимать цифры, которые тот передает в виде сигналов DTMF. Принятые цифры пересылаются к SCF. Когда специализированные ресурсы больше не требуются, SCF передает к SSF/CCF запрос освободить SRF.



Рис. 2.3.8 Модель SRF

В SRF содержатся следующие функциональные компоненты: менеджер доступа FEAM (Functional entity access manager), менеджер ресурсов RM (Resource manager) и специализированные ресурсы. FEAM обеспечивает обмен информацией с другими функциональными объектами. RM содержит средства поиска свободного ресурса нужного типа, контроля состояния каждого ресурса (занят, свободен, блокирован) и управления их работой. В CS-1 предусматриваются ресурсы четырех типов: приемники многочастотного набора, генераторы акустических сигналов и автоинформаторы, приемопередатчики сообщений (речевых, электронной почты и др.), синтезаторы и распознаватели речи.

Для управления соединением SSF/CCF со специализированными ресурсами между SSF/CCF и SRF имеется логическая связь. Информация, касающаяся управления соединением, передается к CCF/SSF от SCF. После установления соединения SCF направляет к SRF инструкции для управления ресурсами.

2.3.5 Модель SCF

Основное назначение SCF состоит в исполнении программ логики услуг. Модель функций управления услугами приведена на рисунке 2.3.9. Показанная на рисунке платформа образует среду, в которой работают программы логики услуг, реализующие управление услугами, причем одновременно могут работать несколько таких программ.

Менеджер исполнения логики услуг SLEM (Service logic execution manager) управляет всеми действиями, обеспечивающими работу логики услуг. В состав SLEM входят отдельные программы логики

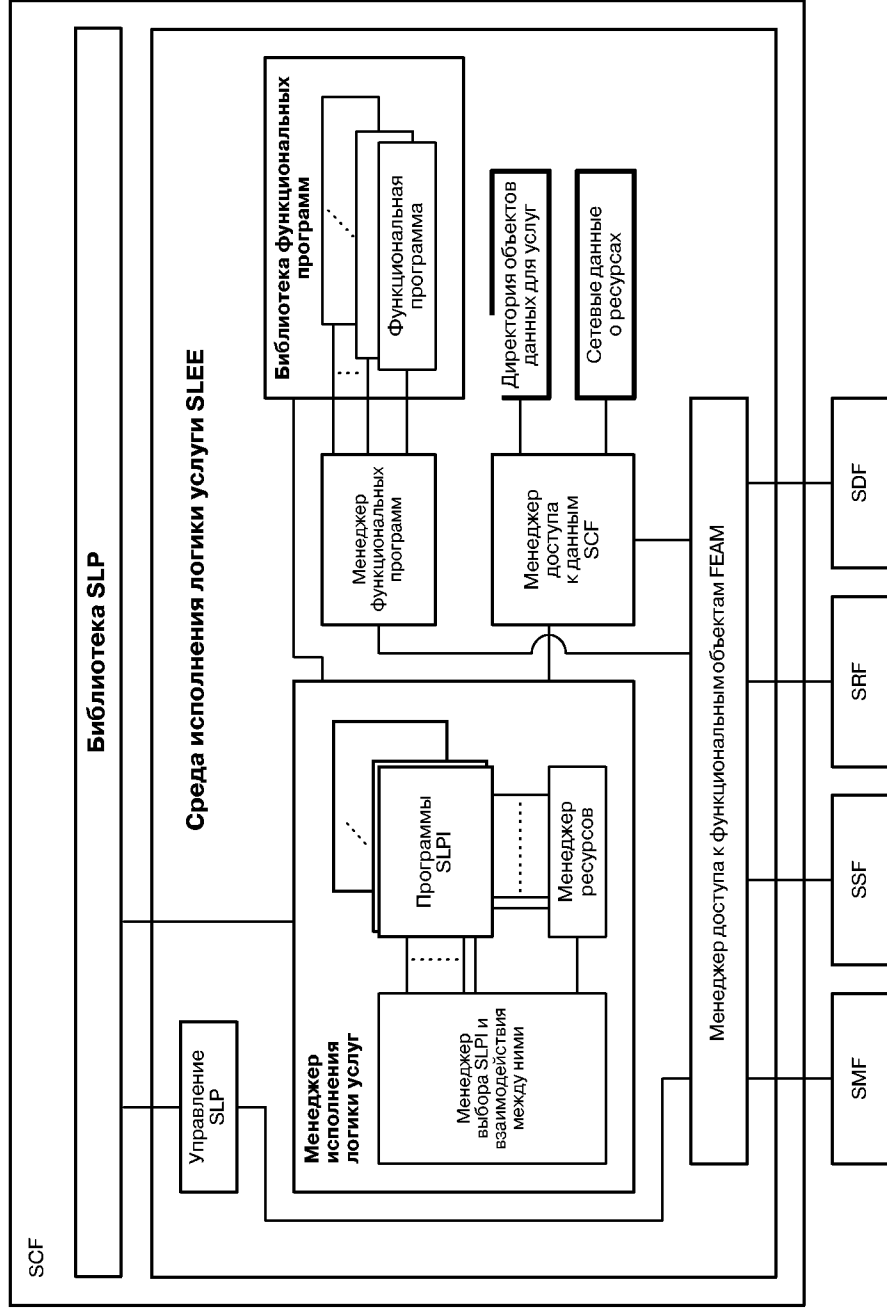


Рис. 2.3.9 Модель SCF

услуг SLPI (Service logic program instances), менеджер выбора программ и взаимодействия между ними SLSIM (Service logic selection/interaction manager), а также менеджер ресурсов. При выполнении программ логики услуг менеджер исполнителя SLEM взаимодействует с имеющимися в SCF менеджерами доступа к данным и к объектам.

Менеджер выбора/взаимодействия SLSIM выбирает нужную программу логики услуг и управляет одновременным исполнением и/или очередностью исполнения нескольких таких программ внутри одного SCF.

Программа логики услуги SLP (Service logic program) представляет собой прикладную программу (или группу программ), которая работает в среде исполнения логики услуги SLEE (Service logic execution environment). SLP содержит логические конструкции, которые управляют последовательностью выполнения атрибутов услуги и запускают программы доступа к сетевым ресурсам и к данным, необходимым для формирования услуги. В противоположность SLP, представляющей собой, как бы статическую библиотечную программу, SLPI есть динамический объект – программа, запущенная для выполнения с конкретными параметрами.

Как мы помним, формирование и предоставление услуги IN сводится к выполнению функциональными объектами сети определенных (стандартизованных) действий. Для того, чтобы вызвать заданную последовательность действий внутри функциональных объектов глобальной распределенной плоскости, SLPI используют функциональные программы. Вызванная таким образом последовательность действий этих объектов обеспечивает выполнение множества функций, определенного на глобальной функциональной плоскости для независимых конструктивных блоков SIB.

Менеджер ресурсов управляет распределением ресурсов внутри SCF и доступом к сетевым ресурсам при исполнении SLPI. Менеджер доступа к данным SCF выполняет функции, необходимые для хранения в SCF постоянной информации, для управления этой информацией и для доступа к ней, а также для доступа к информации, которая размещена в SDF. Существует две структуры, в которых содержатся данные, относящиеся к SDF: директория объектов данных для услуг и сетевые данные о ресурсах.

Директория объектов данных для услуг содержит средства адресации SCF к нужному объекту данных. Менеджер доступа к данным, используя директорию, обеспечивает SLEM глобальным и унифицированным представлением объектов данных в сети.

Сетевые данные о ресурсах представляет собой структуру, в которой находится информация о размещении в сети ресурсов, доступных программам логики услуг, и о функциональных возможностях

каждого из этих ресурсов. Структура предусматривает средства адресации соответствующего функционального объекта (например, SRF) к ресурсам, обладающим нужными возможностями.

Менеджер функциональных программ служит для приема функциональных программ и размещения их в библиотеке. Менеджер доступа к функциональным объектам необходим для обмена информацией (посредством сообщений) между SLEM и другими функциональными объектами. Менеджер программ логики услуг SLP управляет функциями приема и распределения программ SLP при связи с другими объектами, а также добавлением и удалением определенных SLP.

2.3.6 Модель SDF

SDF содержит данные, относящиеся к программам SLPI, и управляет этими данными. Приведенная на рисунке 2.3.10 модель SDF содержит следующие компоненты.



Рис. 2.3.10 Модель SDF

Менеджер данных SDF выполняет функции, необходимые для хранения внутренней информации SDF, для управления этой информацией и для доступа к ней. Например, если данные физически структурированы в виде базы данных, он управляет доступом к ней с использованием стандартного языка доступа к базам данных SQL. Менеджер доступа к функциональным объектам выполняет функции, необходимые менеджеру данных SDF для обмена информации

ей с SCF, SDF и SMF. Менеджер защиты содержит функции, необходимые, например, для проверки прав доступа к данным того или иного вида.

Данные, с которыми работает SDF, делятся на статические (только читаемые программами логики услуг) и динамические (изменяемые этими программами). Более детально классификация выглядит следующим образом (см. рис. 2.3.10).

В SDF хранятся три типа данных. Аутентификационные данные вида PIN-кодов и счетчиков неудачных попыток доступа относятся к группе данных, связанных с правами пользователя. Рабочие данные (такие как ссылка на класс объекта, данные управления доступом и др.) представляют собой данные, используемые для внутренних целей SDF. Данные об услугах задаются при введении каждой услуги в эксплуатацию и представляют собой сведения об абонентских профилях услуг и информацию, определяемую на этапе заключения соглашения с поставщиком услуги.

2.3.7 Стадия 2 описания блоков SIB

2.3.7.1 Принципы декомпозиции

Глобальная функциональная плоскость дает абстрактное представление сетевых возможностей в виде, наиболее удобном для восприятия разработчиком услуги. Напомним, что на этой плоскости определены базовые единицы модульности – независимые от услуг конструктивные блоки SIB, которые используются при создании услуг. Такие блоки являются абстракцией независимых от конкретной услуги сетевых ресурсов и функций.

Любой конструктивный блок представляет одну из множества функциональных возможностей сети связи, реализуемую посредством заданной группы элементарных операций. Например, если способность элемента сети взаимодействовать с конечным пользователем услуги моделируется как SIB, то элементарными операциями, формирующими такой конструктивный блок, могут быть операции «передать речевое сообщение», «принять набранные пользователем цифры» и т.п.

Для доступа к внутренним ресурсам каждого блока SIB определен стандартный интерфейс. Через такой интерфейс производится управление очередностью вызова элементарных операций программами логики услуги, за счет чего и обеспечивается выполнение множества функций, составляющих данную услугу. Так как глобальная функциональная плоскость скрывает логическое и/или физическое распределение сетевых ресурсов, то на ней блоки SIB представляются пользователю (конструктору услуги) как «монолитные» объекты.

В действительности IN не является такой монолитной структурой, как она представлена на глобальной функциональной плоскости. Скорее, это – распределенная конфигурация, представленная на физической плоскости. Конструктивный блок, который, с целью упрощения конструирования услуги, определен на глобальной плоскости как неделимый объект, в действительности может быть реализован несколькими программными процессами, размещенными в разных физических элементах сети. Следовательно, для координации действий, которые выполняет такая распределенная конфигурация логических объектов, расположенных в разных физических элементах сети, необходим протокол (набор правил обмена информацией).

Распределенная функциональная плоскость, как часть концептуальной модели INCM, задумана с целью облегчить спецификацию требований к физическим объектам и протоколам, необходимым для реализации каждого SIB. Эта плоскость аналогична модели, создаваемой на шаге 2.1 Этапа 2 трехэтапной модели спецификации услуг связи, определенной рекомендацией I.130 ITU-T (см. главу 1.3).

Модель спецификации услуг, введенная в рекомендации I.130, определяет некоторое количество типов функциональных объектов. В соответствии с этой моделью функциональный объект представляет собой группу функций, используемых при реализации какого-либо функционально законченного аспекта услуги связи. Группировка функций по объектам произведена так, что модель, с одной стороны, позволяет моделировать атрибуты услуг, раскрывая распределенную природу конструктивных блоков SIB, а с другой стороны, постулирует невозможность дальнейшего разделения функциональных объектов по нескольким физическим узлам.

При представлении SIB на распределенной функциональной плоскости производится их декомпозиция на группы взаимодействующих функциональных средств, которые могут моделироваться в виде пар «клиент-сервер», причем члены каждой пары находятся в функциональных объектах разного типа (см. рис. 2.3.11).

Концепция SIB тесно связана с принципами объектно-ориентированного программирования, которые предполагают, что объект получает доступ к нужным ему функциональным возможностям через стандартный интерфейс. Предоставляемые объектом услуги в действительности могут обеспечиваться внутренней декомпозицией этого объекта на группу объектов, взаимодействующих между собой для достижения результирующего эффекта.

В частности, при декомпозиции вида «клиент-сервер», два взаимодействующих объекта определяются так, что один из них, называемый клиентом, всегда использует услуги, предоставляемые другим объектом – сервером. Взаимодействие между этими объектами реализуется средствами специального протокола. Интерфейс «кли-

ент-сервер» представляет собой набор операций, которые объект «клиент» привлекает для выполнения в объекте «сервер» и который необходим для инкапсуляции всего объекта таким образом, что его результирующий интерфейс остается неизменным.

Интерфейс между программой логики услуги и блоком SIB определен на глобальной функциональной плоскости. Этот интерфейс скрывает внутреннюю структуру блока (см. левую часть рис. 2.3.11), и, вместе с тем, он достаточно удобен для распределенного представления (см. правую часть рис. 2.3.11) функциональных возможностей, предоставляемых SIB, в виде пары «клиент-сервер».

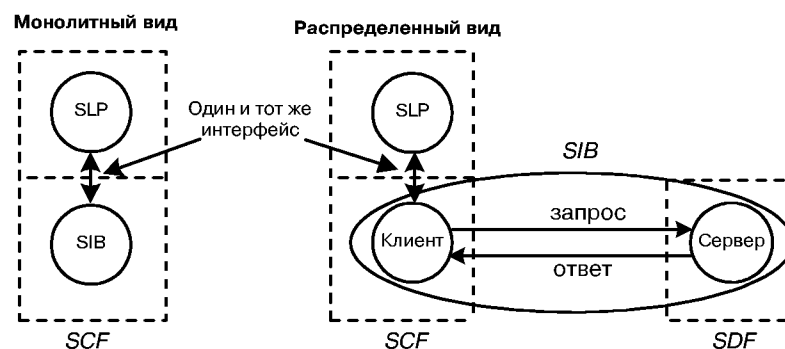


Рис. 2.3.11 Декомпозиция блока SIB в виде пары "клиент-сервер"

Например, как будет показано ниже, блок SIB TRANSLATE, выполняющий пересчет номерной информации, может быть смоделирован как реализуемый частично в SCF и частично в SDF. Запрос пересчета производится со стороны SCF (клиент), а выполнение пересчета производится средствами SDF (сервер).

Программа логики услуги взаимодействует с SIB через стандартный интерфейс, и ей «кажется», что она имеет дело с монолитной структурой. Это обеспечивается тем, что даже в распределенном представлении программа взаимодействует только с частью блока «клиент», реализованной в SCF. Для выполнения нужной процедуры клиент взаимодействует со своим сервером в SDF. В нашем примере вся обработка будет произведена сервером в SDF, при этом часть «клиент» в SCF всего лишь обеспечивает удаленный доступ к серверу. Таким образом, для обеспечения прозрачности переноса запрошенных от SIB функций и результатов их выполнения специфицируются информационные потоки между частью «клиент» и частью «сервер».

Так как всем блокам SIB соответствует одна и та же модель – распределенная функциональная плоскость, то каждому типу функционального объекта назначаются части «клиент» и части «сервер» из

разных SIB. Например, SSF содержит серверы для блоков BCP, CHARGE и STATUS NOTIFICATION, а SCF является частью «клиент» элементарной операции Connect (Соединить) блока BCP и частью «сервер» для элементарной операции Translate (Пересчитать) блока TRANSLATE.

Ниже приведено распределенное представление всех SIB, введенных на глобальной функциональной плоскости CS-1. Название информационного потока на диаграммах пишется заглавными буквами над стрелкой, указывающей направление потока, тогда как этикетка, расположенная под стрелкой, пишется строчными буквами. Название потока и его этикетка определяются тем действием, для выполнения которого он предназначен.

В каждой колонке, отражающей поведение определенного функционального объекта, действия, которые показаны ниже линии, связанной с приемом запроса от пользователя SIB (в большинстве случаев в качестве пользователя выступает логика услуги SL) или информационного потока от другого функционального объекта, не могут быть выполнены раньше, чем произойдут эти события. Действия, которые показаны над линией, связанной с реакцией пользователя SIB или с формированием информационного потока, должны быть выполнены прежде, чем произойдут эти события.

Входными и выходными событиями, используемыми при формировании диаграмм информационных потоков, являются данные Этапа 1 спецификации услуги, то есть запрос услуги, направляемый к сети от пользователя, и соответствующая реакция сети в процессе предоставления услуги. Информационные потоки, которые передаются в направлении от пользователя, на Этапе 1 имеют имена вида XXXX.req или XXXX.resp, а передаваемые в обратном направлении – имена вида XXXX.ind или XXXX.conf.

Этикетки потоков на диаграммах ассоциируются с элементами request/indication (запрос/индикация) и response/confirmation (отклик/подтверждение) имен примитивов, используемых при описании взаимодействия верхних уровней модели OSI с нижележащими уровнями систем сигнализации, которые обеспечивают поддержку информационных потоков на физической плоскости.

Если информационный поток идентифицируется как подтверждаемый, он требует отклика в обратном направлении. Подтверждаемый информационный поток несет запрос какого-то действия (в одном направлении), а отклик несет подтверждение того, что это действие выполнено (в обратном направлении). Такие потоки обычно требуются для целей синхронизации, например, при запросе размещения и/или освобождения разделяемых ресурсов.

Когда взаимодействующие функциональные объекты реализованы в разных физических элементах, для переноса информационных

потоков используется ресурс системы сигнализации. Если же объекты реализованы в одном сетевом узле, то потоки являются внутренними и использование системы сигнализации не требуется.

В соответствии со своим определением, каждый независимый конструктивный блок, созданный для определенной цели, может впоследствии быть многократно использован совместно с другими SIB при создании новых услуг. Перечень SIB и свойства каждого SIB определены на глобальной функциональной плоскости. Как мы помним, для CS-1 определено 14 блоков SIB и SIB базового процесса обслуживания вызова BCP.

На распределенной функциональной плоскости определены требования к интерфейсу каждого SIB, а также реализующие этот SIB функциональные объекты, информационные потоки между объектами и их действия, необходимые для реализации функциональных возможностей каждого конструктивного блока.

Возможные виды связи между функциональными объектами и соответствующие информационные потоки показаны на рисунке 2.3.12. Для их обозначения используется латинская буква «r» с числовым индексом. Все возможные информационные потоки между двумя функциональными объектами характеризуют тип связи этой пары объектов. Связи, относящиеся к базовому процессу обслуживания вызова, определены в рекомендации Q.71. Мы же далее акцентируем внимание на тех связях, которые стандартизованы применительно к услугам IN. Для CS-1 таковыми являются:

- r3: поток между SSF и SCF;
- r4: поток между SCF и SDF;
- r5: поток между SCF и SRF;
- r6: поток между CCF и SRF.

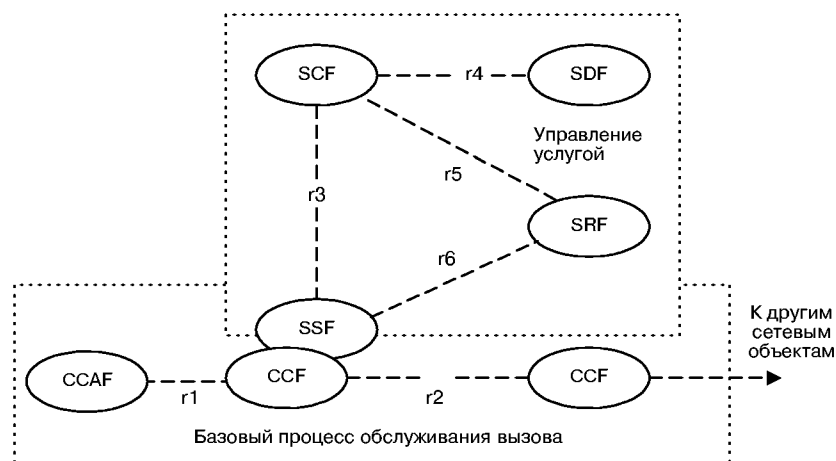


Рис. 2.3.12 Связи между функциональными объектами

Все действия, выполняемые функциональными объектами в связи с приемом информационного потока, нумеруются четырехзначными числами XYZ, цифры которых обозначают:

X – номер функционального объекта, который равен 2 для CCF/SSF, 3 для SRF, 4 для SDF, 9 для SCF;

YY – порядковый номер, присвоенный каждому из SIB (например, YY для BCP равен 00, для блока ALGORITHM – 01 и т.д.);

Z – порядковый номер, присвоенный каждому из действий, которые выполняет функциональный объект X для поддержки SIB с номером YY.

Стадия 2 описания каждого из 14 SIB выполняется по стандартному шаблону, содержащему следующие элементы: краткое определение, информационные потоки и вызываемые ими действия внутри функциональных объектов, SDL-диаграммы. Важно отметить, что информационные потоки определяют лишь смысловое содержание передаваемой информации. Способ передачи потоков определяется протоколом физического уровня.

Детальному рассмотрению прикладного протокола Интеллектуальной сети INAP посвящена часть 3 данной книги. Чтобы чрезмерно не утомлять читателя, в приводимые ниже описания информационных потоков включены только те информационные элементы, которые поддерживаются вариантом ETSI протокола INAP, SDL-диаграммы опущены, а действия объектов показаны непосредственно на диаграммах информационных потоков.

2.3.7.2 Спецификация информационных потоков

Блок ALGORITHM обеспечивает реализацию математического алгоритма обработки данных, поступивших на его вход, и выдает полученный результат. В наборе CS-1 такие возможности полностью реализуются в SCF, а потому для этого блока никаких информационных потоков не определено.

Блок CHARGE определяет некоторые характеристики начисления платы за услуги IN (например, специальный тариф, оплата вызванной стороной, разделение платы между сторонами). Для этого блока определено четыре типа информационных потоков (см. рис. 2.3.13 – 2.3.16):

- тип 1 – поддерживает создание записи о начисленной плате в SSF;
- тип 2 – поддерживает передачу информации о начисленной плате от SSF к сетевым функциям (например, в исходящей ATC);
- тип 3 – запрашивает уведомление о возникновении в SSF событий, связанных с начислением платы;

- тип 4 – запрашивает отчет о созданных в SSF записях о начисленной плате.

Потоки перечисленных типов могут поддерживать разные сценарии начисления платы, применяться в любой комбинации и использоваться неоднократно в течение одного сеанса связи.

Примечание: далее при описании информационных элементов потоков будут применяться сокращения (O)-обязательный и (HO)-необязательный информационный элемент.

Поток *Furnish Charging Info* (создать запись о начисленной плате) содержит следующие информационные элементы:

Call ID	
(идентификатор соединения)	(O)
Billing Charging Characteristics	
(данные для счета)	(O)

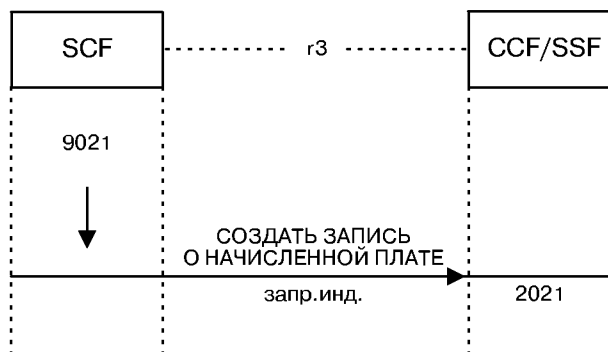


Рис. 2.3.13 Диаграмма информационного потока для блока CHARGE - Тип 1

Send Charging Information (передать данные об оплате) – поток от SCF к SSF, предназначенный для запроса передачи сообщений о начисляемой плате от SSF к сетевым функциям, и содержит следующие информационные элементы:

Call ID	
(идентификатор соединения)	(O)
Billing Charging Characteristics	
(данные для выписки счета)	(O)
Leg ID	
(идентификатор стороны)	(HO)

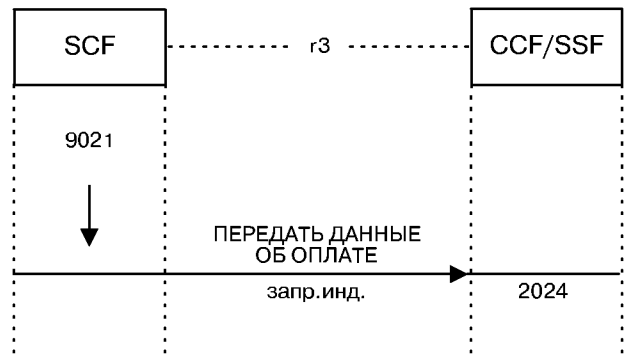


Рис. 2.3.14 Диаграмма информационного потока для блока CHARGE - Тип 2

Поток *Request Notification Charging Event* (запрос уведомления о событии начисления платы) направляется от SCF к SSF для отслеживания событий, связанных с начислением платы, о которых необходимо уведомить SCF, и состоит из одного информационного элемента

SequenceOfChargingEvent
(список событий, связанных с начислением платы) (O)

Поток *Event Notification Charging* (уведомление о событии начисления платы) направляется от SSF к SCF, чтобы сообщить о событии, связанном с начислением платы. Его содержание:

Call ID
(идентификатор соединения) (O)

Event Type Charging
(тип события) (O)

Monitor Mode
(режим наблюдения) (O)

Event Specific Information Charging
(специфическая информация о событии) (NO)

Leg ID
(идентификатор стороны соединения) (NO)

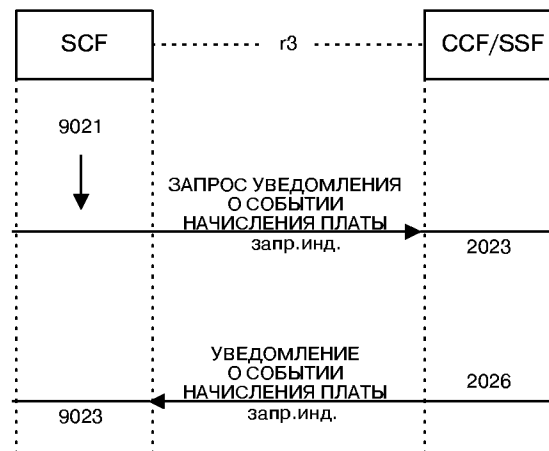


Рис. 2.3.15 Диаграмма информационного потока для блока CHARGE - Тип 3

Поток *Apply Charging* (начать начисление платы) представляет собой запрос от SCF к SSF для взаимодействия с процедурами начисления платы, проводимыми SSF в реальном времени, и содержит следующие информационные элементы:

CallID	
(идентификатор соединения)	(O)
BillingChargingCharacteristics	
(данные для выписки счета)	(O)
PartyToCharge	
(оплачивающая сторона)	(HO)

Ответом SSF на него является направляемый к SCF поток *Apply Charging Report* (отчет о начисленной плате), содержащий информационный элемент:

Call Result	
(Результат)	(O)

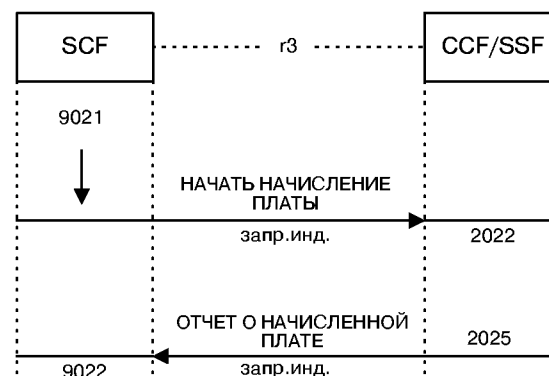


Рис. 2.3.16 Диаграмма информационного потока для блока CHARGE - Тип 4

Блок COMPARE предусматривает сравнение идентификатора с заданным контрольным значением и возвращает одно из решений (больше, меньше или равно). В наборе CS-1 такие возможности полностью реализуются в SCF, а потому для этого блока никаких информационных потоков не определено.

Блок DISTRIBUTION обеспечивает распределение вызовов по разным логическим точкам в зависимости от заданных пользователем параметров. Результатом выполнения блока является один из нескольких логических адресов, к которому вызов должен быть направлен для завершения его обслуживания. В наборе CS-1 такие возможности полностью реализуются в SCF, а потому для этого блока никаких информационных потоков не определено.

Блок LIMIT ограничивает количество вызовов в сети со структурой IN путем просеивания их по определенным параметрам. Просеиванию подлежат только те вызовы, которые требуют привлечения к их обслуживанию функций IN (т.е. процедура просеивания применяется ко всем TDP). Сущность процедуры состоит в том, что в течение заданного периода времени SSF блокирует вызовы, поступающие внутри определенных временных интервалов. Диаграмма информационных потоков блока LIMIT представлена на рисунке 2.3.17.

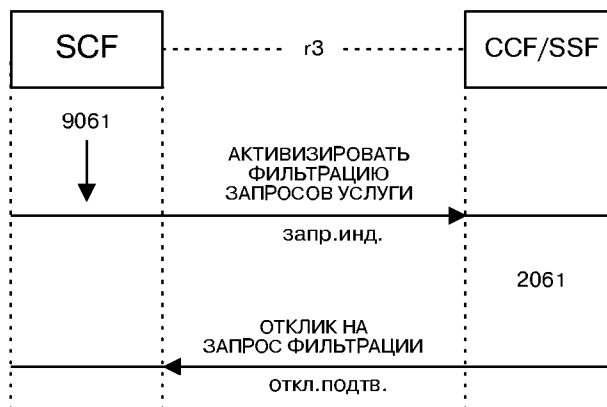


Рис. 2.3.17 Диаграмма информационного потока блока LIMIT

Подтверждаемый информационный поток от SCF к SSF *Activate Service Filtering* (активизировать просеивание запросов услуги) требует просеивания вызовов, которые связаны с обращением к определенной услуге, и подсчета количества таких вызовов. Результат подсчета возвращается в SCF по окончании периода просеивания

(задается либо длительность этого периода, либо момент его окончания). Поток может содержать следующие элементы.

Filtering Timeout (окончание периода просеивания)	(O)
Filtered Call Treatment (действия с заблокированным вызовом)	(O)
Filtering Characteristics (характеристики просеивания)	(O)
Filtering Criteria (критерий просеивания)	(HO)
Start Time (момент начала просеивания)	(HO)

Поток *Service Filtering Response* (отклик на запрос просеивания обращений к услуге) направляется от SSF к SCF по окончании периода просеивания и включает в себя:

Counters Value (значения счетчиков)	(O)
Filtering Criteria (критерий просеивания)	(O)
Response Condition (условия передачи отклика)	(HO)

Блок LOG CALL INFORMATION производит запись подробной информации о каждом вызове. С блоком связаны четыре информационных потока (см. рис. 2.3.18).

Поток *Call Information Request* (запрос информации о вызове) направляется от SCF к SSF с указанием сохранить определенную информацию об одном вызове (и о связи, которую запрашивал этот вызов) и передать эту информацию в SCF по окончании связи и содержит следующие информационные элементы:

Requested Information Type List (запрашиваемая информация)	(O)
Correlation ID (связующий идентификатор)	(HO)

Поток *Call Information Report* (информация о вызове) несет от SSF к SCF запрошенную информацию о вызове и содержит:

Requested Information List (запрошенная информация)	(O)
Correlation ID (связующий идентификатор)	(HO)

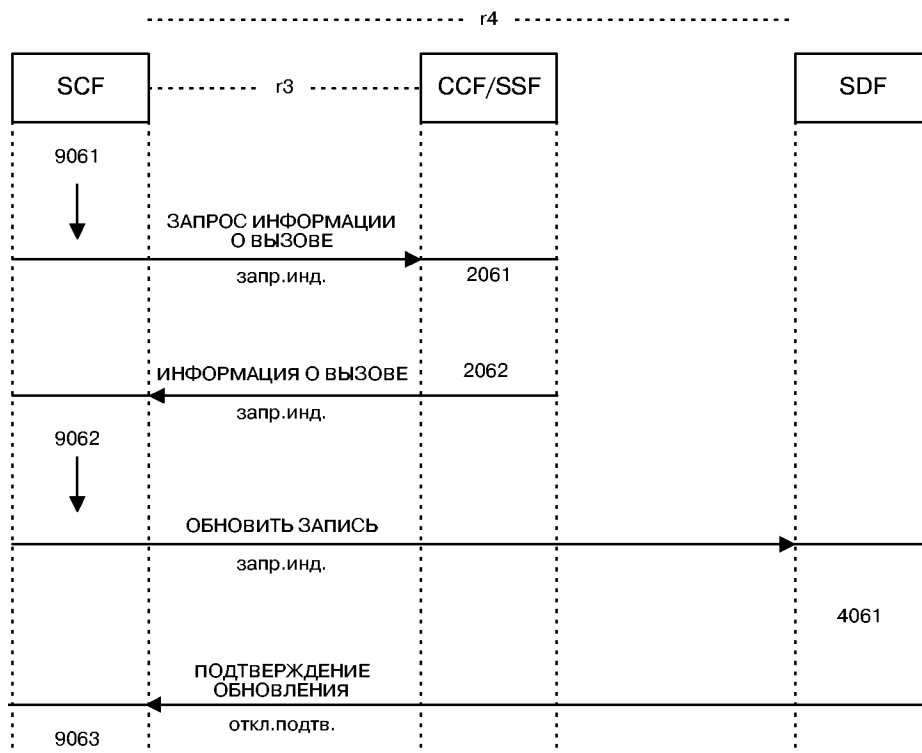


Рис. 2.3.18 Диаграмма информационных потоков блока
LOG CALL INFORMATION

Modify Entry (обновить запись) – подтверждаемый информационный поток (необязательный), передаваемый от SCF к SDF для обновления данных, который содержит следующие информационные элементы:

Authorized Relationships ID (идентификатор разрешенной связи между SL и базой данных, соответствует идентификатору транзакции TCAP)	(O)
Object (объект)	(O)
Changes (изменения в объекте)	(O)
Selection (данные, подлежащие изменению)	(HO)

Modify Entry Result (подтверждение обновления) – поток, передаваемый от SDF к SCF в качестве подтверждения проведенного обновления данных. Содержит один или два элемента:

Authorized Relationships ID
(идентификатор разрешенной связи
между SL и базой данных) (O)

Information
(информация о результате
выполнения изменений) (HO)

Блок QUEUE обеспечивает установку вызовов на ожидание. Это включает в себя: предоставление вызову ресурса, если он доступен, установку в очередь при недоступности ресурса, передачу речевого уведомления пользователю, выбор из очереди с предоставлением ресурса, который стал доступен (см. рис. 2.3.19 – 2.3.20). Состояние ресурсов определяется путем наблюдения за событиями в BCSM.

Поток *Hold Call In Network* (удержать вызов) направлен от SCF к CCF/SSF и используется, чтобы информировать CCF/SSF о том, что вызов устанавливается на ожидание, в связи с чем CCF/SSF должны принять на себя все действия, связанные с поддержкой ожидания в сети. Информационные элементы потока:

Call ID
(идентификатор вызова) (O)

Hold Cause
(причина удержания) (HO)

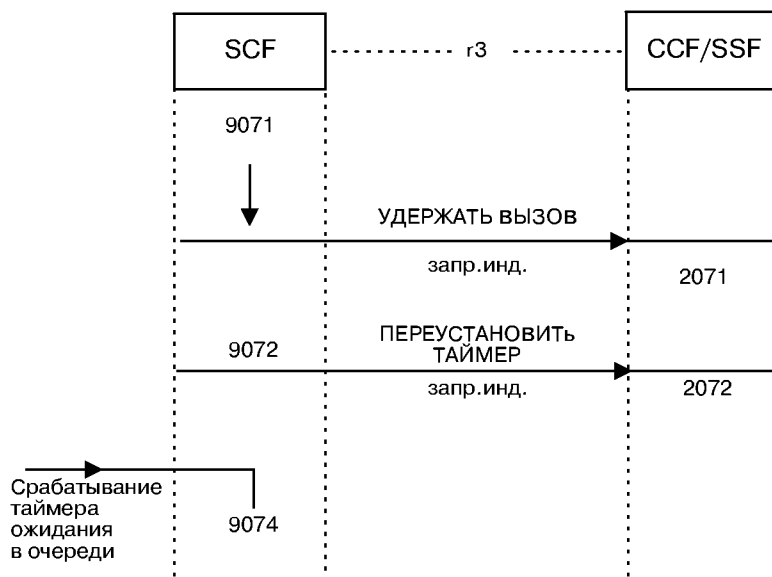


Рис. 2.3.19 Диаграмма информационных потоков блока QUEUE (без передачи речевого сообщения пользователю)

Поток *Reset Timer* (переустановить таймер) используется SCF для передачи в CCF/SSF запроса перезапустить таймер и содержит следующие информационные элементы.

Call ID (идентификатор вызова)	(O)
Timer ID (идентификатор таймера)	(O)
Timer value (выдержка времени таймера)	(O)

Поток *Event Report BCSM* (отчет о событии BCSM) используется CCF/SSF, чтобы уведомить SCF о событии в процессе обслуживания вызова, например, об отказе пользователя от ожидания связи.

Call ID (идентификатор вызова)	(O)
Event type BCSM (тип события)	(O)
Misc Call Info (дополнительная информация о вызове, напр. Список триггерных точек)	(HO)
Event Specific Info BCSM (информация о событии)	(HO)
Leg Id (идентификатор стороны соединения)	(HO)

Поток *Request Report BCSM Event* (запрос уведомления о событии BCSM) используется, чтобы запросить CCF/SSF уведомлять SCF о некоторых событиях в BCSM.

Call ID (идентификатор вызова)	(O)
BCSM Event list (список событий)	(O)
BCSM Event Correlation Id (идентификатор, связывающий данный запрос с определенным откликом)	(HO)

Поток *Connect To Resource* (подключить к ресурсу) используется SCF, чтобы дать указание SSF/CCF установить соединение между абонентом и SRF.

Call ID (идентификатор вызова)	(O)
IP Routing Address (адрес IP)	(HO)
Leg ID (идентификатор стороны соединения)	(HO)
Service Interaction Indicators (индикаторы взаимодействия услуг)	(HO)

Поток *Disconnect Forward Connection* (отключить ресурс) используется SCF, чтобы дать указание SSF/CCF нарушить соединение с SRF.

Call ID
(идентификатор вызова) (O)

Поток *PlayAnnouncement* (передать речевое сообщение) используется SCF, чтобы дать указание SRF воспроизвести нужное речевое сообщение.

SRF Connect ID
(идентификатор соединения с SRF,
соответствует идентификатору
транзакции TCAP) (O)

Information To Send
(тип речевого сообщения) (O)

Disconnect From IP Forbidden
(запрет разъединения со стороны IP) (HO)

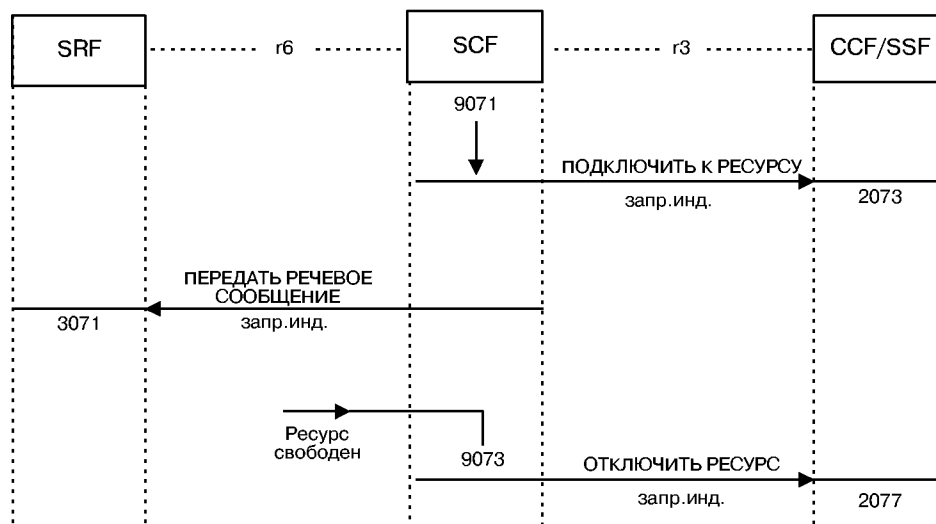


Рис. 2.3.20 Диаграмма информационных потоков блока QUEUE (с передачей речевого сообщения пользователю)

Блок SCREEN обеспечивает для SCF возможность сравнения идентификатора со списком, размещенным в SDF. Диаграмма информационного потока данного блока приведена на рисунке 2.3.21.

Search (поиск) – это подтверждаемый информационный поток, который направляется от SCF к SDF для запроса сравнения идентификатора со списком. *Search Result* (результат поиска) – подтверждение со стороны SDF, содержащее результат выполнения запроса. Потoki содержат следующие информационные элементы:

Authorized Relationship ID (идентификатор разрешенной связи между SL и базой данных)	(O)
Base Object (объект в базе данных)	(O)
Subset (тип запрашиваемой информации)	(O)
Selection (искомые данные внутри объекта)	(O)
Filter (правило просеивания)	(HO)
Matched Values Only (только совпавшие значения)	(HO)
Search Info (искомая информация)	(O)

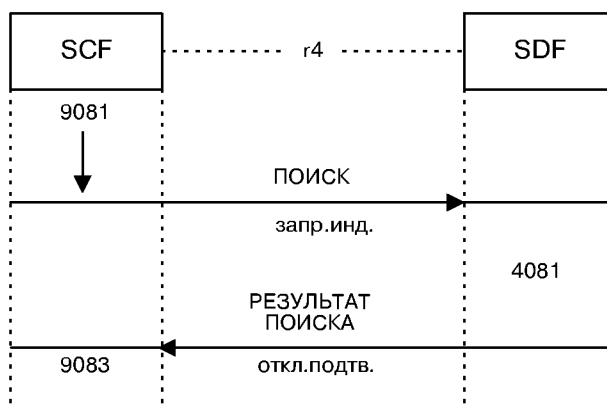


Рис. 2.3.21 Диаграмма информационного потока блока SCREEN

Блок SERVICE DATA MANAGEMENT предоставляет SCF функции запроса, замены, увеличения или уменьшения значений данных, находящихся в SDF. Диаграммы информационных потоков этого блока приведены на рисунке 2.3.22 (запрос данных) и рисунке 2.3.23 (операции с данными).

Search (поиск) – подтверждаемый поток, который передает SCF, чтобы получить данные от SDF, *Search Result* (результат поиска) – соответствующий отклик SDF. *Update* (обновить) – подтверждаемый информационный поток от SCF к SDF с требованием выполнить нужные действия с данными внутри идентифицированного объекта, *Update Result* (подтверждение обновления) – подтверждение выполнения этих действий. Пара *Update Data* и *Update Result* может быть сле-

дующего вида: *Modify Entry/Modify Entry Result* (изменить запись/результат изменения), *Add Entry/Add Entry Result* (добавить запись/результат добавления) или *Remove Entry/Remove Entry Result* (удалить запись/результат удаления). Поток *Remove Entry Result* не содержит информационных элементов.

Информационные элементы потока *Add Entry*:

Authorized Relationship ID (идентификатор разрешенной связи между SL и базой данных)	(O)
Object (объект)	(O)
Entry (создаваемая запись)	(O)

Информационный элемент потока *Add Entry Result*:

Authorized Relationship ID (идентификатор разрешенной связи между SL и базой данных)	(O)
--	-----

Информационные элементы потока *Remove Entry*:

Authorized Relationship ID (идентификатор разрешенной связи между SL и базой данных)	(O)
Object (объект)	(O)
Entry (удаляемая запись)	(O)

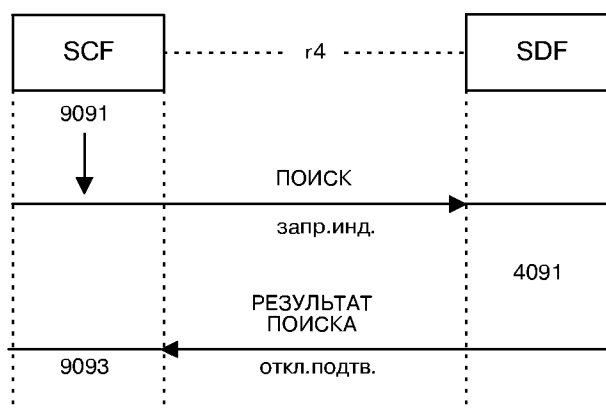


Рис. 2.3.22 Диаграмма информационного потока блока **SERVICE DATA MANAGEMENT** (запрос данных)

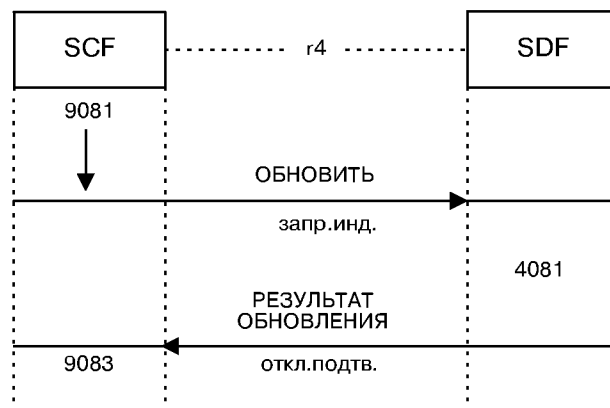


Рис. 2.3.23 Диаграмма информационного потока блока **SERVICE DATA MANAGEMENT** (операции с данными)

Блок STATUS NOTIFICATION предоставляет SCF возможность следить за статусом соединений или ресурсов, и, в качестве дополнительной возможности, сохранения этой информации в SDF (см. рис. 2.3.24). Например, блок используется, чтобы определить, свободен или занят вызываемый абонент. Для этого SCF запрашивает CCF/SSF сведения о статусе соединения или ресурса путем передачи информационного потока *Request Status Report* (запрос данных о статусе). Запрос требует действия одного из трех типов (определить статус в данный момент, обнаружить переход в определенный статус, вести непрерывный мониторинг) и содержит следующие информационные элемент:

Monitor Type (тип мониторинга)	(O)
Monitor Duration (длительность мониторинга)	(HO)
Resource ID (идентификатор ресурса)	(O)
Resource Status (статус ресурса)	(HO)
Correlation ID (связующий идентификатор)	(HO)

В зависимости от запрошенного типа действия уведомление *Status Report* (отчет о статусе) формируется либо немедленно, либо после того, как ресурс перейдет в указанный статус, либо всякий раз, когда статус ресурса изменяется. Отчет о статусе содержит

Resource Status (статус ресурса)	(O)
Correlation ID (связующий идентификатор)	(HO)

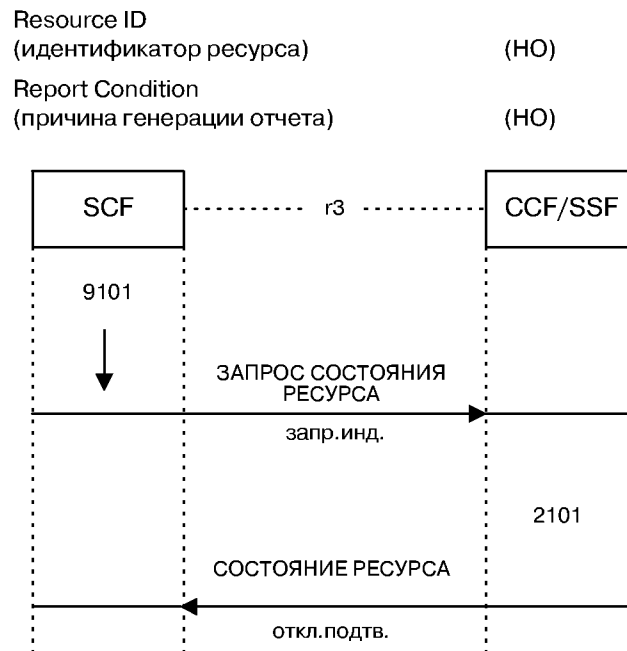


Рис. 2.3.24 Диаграмма информационного потока блока STATUS NOTIFICATION (извещение о статусе в данный момент)

Блок TRANSLATE обеспечивает предоставление данных, необходимых для пересчета адресной информации. Наиболее часто используется для преобразования логического номера в физический номер вызываемой стороны на основании определенных критериев. Диаграмма информационного потока этого блока приведена на рисунке 2.3.25.

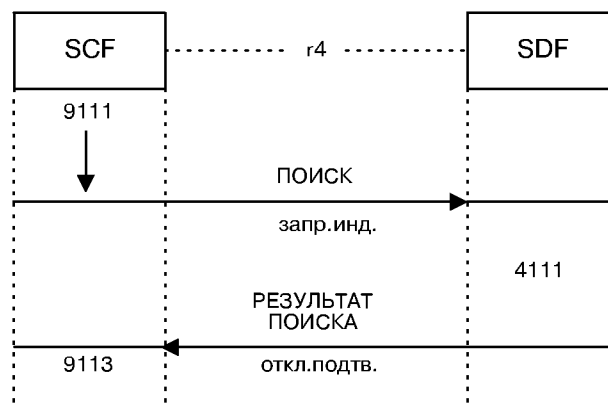


Рис. 2.3.25 Диаграмма информационного потока блока TRANSLATE

Поток *Search* (поиск) генерируется в SCF с целью поиска находящегося в SDF объекта заданного типа (например, объекта «физический номер») по имеющимся неполным данным, например, по логическому номеру. Подтверждением служит поток *Search Result* (результат поиска), содержащий результат выполнения этой операции (см. описание блока ОЧЕРЕДЬ).

Блок VERIFY обеспечивает синтаксическую проверку входной информации. В наборе CS-1 такие возможности полностью реализуются в SCF, а потому для этого блока никаких информационных потоков не определено.

Блок AUTHENTICATE предоставляет SCF возможность установить от имени пользователя связь между логикой услуги и SDF, запросив тот или иной механизм проверки подлинности данных, которыми пользователь себя идентифицировал. Диаграмма информационного потока блока AUTHENTICATE представлена на рисунке 2.3.26.

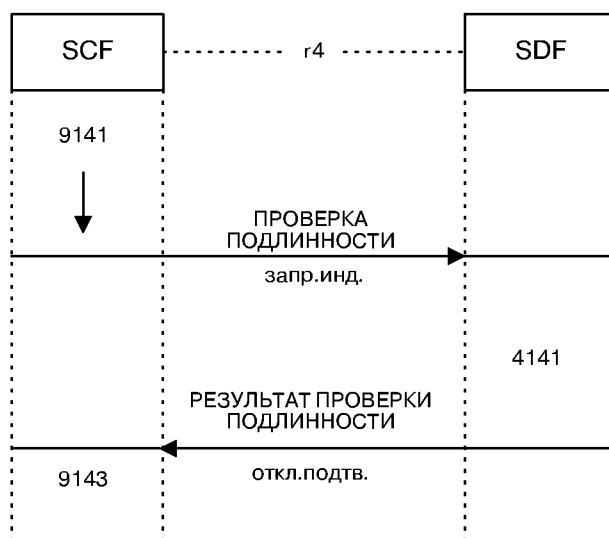


Рис. 2.3.26 Диаграмма информационного потока блока AUTHENTICATE

Информационный поток *Authenticate* (проверка подлинности) передается от SCF к SDF с целью установить разрешенную связь. *Authentication Result* (результат проверки подлинности) передается в обратном направлении и содержит

Authorized Relationship ID (идентификатор разрешенной связи между SL и базой данных)	(O)
Authentication Information (информация, необходимая для проведения проверки подлинности)	(O)

2.3.7.3 Блок USER INTERACTION

2.3.7.3.1 Основные функции

Блок USER INTERACTION выполняет функции, используя которые, SCF управляет подключением пользователя к SRF, передачей речевых сообщений и, в некоторых случаях, приемом информации от пользователя в форме многочастотных сигналов. Диаграммы информационных потоков этого блока представлены на рисунках 2.3.27 и 2.3.28. Поток *Connect To Resource* (подключить к ресурсу) используется SCF, чтобы запросить у CCF/SSF соединение между пользователем и SRF, и содержит

Call ID (идентификатор вызова)	(O)
IP Routing Address (адрес IP)	(HO)
Leg ID (идентификатор стороны)	(HO)
Service Interaction Indicators (индикаторы взаимодействия услуг)	(HO)

Поток *Setup* (установить соединение) – это подтверждаемый информационный поток, несущий сведения, которые нужны для организации соединения с нужным выходом SRF. На физическом уровне для переноса потока *Setup* преимущественно используется система сигнализации DSS1.

Prompt And Collect User Information (уведомление пользователя о том, какая нужна информация, и прием ее от пользователя) представляет собой подтверждаемый информационный поток, передаваемый от SCF к SRF с указанием дать пользователю речевую подсказку (используя установленное между ним и SRF соединение) и принять от него информацию для последующей передачи к SCF. Содержит следующие информационные элементы:

SRF Connect ID (идентификатор соединения с SRF)	(O)
Collected Info (принимается информация)	(O)
Disconnection from IP Forbidden (разъединение со стороны IP запрещено)	(O)
Information To Send (тип информации для передачи)	(HO)

Подтверждением служит поток *Collected User Information* (полученная от пользователя информация), состоящий из

SRF Connect ID (идентификатор соединения с SRF)	(O)
Received Information (принятая от пользователя информация)	(O)

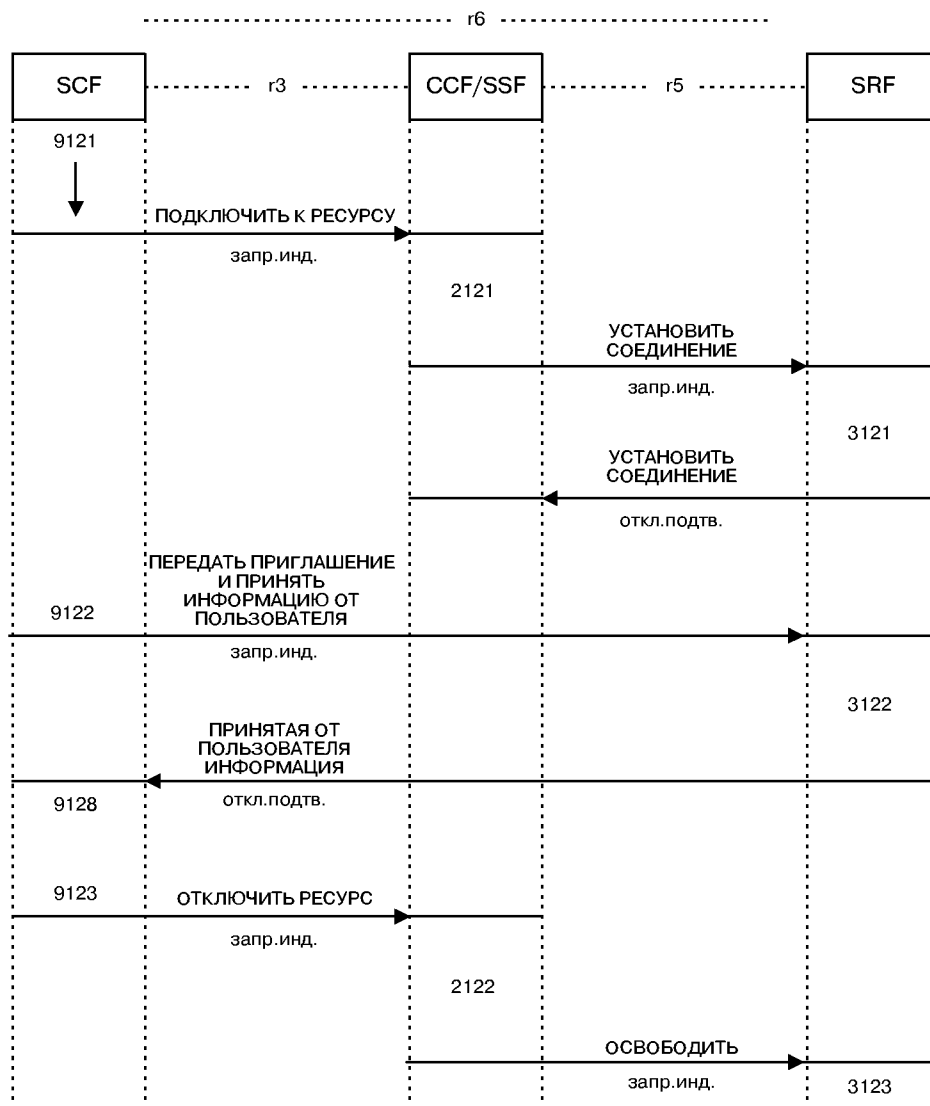


Рис. 2.3.27 Диаграмма информационных потоков блока USER INTERACTION

Информационный поток *Play Announcement* (передать речевое сообщение) направляется от SCF в сторону SRF с целью указать, какое именно речевое сообщение следует передать пользователю по установленному между ним и SRF соединению. Поток состоит из

SRF Connect ID	
(идентификатор соединения с SRF)	(O)
Information To Send	
(тип информации для передачи)	(O)

Disconnection from IP Forbidden
(разъединение со стороны IP запрещено) (O)

Request Announcement Completed Indication
(запрос индикации окончания
передачи речевого сообщения) (O)

В качестве опции к SCF (по окончании передачи речевого сообщения) может быть передано подтверждение – поток *SRF Report* (отчет SRF) с информационным элементом.

SRF Connect ID
(идентификатор соединения с SRF) (O)

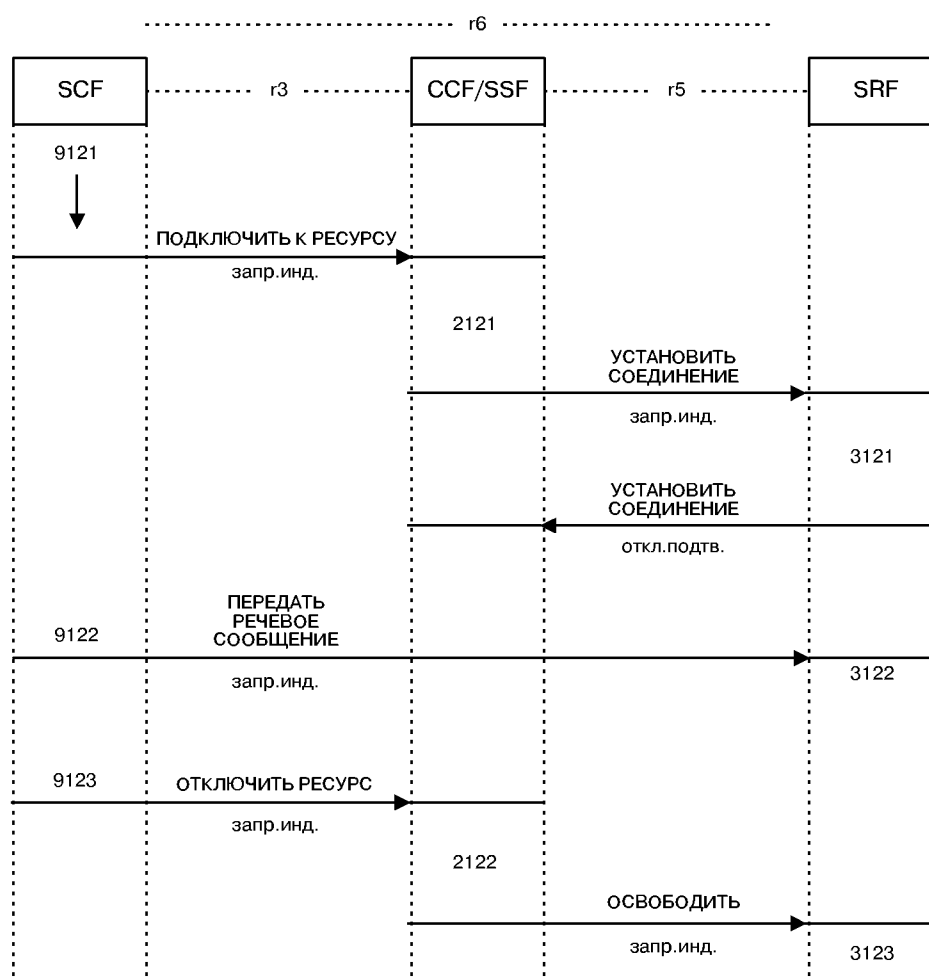


Рис. 2.3.28 Диаграмма информационных потоков блока USER INTERACTION

Информационный поток *Cancel Announcement* (отмена передачи речевого сообщения) используется SCF, чтобы прекратить передачу пользователю речевого сообщения со стороны SRF. Содержит информационный элемент:

Operation Identifier
(идентификатор операции) (O)

Disconnect Forward Connection (нарушить соединение) используется SCF для передачи к SSF/CCF указания нарушить соединение между пользователем и SRF и содержит

SRF Connect ID
(идентификатор соединения с SRF) (O)

Release (освободить) – подтверждаемый информационный поток, передаваемый от CCF к SRF и содержащий идентификатор того выхода SRF, соединение с которым должно быть нарушено.

2.3.7.3.2 Процедура ассистирования (Service assist)

Процедура ассистирования (Service assist) используется в тех случаях, когда иницирующий (т.е. обнаруживший запрос услуги IN) CCF/SSF не имеет прямого доступа к SRF, и предусматривает создание в сети временного соединения с таким CCF/SSF, которому SRF доступен (с так называемым ассистирующим CCF/SSF). Это соединение нарушается по окончании использования функций SRF, а дальнейшее обслуживание вызова продолжает иницирующий CCF/SSF. Функциональная модель процедуры представлена на рисунке 2.3.29. На приведенных диаграммах информационных потоков функции CCF/SSF и SRF в ассистирующем SSF показаны как единый объект (рис. 2.3.30).

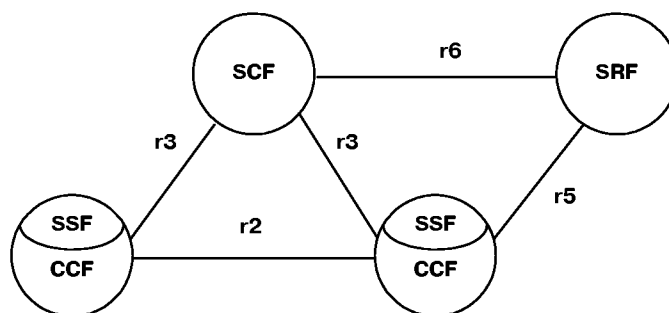


Рис. 2.3.29 Функциональная модель процедуры ассистирования (Service assist)

Поток *Establish Temporary Connection* (установить временное соединение) несет от SCF к иницирующему CCF/SSF инструкции с информацией, необходимой для установления временного соединения с ассистирующим CCF/SSF. Содержит следующие информационные элементы:

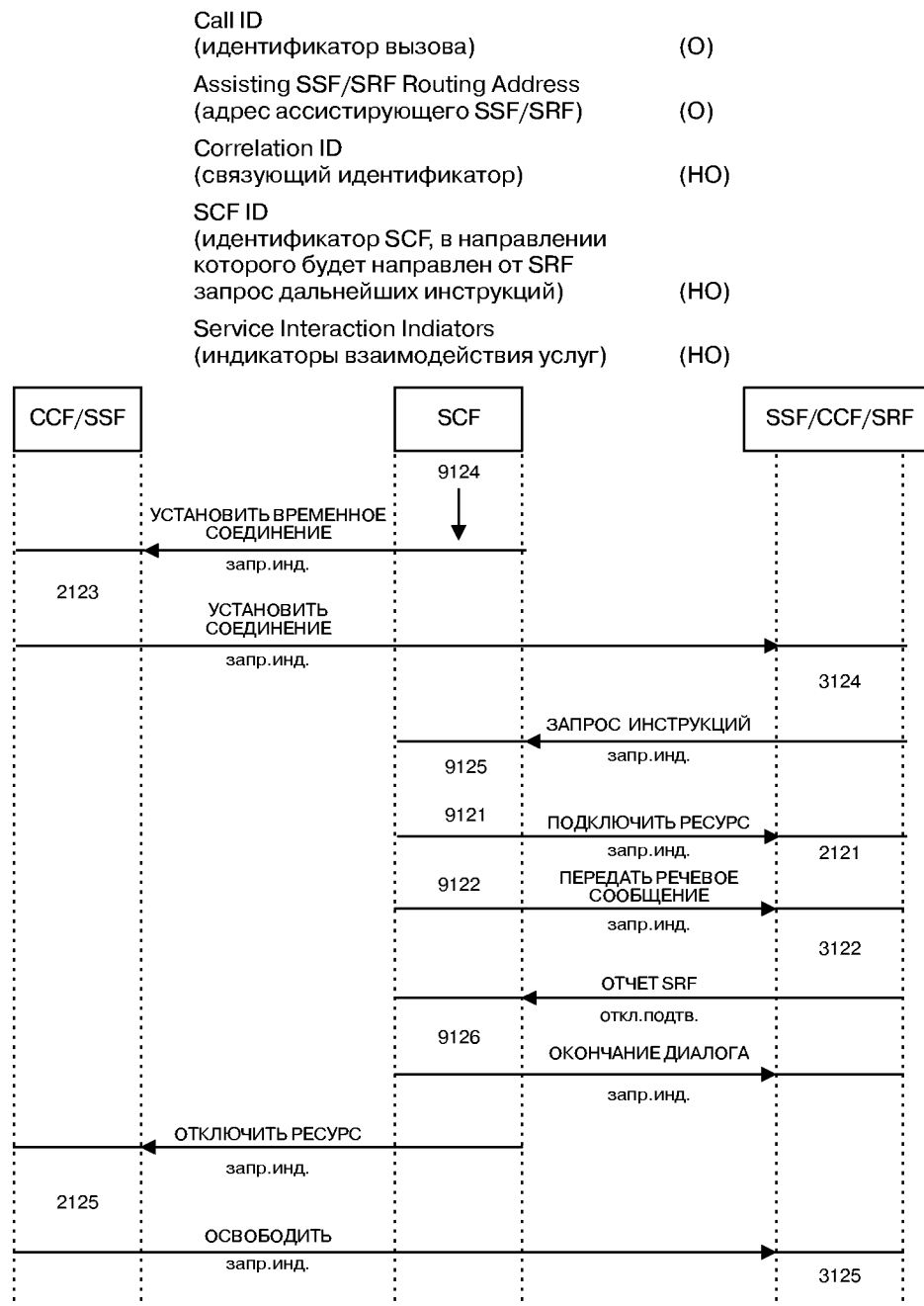


Рис. 2.3.30 Информационные потоки при процедуре ассистирования

Assist Request Instruction (запрос инструкций) поток, посредством которого ассистирующий CCF/SSF (или непосредственно SRF) запрашивает от SCF инструкции для ведения диалога с пользователем. Содержит следующие информационные элементы:

Call ID (идентификатор вызова)	(O)
Correlation ID (связующий идентификатор)	(O)
SRF Available (информация о доступности SRF)	(HO)
SSF/SRF Capabilities (функциональные возможности SSF/SRF)	(HO)

2.3.7.3.3 Процедура передачи управления услугой (Service hand-off)

Процедура передачи управления услугой (Service hand-off) используется в тех же случаях, что и процедура ассистирования. Различие между ними заключается в том, что по окончании использования SRF тот CCF/SSF, который имеет к нему прямой доступ, продолжает обслуживать вызов, не возвращая управление иницирующему CCF/SSF. Функциональная модель процедуры аналогична приведенной на рисунке 2.3.29, а показанные на рисунке 2.3.31 информационные потоки, связанные с этой процедурой, уже были рассмотрены выше.

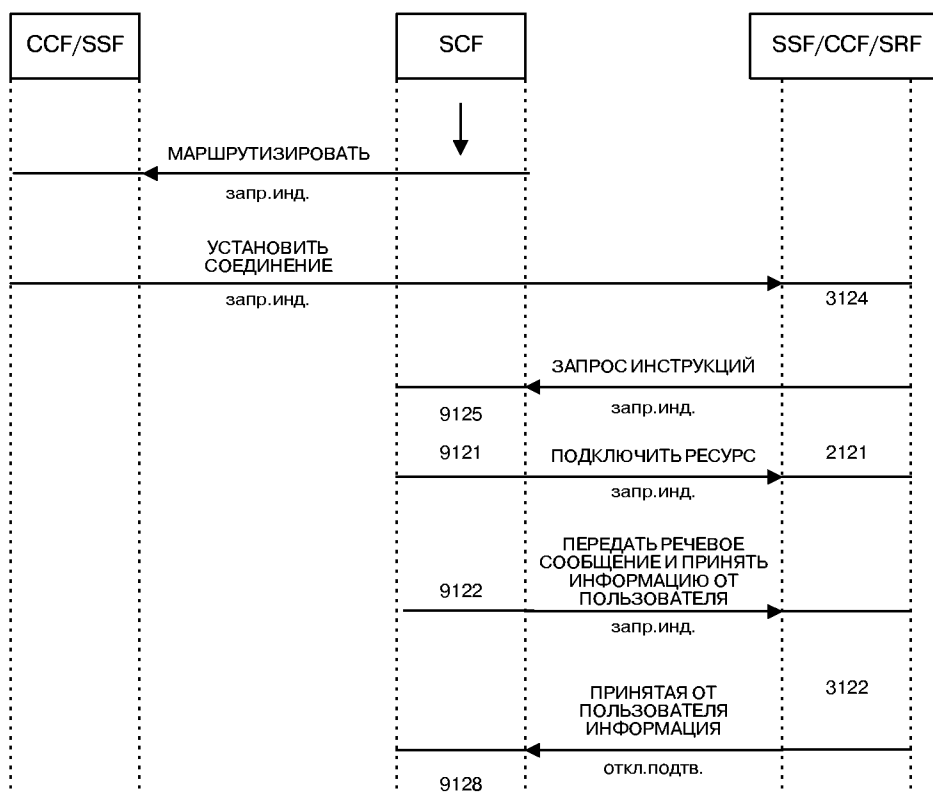


Рис. 2.3.31 Информационные потоки при процедуре передачи управления услугой Service hand-off

2.3.8 Стадия 2 описания блока BCP

Блок базового процесса обслуживания вызова BCP обеспечивает возможность доступа SCF к внутренним ресурсам SSF/CCF. Для CS-1 эти ресурсы включают в себя средства:

- обработки запросов связи – воздействие на исходящий или входящий запрос двусторонней связи (с целью гибкой маршрутизации, управления очередями, переадресации);
- инициирования вызова (для организации связи между двумя участниками);
- нарушения связи;
- назначения точек EDP с целью получения и предоставления сведений о таких событиях как, например, отказ вызывающей стороны от продолжения обслуживания вызова, занятость или отсутствие ответа вызываемой стороны.

SSF/CCF обращается к средствам BCP при обнаружении триггерной точки TDP. К SCF передается соответствующая индикация (начальный информационный поток). Для точки типа TDP-R такой поток устанавливает управляющую связь между SSF/CCF и SCF, позволяя последнему возвращать инструкции в сторону SSF/CCF. Для точек типа TDP-N управляющая связь этим потоком не устанавливается.

SCF обращается к средствам BCP в случае, когда он передает к SSF/CCF информационный поток *Initiate Call Attempt* вне контекста уже существующей управляющей связи. При этом по указанию SCF может быть установлена новая управляющая связь между SSF/CCF и SCF.

Используя установленную управляющую связь, SCF может передавать к SSF/CCF инструкции для обслуживания вызова в виде одного или нескольких информационных потоков. Инструкции могут быть переданы в ответ на начальный информационный поток или вслед за переданными ранее инструкциями.

Диаграмма информационных потоков блока BCP представлена на рисунке 2.3.32. В контексте управляющей связи SCF может потребовать от SSF/CCF передавать ему сведения о последующих событиях в процессе обслуживания вызова (такими событиями могут быть, например, ответ или занятость вызываемой стороны, отказ абонента от продолжения обслуживания и т.п.). Это требование содержится в информационном потоке *Request Report BCSM Event* (запрос отчета о событии в BCSM), назначающем нужные точки EDP.

При обнаружении назначенной EDP от SSF/CCF к SCF передается поток *Event Report BCSM* (отчет о событии в BCSM). Если обнаружена точка типа EDP-R, то CCF/SSF приостанавливает обслуживание вызова.

ва до получения инструкций от SCF, если же это точка типа EDP-N, обслуживание вызова продолжается. Потоки, связанные с запросами сведений о событиях, были описаны выше (см. SIB «ОЧЕРЕДЬ»).

Содержание информационных потоков, которые могут быть переданы от SCF к SSF/CCF в некоторый момент времени, зависит от того, в каком состоянии находится в этот момент соединение. Данные, которые передаются к SCF от SSF/CCF в начальном информационном потоке, зависят от точки обнаружения. Например, если это точка – *Originating_Attempt Authorized* (разрешенный исходящий вызов), то необходимо передать набранные абонентом цифры, если же это точка – *T_Disconnect* (разъединение), то ясно, что никаких цифр передавать не нужно.

Данное обстоятельство определило существование двух подходов к запросу управляющих инструкций от SCF, разница между которыми состоит в следующем. При первом подходе для всех точек обнаружения используется один информационный поток *Initial DP* (начальная DP); ясно, что его содержание (набор информационных элементов) зависит от того, к какой именно DP он относится:

Call ID (идентификатор вызова)	(O)
Service key (тип услуги)	(O)
Call gapping encountered (имеет место прореживание потока вызовов)	(HO)
Called party number (номер вызываемой стороны)	(HO)
Calling party number (номер вызывающей стороны)	(HO)
Calling party category (категория вызывающей стороны)	(HO)
SSF/SRF capabilities (возможности SSF/SRF)	(HO)
SRF available (доступность SRF)	(HO)
Location number (номер местонахождения)	(HO)
Original called party ID (номер первоначально вызывавшейся стороны)	(HO)
Forward call indicators (индикаторы переадресации вызова)	(HO)
Bearer capability (характеристики средств доставки информации)	(HO)

Event Type BCSM (тип события BCSM)	(НО)
Redirecting party ID (номер, с которого произведена переадресация вызова)	(НО)
Redirection information (информация о переадресации)	(НО)
Additional calling party number (дополнительный номер вызывающей стороны)	(НО)
Service interaction indicators (индикаторы взаимодействия услуг)	(НО)
High layer compatibility (совместимость в верхних уровнях)	(НО)

Второй подход состоит в использовании семейства начальных информационных потоков, причем для каждой DP применяется свой поток, содержание которого однозначно определяется вызвавшим его событием. Название каждого потока из семейства соответствует той точке обнаружения, с которой он связан, а содержимое состоит как из информационных элементов, общих для всех точек DP, так и из элементов, специфических только для данной точки.

Полный перечень информационных потоков, специфических для одноименных точек обнаружения, выглядит следующим образом: *Analysed Information* (проанализированная информация), *Route Select Failure* (невозможность выбора маршрута), *O_Called Party Busy* (исходящая часть, вызываемая сторона занята), *O_No Answer* (исходящая часть, отсутствие ответа), *O_Answer* (исходящая часть, ответ), *O_MidCall* и *T_MidCall* (исходящая и входящая части, активное состояние), *O_Disconnect* (исходящая часть, разъединение), *Term Attempt Authorized* (входящий вызов разрешен), *T_Called Party Busy* (входящая часть, вызываемая сторона занята), *T_No Answer* (входящая часть, отсутствие ответа), *T_Answer* (входящая часть, ответ), *T_Disconnect* (входящая часть, разъединение). С учетом того, что в версии ETSI протокола INAP поддерживаются только первый, вариант подробное рассмотрение этих потоков не приводится.

Поток *Initiate Call Attempt* (инициировать вызов) используется SCF для передачи к SSF запроса создать соединение по инициативе SCF (например, для автоматической побудки или для определенной заранее конференцсвязи) и содержит:

Call ID (идентификатор вызова)	(О)
Destination routing address (адрес назначения)	(НО)

Alerting pattern
(тип оповещения о передаче сигнала
вызова вызываемой стороне) (НО)
Calling party number
(номер вызывающей стороны) (НО)
Service interaction indicators
(индикаторы взаимодействия услуг) (НО)

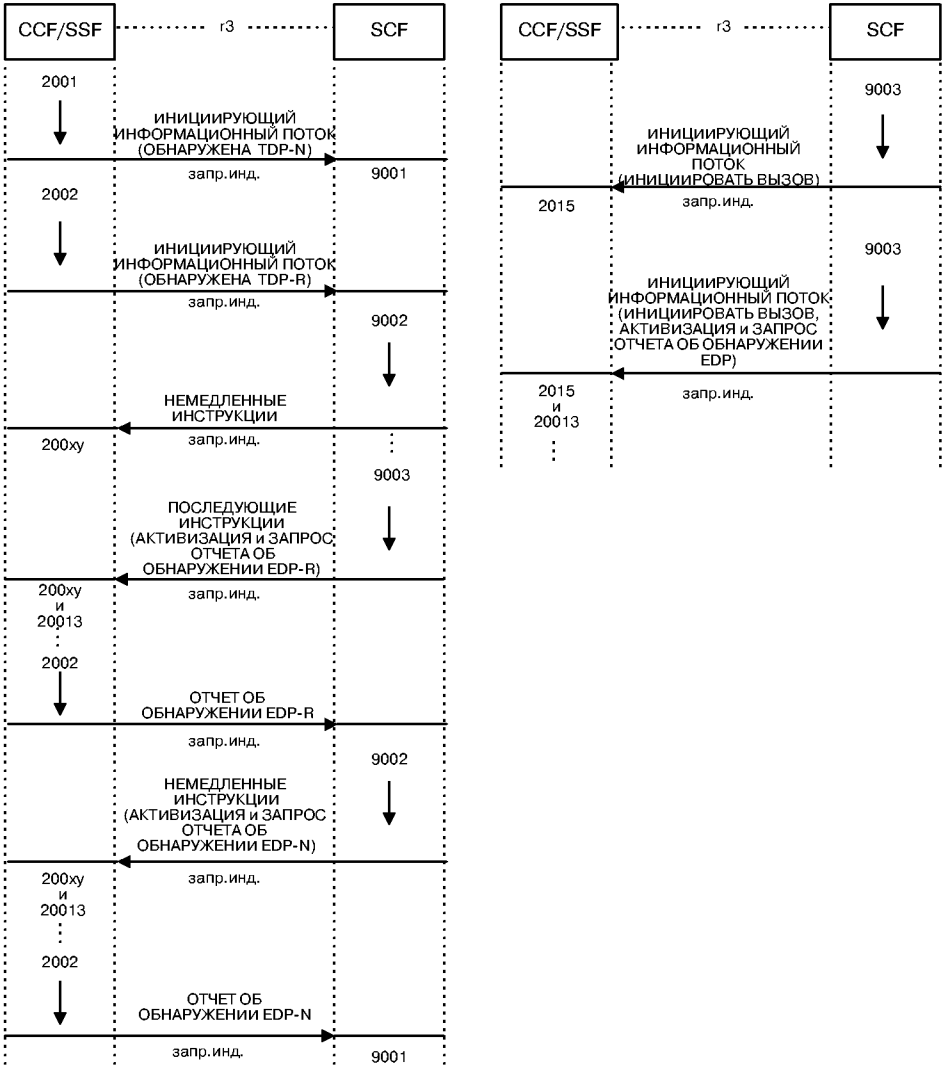


Рис. 2.3.32 Диаграмма информационных потоков блока ВСП

Для потоков *Request Report BCSM Event* (запрос отчета о событии) и *Event Report BCSM* (отчет о событии) в качестве альтернативы также определено семейство информационных потоков, каждый из которых предназначен для запроса и передачи отчета только об одном определенном событии. Принцип формирования этих потоков и содержание каждого из них читатель может найти в рекомендации Q.1214.

Поток *Connect* (соединить) используется SCF в течение фазы установления соединения для передачи к SSF информации о маршруте для завершения соединения и содержит

Call ID (идентификатор вызова)	(O)
Destination routing address (адрес назначения)	(O)
Alerting pattern (тип оповещения о передаче сигнала вызова вызываемой стороне)	(HO)
Route list (список маршрутов)	(HO)
Correlation ID (связующий идентификатор)	(HO)
SCF ID (идентификатор SCF)	(HO)
Cut and paste (удалить и вставить заданные цифры)	(HO)
Original called party ID (первоначально вызываемая сторона)	(HO)
Service interaction indicators (индикаторы взаимодействия услуг)	(HO)
Calling party number (номер вызывающей стороны)	(HO)
Calling party category (категория вызывающей стороны)	(HO)
Redirecting party ID (номер переадресующей стороны)	(HO)
Redirection information (информация переадресации)	(HO)

Поток *Continue* (продолжить) передается от SCF к SSF с указанием возобновить обслуживание вызова, начиная с той DP, где оно было приостановлено в ожидании инструкций от SCF, и содержит

Call ID (идентификатор вызова)	(O)
-----------------------------------	-----

Поток *Release Call* (разъединить) предназначен для передачи от SCF к SSF запроса разъединения на любой фазе обслуживания вызова. Состоит из:

Call ID (идентификатор вызова)	(O)
Cause (причина)	(O)

Завершает перечень потоков, непосредственно связанных с блоком базового процесса обслуживания вызова, семейство информационных потоков, передаваемых от SCF к SSF для возобновления обслуживания вызова, которое было приостановлено в одной из определенных точек DP. Процесс обслуживания возобновляется в определенном состоянии PIC, причем название потока соответствует этому PIC. Перечень информационных элементов для этой группы читатель может найти в Q.1214.

Поток *Collect Information* (накопить информацию) требует от SSF пригласить пользователя к набору номера и принять от него адресную информацию в соответствии с заданным планом нумерации (например, для услуги виртуальной частной сети). Поток *Collect Information* имеет смысл, если процесс обслуживания вызова был приостановлен в DP: *Origination_Attempt_Authorized*, *Collected_Info*, *Analysed_Info*, *Route_Select_Failure*, *O_Called_Party_Busy*, *O_No_Answer* или *O_Disconnect*.

Поток *Analyse Information* (проанализировать информацию) предписывает SSF произвести анализ адресной информации, принятой от пользователя или полученной от SCP (например, после пересчета), включая проверку ее соответствия плану нумерации, определение номера вызываемой стороны и типа адреса. Поток *Analyse Information* имеет смысл, если процесс обслуживания вызова был приостановлен в следующих точках DP: *Origination_Attempt_Authorized*, *Collected_Info*, *Analysed_Info*, *Route_Select_Failure*, *O_Called_Party_Busy*, *O_No_Answer*, *O_Disconnect*.

Поток *Select Route* (выбрать маршрут) содержит предписание выбрать маршрут (в частности, альтернативный) в соответствии с информацией, полученной от пользователя или от SCF. Поток *Select Route* правомерен, если процесс обслуживания вызова был приостановлен в следующих точках DP: *Origination_Attempt_Authorized*, *Collected_Info*, *Analysed_Info*, *Route_Select_Failure*, *O_Called_Party_Busy*, *O_No_Answer*, *O_Disconnect*.

Поток *Select Facility* (выбрать обслуживающую линию) используется для выбора свободной линии (возможно, из группы линий) вызываемой стороны. Поток *Select Facility* имеет смысл, если процесс обслуживания вызова был приостановлен в следующих точках обнаружения DP: *Termination_Attempt_Authorized*, *T_Called_Party_Busy* и *T_No_Answer*.

2.3.9 Дополнительные функции

При описании соответствия между функциями глобальной и распределенной функциональных плоскостей, представленными монолитными блоками SIB на первой из них и информационными потоками между функциональными объектами на второй, возникает необходимость в описании дополнительных распределенных функций, никак не отражаемых в блоках SIB. Эти дополнительные функции необходимы для защиты сети от перегрузки, от ошибок распределенной обработки или от отказов физических элементов. В CS-1 для таких целей определены две процедуры: проверка активности (Activity test) и прореживание потока вызовов (Call gap).

2.3.9.1 Проверка активности (Activity test)

Процедура «проверка активности» позволяет функциональному объекту SCF проверить существование логической связи между ним и SSF. Предназначенный для этой цели подтверждаемый информационный поток *Activity Test*, как и отклик на него со стороны SSF (*Activity Test Response*), не содержит никаких информационных элементов. Диаграмма информационных потоков для данной процедуры представлена на рисунке 2.3.33.

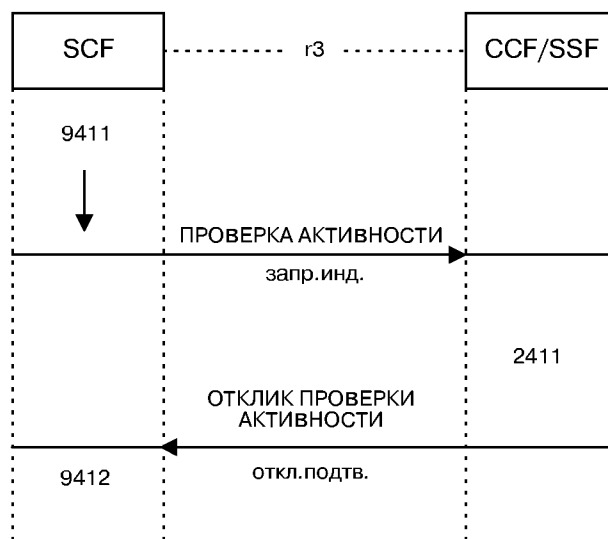


Рис. 2.3.33 Диаграмма информационных потоков процедуры проверки активности (Activity test)

2.3.9.2 Прореживание потока вызовов (Call gap)

Необходимость прореживания потока вызовов (Call gap) обусловлена тем, что в течение относительно коротких периодов времени со стороны CCF/SSF к SCF может поступать особенно интенсивный по-

ток запросов инструкций. Процедура применяется, когда нагрузка SCF этими запросами превышает расчетную, что может вызывать недопустимые задержки при обмене сообщениями и привести к отказу абонента от обслуживания. Обнаружив перегрузку, SCF активизирует в CCF/SSF средства прореживания потока тех вызовов, обслуживание которых связано с обращениями за инструкциями. Диаграмма информационного потока для этой процедуры представлена на рисунке 2.3.34.

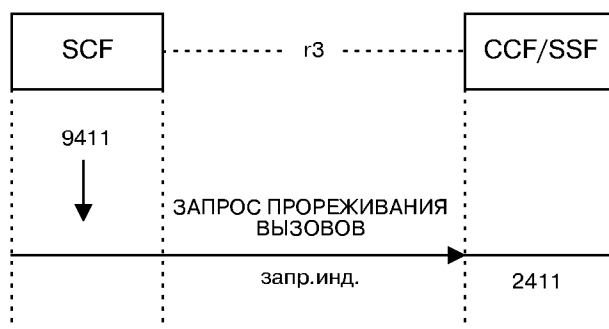


Рис. 2.3.34 Диаграмма информационного потока процедуры прореживания (Call Gap)

Процедура прореживания ограничивает количество вызовов, допускаемых к обработке с привлечением функций IN, за счет блокирования вызовов с определенными характеристиками. Такие вызовы блокируются в CCF/SSF в течение заданного периода времени. Информационные потоки, связанные с процедурой прореживания, передаются асинхронно по отношению к любой программе логики услуг в составе отклика на полученные от SCF инструкции. Поток *Call Gap* содержит:

Control Type (тип управления)	(NO)
Gap Indicators (индикаторы прореживания)	(O)
Gap Criteria (критерии прореживания)	(O)
Gap Treatment (обработка заблокированных вызовов)	(NO)

2.3.10 Соответствие между POI/POR и DP/PIC

Взаимодействие между блоком базового процесса обслуживания вызова (BCP) и глобальной логикой услуг (GSL) на глобальной функциональной плоскости происходит в точках обращения BCP к дру-

гим SIB (точках POI) и в точках возврата (точках POR). На распределенной функциональной плоскости взаимодействие происходит между менеджером состояний базового процесса BCSM (внутренним объектом CCF/SSF) и программой логики услуг SLPI (внутренним объектом SCF). Для этой плоскости определены точки состояний (PIC) и точки обнаружения (типа TDP-R и EDP-R).

Установить однозначное соответствие между точками POI/POR и точками PIC/DP не всегда возможно из-за того, что ресурсы распределенной функциональной плоскости раздроблены. Например точка возврата PROCEED WITH NEW DATA (ПРОДОЛЖИТЬ РАБОТУ С НОВЫМИ ДАННЫМИ), однозначно определенная на глобальной функциональной плоскости, здесь может соответствовать той же точке TDP-R, которая ранее вызвала запрос инструкций от SCF. Ниже приведено примерное соответствие между точками POI/POR и точками PIC/DP. Точное же соответствие может быть установлено отдельно для каждой услуги (т.е. с учетом используемых в ней блоков SIB).

POI	DP-R
CALL ORIGINATED (ВЫЗОВ СТАНЦИИ)	Orig.Attempt_Authorized (исходящий вызов разрешен)
ADDRESS COLLECTED (АДРЕС ПРИНЯТ)	Collected_Info. (накопленная информация)
ADDRESS ANALYSED (АДРЕС ПРОАНАЛИЗОВАН)	Analysed_Info. (информация проанализирована)
PREPARED TO COMPLETE CALL (ГОТОВНОСТЬ ОРГАНИЗОВАТЬ СВЯЗЬ)	Term_Attempt_Authorized (входящий вызов разрешен)
BUSY (ЗАНЯТОСТЬ)	O_Called_Party_Busy (исходящая часть, абонент занят) T_Busy (входящая часть, абонент занят) Route_Select_Failure (невозможность выбрать маршрут)
NO ANSWER (НЕТ ОТВЕТА)	O_No_Answer (исходящая часть, отсутствие ответа) T_No_Answer (входящая часть, отсутствие ответа)
CALL ACCEPTANCE (ВЫЗОВ ПРИНЯТ)	O_Answer (исходящая часть, ответ) T_Answer (входящая часть, ответ)
ACTIVE STATE (АКТИВНОЕ СОСТОЯНИЕ)	O_Mid_Call (исходящая часть, активное состояние) T_Mid_Call (входящая часть, активное состояние)
END OF CALL (ОКОНЧАНИЕ СВЯЗИ)	O_Abandon (исходящая часть, отказ от обслуживания) T_Abandon (входящая часть, отказ от обслуживания) O_Disconnect (исходящая часть, разъединение) T_Disconnect (входящая часть, разъединение)

POR	DP/PIC
CONTINUE WITH EXISTING DATA (ПРОДОЛЖИТЬ РАБОТУ С ИМЕЮЩИМИСЯ ДАННЫМИ)	Несколько разных точек DP (возврат к той DP, в которой был сделан запрос логики услуги)
PROCEED WITH NEW DATA (ПРОДОЛЖИТЬ РАБОТУ С НОВЫМИ ДАННЫМИ)	Несколько разных точек PIC (переход к PIC согласно полученным инструкциям)
HANDLE AS TRANSIT (ОБРАБАТЫВАТЬ КАК ТРАНЗИТНЫЙ)	Точки PIC: Analyse_Info (анализ информации) или Routing&Alerting (маршрутизация и оповещение)
CLEAR CALL (ПРЕКРАТИТЬ ОБСЛУЖИВАНИЕ ВЫЗОВА)	O_Null (исходящая часть, исходное состояние); T_Null (входящая часть, исходное состояние);
ENABLE CALL PARTY HANDLING (РАЗРЕШИТЬ УПРАВЛЕНИЕ СОЕДИНЕНИЕМ СО СТОРОНЫ УЧАСТНИКОВ СВЯЗИ)	Несколько разных точек DP (возврат к той DP, в которой был сделан запрос логики услуги)
INITIATE CALL (ИНИЦИИРОВАТЬ ВЫЗОВ)	Точки PIC: Analyse_Info или Routing&Alerting PICs в новой BCSM

Глава 2.4

Физическая плоскость

2.4.1 Физические элементы в CS-1

В концептуальной модели IN физическая плоскость определяет физические элементы сети и интерфейсы между ними. Применительно к набору возможностей CS-1 физическая плоскость описана в рекомендации ITU-T Q.1215.

В CS-1 не рассматриваются такие физические элементы как среда создания услуг (SCEP), система эксплуатационного управления услугами (SMP) и пункт доступа к системе эксплуатационного управления (SMAP); не рассматриваются и соответствующие интерфейсы. В то же время, на физической плоскости CS-1 может присутствовать узел сетевого доступа NAP (Network access point), ориентированный на ранние стадии развертывания услуг IN. Основные требования в отношении архитектуры физической плоскости CS-1 состоят в следующем:

- функциональные объекты распределенной функциональной плоскости CS-1 должны размещаться в физических элементах, определенных для CS-1;
- в одном физическом элементе может размещаться один или несколько функциональных объектов;
- один функциональный объект не может разделяться между двумя физическими элементами, т.е. функциональный объект должен целиком размещаться в одном физическом элементе;
- функциональный объект может быть дублирован в разных физических элементах, однако дублирование функционального объекта в одном физическом элементе недопустимо;
- физические элементы можно группировать для образования физической архитектуры;
- физические элементы должны иметь стандартные интерфейсы.

Для поддержки набора CS-1 Интеллектуальной сети определены следующие физические элементы: узел коммутации услуг (SSP), узел управления услугами (SCP), узел хранения данных для услуг (SDP), узел коммутации и управления услугами (SSCP), вспомогательный узел управления (AD), интеллектуальная периферия (IP), узел услуг (SN) и узел сетевого доступа (NAP).

Узел сетевого доступа (NAP) – это физический элемент сети (коммутационная система), применяемый на начальном этапе внедрения и развития услуг Интеллектуальной сети. Узел NAP содержит только

функции CCAF и функции CCF. Он не содержит функции SSF и поэтому не имеет возможности прямой связи с SCP. Однако NAP может определять, требуется ли привлекать к обслуживанию вызова средства IN, и, если это так, то отправлять вызов для дальнейшей обработки в SSP.

На рисунке 2.4.1 представлены архитектура физической плоскости CS-1 и проекция на нее распределенной функциональной плоскости, т.е. все перечисленные выше физические элементы и размещаемые в них функции или комбинации функций.

Таблица 2.4.1

Физические элементы	Функциональные объекты			
	SCF	SSF/CCF	SDF	SRF
SSP	HO	O	HO	HO
SCP	O	-	O	-
SDP	-	-	O	-
IP	-	-	-	O
AD	O	-	O	-
SN	O	O	O	O
SSCP	O	O	O	HO
NAP	-	O (только CCF)	-	-
O Обязательно HO Не обязательно - Не допускается				

Проекция функциональных объектов в физические элементы, определенные для CS-1, имеют два главных отличия от определенных в рекомендации Q.1205, а именно: физический элемент SCP обязательно должен содержать функциональные объекты SCF и SDF; и физический элемент IP должен содержать только функциональный объект SRF.

В таблице 2.4.1 приведены наиболее типичные для CS-1 варианты размещения функциональных объектов в физических элементах.

2.4.2 Интерфейсы между физическими элементами и протоколы

В CS-1 определены следующие интерфейсы между элементами физической плоскости: SCP-SSP; AD-SSP; IP-SSP; SN-SSP; SCP-IP; AD-IP; SCP-SDP.

Для переноса через перечисленные интерфейсы сообщений прикладного уровня, обеспечивающих поддержку услуг IN, было предложено использовать существующие протоколы нижних уровней.

Поэтому при стандартизации CS-1 удалось сосредоточить основные усилия на спецификации протокола прикладного уровня. Сообщения прикладного уровня, передаваемые через разные интерфейсы, должны иметь одинаковое семантическое содержание, даже если для них применяются разные способы кодирования. Например, сообщения между SSF и SCF должны содержать одинаковую информацию, независимо от того, через какой интерфейс происходит обмен – SCP-SSP, AD-SSP или SN-SSP. Ниже приводятся сведения о протоколах нижних уровней, которые предлагается использовать в вышеперечисленных интерфейсах.

Интерфейсы SCP-SSP, SCP-IP и SCP SDP. Для этих интерфейсов в качестве протоколов нижних уровней предлагаются следующие подсистемы ОКС-7: подсистема TCAP, подсистема SCCP и подсистема MTP.

Интерфейс AD-SSP. Для этого интерфейса в качестве протокола нижележащего уровня предлагается TCAP. В качестве протокола физического уровня может использоваться любой стандартный протокол.

Интерфейс IP-SSP. Интерфейс используется для взаимодействия IP и SSP, а также IP и SCP через SSP. Для этого интерфейса в качестве протоколов нижних уровней предлагаются протоколы базового/первичного доступа ISDN (BRI, PRI) или протоколы ОКС-7 (MTP/SCCP/TCAP).

При использовании BRI или PRI D-канал ISDN, соединяющий IP и SSP, переносит информацию прикладного уровня между SCF и SRF и сигнальную информацию для установления соединений по В-каналу к IP. Информация, передаваемая от SCF к SRF (например, номер речевого оповещения и число дополнительных цифр номера) и обратно (например, накопленные цифры), должна содержаться в информационном элементе «Facility» сообщений SETUP, DISCONNECT или FACILITY (ITU-T, Q.931).

Интерфейс SN-SSP. Для этого интерфейса в качестве протоколов нижнего уровня предлагаются протоколы базового/первичного доступа ISDN (BRI, PRI). SN и SSP обмениваются сообщениями прикладного уровня по D-каналу ISDN с использованием процедур, определенных в рекомендации ITU-T Q.932.

Интерфейс AD-IP. Для этого интерфейса в качестве протокола прикладного уровня предлагается TCAP. В качестве протокола физического уровня может использоваться любой стандартный протокол.

Для взаимодействия с физическими объектами, находящимися за пределами сети, может использоваться конвертер, преобразующий протокол TCAP ОКС-7 в протокол передачи данных (например, X.25).

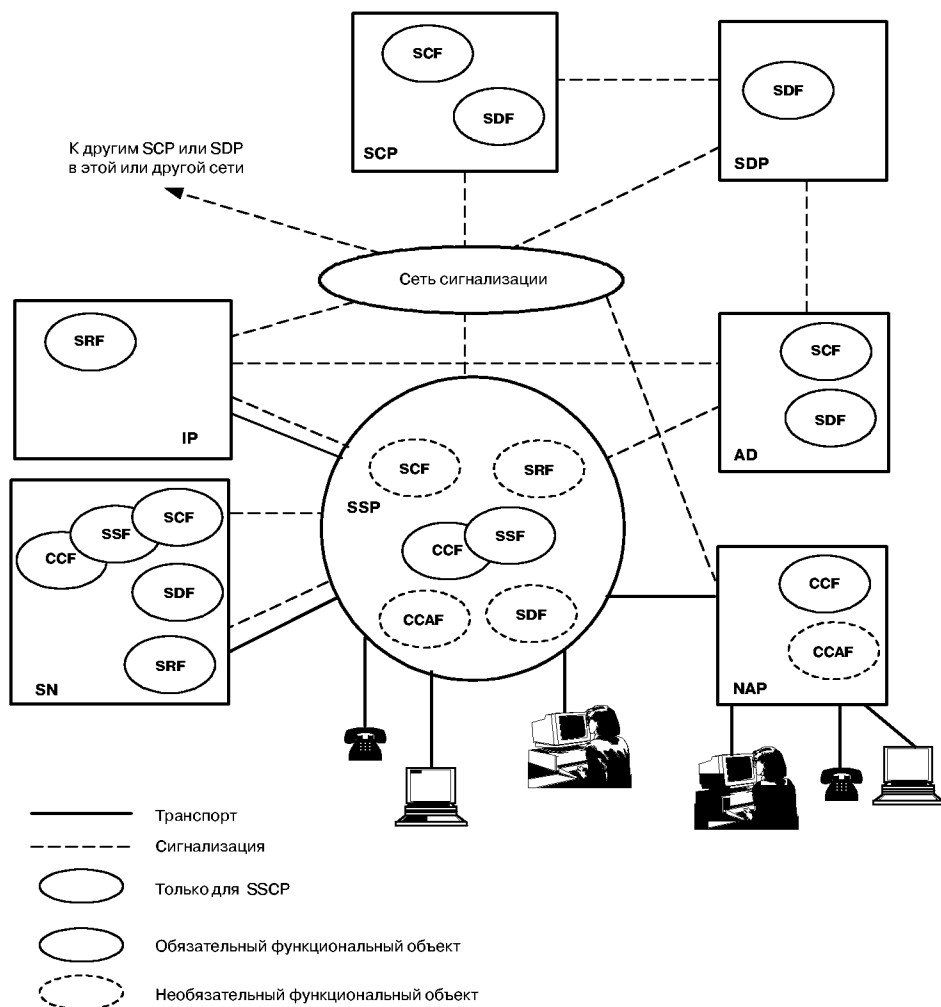


Рис. 2.4.1 Архитектура физической плоскости CS-1

Часть 3

Интерфейсы и протоколы

Глава 3.1

Интерфейсы IN

3.1.1 Процесс стандартизации INAP

Спецификации прикладного протокола Интеллектуальной сети (INAP) в части поддержки услуг CS-1 для интерфейсов SSF-SCF, SCF-SDF и SCF-SRF определены в рекомендации ITU-T Q.1218. Первая редакция этой рекомендации была опубликована в марте 1993 года и содержала около 100 страниц.

Процесс усовершенствования рекомендаций, касающихся Интеллектуальной сети, наиболее сильно затронул спецификации протокола INAP. Этот процесс был начат Европейским институтом стандартизации ETSI как часть работ по созданию европейского стандарта Интеллектуальной сети и был оформлен в виде набора стандартов ETSI, одним из которых является ETS 300 374, определяющий модификацию протокола INAP (Q.1218 редакции 1993 года) для использования в Европе (протокол Core INAP). Из протокола INAP был удален ряд операций, в основном, из числа специфических для точек обнаружения (DP specific). Для оставшихся операций и их параметров ETSI предоставил подробное описание их назначения и использования. Кроме того, была усовершенствована процедурная модель. Стандарт ETSI, изданный в 1994 году, был представлен в качестве вклада в процесс обновления рекомендаций ITU-T по Интеллектуальной сети.

В обновленной версии Q.1218, одобренной в 1995 году и содержащей уже около 400 страниц, были оставлены все ранее присутствовавшие операции, однако для их описания была использована методология, предложенная ETSI. На базе работ ETSI была также улучшена процедурная модель. Кроме того, в обновленную версию Q.1218 вошла спецификация интерфейса SCF-SDF.

Итак, окончательные версии Q.1218 и ETS 300 374 содержат:

- детальное описание стека протоколов, необходимого для поддержки INAP и соответствующих стандартов ITU-T и ISO;
- подробный словарь протокола INAP (операции и параметры) на языке ASN.1; описание операций сопровождается описанием соответствующих таймеров;
- спецификации прикладных сервисных элементов (ASE) протокола INAP;
- спецификации прикладных контекстов (AC) протокола INAP;

- модель распределенного окружения IN и описание процедур на основе машин конечных состояний (FSM);
- подробное описание назначения и использования всех операций, а также их параметров.

Рекомендация Q.1218 содержит дополнение, где приводится SDL-описание процедурной модели, и два приложения, в одном из которых описывается модель данных для услуг, а в другом приведены материалы, касающиеся дальнейших исследований.

С учетом все большей ориентации России на европейские стандарты, происходящей в последние годы, и того факта, что вариант, предложенный ETSI, на сегодня представляется более стабильным, чем вариант ITU-T, в дальнейшем подробно будет рассматриваться именно протокол Core INAP ETSI.

3.1.2 Сценарии взаимодействия элементов IN

Прикладной протокол INAP для CS-1 поддерживает взаимодействие между тремя следующими функциональными объектами, определенными функциональной моделью IN:

- SSF (Service switching function);
- SCF (Service control function);
- SRF (Specialized resource function).

Интерфейс между функциональными объектами SCF и SDF, размещенными в территориально обособленных PE, поддерживает INAP, использующий подсистему TCAP, которая, в свою очередь, использует не ориентированные на соединение услуги подсистемы SCCP и услуги подсистемы MTP системы ОКС-7 (см. рис. 3.1.1). За любое сопряжение с другими протоколами при доступе к сетям других типов отвечает SDF.

Далее приведены примеры реализации связей между функциональными объектами коммутации услуг (SSF), управления услугами (SCF) и специализированных ресурсов (SRF) для случая, когда все эти объекты размещены в разных физических элементах, т.е. SSF размещен в узле коммутации услуг SSP, SCF размещен в узле управления услугами SCP, SRF размещен в физическом элементе – интеллектуальная периферия IP.

Все варианты реализации связей иллюстрированы схемами, представленными на рисунках 3.1.2 – 3.1.6. Варианты различаются способом поддержки связи SCF-SRF и типом системы сигнализации, используемой между SSF и SRF. Таблица 3.1.1 суммирует особенности каждого варианта с точки зрения вышеуказанных аспектов.

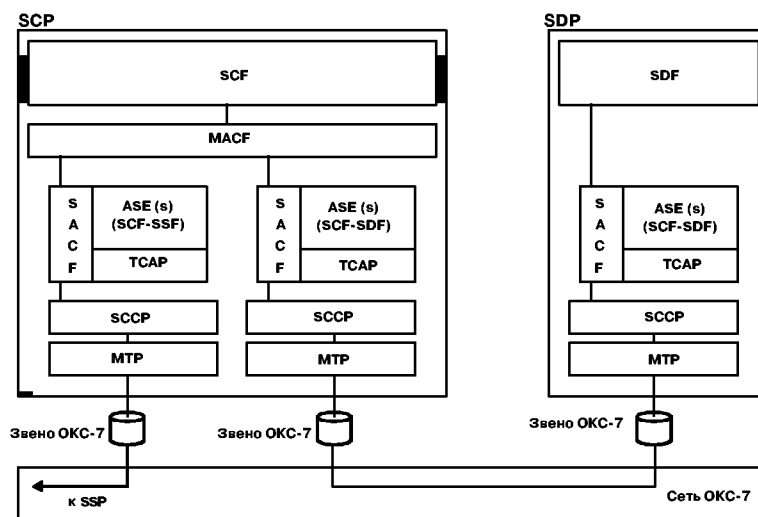


Рис. 3.1.1 Физический интерфейс между SCP и SDP

Таблица 3.1.1

Система сигнализации SSF-SRF	Способ поддержки связи SCF-SRF	
	Прямое звено TCAP	Взаимодействие через SSP
ISUP	Рис. 3.1.2	Рис. 3.1.5
DSS1	Рис. 3.1.6	Рис. 3.1.3
Другая	Как на рис. 3.1.2 или рис. 3.1.6, но с интерфейсом SCP-IP, соответствующим используемой системе сигнализации	Рис. 3.1.4

Первый вариант, изображенный на рисунке 3.1.2., относится к случаю, когда интеллектуальная периферия – IP (функциональный объект специализированных ресурсов – SRF) соединена с узлом коммутации услуг SSP через сеть ТФОП или ISDN. Все ассоциации поддерживаются системой сигнализации ОКС-7 (TCAP или ISUP). В этом случае IP является одним из узлов сети, а SCP получает доступ к IP по звену ОКС-7. Могут использоваться другие системы сигнализации, но ими должен поддерживаться перенос информации о корреляции. Подсистема ISUP реализует такое требование без введения в нее новых параметров.

Второй вариант, изображенный на рисунке 3.1.3, относится к случаю, когда интеллектуальная периферия – IP соединена с узлом коммутации услуг SSP через интерфейс пользователь-сеть ISDN. Узел управления услугами SCP получает доступ к IP через SSP по D-кана-

лу цифровой абонентской сигнализации DSS1. При этом IP может являться самостоятельным физическим элементом, размещенным вне сети ОКС-7. Информационные потоки между SCF и SRF поддерживаются протоколом ROSE, функции трансляции возложены на MACF или прикладной процесс SSP.

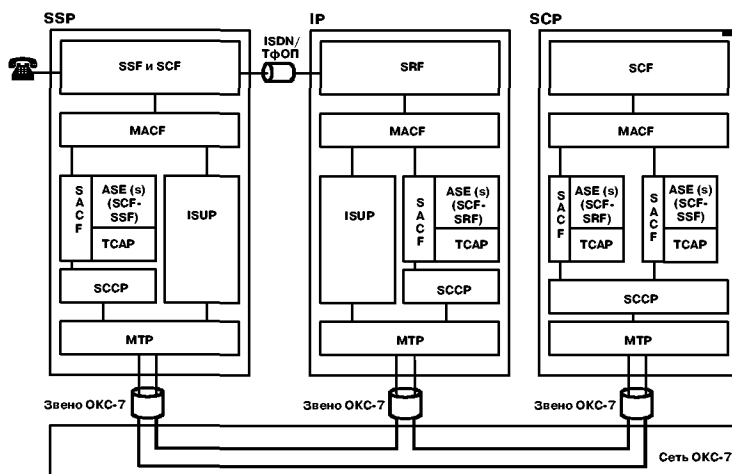


Рис. 3.1.2 Пример архитектуры для поддержки SRF, вариант 1

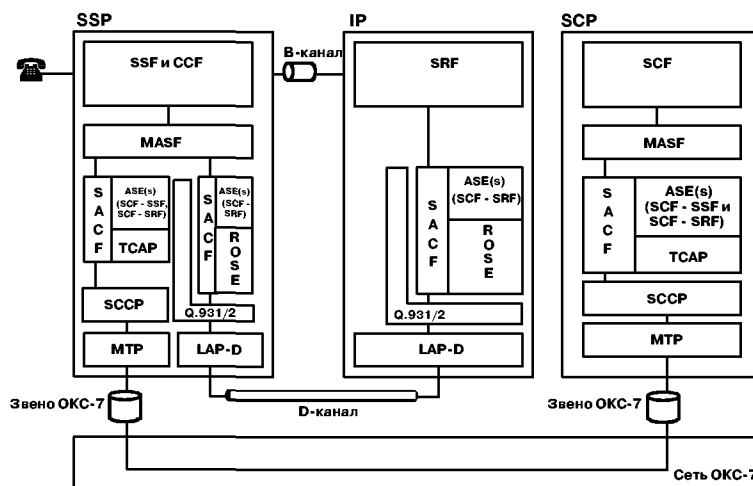


Рис. 3.1.3 Пример архитектуры для поддержки SRF, вариант 2

Третий вариант, изображенный на рисунке 3.1.4, описывает случай, когда IP интегрирована с узлом коммутации услуг SSP, в котором размещаются функциональные объекты CCF, SSF и SRF. Узел управления услугами SCP имеет возможность доступа к SRF при помощи прикладного процесса в SSP. Взаимодействие SCP с SRF может быть аналогичным варианту 2 (рис 3.1.3).

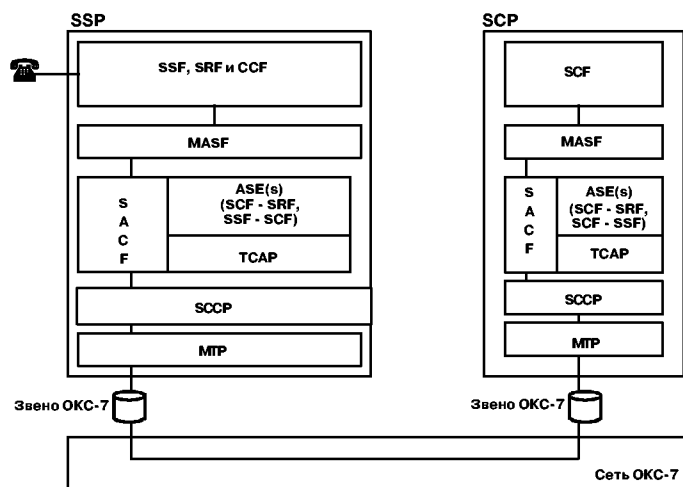


Рис. 3.1.4 Пример архитектуры для поддержки SRF, вариант 3

Четвертый вариант (рис. 3.1.5) относится к случаю, когда IP соединена с узлом коммутации услуг SSP через сеть ТФОП или ISDN. Взаимодействие SCP с SRF может быть аналогичным варианту 2 (рис. 3.1.3). Информационные потоки SCF-SRF, как и управление соединением, косвенно поддерживаются подсистемой ISUP, которая предоставляет средства транспортировки информации ROSE.

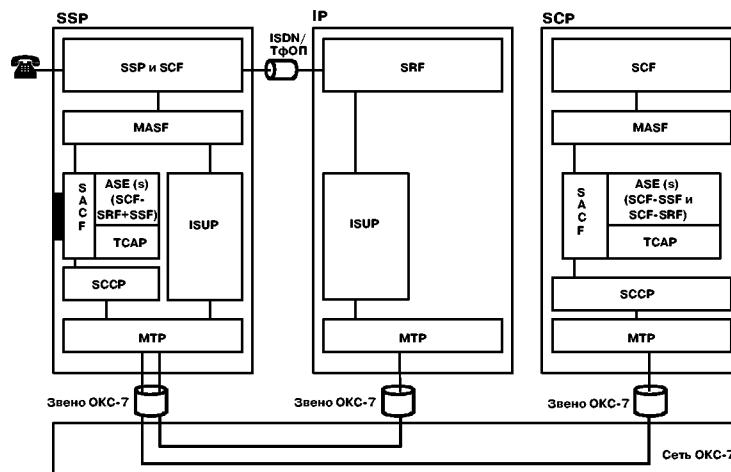


Рис. 3.1.5 Пример архитектуры для поддержки SRF, вариант 4

Пятый вариант 5 (рис. 3.1.6) относится к случаю, когда интеллектуальная периферия – IP соединена с SCP звеном ОКС-7 и с SSP через интерфейс пользователь-сеть ISDN. Взаимодействие SCP с SRF может быть аналогичным варианту 1 (рис 3.1.2).

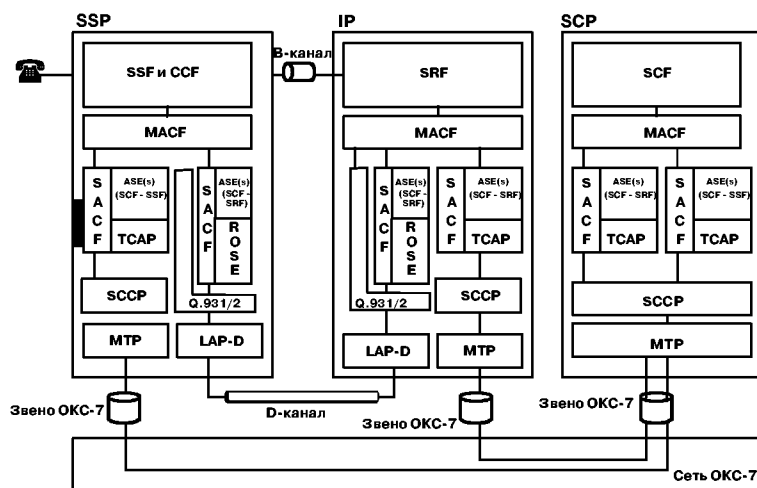


Рис. 3.1.6 Пример архитектуры для поддержки SRF, вариант 5

3.1.3 Архитектура и адресация INAP

Чтобы блоки данных протокола PDU (Protocol data units) могли достичь физического пункта назначения независимо от того, в какой сети он находится, INAP использует адресацию SCCP (параметр «глобальный заголовок» – global title) и MTP (поле «код пункта сигнализации» – point code). Выбор номеров подсистем SSN (Subsystem numbers), присваиваемых INAP внутри узла, производится оператором сети по своему усмотрению.

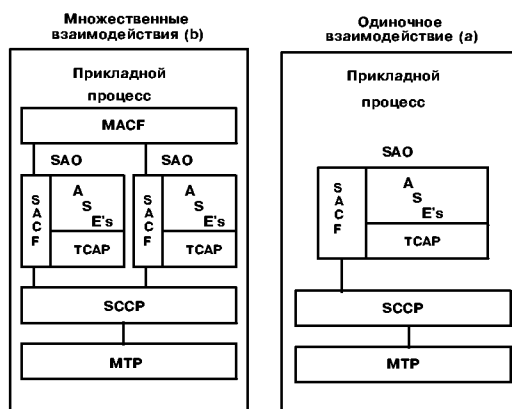


Рис. 3.1.7 Архитектура протокола INAP

Архитектура протокола INAP представлена на рисунке 3.1.7. INAP представляет собой совокупность всех прикладных сервисных элементов (ASE) IN. Физический элемент может взаимодействовать все-

го с одним другим физическим элементом (случай (а) рис.3.1.7) или с несколькими другими физическими элементами (случай (б) рис.3.1.7). В случае (а) координацию использования разных ASE (т.е. организацию очередности поддерживаемых этими ASE операций согласно очередности приема соответствующих примитивов) выполняет функция управления одиночной ассоциацией SACF. Эту функцию и относящиеся к ней ASE представляет SAO – объект одиночной логической связи. В случае (б) координацию взаимодействия во всех установленных ассоциациях выполняет MACF-функция управления множественными ассоциациями, синхронизирующая работу нескольких разных SAO, каждый из которых взаимодействует с SAO в одном из нескольких удаленных физических объектов.

Каждый ASE поддерживает одну или несколько операций. Здесь требуются некоторые пояснения. Согласно рекомендации ITU-TX.219 под операцией (operation) понимается совокупность действий, которые должен выполнить функциональный объект, получив соответствующий запрос (request) от другого функционального объекта. В ответ на запрос может последовать отклик (response), несущий информацию либо о результате выполнения этих действий, либо о невозможности их выполнить. Однако в рекомендации Q.1218 и в стандарте ETS 300 374 термин *операция* трактуется более широко и обозначает следующие понятия:

1. задачу, которая определена и содержится в управляющем функциональном объекте;
2. информационный блок, передаваемый от управляющего функционального объекта к управляемому и предписывающий выполнить эту задачу;
3. совокупность действий, которые должен произвести управляемый объект для решения указанной задачи;
4. информационный блок, передаваемый от управляемого функционального объекта к управляющему и сообщающий об успешном выполнении этих действий (заметим, что неудача в выполнении требуемых действий, как и информационный блок, сообщающий об этом, в Q.1218 и в ETS 300 374 называется не операцией, а ошибкой – error);
5. информационный блок, передаваемый от управляемого функционального объекта к управляющему без предварительного запроса и сообщающий о некотором событии.

Такая многозначность термина для читателя, свободно владеющего английским языком, особых трудностей не представляет, поскольку в значительной степени компенсируется смыслом имени той или иной операции. Стремясь сделать содержание книги доступным тем читателям, которые с языком знакомы недостаточно, и, в то же

время, как можно меньше отступать от системы понятий, используемой ETSI, авторы решили закрепить за словом *операция* все вышеперечисленные значения термина operation, кроме значения, приведенного в списке третьим (действия, предписываемые операцией, так и будут называться *действиями*). Каждая операция специфицируется посредством макроописания, которое представлено на рисунке 3.1.8.

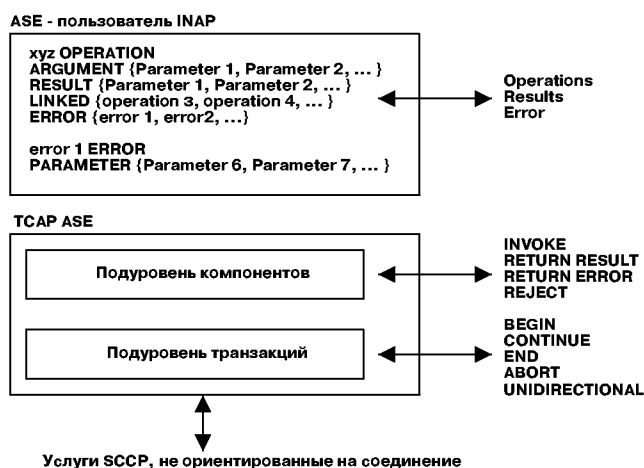


Рис. 3.1.8 Макроописание OPERATION

Использование механизма согласования прикладного контекста AC (Application context), определенного в рекомендациях ITU-T серии Q.77X, позволяет двум взаимодействующим элементам точно идентифицировать свои характеристики, а также и те характеристики, которыми должен обладать используемый для взаимодействия интерфейс.

3.1.4 Информационные потоки и операции

Полный перечень информационных потоков и соответствующих им операций представлен в таблице 3.1.2.

Правила согласования прикладного контекста (AC), принятые в TCAP, требуют, чтобы предложенный AC, если он приемлем, был отражен в первом сообщении обратного направления. Если AC неприемлем, и пользователь подсистемы TC не желает продолжать данный диалог, то инициатору диалога может быть предложен альтернативный AC, который можно использовать, чтобы начать новый диалог. Согласование прикладного контекста применимо только к интерфейсам SCF.

Таблица 3.1.2

Q.1214	Информационные потоки	Операции
6.4.2.1	Activate Service Filtering	ActivateServiceFiltering
6.4.2.2	Activity Test	ActivityTest
6.4.2.3	Activity Test Response	Return Result from Activity Test
6.4.2.4	Analized Information	InitialDP, EventReportBCSM
6.4.2.5	Analyze Information	Connect
6.4.2.6	Apply Charging	ApplyCharging
6.4.2.7	Apply Charging Report	ApplyChargingReport
6.4.2.8	Assist Request Instructions	AssistRequestInstructions
6.4.2.9	Call Gap	CallGap
6.4.2.10	Call Information Report	CallInformationReport
6.4.2.11	Call Information Request	CallInformationRequest
6.4.2.13	Cancel Status Report Request (ITU-T)	CancelStatusReportRequest (ITU-T)
6.4.2.14	Collected Information	InitialDP, EventReportBCSM
6.4.2.15	Collect Information	CollectInformation
6.4.2.16	Connect	Connect
6.4.2.17	Connect to Resource	ConnectToResource
6.4.2.18	Continue	Continue
6.4.2.19	Disconnect Forward Connection	DisconnectForwardConnection
6.4.2.20	Establish Temporary Connection	EstablishTemporaryConnection
6.4.2.21	Event Notification Charging	EventNotificationCharging
6.4.2.22	Event Report BCSM	EventReportBCSM
6.4.2.23	Furnish Charging Information	FurnishChargingInformation
6.4.2.24	Hold Call in network	ResetTimer and FurnishChargingInformation
6.4.2.25	Initial DP	InitialDP
6.4.2.26	Initiate Call Attempt	InitiateCallAttempt
6.4.2.27	O_Answer	InitialDP, EventReportBCSM
6.4.2.28	O_CalledPartyBusy	InitialDP, EventReportBCSM
6.4.2.29	O_Disconnect	InitialDP, EventReportBCSM
6.4.2.30	O_MidCall	InitialDP, EventReportBCSM
6.4.2.31	O_No_Answer	InitialDP, EventReportBCSM
6.4.2.32	Origination Attempt Authorized	InitialDP
6.4.2.33	Release Call	ReleaseCall
6.4.2.34	Request Notification Charging Event	RequestNotificationChargingEvent
6.4.2.35	Request Report BCSM Event	RequestReportBCSMEvent
6.4.2.36	Request Status Report (ITU-T)	RequestCurrentStatusReport (ITU-T) RequestFirstStatusMatchReport (ITU-T) RequestEveryStatusChangeReport (ITU-T)
6.4.2.37	Reset Timer	ResetTimer

Продолжение таблицы 3.1.2

Q.1214	Информационные потоки	Операции
6.4.2.38	Route Select Failure	InitialDP, EventReportBCSM
6.4.2.39	Select Facility (ITU-T)	SelectFacility (ITU-T)
6.4.2.40	Select Route	Connect
6.4.2.41	Send Charging Information	SendChargingInformation
6.4.2.42	Service Filtering Response	ServiceFilteringResponse
6.4.2.43	Status Report (ITU-T)	StatusReport (ITU-T)
6.4.2.44	T_Answer	InitialDP, EventReportBCSM
6.4.2.45	T_Called Party Busy	InitialDP, EventReportBCSM
6.4.2.46	T_Disconnect	InitialDP, EventReportBCSM
6.4.2.47	Term Attempt Authorized	InitialDP
6.4.2.48	T_MidCall	InitialDP, EventReportBCSM
6.4.2.49	T_NoAnswer	InitialDP, EventReportBCSM
6.5.2.1	AssistRequestInstructions from SRF	AssistRequestInstructions
6.5.2.2	Cancel Announcement	Cancel
6.5.2.3	CollectedUserInformation	Return Result from PromptAndCollectUserInformation
6.5.2.4	Play Announcement	PlayAnnouncement
6.5.2.5	Prompt and collect user information	PromptAndCollectUserInformation
6.5.2.6	Specialized Resource Report	SpecializedResourceReport
6.6.2.1	Search (ITU-T)	Search (ITU-T)
6.6.2.2	Search Result (ITU-T)	Return Result from Search (ITU-T)
6.6.2.3	Modify Entry (ITU-T)	ModifyEntry (ITU-T)
6.6.2.4	Modify Entry Result (ITU-T)	Return Result from ModifyEntry (ITU-T)
6.6.2.5	Authenticate (ITU-T)	Bind (ITU-T)
6.6.2.6	Authenticate Result (ITU-T)	Return Result from Bind (ITU-T)
6.6.2.7	AddEntry (ITU-T)	AddEntry (ITU-T)
6.6.2.8	Add Entry Result (ITU-T)	Return Result from AddEntry (ITU-T)
6.6.2.9	RemoveEntry (ITU-T)	RemoveEntry (ITU-T)
6.6.2.10	Remove Entry Result (ITU-T)	Return Result from RemoveEntry (ITU-T)

В некоторых случаях может оказаться необходимым различать, как должны выполняться действия, предписываемые разными операциями: последовательно или параллельно (т.е. быть синхронизированными). Действия, связанные с начислением платы (Charging operations), могут быть синхронизированы с действиями, предписываемыми любой другой операцией.

Чтобы указать на то, что некие действия должны быть синхронизированы, соответствующие операции включаются в одно сообщение. Если действия, которые предписаны одной из операций, невозможно выполнить до момента, пока не будут выполнены действия (или некоторая часть действий), предписанные какой-либо другой операцией, то инициирующий физический элемент (обычно SCP) имеет возможность включить эти две операции в два разных сообщения.

Данный метод синхронизации не предполагает, что выполняться одновременно должны все действия, операции управления которыми переданы в одном сообщении. Имеется в виду, что в случае, когда это имеет смысл, действия должны быть синхронизированы.

3.1.5 Абстрактный синтаксис протокола INAP для CS-1

3.1.5.1 Операции

Для спецификации INAP используется язык абстрактного описания ASN.1, определенный в рекомендации ITU-T X.208. Кодирование осуществляется по базовым правилам кодирования, определенным в рекомендации ITU-T X.209. Дополнительные правила кодирования устанавливают, что для параметров, используемых в существующих спецификациях ISUP и DSS1, значения параметров кодируются согласно этим спецификациям, тогда как идентификаторы параметров заменяются идентификаторами, определенными для INAP.

Проекции операций в компоненты подсистемы TCAP определены рекомендацией Q.773 и стандартом ETS 300 287. Класс операции специфицируется в ее макроописании ASN.1 OPERATION MACRO следующим образом:

- **класс 1** – в макроописании присутствуют и поле результата (RESULT), и поле ошибок (ERRORS);
- **класс 2** – в макроописании присутствует только поле ошибок (ERRORS);
- **класс 3** – в макроописании присутствует только поле результата (RESULT);
- **класс 4** – в макроописании отсутствуют и поле результата (RESULT), и поле ошибок (ERRORS).

Абстрактный синтаксис протокола INAP состоит из нескольких модулей ASN.1, описывающих операции, ошибки и типы связанных с ними данных. Коды операций и коды ошибок определяются в отдельном модуле.

- Модуль **IN-CS-1-Operations** содержит определения типов операций INAP.
- Модуль **IN-CS-1-Errors** содержит определения типов ошибок INAP.
- Модуль **IN-CS-1-Data Types** содержит определения типов данных INAP.
- Модуль **IN-CS-1-Codes** содержит коды операций и коды ошибок.

Спецификация операции INAP, в зависимости от ее класса, может содержать следующие поля.

- Поле ARGUMENT с именем операции.
- Поле RESULT с параметрами, возвращаемыми после выполнения предписанных действий. Это поле присутствует только в спецификации операций классов 1 или 2.
- Поле ERRORS с параметрами, возвращаемыми после неудачного завершения действий. Это поле присутствует только в спецификации операций классов 1 или 3.
- Поле LINKED является дополнительным и указывает названия операций, которые связаны с данной операцией и могут передаваться встречной стороной.

Для каждой операции назначается таймер. Выдержка времени, определяемая таймером, может быть короткой (до 10 секунд), средней (до 60 секунд) или длинной (до 30 минут). Минимальное значение выдержки времени для таймеров всех трех категорий – 1 секунда. Для каждого таймера выдержка времени может быть разной в разных сетях и должна устанавливаться оператором. В таблице 3.1.3 перечислены таймеры операций и их категории. Категории некоторых таймеров стандартом не специфицированы (имеют статус «для дальнейшего исследования»), а потому в таблице не приводятся.

Таблица 3.1.3

Операция	Таймер	Выдержка времени
<i>ActivateServiceFiltering</i>	T_{asf}	Средняя
<i>ActivityTest</i>	T_{at}	Короткая
<i>ApplyCharging</i>	T_{ac}	Короткая
<i>ApplyChargingReport</i>	T_{acr}	Короткая
<i>AssistRequestInstructions</i>	T_{ari}	Короткая
<i>CallGap</i>	T_{cq}	Короткая
<i>CallInformationReport</i>	T_{cirp}	Короткая
<i>CallInformationRequest</i>	T_{cirq}	Короткая
<i>Cancel</i>	T_{can}	Короткая
<i>CollectInformation</i>	T_{ci}	Средняя
<i>Connect</i>	T_{con}	Короткая
<i>ConnectToResource</i>	T_{ctr}	Короткая
<i>Continue</i>	T_{cue}	Короткая
<i>DisconnectForwardConnection</i>	T_{dfc}	Короткая
<i>EstablishTemporaryConnection</i>	T_{etc}	Средняя
<i>EventNotificationCharging</i>	T_{enc}	Короткая
<i>EventReportBCSM</i>	T_{erb}	Короткая
<i>FurnishChargingInformation</i>	T_{fci}	Короткая
<i>InitialDP</i>	T_{idp}	Короткая
<i>InitiateCallAttempt</i>	T_{ica}	Короткая
<i>ReleaseCall</i>	T_{rc}	Короткая
<i>RequestEveryStatusChangeReport (ITU-T)</i>	T_{res}	Короткая
<i>RequestFirstStatusMatchReport (ITU-T)</i>	T_{rfs}	Короткая
<i>RequestNotificationChargingEvent</i>	T_{rnc}	Короткая
<i>RequestReportBCSMEvent</i>	T_{rrb}	Короткая
<i>ResetTimer</i>	T_{rt}	Короткая
<i>SendChargingInformation</i>	T_{sci}	Короткая
<i>ServiceFilteringResponse</i>	T_{sfr}	Короткая
<i>PlayAnnouncement</i>	T_{pa}	Длинная
<i>PromptAndCollectUserInformation</i>	T_{pc}	Длинная
<i>SpecializedResourceReport</i>	T_{srr}	Короткая

В таблице 3.1.4 приведены функциональное описание и данные о направлении передачи, классах и таймерах для операций SCF-SSF.

Таблица 3.1.4

Операция	Направление Класс Таймер	Описание
<i>ActivateServiceFiltering</i>	SCF -> SSF Класс: 1 Таймер: Tasf	При приеме этой операции SSF маршрутизирует вызовы определенным образом, не передавая запроса о каждом обнаруженном вызове. Операция используется, например, при предоставлении услуг Телеголосование или услуг Массовые вызовы. Простые функции регистрации и управления оповещениями могут размещаться в SSF. Эта операция запускает специальные счетчики в SSF.
<i>ActivityTest</i>	SCF -> SSF Класс: 3 Таймер: Tat	Эта операция используется для проверки наличия связи между SCF и SSF. Если связь по-прежнему существует, SSF передает ответ. Если ответа не получено, SCF считает, что связь с SSF по тем или иным причинам нарушена, и предпринимает соответствующие действия.
<i>ApplyCharging</i>	SCF -> SSF Класс: 2 Таймер: Tac	Эта операция используется SCF для взаимодействия со средствами начисления платы, имеющимися в SSF. Операция <i>ApplyChargingReport</i> обеспечивает обратную связь от SSF к SCF.
<i>ApplyChargingReport</i>	SCF <- SSF Класс: 2 Таймер: Tacr	Эта операция используется SSF, чтобы сообщить SCF о возникновении связанного с начислением платы события, о котором SCF запрашивал с помощью операции <i>ApplyCharging</i> .
<i>AssistRequestInstructions</i>	SCF <- SSF SCF <- SRF Класс: 2 Таймер: Tari	Эта операция используется, когда существует процедура ассистирования или процедура передачи управления, и может передаваться со стороны SSF или SRF к SCF, когда SSF установил соединение с SRF или с ассистирующим SSF после приема от SCF операции <i>ConnectToResource</i> или <i>EstablishTemporaryConnection</i> .
<i>CallGap</i>	SCF -> SSF Класс: 4 Таймер: Tcg	Эта операция служит для передачи в SSF указания снизить интенсивность передаваемых в SCF запросов определенной услуги.
<i>CallInformationReport</i>	SSF -> SCF Класс: 4 Таймер: Tcirp	Эта операция используется для передачи в SCF определенной информации об отдельном вызове, которую SCF запросил с помощью операции <i>CallInformationRequest</i> .
<i>CallInformationRequest</i>	SCF -> SSF Класс: 2 Таймер: Tciq	Эта операция предписывает SSF записать определенную информацию об отдельном вызове и передать ее SCF с помощью операции <i>CallInformationReport</i> .
<i>Cancel</i>	SCF -> SRF SCF -> SSF Класс: 2 Таймер: Tcan	Эта общая операция отменяет предыдущие взаимосвязанные операции или все запросы. Могут быть отменены операции <i>PromptAndCollectUserInfo</i> и <i>PlayAnnouncement</i> .
<i>CollectInformation</i>	SCF -> SSF Класс: 2 Таймер: Tci	Эта операция предписывает SSF выполнить в процессе обслуживания вызова действия, которые связаны с запросом от вызывающей стороны информации о пункте назначения и с ее накоплением в соответствии с определенным планом нумерации (например, виртуальной частной сети).
<i>Connect</i>	SCF -> SSF Класс: 2 Таймер: Tcon	Эта операция предписывает SSF выполнить действия, нужные для маршрутизации или переадресации вызова к указанному пункту назначения. Для выполнения этих действий SSF, в зависимости от информации, принятой от SCF, может использовать или не использовать информацию, полученную от вызывающей стороны (набранные цифры номера), и существующую информацию для установления соединений (например, индекс выхода к пучку соединительных линий).

Продолжение таблицы 3.1.4

Операция	Направление Класс Таймер	Описание
<i>ConnectToResource</i>	SCF -> SSF Класс: 2 Таймер: Tctr	Эта операция используется для установления соединения SSP с физическим элементом, содержащим SRF.
<i>Continue</i>	SCF -> SSF Класс: 4 Таймер: Tcue	Эта операция предписывает SSP продолжить работу с вызовом, процесс обслуживания которого был приостановлен в точке обнаружения (DP) до получения инструкций от SCF. SSF продолжает обслуживать вызов без подстановки новых данных от SCF.
<i>DisconnectForwardConnection</i>	SCF -> SSF Класс: 2 Таймер: Tdfe	Эта операция служит для того, чтобы нарушить соединение, установленное временно (например, соединение с ресурсами интеллектуальной периферии).
<i>EstablishTemporaryConnection</i>	SCF -> SSF Класс: 2 Таймер: Tetc	Эта операция используется для установления соединения с ресурсом интеллектуальной периферии на ограниченное время (например, для воспроизведения речевого сообщения, для получения информации от пользователя). Операция подразумевает использование процедуры ассистирования.
<i>EventNotificationCharging</i>	SSF -> SCF Класс: 4 Таймер: Tenc	Эта операция используется для уведомления SCF о событиях, связанных с начислением платы, о чем SCF запросил ранее с помощью операции <i>RequestNotificationChargingEvent</i> .
<i>EventReportBCSM</i>	SSF -> SCF Класс: 4 Таймер: Terb	Эта операция используется для уведомления SCF о событиях в BCSM (например, таком как ЗАНЯТО или НЕТ ОТВЕТА), о чем SCF запросил ранее с помощью операции <i>RequestReportBCSMEvent</i> .
<i>FurnishChargingInformation</i>	SCF -> SSF Класс: 2 Таймер: Tfci	Эта операция используется, чтобы задать характеристики записей, создаваемых SSF для начисления платы.
<i>InitialDP</i>	SSF -> SCF Класс: 2 Таймер: Tidp	Эта операция используется после обнаружения TDP для индикации запроса услуги.
<i>InitiateCallAttempt</i>	SSF -> SCF Класс: 2 Таймер: Tica	Эта операция предписывает SSF создать новое соединение с одним из участников связи, используя адресную информацию, предоставляемую SCF (например, побудка).
<i>ReleaseCall</i>	SCF -> SSF Класс: 4 Таймер: Trc	Эта операция используется для нарушения соединения на любой его фазе с отключением всех его участников.
<i>RequestNotificationChargingEvent</i>	SCF -> SSF Класс: 2 Таймер: Trnc	Эта операция служит для передачи от SCF к SSF инструкций о том, как обрабатывать данные о тех событиях, связанных с начислением платы, сведения о которых поступили от других функциональных элементов не под управлением логики услуг данного SCF. Операция поддерживает взаимодействие между сетями при начислении платы за связь через несколько сетей.
<i>RequestReportBCSMEvent</i>	SCF -> SSF Класс: 2 Таймер: Trrb	Эта операция дает указание SSF отслеживать события в BCSM (например, такие как ЗАНЯТО или НЕТ ОТВЕТА). Когда такое событие обнаружится, к SCF передается операция <i>EventReportBCSM</i> .
<i>ResetTimer</i>	SCF -> SSF Класс: 2 Таймер: Trt	Эта операция предписывает SSF установить в нуль таймер приложения SSF.
<i>SendChargingInformation</i>	SCF -> SSF Класс: 2 Таймер: Tsci	Эта операция инструктирует SSF в отношении передачи им информации о начислении платы. Информация может быть либо отправлена к исходящей АТС, либо обрабатываться в SSF, если он размещен в оконечной АТС. Эта информация может использоваться оконечной станцией для обновления счетчиков или для создания стандартной записи.
<i>ServiceFilteringResponse</i>	SSF -> SCF Класс: 4 Таймер: Tsfr	Эта операция используется для передачи в SCF значений тех счетчиков, которые были ранее указаны в принятой от SCF операции <i>ActivateServiceFiltering</i> .

В таблице 3.1.5, приведены функциональное описание и данные о направлении передачи, классах и таймерах для операций SCF-SRF.

Таблица 3.1.5

Операция	Направление Класс Таймер	Описание
<i>AssistRequestInstructions</i>	SRF -> SCF Класс: 2 Таймер: T _{ari}	Эта операция используется, когда существует процедура ассистирования или процедура передачи управления, и может передаваться со стороны SSF или SRF к SCF, когда SSF установил соединение с SRF или с ассистирующим SSF после приема от SCF операции <i>ConnectToResource</i> или <i>EstablishTemporaryConnection</i> .
<i>Cancel</i>	SCF -> SRF Класс: 2 Таймер: T _{can}	Эта общая операция отменяет предыдущие взаимосвязанные операции или все запросы. Могут быть отменены операции <i>PromptAndCollectUserInfo</i> и <i>PlayAnnouncement</i> .
<i>PlayAnnouncement</i>	SCF -> SRF Класс: 2 Таймер: T _{pa}	Эта операция должна использоваться после операций <i>EstablishTemporaryConnection</i> (процедура ассистирования) или <i>ConnectToResource</i> . Операция применима как при взаимодействии с пользователем, имеющим аналоговый аппарат, так и при взаимодействии с пользователем ISDN. Выдержка времени, определяемая таймером этой операции, должна быть достаточной, чтобы обеспечить корреляцию с другими связанными с ней операциями.
<i>PromptAndCollectUserInfo</i>	SCF -> SRF Класс: 1 Таймер: T _{pc}	Эта операция служит для взаимодействия с пользователем в процессе накопления передаваемой им информации.
<i>SpecializedResourceReport</i>	SRF -> SCF Класс: 4 Таймер: T _{srr}	Эта операция используется в качестве ответа на операцию <i>PromptAndCollectUserInfo</i> или <i>PlayAnnouncement</i> , когда готов отчет о завершении затребованных действий.

3.1.5.2 Прикладные сервисные элементы (ASE)

Прикладные сервисные элементы ASE – это всего лишь группы операций. В настоящее время операции группируются в ASE в соответствии с выполняемыми функциями. В некоторых случаях такое объединение выглядит весьма логичным, а в некоторых – носит, в известной степени, произвольный характер. В будущем ASE могут быть перегруппированы, например, таким образом, чтобы ASE, который соответствует некоторому блоку SIB, содержал только операции, необходимые для выполнения задач этого SIB.

Описание тех ASE, которые содержат операции, относящиеся к интерфейсам SSF/CCF-SCF и SRF-SCF, отличается от описания ASE с операциями, относящимися к интерфейсу SCF-SDF. Первая из этих двух групп ASE специфицируется с помощью макроописаний, в то время как для спецификации второй группы ASE используются информационные объекты последней версии языка ASN.1.

Макроописание ASE с операциями в интерфейсах SSF/CCF-SCF и SRF-SCF содержит два поля – CONSUMER-INVOKES и SUPPLIER-INVOKES, – относящихся, соответственно, к потребителю и к поставщику услуги ASE. В каждом из этих двух полей перечислены операции, которые запрашиваются объектом, соответствующим названию поля.

В таблице 3.1.6 перечислены типы ASE и сведения о содержащихся в них операциях.

Таблица 3.1.6

Тип ASE	Операции	Направление
SCF activation ASE	<i>InitialDP</i>	SSF-SCF
SCF/SRF activation of assist ASE	<i>AssistRequestInstructions</i>	SRF-SCF, SSF-SCF
Assist connection establishment ASE	<i>EstablishTemporaryConnection</i>	SCF-SSF
Generic disconnect resource ASE	<i>DisconnectForwardConnection</i>	SCF-SSF
Non-assisted connection establishment ASE	<i>ConnectToResource</i>	SCF-SSF
Connect ASE (elementary SSF function)	<i>Connect</i>	SCF-SSF
Call handling ASE (elementary SSF function)	<i>ReleaseCall</i>	SCF-SSF
BCSM Event handling ASE	<i>RequestReportBCSMEvent</i> <i>EventReportBCSM</i>	SCF-SSF SSF-SCF
Charging Event handling ASE	<i>RequestNotificationChargingEvent</i> <i>EventNotificationCharging</i>	SCF-SSF SSF-SCF
SSF call processing ASE	<i>CollectInformation</i> <i>Continue</i>	SCF-SSF
SCF call initiation ASE\	<i>InitiateCallAttempt</i>	SCF-SSF
Timer ASE	<i>ResetTimer</i>	SCF-SSF
Billing ASE	<i>FurnishChargingInformation</i>	SCF-SSF
Charging ASE	<i>ApplyCharging</i> <i>ApplyChargingReport</i>	SCF-SSF SSF-SCF
Traffic management ASE	<i>CallGap</i>	SCF-SSF
Service management ASE	<i>ActivateServiceFiltering</i> <i>ServiceFilteringResponse</i>	SCF-SSF SSF-SCF
Call report ASE	<i>CallInformationRequest</i> <i>CallInformationReport</i>	SCF-SSF SSF-SCF
Signalling control ASE	<i>SendChargingInformation</i>	SCF-SSF
Specialized resource control ASE	<i>PlayAnnouncement</i> <i>PromptAndCollectUserInfo</i> <i>SpecializedResourceReport</i>	SCF-SSF (SRF) SRF-SCF
Cancel ASE	<i>Cancel</i>	SCF-SSF
Activity Test ASE	<i>ActivityTest</i>	SCF-SSF

3.1.5.3 Прикладные контексты (AC)

Прикладные контексты представляют собой группы прикладных сервисных элементов. Прикладные сервисные элементы сгруппированы в прикладные контексты так, чтобы они покрывали настолько много вариантов реализаций, насколько это виделось компаниям, участвовавшим в разработке стандарта.

Таблица 3.1.7

AC	Связанные ASE	Инициатор диалога
Core-INAP-CS1-SSP-to-SCP-AC	SCF-activation-ASE Assist-connection-establishment-ASE Generic-disconnect-resource-ASE Non-assisted-connection-establishment-ASE Connect-ASE Call-handling-ASE BCSM-event-handling-ASE Charging-event-handling-ASE SSF-call-processing-ASE Timer-ASE Billing-ASE Charging-ASE Traffic-management-ASE Call-report-ASE Signalling-control-ASE Specialized-resource-control-ASE Cancel-ASE Activity-test-ASE	SSP (InitialDP)
Core-INAP-CS1-assist-handoff-SSP-to-SCP-AC	SCF-SRF-activation-of-assist-ASE Generic-disconnect-resource-ASE Non-assisted-connection-establishment-ASE Call-handling-ASE Timer-ASE Billing-ASE Charging-ASE Specialized-resource-control-ASE Cancel-ASE Activity-test-ASE	SSP (AssistRequestInstructions)
Core-INAP-CS1-IP-to-SCP-AC	SCF-SRF-activation-of-assist-ASE Timer-ASE Specialized-resource-control-ASE Cancel-ASE Activity-test-ASE	IP (AssistRequestInstructions)
Core-INAP-CS1-SCP-to-SSP-AC	Assist-connection-establishment-ASE Generic-disconnect-resource-ASE Non-assisted-connection-establishment-ASE Connect-ASE Call-handling-ASE BCSM-event-handling-ASE Charging-event-handling-ASE SSF-call-processing-ASE SCF-call-initiation-ASE Timer-ASE Billing-ASE Charging-ASE Call-report-ASE Signalling-control-ASE Specialized-resource-control-ASE Cancel-ASE Activity-test-ASE	SCP (InitiateCallAttempt)
Core-INAP-CS1-SCP-to-SSP-traffic-management-AC	traffic-management-ASE	SCP (CallGap)
Core-INAP-CS1-SCP-to-SSP-service-management-AC	service-management-ASE	SCP (ActivateServiceFiltering)
Core-INAP-CS1-SSP-to-SCP-service-management-AC	service-management-ASE	SSP (ServiceFilteringResponse)

Так же, как и в случае с ASE, описание прикладных контекстов AC, относящихся к интерфейсам SSF/CCF-SCF и SRF-SCF, отличается от описания контекстов, относящихся к интерфейсу SCF-SDF. Здесь тоже первая группа AC специфицируется с помощью макроописаний, а для спецификации второй группы AC используются информационные объекты.

Макроописание прикладного контекста AC для интерфейсов SSF/CCF-SCF и SRF-SCF содержит два поля, которые идентифицируют прикладные сервисные элементы, используемые, соответственно, инициатором диалога и отвечающей стороной. В таблице 3.1.7 перечислены прикладные контексты CS-1 INAP и типы содержащихся в каждом из них прикладных сервисных элементов.

Глава 3.2

Процедуры прикладного объекта SSF

3.2.1 Функциональная модель и интерфейсы SSF-AE

Процедуры прикладного объекта SSF-AE относятся к интерфейсу между SSP и SCP. Процедуры основаны на использовании системы сигнализации № 7 (ОКС-7), однако могут быть использованы и другие системы сигнализации, например, DSS1.

В соответствии с архитектурой, предложенной в рекомендациях ITU-T Q.700, Q.771 и Q.1400, прикладной объект AE (Application entity) включает в себя подсистему TCAP ОКС-7 и один или более прикладных сервисных элементов ASE (Application service element), называемых ТС-пользователями. Эти ASE взаимодействуют с подсистемой TCAP, используя примитивы, описанные в стандарте ETSI ETS 300 287 (ITU-T Q.771).

Функциональная модель прикладного объекта SSF-AE представлена на рисунке 3.2.1. Сервисные элементы ASE взаимодействуют с SCF, используя интерфейс с подсистемой TCAP, а также интерфейсы с CCF и с функциями эксплуатационного управления. Предметом рассмотрения данной главы является область, которая на рисунке 3.2.1 затемнена.

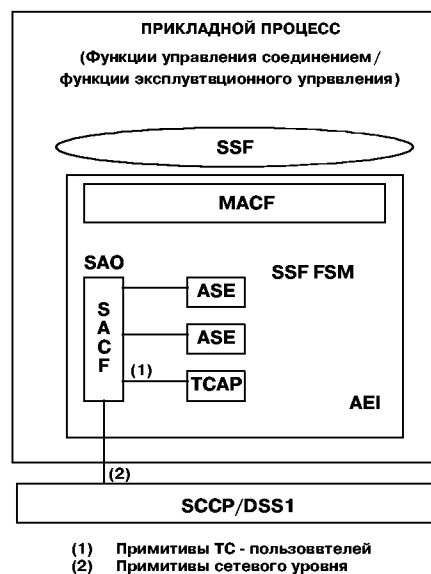


Рис. 3.2.1 Функциональная модель прикладного объекта SSF-AE

В интерфейсе (1) используются примитивы ТС-пользователей, определенные в ETS 300 287 (ITU-T Q.771), а в интерфейсе (2) – примитивы сетевого уровня (N-Primitives), определенные в ETS 300 009 (ITU-T Q.771).

3.2.2 Связь между SSF-FSM и функциями CCF/функциями эксплуатационного управления

Интерфейс, через который происходит обмен примитивами между машиной конечных состояний SSF-FSM и функциями CCF/функциями эксплуатационного управления – внутренний и не подлежит стандартизации в CS-1. Тем не менее, концептуально и функционально этот интерфейс должен соответствовать модели BCSM, подробно рассмотренной в предыдущем разделе.

Функции административного управления, которые относятся к исполнению операций, принятых от SCF, выполняет объект административного управления SSME (SSF management entity). Этот объект содержит управляющую часть (SSME-Control) и несколько машин конечных состояний (SSME-FSM). Управляющая часть имеет интерфейсы с разными SSF-FSM и, соответственно, с разными SSME-FSM, а также интерфейс с администратором доступа FEAM (Functional entity access manager) к функциональным объектам. Интерфейсы SSF показаны на рисунке 3.2.2.

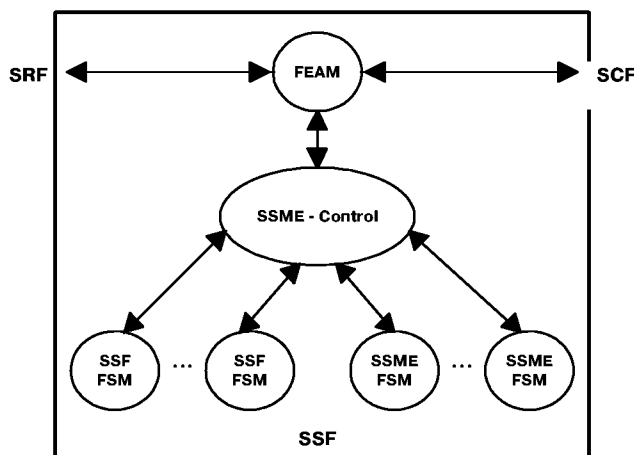


Рис. 3.2.2 Интерфейсы SSF

Администратор доступа FEAM выполняет функции нижнего уровня поддержки интерфейсов. В число этих функций входят создание и поддержка интерфейсов с SCF и SRF, контроль прохождения (и, если это требуется, – установка в очередь) сообщений, поступающих от

SCF и SRF к SSME-control, форматирование, установка в очередь (если требуется) и передача к SCF и SRF сообщений, принятых от SSME-Control.

Управляющая часть SSME-Control поддерживает диалоги с SCF и SRF всех машин SSF-FSM, образующихся по мере возникновения вызовов. Тот факт, что вызовы возникают асинхронно, требует, чтобы существовал отдельный объект, ответственный за выполнение задач создания, привлечения и поддержки машин конечных состояний SSF-FSM. SSME-Control интерпретирует сообщения, которые поступают от других функциональных объектов, и преобразует их в соответствующие события SSF-FSM; преобразует выходные данные SSF-FSM в сообщения, подлежащие передаче к другим функциональным объектам; выделяет асинхронные (по отношению к обслуживанию вызовов) действия, относящиеся к функциям административного управления и контроля в SSF, и создает машины конечных состояний SSME-FSM. Например, объект SSME производит не связанные с конкретным вызовом действия, обусловленные изменениями правил просеивания (Filtering) или условий прореживания (Call gap) потока вызовов, а SSME-Control отделяет SSF-FSM от функций просеивания и прореживания путем создания машин конечных состояний (SSME-FSM) отдельно для каждого контекста операций административного управления.

Разные контексты машин SSME-FSM могут отличаться один от другого адресной информацией, которая содержится в операциях, инициирующих диалог. В случае операции просеивания (*ActivateServiceFiltering*) такая информация содержится в параметре *filteringCriteria*, и все операции *ActivateServiceFiltering*, использующие тот же адрес, направляются к той машине конечных состояний SSME-FSM, которая создана для этого контекста просеивания. Операции *ActivateServiceFiltering*, предписывающие другие критерии просеивания, вызывают создание других SSME-FSM.

SSF-FSM передает инструкции для работы с вызовом в соответствующую BCSM. Точки DP могут динамически определяться как точки обнаружения событий (Event DP), требующие, чтобы SSF-FSM оставалась активной. В точке, где дальнейшее взаимодействие с SCF не требуется, машина SSF-FSM может быть остановлена, а BCSM будет продолжать работу с данным вызовом. Последующие триггерные точки обнаружения TDP в модели BCSM могут привести к созданию для того же самого вызова новых машин конечных состояний SSF-FSM.

Поскольку услуги и атрибуты услуг IN для CS-1 относятся к типу A, то одна SSF-FSM соотносится только с одной из функционально разделенных частей процесса обслуживания вызова, т.е. только с исходящей или только с входящей BCSM, но не с обеими одновременно.

3.2.3 Машина конечных состояний объекта SSME-FSM

Диаграмма состояний SSME-FSM приведена на рисунке 3.2.3. Заметим, что SSME-FSM независима от индивидуальных SSF-FSM.

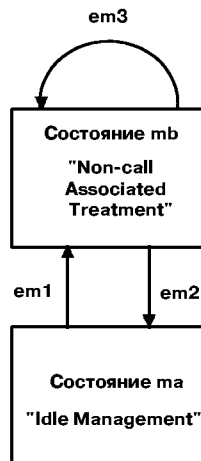


Рис. 3.2.3 Диаграмма состояний SSME-FSM

Переход в состояние «Non-call Associated Treatment» (обработка, не связанная с конкретным вызовом) из состояния «IdleManagement» происходит, когда принимается одна из следующих не связанных с обработкой вызова операций (переход em1): *ActivateServiceFiltering*, *CallGap*, *ActivityTest*.

Операция *CallGap* может быть принята как при транзакции в контексте вызова, так и вне такой транзакции. Операция *ActivityTest* принимается только при транзакции в контексте вызова. Операция *ActivateServiceFiltering* может быть принята только при транзакции вне контекста вызова.

В состоянии «Non-call Associated Treatment» могут произойти следующие события:

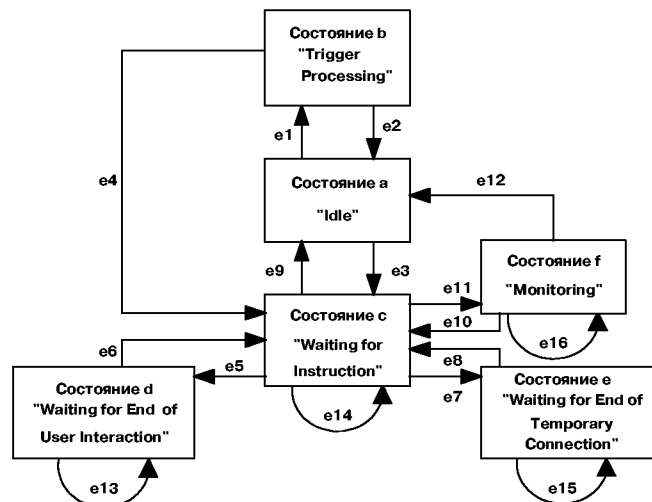
- если услуга Service Filtering активизирована, SSF должен передать к SCF операцию *ServiceFilteringResponse*; SSME-FSM остается в том же состоянии (переход em3);
- если услуга Service Filtering активизирована, но время просеивания истекло, SSME-FSM должна перейти в состояние «IdleManagement» (переход em2) и передать к SCF операцию *ServiceFilteringResponse*;
- если сработал таймер процедуры прореживания вызовов (*CallGap*), SSME-FSM должна перейти в состояние «IdleManagement» (переход em2);

- если активизирована услуга прореживания вызовов (Call gap) или просеивания (Service filtering), а SSF принял операцию *CallGap/ActivateServiceFiltering* с теми же самыми критериями, то предыдущие параметры прореживания или просеивания заменяются новыми (переход em3) при условии, что текущее значение таймера не равно нулю; в противном случае SSME-FSM должна перейти в состояние «IdleManagement» (переход em2).

Все другие операции не влияют на состояния SSME-FSM; они проходят под контролем SSME-control к соответствующей SSF-FSM.

3.2.3.1 Диаграмма состояний SSF

На рисунке 3.2.4 показана диаграмма состояний SSF в узле коммутации услуг SSP при обработке вызова, требующего услуги Интеллектуальной сети.



Примечание: переходы, связанные с отказом от обслуживания, на схеме не показаны.

Рис. 3.2.4 Диаграмма состояний SSF-FSM

Рассмотрим некоторые общие правила, относящиеся к описываемым ниже состояниям SSF-FSM. Один или более компонентов, принятых в одном или нескольких сообщениях TCAP, могут включать в себя одну операцию или последовательность нескольких операций и должны обрабатываться следующим образом.

Операции должны обрабатываться в порядке их поступления. Каждая операция может вызвать изменение состояния FSM, независимо от того, сколько операций было принято в одном сообщении. SSF поочередно проверяет принимаемые операции и выполняет предписываемые ими действия до тех пор, пока при их выполнении FSM остае-

ся в том же состоянии. Если какая-либо операция в последовательности вызывает переход FSM в другое состояние, то все следующие операции сохраняются в буфере до момента окончания выполнения текущей операции. Во всех остальных случаях ожидается событие, которое может вызвать изменение текущего состояния FSM. Таким событием может быть окончание выполнения действий, предписанных операцией, или внешнее событие.

Например, SSF может принять операции *FurnishChargingInformation*, *ConnectToResource* и *PlayAnnouncement* в последовательности компонентов внутри одного сообщения TCAP. После приема этого сообщения выполняются операции *FurnishChargingInformation* и *ConnectToResource*, в то время как SSF находится в состоянии ожидания инструкций («Waiting For Instructions»). После выполнения действий согласно операции *ConnectToResource* (и действий, предписанных операцией *FurnishChargingInformation*) машина состояний SSF-FSM переходит в состояние ожидания окончания взаимодействия с пользователем («Waiting For End Of User Interaction»). Операция *PlayAnnouncement* транслируется в SRF, а SSF остается в состоянии «Waiting For End Of User Interaction».

Если при обработке одной из операций последовательности обнаруживается ошибка, SSF-FSM приступает к обработке ошибки и сбрасывает все оставшиеся операции этой последовательности. Если смысл операции не ясен или выходит за рамки контекста, диалог прекращается.

В любом состоянии SSF-FSM информация об ошибке, обнаруженной в принятой операции, передается функциям техобслуживания, а SSF-FSM остается в том же состоянии. В зависимости от класса операции SSF может информировать об ошибке SCF, используя для этого соответствующий компонент TCAP.

В любом состоянии (кроме исходного состояния «Idle») при отказе вызывающей стороны от обслуживания до получения ответа (т.е. перед точкой PIC «Active» в модели BCSM) SSF-FSM должна дать CCF инструкцию прекратить обслуживание вызова и убедиться, что все назначенные для его обслуживания ресурсы CCF освобождены, а затем завершить работу в соответствии с установленным алгоритмом и перейти в состояние «Idle».

В любом состоянии (исключая состояние «Idle»), если одна из сторон дала отбой после того как связь была установлена (т.е. в точке PIC «Active» в модели BCSM), SSF-FSM должна завершить работу в соответствии с установленным алгоритмом и перейти в состояние «Idle».

В SSF имеется таймер T_{SSF} , который предназначен для предотвращения чрезмерно длительной приостановки обслуживания вызова, а также для защиты ассоциации между SSF и SCF. Таймер запускается в следующих случаях:

- когда SSF передает операцию *InitialDP* (состояние с: «Waiting For Instructions» на рис. 3.2.4) или операцию *AssistRequestInstructions* (состояние b: «Waiting For Instructions» на рис. 3.2.5, рассматриваемом ниже для случая ассистирования/передачи управления). До получения первого ответа от SCF таймер T_{SSF} может быть переустановлен только один раз с помощью операции *ResetTimer*. После приема первого ответа таймер может переустанавливаться сколь угодно много раз.
- когда SSF входит в состояние «Waiting For Instructions» при любых других условиях. SCF может переустанавливать таймер T_{SSF} с помощью операции *ResetTimer* любое число раз.
- когда SSF входит в состояние «Waiting For End of User Interaction» или в состояние «Waiting For End of Temporary Connection». SCF может переустанавливать таймер T_{SSF} с помощью операции *ResetTimer* любое число раз.

В трех описанных случаях таймер T_{SSF} может быть настроен, соответственно, на три различные выдержки времени, определенные прикладным процессом. Приняв или передав операцию, отличную от приведенных выше, SSF должен установить таймер T_{SSF} в нуль, сохранив его настройку на последнюю из использованных выдержек времени. Эта настройка может быть либо той, которая была принята для одного из трех случаев, описанных выше, либо той, которая указывается в операции *ResetTimer* (смотря по тому, какое из событий наступило последним). В состоянии «Monitoring» таймер T_{SSF} не используется. По истечении выдержки времени таймера T_{SSF} SSF-FSM переходит в состояние «Idle» и прерывает взаимодействие с SCF.

Диаграмма состояний SSF включает в себя следующие переходы (события):

e1	TDP encountered	Встречена точка TDP
e2	Trigger fail (ETSI)	Сбой при обработке TDP
e3	<i>InitiateCallAttempt</i> received	Принята операция <i>InitiateCallAttempt</i>
e4	Trigger detected (ETSI)	Точка TDP обработана
e5	User interaction requested	Затребовано взаимодействие с пользователем
e6	User interaction ended	Взаимодействие с пользователем закончено
e7	Temporary connection created	Создано временное соединение
e8	Temporary connection ended	Временное соединение разрушено
e9	Idle return from wait for instruction	Переход в "Idle" из "Wait for instruction"
e10	EDP-R encountered	Встречена точка EDP-R

e11	Routing instruction received	Инструкции для маршрутизации получены
e12	EDP-N last encountered	Встречена последняя точка EDP-N
e13	Waiting for End of User Interaction state no change	Состояние "Waiting for End of User Interaction" без изменений
e14	Waiting for Instructions state no change	Состояние "Waiting for Instructions" без изменений
e15	Waiting for End of Temporary Connection state no change	Состояние "Waiting for End of Temporary Connection" без изменений
e16	Monitoring state no change	Состояние "Monitoring" без изменений
e17	Abandon (from any state) (not shown in the SSF state diagram)	Отбой в любом состоянии (на диаграмме не показано)
e18	Disconnect (from any state) (not shown in the SSF state diagram)	Разъединение в любом состоянии (на диаграмме не показано)
e19	Non-Call Associated Treatment from any state (not shown in the SSF state diagram)	Не связанная с вызовом обработка в любом состоянии (на диаграмме не показано)

Диаграмма состояний SSF включает в себя следующие состояния:

Состояние a	Idle	Исходное
Состояние b	Trigger Processing (ETSI)	Обработка точек TDP
Состояние c	Waiting for Instructions	Ожидание инструкций
Состояние d	Waiting for End of User Interaction	Ожидание окончания взаимодействия с пользователем
Состояние e	Waiting for End of Temporary Connection	Ожидание окончания временного соединения
Состояние f	Monitoring	Мониторинг

3.2.3.1.1 Состояние a – «Idle»

SSF-FSM переходит в состояние «Idle» из любого состояния при приеме или передаче примитива ABORT TCAP вследствие возникновения некорректных условий. Так, SSF-FSM переходит в состояние «Idle», когда происходит одно из следующих событий:

- вызывающая сторона дает отбой до окончания обслуживания или одна или более сторон, участвующих в связи, нарушает соединение;
- в состоянии «Waiting for Instructions» выполняются действия, которые предписаны операцией Connect, и не существует назначенных точек обнаружения событий EDP, а также отсутствуют запросы отчетов (переход e9);

- выдержка времени, определяемая таймером T_{SSF} , истекает в одном из следующих состояний: «Waiting for Instructions», «Waiting for End of User Interaction» или «Waiting for End of Temporary Connection»;
- выполняются действия, предписанные операцией *ReleaseCall*, в состоянии «Waiting for Instructions» (переход e9) или в состоянии «Monitoring» (переход e12);
- в состоянии «Monitoring» встретилось последнее событие, связанное с начислением платы, и не существует назначенных точек обнаружения событий EDP (переход e12).

При наличии полученной ранее и не отработанной операции *Call-InformationRequest*, SSF должен передать операцию *CallInformation-Report* в SCF до перехода в состояние «Idle». В состоянии «Idle» могут произойти следующие связанные с вызовом события:

- от CCF поступает индикация того, что встречена активная точка TDP, которая, возможно, связана с обращением к услуге IN: SSF-FSM переходит в состояние «Trigger Processing» (переход e1)
- от SCF принято сообщение, относящееся к новой транзакции и содержащее операцию *InitiateCallAttempt*: SSF-FSM переходит в состояние «Waiting for Instructions» (переход e3).

Любые другие операции, принятые от SCF, когда SSF находится в состоянии «Idle», должны расцениваться и обрабатываться как ошибочные. О таком событии должно быть сообщено функциям техобслуживания, а транзакция должна быть прервана в соответствии с процедурой, определенной для подсистемы TCAP.

3.2.3.1.2 Состояние b – «Trigger Processing»

Как только встречается активная точка TDP, автомат SSF-FSM активизируется и переходит из состояния «Idle» в состояние «Trigger Processing» (переход e1). Находясь в этом состоянии, SSF/CCF должен выполнить действия, связанные с обработкой триггерных точек DP, а именно:

- проверить, не активны ли функции просеивания или прореживания вызовов;
- убедиться в доступности SCF;
- определить, удовлетворяются ли критерии DP;
- обеспечить корректность взаимодействия атрибутов.

Кроме того, SSF/CCF должен собрать и проверить параметры, необходимые для передачи в SCF операции *InitialDP*: если DP является точкой TDP-R, то, в соответствии с процедурой обработки DP, передать операцию *InitialDP* в SCF, и перейти в состояние «Waiting for

Instructions» (переход e4); если же обработка DP привела к неудаче, – вернуться в состояние «Idle» (переход e2).

Неуспешным результатом обработки DP считаются следующие ситуации:

- сказалось действие функции прореживания CallGapping: SSF-FSM передает в CCF инструкцию завершить обслуживание данного вызова;
- сказалось действие функции просеивания Service Filtering: вызов учитывается (если это требуется), и SSF-FSM передает в CCF инструкцию завершить его обслуживание;
- не обнаружен соответствующий критерий: SSF-FSM возвращает управление обслуживанием вызова в CCF;
- вызывающая сторона отказалась от обслуживания (отбой): SSF-FSM возвращает управление в CCF и завершает работу в соответствии с описанным выше алгоритмом;
- отсутствует достаточная для обслуживания вызова информация: SSF-FSM передает в CCF инструкцию завершить обслуживание данного вызова в соответствии со стандартным алгоритмом;
- недоступны функции управления услугами SCF: SSF-FSM передает в CCF инструкцию маршрутизировать вызов, например, к автоинформатору;
- для данного вызова уже имеется управляющая логическая связь: SSF-FSM возвращает управление обслуживанием вызова в CCF.

3.2.3.1.3 Состояние с – «Waiting for Instructions»

В состояние «Waiting for Instructions» SSF-FSM может перейти из состояния «Trigger Processing» (переход e4), или прямо из состояния «Idle», если SSF-FSM принимает от SCF примитив-индикацию TC-BEGIN, содержащий операцию *InitiateCallAttempt* (переход e3), или из состояния «Monitoring» при обнаружении точки EDP-R (переход e10). Во втором случае (e3) SSF-FSM немедленно переходит в состояние «Waiting for Instructions», а затем продолжает выполнять функции в соответствии с содержанием операции *InitiateCallAttempt* и любых других принимаемых операций.

В состоянии «Waiting for Instructions» SSF-FSM ожидает инструкций от SCF; обработка вызова приостанавливается и в момент прихода в это состояние запускается таймер T_{SSF} . Затем могут произойти следующие события:

- пользователь набрал дополнительные цифры (при использовании в сети открытой системы нумерации): CCF должен сохранить эти цифры.

- пользователь отказывается от обслуживания (дает отбой): данная ситуация должна обрабатываться в соответствии с общим алгоритмом.
- истекла выдержка времени, определяемая таймером T_{SSF} : SSF-FSM переходит в состояние «Idle», CCF, если это возможно, маршрутизирует вызов (по умолчанию – к автоинформатору), информация о срабатывании таймера T_{SSF} передается функциям техобслуживания и транзакция прекращается.
- поступила операция от SCF. SSF-FSM выполняет действия, предписанные принятой операцией, как это описывается ниже.

При приеме от SCF операций: *RequestReportBCSMEEvent*, *RequestNotificationChargingEvent*, *ResetTimer*, *FurnishChargingInformation*, *ApplyCharging*, *CallInformationRequest*, *SendChargingInformation*, *Cancel* SSF выполняет предписываемые ими действия и остается в том же состоянии (переход e14).

Прием от SCF операции *ConnectToResource* и выполнение предписанных ею действий приводит к переходу SSF в состояние «Waiting for End of User Interaction» (переход e5).

Прием от SCF операции *EstablishTemporaryConnection* и выполнение предписанных ею действий приводит к переходу SSF в состояние «Waiting for End of Temporary Connection» (переход e7).

Прием от SCF операций *Connect*, *CollectInformation*, *Continue*, *ReleaseCall* и выполнение предписанных ими действий приводит к переходу SSF либо в состояние «Monitoring» (если были встречены какие-либо точки EDP, или были запрошены какие-либо отчеты) (переход e11), либо в состояние «Idle» (переход e9).

Прием от SCF операции *ReleaseCall* и выполнение предписанных ею действий приводит к переходу SSF в состояние «Idle» (переход e9). Если имеется ожидающий ответа запрос *CallInformationRequest*, то, прежде чем вернуться в состояние «Idle», SSF передает к SCF операцию *CallInformationReport*.

В состоянии «Waiting for Instructions» должны выполняться действия, которые предписаны операцией *InitiateCallAttempt*, принятой в состоянии «Idle» (переход e3).

Действия, связанные с любой другой принятой в этом состоянии операцией, выполняются в соответствии с описанным выше общим алгоритмом.

3.2.3.1.4 Состояние d – «Waiting for End of User Interaction»

SSF переходит в состояние «Waiting for End of User Interaction» из состояния «Waiting for Instructions» (переход e5) при приеме операции *ConnectToResource*. В этом состоянии могут произойти следующие события:

- от SCF принята одна из следующих операций – *PlayAnnouncement*, *PromptAndCollectUserInfo* или *CancelAnnouncement*: при условии, что операция корректна, она передается в SRF для исполнения, а SSF-FSM остается в состоянии «Waiting for End of User Interaction» (переход e13).
- от SRF принята операция *SpecializedResourceReport* или возврат результата исполнения операции *PromptAndCollectUserInfo*: при условии, что операция корректна, она транслируется в SCF, а SSF-FSM остается в состоянии «Waiting for End of User Interaction» (переход e13).
- истекла выдержка времени, определяемая таймером T_{SSF} : SSF-FSM переходит в состояние «Idle», CCF, если это возможно, маршрутизирует вызов (по умолчанию – к автоинформатору), информация о срабатывании таймера T_{SSF} передается функциям техобслуживания и транзакция прекращается.
- пользователь отказывается от обслуживания (дает отбой): данная ситуация должна обрабатываться в соответствии с общим алгоритмом.
- поступила операция от SCF: SSF-FSM выполняет действия, предписанные принятой операцией, как это описывается ниже.

При приеме от SCF операций *RequestNotificationChargingEvent*, *ResetTimer*, *FurnishChargingInformation*, *ApplyCharging* и *SendChargingInformation* SSF выполняет предписываемые ими действия и остается в том же состоянии (переход e13).

SSF может принять от SCF операцию *DisconnectForwardConnection* и выполнить требуемые действия. Кроме того, может поступить сигнал о разъединении от SRF. В обоих случаях соединение с SRF нарушается и SSF-FSM переходит в состояние «Waiting for Instructions» (переход e6). Связь с пользователем при этом не нарушается.

Действия, связанные с любой другой принятой в этом состоянии операцией, выполняются в соответствии с описанным выше общим алгоритмом.

3.2.3.1.5 Состояние e – «Waiting for End of Temporary Connection»

SSF переходит в состояние «Waiting for End of Temporary Connection» из состояния «Waiting for Instructions» (переход e7) при приеме операции *EstablishTemporaryConnection*. Запускается таймер T_{SSF} . Вызов направляется к ассистирующему SSF/SRF, а его обслуживание данным SSF приостанавливается до окончания процедуры ассистирования.

В состоянии «Waiting for End of Temporary Connection» могут произойти следующие события:

- истекла выдержка времени, определяемая таймером T_{SSF} : SSF-FSM переходит в состояние «Idle», CCF, если это возможно, маршрутизирует вызов (по умолчанию – к автоинформатору), информация о срабатывании таймера T_{SSF} передается функциям техобслуживания и транзакция прекращается.
- от CCF поступила индикация прекращения временного соединения: SSF переходит в состояние «Waiting for Instructions» (переход e8). Связь с вызывающим пользователем при этом не нарушается.
- пользователь отказывается от обслуживания (дает отбой): данная ситуация должна обрабатываться в соответствии с общим алгоритмом.
- поступила операция от SCF: SSF-FSM выполняет действия, предписанные принятой операцией, как это описывается ниже.

При приеме от SCF операций *RequestNotificationChargingEvent*, *ResetTimer*, *FurnishChargingInformation*, *ApplyCharging* и *SendChargingInformation* SSF выполняет предписываемые ими действия и остается в том же состоянии (переход e15).

SSF может принять от SCF операцию *DisconnectForwardConnection* и выполнить требуемые действия. Кроме того, может поступить сигнал о разъединении от SRF. В обоих случаях соединение с SRF нарушается и SSF-FSM переходит в состояние «Waiting for Instructions» (переход e8). Связь с пользователем при этом не нарушается.

Действия, связанные с любой другой принятой в этом состоянии операцией, выполняются в соответствии с описанным выше общим алгоритмом.

3.2.3.1.6 Состояние f – «Monitoring»

SSF переходит в состояние «Monitoring» из состояния «Waiting For Instructions» (переход e11) при приеме операций *Connect*, *CollectInformation*, *Continue*, когда назначены одна или более точек обнаружения событий EDP. Таймер T_{SSF} в состоянии «Monitoring» не используется, т.е. срабатывание этого таймера не оказывает никакого влияния на SSF-FSM. В состоянии «Monitoring» могут произойти следующие события:

- встречена точка EDP-N (нотификация): SSF должен уведомить об этом SCF, передав операцию *EventReportBCSM*, и остаться в состоянии «Monitoring» (переход e16), если назначены еще одна или более точек EDP, или перейти в состояние «Idle» (переход e12), если назначенных точек EDP больше не осталось.
- встречена точка EDP-R (запрос): SSF должен уведомить об этом SCF, передав операцию *EventReportBCSM*, и перейти в состояние «Waiting For Instructions» (переход e10).

- прием примитивов END или ABORT от подсистемы TCAP не оказывает никакого влияния на процесс обслуживания вызова: процесс может быть продолжен или завершен с доступной информацией. В этом случае SSF-FSM переходит в состояние «Idle» (переход e12), прекращая свою связь с процессом.
- пользователь отказывается от обслуживания (дает отбой): данная ситуация должна обрабатываться в соответствии с общим алгоритмом.
- поступила операция от SCF. SSF-FSM выполняет действия, предписанные принятой операцией, как это описывается ниже.

При приеме от SCF операций *RequestReportBCSMEvent*, *ResetTimer* и *SendChargingInformation* SSF выполняет предписываемые ими действия и остается в том же состоянии (переход e16). Прием от SCF операции *Cancel* и выполнение соответствующих действий переводит SSF в состояние «Idle» (переход e12).

При приеме от SCF операции *ReleaseCall* SSF-FSM выполняет предписываемые этой операцией действия, убеждается, что все ресурсы CCF, использовавшиеся для обработки вызова, освобождены, и переходит в состояние «Idle» (переход e12).

Действия, связанные с любой другой принятой в этом состоянии операцией, выполняются в соответствии с описанным выше общим алгоритмом.

3.2.3.2 Машина состояний SSF-FSM для случая Ассистирования/Передачи управления (Assisting/Handoff)

В CS-1 машина состояний Handoff SSF-FSM применима только к случаям передачи управления с целью окончательной обработки запроса услуги. Диаграмма состояний машины Assisting/Handoff SSF-FSM содержит следующие переходы (см. рис. 3.2.5):

ea1	Assist/Hand-off detected	Запрошена процедура Assist/Handoff
ea2	Assist/Hand-off ended (fail or success)	Процедура Assist/Handoff завершена (успешно или неуспешно)
ea3	User interaction requested	Затребовано взаимодействие с пользователем
ea4	User interaction ended	Взаимодействие с пользователем закончено
ea5	Waiting For Instruction state no change	Состояние ожидания инструкций «Waiting For Instruction» без изменения
ea6	Waiting For End Of User Interaction state no change	Состояние ожидания окончания процедуры взаимодействия с пользователем «Waiting For End Of User Interaction» без изменения

Диаграмма состояний Assisting/Handoff SSF-FSM содержит следующие состояния:

Состояние a	Idle	Исходное
Состояние b	Waiting for Instructions	Ожидание инструкций
Состояние c	Waiting for End of User Interaction	Ожидание окончания процедуры взаимодействия с пользователем

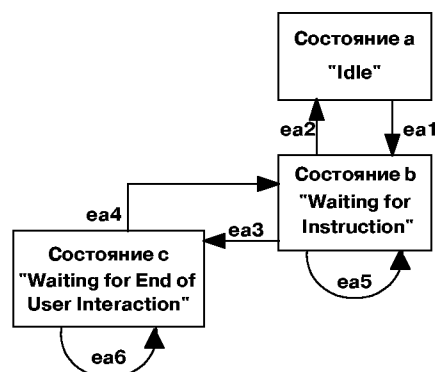


Рис. 3.2.5 Диаграмма состояний машины Assisting/Handoff SSF-FSM

3.2.3.2.1 Состояние a – «Idle»

SSF-FSM переходит в исходное состояние «Idle» в двух случаях:

- при приеме или передаче примитива ABORT подсистемы TCAP из-за некорректных условий в любом состоянии;
- при установленном временном соединении между инициирующим SSF и ассистирующим SSF, когда от инициирующего SSF поступает сигнал отключения разговорного тракта (переход ea2).

Войдя в состояние «Idle», ассистирующий SSF сбрасывает все ответы, которые подлежали передаче в SCF, но остались не переданными.

Любая операция, поступающая от SCF, когда ассистирующий SSF находится в состоянии «Idle», воспринимается им как ошибка.

3.2.3.2.2 Состояние b – «Waiting for Instructions»

В состояние «Waiting for Instructions» ассистирующая SSF-FSM переходит из состояния «Idle» при приеме ассистирующим SSF от другого SSF указания на необходимость ассистирования (переход ea1). В этом состоянии ассистирующий SSF передает в SCF операцию *AssistRequestInstruction*, ассистирующая SSF-FSM ожидает от SCF инструкций; обработка вызова приостанавливается и запускается таймер T_{SSF} , регламентирующий время ожидания. В состоянии «Waiting for Instructions» могут произойти следующие события:

- истекла выдержка времени, определяемая таймером T_{SSF} : ассистирующая SSF-FSM возвращается в состояние «Idle» (переход ea2), информация о срабатывании таймера T_{SSF} передается функциям техобслуживания и транзакция прекращается.
- со стороны инициирующего SSF поступил сигнал отключения несущего канала: SSF-FSM переходит в состояние «Idle» (переход ea2).
- поступила операция от SCF: SSF-FSM выполняет действия, предписанные принятой операцией, как это описывается ниже.

При приеме от SCF операций *ResetTimer*, *FurnishChargingInformation*, *Apply Charging* и *SendChargingInformation* ассистирующий SSF выполняет предписываемые ими действия, оставаясь в том же состоянии (переход ea5).

Прием ассистирующим SSF от SCF операции *ConnectToResource* и выполнение соответствующих действий вызывает переход FSM в состояние «Waiting for End of User Interaction» (переход ea3). Прием от SCF операции *ReleaseCall* и выполнение соответствующих действий вызывает переход FSM в состояние «Idle» (переход ea2).

Заметим, что операция *ReleaseCall* разрешена только в том SSF, которому было передано управление (в Handoff SSF). Однако, в случае, когда реализация оборудования не позволяет различать машины Handoff SSF и Assisting SSF, разрешается выполнять операцию *ReleaseCall* и в ассистирующем SSF.

Действия, связанные с любой другой принятой в этом состоянии операцией, выполняются в соответствии с описанным выше общим алгоритмом.

3.2.3.2.3 Состояние с – «Waiting for End of User Interaction»

Ассистирующий SSF переходит в состояние «Waiting for End of User Interaction» из состояния «Waiting for Instructions» (переход ea3) при приеме операции *ConnectToResource*. В состоянии «Waiting for End of User Interaction» могут произойти следующие события:

- от SCF принята одна из следующих операций – *PlayAnnouncement*, *PromptAndCollectUserInformation* или *Cancelannouncement*: при условии, что операция корректна, она передается в SRF для исполнения, а SSF-FSM остается в состоянии «Waiting for End of User Interaction» (переход ea6).
- от SRF принята операция *SpecializedResourceReport* или возврат результата исполнения операции *PromptAndCollectUserInformation*: при условии, что операция корректна, она транслируется в SCF, а SSF-FSM остается в состоянии «Waiting for End of User Interaction» (переход ea6).

- SRF указывает принявшему управление SSF на окончание процедуры взаимодействия с пользователем, инициируя разъединение: Handoff SSF-FSM возвращается в состояние «Waiting for Instructions» (переход ea4).
- истекла выдержка времени, определяемая таймером T_{SSF} : SSF-FSM переходит в состояние «Idle», CCF, если это возможно, маршрутизирует вызов (по умолчанию – к автоинформатору), информация о срабатывании таймера T_{SSF} передается функциям техобслуживания и транзакция прекращается.
- со стороны инициирующего SSF поступил сигнал отключения: SSF-FSM переходит в состояние «Idle», соединение с SRF нарушается и транзакция прекращается.
- поступила операция от SCF: SSF-FSM выполняет действия, предписанные принятой операцией, как это описывается ниже.

При приеме от SCF операции *ResetTimer* SSF выполняет предписываемые ею действия и остается в том же состоянии (переход ea6):

Прием от SCF операции *DisconnectForwardConnection* и выполнение машиной Assisting/handoff SSF соответствующих действий вызывает переход SSF-FSM в состояние «Waiting for Instructions» (переход ea4). Эта процедура действует только в том случае, если переход в состояние «Waiting for End of User Interaction» явился результатом выполнения действий, связанных с приемом операции *ConnectToResource*.

Действия, связанные с любой другой принятой в этом состоянии операцией, выполняются в соответствии с описанным выше общим алгоритмом.

Глава 3.3

Процедуры прикладного объекта SCF

3.3.1 Функциональная модель и интерфейсы SCF-AE

Процедуры объекта SCF-AE относятся к интерфейсу SCF-SSF. Процедуры основаны на использовании системы сигнализации №7 (ОКС-7), однако могут быть использованы и другие системы сигнализации. Процедуры не накладывают каких-либо ограничений на реализацию программ логики услуг (SLPs). Аналогично SSF-AE, прикладной объект SCF-AE включает в себя подсистему TCAP и один или более ASE, называемых TC-пользователями.

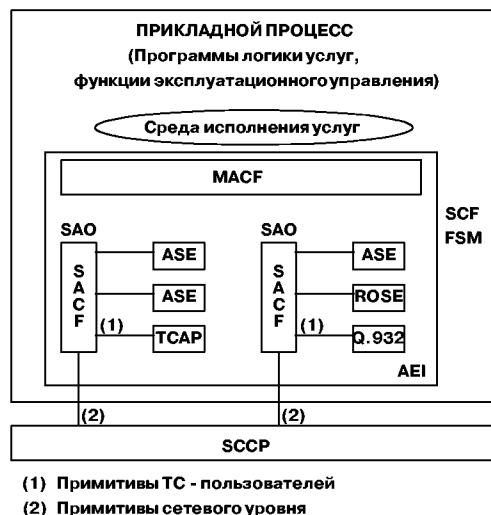


Рис. 3.3.1 Функциональная модель прикладного объекта SCF-AE

На рисунке 3.3.1. показана функциональная модель прикладного объекта SCF-AE и интерфейс с нижележащими уровнями системы сигнализации. Здесь же показан интерфейс с программами SLP и с функциями эксплуатационного управления. Модель представляет собой машину конечных состояний, обозначаемую SCF-FSM. Предметом рассмотрения данной главы является область, которая на рисунке 3.3.1. затемнена.

3.3.2 Связь между SCF-FSM и программами SLP/функциями эксплуатационного управления

Интерфейс, через который происходит обмен примитивами между машиной конечных состояний SCF-FSM и программами SLP/функ-

циями эксплуатационного управления – внутренний и не подлежит стандартизации в CS-1. SCF-FSM отвечает, прежде всего, за взаимодействие с FSM других функциональных объектов (SSF-FSM, SRF-FSM и SDF-FSM) и уведомляет программы SLP о возникающих в них событиях. Структура SCF-FSM приведена на рисунке 3.3.2.

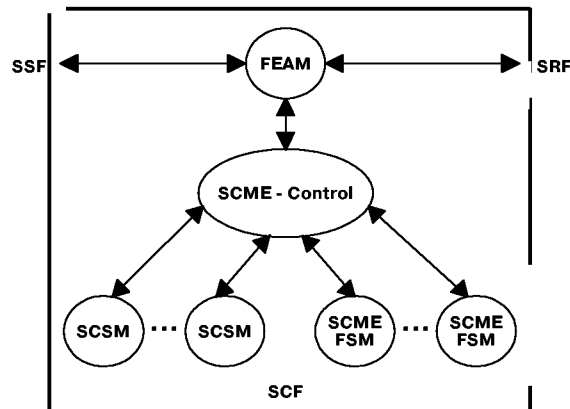


Рис. 3.3.2 Структура SCF-FSM

Каждый раз при обращении к IN со стороны конечного пользователя (т.е. при получении запроса от SSF), или если вызов инициирован запросом со стороны логики услуг, в SCF создается специальный объект – модель состояний процесса обработки вызова (SCF call state model, SCSM). Функции управления, связанные с выполнением действий, которые предписывают получаемые от SCF операции, возлагаются на другой объект, называемый объектом административного управления SCME (SCF management entity). SCME содержит управляющую часть (SCME-Control) и несколько машин конечных состояний административного управления – SCME FSM.

SCF должен обладать способностью одновременно и асинхронно обслуживать большое количество запросов. Этим объясняется необходимость выделения функций SCME-Control в отдельный объект, который решает задачи создания и технического обслуживания остальных объектов, входящих в состав SCF-FSM, а также обращения к этим объектам. Кроме того, SCME-Control служит интерфейсом между объектами SCSM и администратором доступа (FEAM) к функциональным объектам.

В дополнение к вышеперечисленным задачам SCME поддерживает диалоги всех машин состояний с SSF, SDF и SRF. В частности, SCME-control интерпретирует сообщения, поступающие от других функциональных объектов, и преобразует их в соответствующие события SCSM, преобразует получаемые от SCSM данные в сообщения для передачи к другим функциональным объектам, выполняет такие (асинхронные по отношению к обслуживанию вызова или не

связанные с ним) действия, как: управление потоком данных, просеивание, прореживание потока вызовов и наблюдение за ресурсами; поддерживает взаимодействие между SCF и другими функциональными объектами, а также обнаруживает другие ситуации, связанные с необходимостью создания SCME-FSM.

Разные контексты объектов типа SCME-FSM различаются на основании адресной информации, которая содержится в операциях, инициирующих услугу. Так, для процедуры просеивания адресная информация задается параметром «критерий просеивания» (Filtering Criteria), после чего все операции вида *ActivateServiceFiltering*, использующие одинаковый адрес, направляются к одной и той же машине SCME-FSM, созданной для этого контекста просеивания. Операции *ActivateServiceFiltering*, содержащие другие критерии просеивания, вызывают создание новых SCME-FSM.

Администратор доступа FEAM выполняет функции нижнего уровня поддержки интерфейсов, освобождая от этих функций SCME. В число таких функций входят создание и поддержка интерфейсов с SSF, SRF и SDF, контроль прохождения (и, если это требуется, – установка в очередь) сообщений, поступающих от SSF, SRF и SDF к SCME-control, форматирование, установка в очередь (если требуется) и передача к SSF, SRF и SDF сообщений, принятых от SCME-Control

Хотя SCSM имеет состояние и процедуры, обеспечивающие управление очередями, существует и другая альтернатива – управлять очередями вызовов в SSF-CCF, однако технические подробности того, как SSF/CCF осуществляет управление очередью, в концепции IN не рассматриваются.

3.3.3 Диаграмма состояний SCME

Пример диаграммы изменения состояний SCME показан на рисунке 3.3.3.

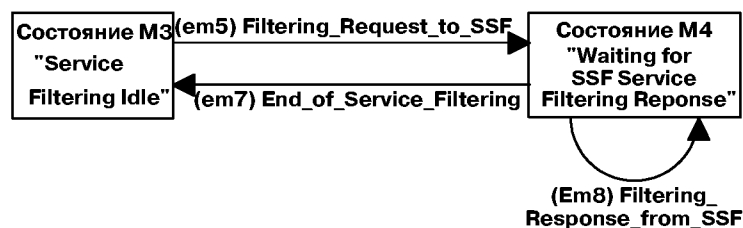


Рис. 3.3.3 Диаграмма состояний машины **Service Filtering** в SCME

SCME работает со следующими операциями: *ActivateServiceFiltering*, *ServiceFilteringReport*, *CallGap*, *ActivityTest*. Генерация операций

CallGap и *ActivityTest* не изменяет состояния SCME. Процедуры, связанные с операциями *ActivateServiceFiltering* и *ServiceFilteringReport*, рассматриваются ниже. При приеме иных операций SCME транслирует их к соответствующему SCSM, не изменяя своего состояния.

Состояние M3 – «Service filtering idle» (Просеивание, исходное состояние). В этом состоянии возможно одно событие:

- (Em5) *Filtering_Request_to_SSF* – внутреннее событие, является результатом того, что логика услуг нуждается в просеивании поступающих к SSF запросов обслуживания. Передается операция *ActivateServiceFiltering* и происходит переход FSM в состояние M4 «Waiting for SSF service filtering response».

Состояние M4 – «Waiting for SSF Service Filtering Response» (Ожидание ответа от SSF об активизации просеивания). В этом состоянии существуют два события:

- (Em7) *End_of_Service_Filtering* – внутреннее событие, состоящее в том, что истекла выдержка времени, определяемая таймером продолжительности просеивания в SCF. Это событие приводит к переходу в состояние M3 «Service filtering idle».
- (Em8) *Filtering_Response_from_SSF* – внешнее событие, состоящее в том, что поступил ответ на операцию *ActivateServiceFiltering*, предварительно отправленную к SSF. Это событие не ведет к переходу в другое состояние.

3.3.4 Диаграмма состояний SCSM

3.3.4.1 Общие принципы, операции и таймеры

На рисунке 3.3.4 приведена диаграмма состояний SCSM в соответствии с процедурами обработки вызовов, требующих услуг IN, при реализации SCF-FSM в любом из узлов вида SCP/AD/SN. Некоторые состояния могут содержать в себе несколько внутренних состояний и, соответственно, несколько машин (sub-FSM).

Рассмотрим общие правила, применимые более чем к одному состоянию.

Если в принятой операции обнаружена ошибка, об этом уведомляются SLP и функции технического обслуживания. Обычно SCSM остается в прежнем состоянии, однако ошибки могут обрабатываться по-разному. В зависимости от класса операции, уведомление о наличии ошибки может быть передано тому объекту, от которого принята операция.

Если со стороны SSF поступает уведомление об окончании диалога, SCSM информирует об этом SLP и возвращается в состояние «Idle». Ресурсы, выделенные для обслуживания вызова, включая ре-

сурсы, поддерживающие диалоги с другими функциональными объектами, должны быть освобождены (на диаграммах, для простоты, соответствующие переходы не показаны).

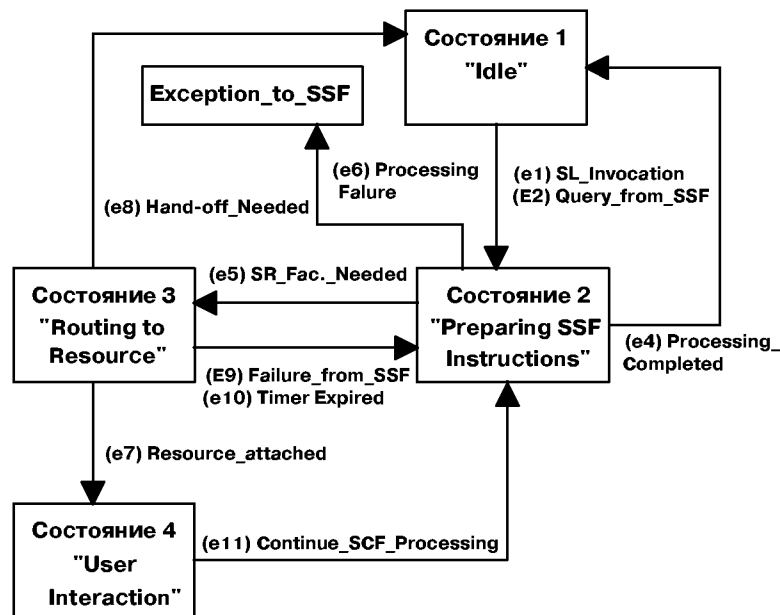


Рис. 3.3.4 Диаграмма состояний SCSM

Всякий раз, когда программа SLP запрашивает информацию о вызове, SCSM передает к SSF операцию *CallInformationRequest* и ожидает ответную операцию *CallInformationReport*. Если операция *CallInformationReport* задерживается, SCSM может принять ее в любом состоянии (кроме Idle). Получив от SLP (в любом состоянии, кроме Idle) информацию о том, что обработка завершена, SCSM остается в том же состоянии до получения операции *CallInformationReport*.

Для SCSM определены три таймера.

Таймер $T_{SCF-SSF}$ предназначен для перезапуска в SSF таймера T_{SSF} , предотвращающего чрезмерно длительную приостановку обслуживания вызова и защищающего ассоциацию между SSF и SCF. $T_{SCF-SSF}$ запускается в следующих случаях:

- когда SCF принимает операции *InitialDP* или *AssistRequestInstructions* (состояния 2.1 и 2.2.1: «Preparing SSF Instructions»). Таймер $T_{SCF-SSF}$ перезапускается при посылке к SSF первого запроса, отличного от операции *ResetTimer*. После первого срабатывания таймер может быть перезапущен, после его второго срабатывания машина SCSM сообщает об этом SLP и функциям техобслуживания и переходит в состояние «Idle»;

- когда SCF входит в состояние «Queueing». После срабатывания $T_{SCF-SSF}$ SCF может перезапускать этот таймер, используя операцию *ResetTimer*, любое число раз;
- когда SCF входит в состояние «Waiting for Assist Request Instructions» или в состояние «User Interaction». После срабатывания $T_{SCF-SSF}$ SCF может перезапускать его, используя операцию *ResetTimer*, любое число раз.

Во всех трех случаях выдержка времени, определяемая таймером $T_{SCF-SSF}$ может иметь разные значения, причем значение выдержки времени по таймеру $T_{SCF-SSF}$ всегда меньше, чем соответствующее значение выдержки времени по таймеру T_{SSF} . При приеме или передаче любой другой операции SCF должен перезапустить $T_{SCF-SSF}$. В состоянии «Waiting for Notification or Request» таймер $T_{SCF-SSF}$ не используется.

Второй таймер, определенный для SCSM, – $T_{ASSIST/HANDOFF}$ – предназначен для предотвращения чрезмерно длительной приостановки обслуживания вызова при выполнении процедур ассистирования/передачи управления (assist/handoff). Таймер $T_{ASSIST/HANDOFF}$ запускается в случае, когда SCSM передает операцию *EstablishTemporaryConnection* или *Connect*, содержащую «связывающий идентификатор» (Correlation ID). Таймер останавливается, когда SCSM принимает от ассистирующего SSF операцию *AssistRequestInstructions*. По истечении выдержки времени, определяемой этим таймером, SCSM информирует логику услуг и функции техобслуживания, после чего переходит в состояние «Idle» (в случае передачи управления) или в состояние «Preparing SSF Instructions» (в случае ассистирования).

Последний связанный с SCSM таймер – $T_{ActTest}$ – предназначен для защиты установленного диалога. Таймер запускается в начале диалога, а его срабатывание инициирует проверку связанных с данным диалогом ресурсов SSF и SCF посредством операции *ActivityTest*.

Все относящиеся к управлению соединением операции в интерфейсе SCF-SSF (кроме операций связанных с SCME) делятся на две категории: операции, связанные с обслуживанием вызова, и операции, не связанные с обслуживанием вызова.

Операции, входящие в первую категорию, образуют две группы: первая включает в себя операции *CollectInformation*, *Connect*, *ReleaseCall*, и *Continue*; а вторая – операции *InitiateCallAttempt*, *ConnectToResource*, *DisconnectForwardConnection* и *EstablishTemporaryConnection*.

SCF не может передавать в направлении к SSF две операции первой группы, используя последовательность сообщений TCAP или последовательность компонентов в одном сообщении. Возможна передача за один раз только одной операции. Две операции первой

группы должны быть разделены по крайней мере одним полученным SCSM сообщением, несущим информацию о возникшем в SSF событии вида EDP-R. Такой же принцип применяется для любой операции из первой группы, если она сопровождается операциями *ConnectToResource* или *EstablishTemporaryConnection*.

Операции, не связанные с обслуживанием вызова, составляют оставшуюся часть операций в интерфейсе SCF-SSF. Когда логике услуг нужно передать несколько операций для параллельной обработки, они передаются в виде последовательности компонентов.

Ниже описывается каждое состояние SCSM-FSM и дается перечень событий, вызывающих переходы из этого состояния в другое.

3.3.4.2 Состояния «Idle» и «Preparing SSF Instruction»

В состоянии 1 – «Idle» – существенны два события, вызывающих переход в состояние 2 – «Preparing SSF Instructions» (подготовка инструкций для SSF).

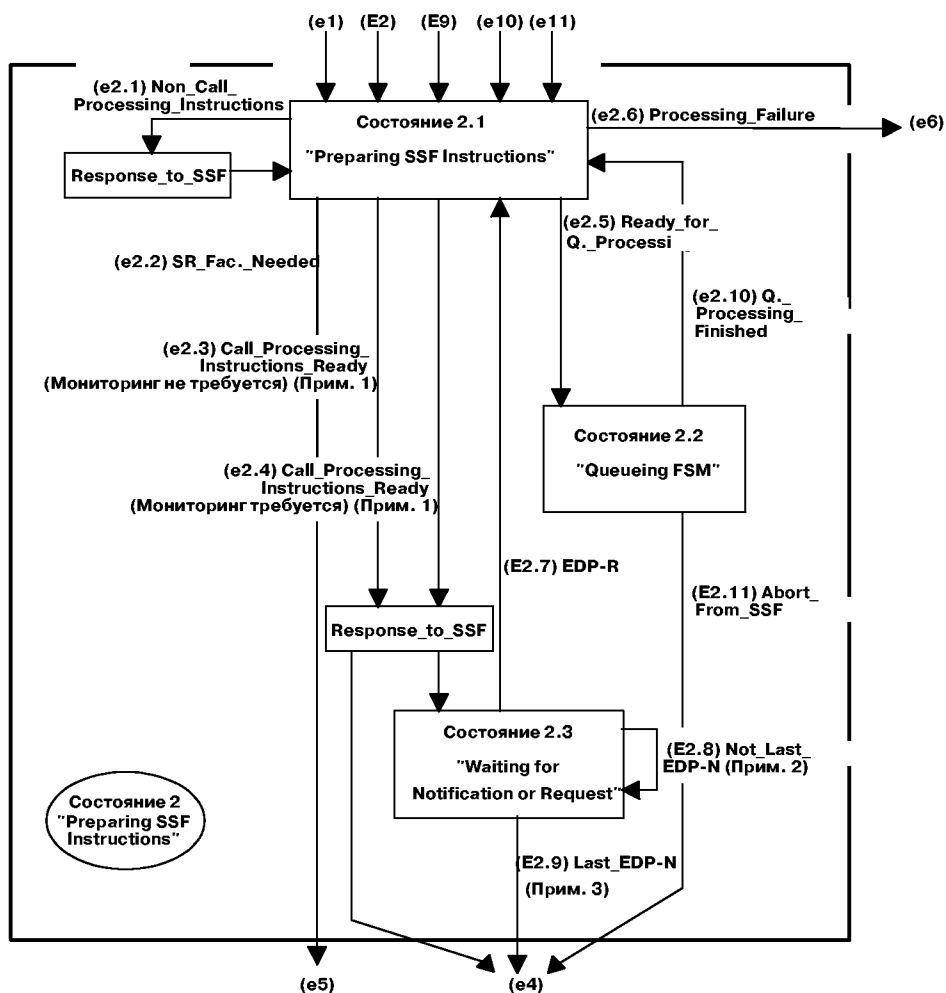
- (e1) *SL_Invocation* – внутреннее событие, обусловленное тем, что логике услуг необходимо инициировать вызов. SCSM передает к SSF операцию *Initiate CallAttempt*.
- (e2) *Query_from_SSF* – внешнее событие, обусловленное приемом операции *InitialDP* или *AssistRequestInstructions* (для случая передачи управления).

В состоянии 2 – «Preparing SSF Instructions» – существенны следующие события:

- (e4) *Processing_Completed* – внутреннее событие, состоящее в том, что подготовка инструкций для SSF завершена. Это событие вызывает передачу соответствующей операции к SSF и переход SCSM FSM в состояние 1 – «Idle».
- (e5) *SR_Facilities_Needed* – внутреннее событие, обусловленное тем, что логике услуг нужна дополнительная информация от пользователя, и, следовательно, нужно установить соединение между этим пользователем и SRF. Это событие вызывает переход в состояние 3 – «Routing to Resource».
- (e6) *Processing_Failure* – внутреннее событие (сбой), требующее обработки особой ситуации и перехода в состояние 1 – «Idle».

Обработка особых ситуаций рекомендациями ITU-T и стандартами ETSI не определена. Предполагается лишь, что при такой обработке необходимо освободить все используемые ресурсы и передать соответствующее сообщение к SSF.

Для описания процедур, относящихся к состоянию 2, его удобно разделить на три состояния (2.1, 2.2 и 2.3), как это показано на рисунке 3.3.5.



Примечание 1: Включая операции *CallInformationRequest*, *ApplyCharging*

Примечание 2: Включая операции *CallInformationReport*, *ApplyChargingReport* и *EventNotificationCharging*.

Примечание 3: Включая операции *CallInformationReport*, *ApplyChargingReport*.

Рис. 3.3.5 Диаграмма, детализирующая Состояние 2

Состояние 2.1 – «Preparing SSF Instructions». В этом состоянии принимается начальное решение о том, нужна ли информация от SDF, требуется ли SRF, поддерживается ли организация очереди, и т.д. Кроме того, в этом состоянии выполняются действия, связанные с событиями вида EDP-R.

При входе в состояние 2.1 SCSM запускает или перезапускает таймер $T_{SCF-SSF}$.

В состоянии 2.1 существенны следующие события:

- (e2.1) *Non-Call_Processing_Instructions* – внутреннее событие, обусловленное тем, что логике услуг нужно передать в SSF одну из следующих операций, не связанных с обработкой вызова: *ApplyCharging*; *CallInformationRequest*; *FurnishChargingInformation*; *Cancel* (применительно ко всем запросам); *RequestReportBCSMEvent*; *ResetTimer*; *RequestNotificationChargingEvent*; *SendChargingInformation*.

Это событие вызывает возврат в состояние 2.1 – «Preparing SSF Instructions».

- (e2.2) *SR_Facilities_Needed* – внутреннее событие, обусловленное тем, что логике услуг необходимо использование ресурсов SRF; соответствует событию (e5) SCSM.
- (e2.3) *Call_Processing_Instruction_Ready* (Monitoring not required) – внутреннее событие, обусловленное работой логики услуг, когда готова заключительная операция, связанная с обработкой вызова, отсутствуют назначенные точки EDP и от SSF приняты все ожидавшиеся операции *CallInformationReport* или *ApplyChargingReport*. Это событие приводит к передаче в SSF одной из следующих операций: *Connect*, *Continue*, *ReleaseCall*.

Кроме того, до передачи операций, перечисленных выше, к SSF может быть передана одна или несколько из следующих операций: *Cancel* (применительно ко всем запросам), *RequestReportBCSMEvent* (чтобы отменить назначение всех назначенных ранее EDP); *FurnishChargingInformation*; *SendChargingInformation*. Событие (e2.3) соответствует событию (e4) SCSM.

- (e2.4) *Call_Processing_Instruction_Ready* (Monitoring required) – внутреннее событие, обусловленное работой логики услуг, когда готова операция, связанная с обработкой вызова, и требуется мониторинг процесса (например, имеется назначенная точка EDP, или не приняты все ожидаемые операции *CallInformationReport* или *ApplyChargingReport*, или же нужно передать соответствующий запрос). Это событие приводит к передаче в SSF одной из следующих операций: *CollectInformation*; *Connect*; *Continue*; *ReleaseCall*.

Кроме того, до передачи операций, перечисленных выше, к SSF может быть передана одна или несколько из следующих операций: *ApplyCharging*; *CallInformationRequest*; *FurnishChargingInformation*; *RequestReportBCSMEvent*; *RequestNotificationChargingEvent*; *SendChargingInformation*.

Событие (e2.4) вызывает переход в состояние 2.3 «Waiting for Notification or Request».

- (e2.5) *Ready_for_Queueing_Processing* – внутреннее событие, обусловленное работой логики услуг в случае, когда вызов нужно поставить на ожидание. Это событие вызывает переход в состояние 2.2 – «Queueing FSM».
- (e2.6) *Processing_Failure* – внутреннее событие, соответствующее событию (e6) SCSM.

Состояние 2.2 – «Queueing FSM» (очередь) соответствует ситуации, которая возникает, когда при обработке запроса от SSF-CCF оказывается, что ресурс, к которому должен быть направлен вызов, недоступен (занят). SCF может поставить вызов на ожидание и возобновить его обслуживание после освобождения ресурса. В этом состоянии могут быть переданы следующие операции: *ApplyCharging*; *CallInformationRequest*; *FurnishChargingInformation*; *RequestReportBCSMEvent*; *RequestNotificationChargingEvent*; *ResetTimer*; *SendChargingInformation*.

В состоянии 2.2 существенными являются два события:

- (e2.10) *Queueing_Processing_Finished* – внутреннее событие, обусловленное тем, что SLP готова сформировать связанную с вызовом операцию для передачи в SSF. Это событие вызывает переход в состояние 2.1 – «Preparing SSF Instructions».
- (e2.11) *Abort_from_SSF* – внешнее событие, обусловленное приемом от SSF сообщения *Abort* (при отказе от ожидания). Оно вызывает переход, который соответствует событию (e4) SCSM.

Состояние 2.2 удобно разделить на два состояния (2.2.1 и 2.2.2), как это показано на рисунке 3.3.6.

В состоянии 2.2.1 – «Preparing SSF Instructions» SCSM подготавливает для SSF команды, необходимые, чтобы завершить обслуживание вызова. В этом состоянии существенными являются следующие события:

- (e2.2.1) *Instruction_Ready* – внутреннее событие, которое наступает только в случае, если требуемый ресурс доступен. SCSM получает адрес доступного ресурса. Это событие вызывает переход в состояние 2.1 – «Preparing SSF Instructions» [переход (e2.10)].
- (e2.2.2) *Non-Call_Processing_Instructions* – внутреннее событие, которое обусловлено работой логики услуг в случае, когда требуется передать в SSF одну из следующих операций, не связанных с обработкой вызова: *ApplyCharging*; *CallInformationRequest*; *FurnishChargingInformation*; *RequestReportBCSMEvent*; *RequestNotificationChargingEvent*; *ResetTimer*; *SendChargingInformation*.

Это событие вызывает возврат в состояние 2.2.1.

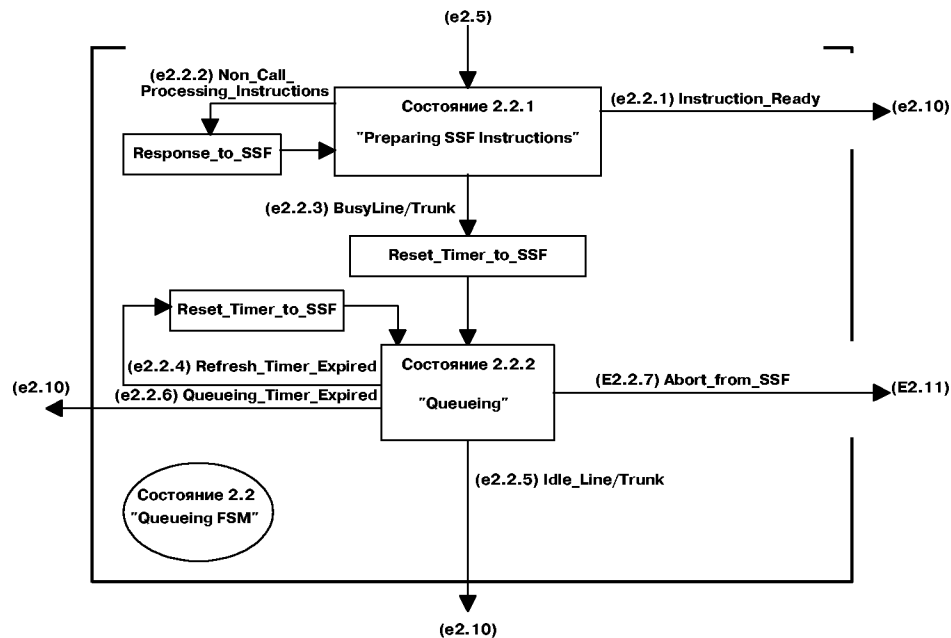


Рис. 3.3.6 Диаграмма, детализирующая состояние 2.2

- (e2.2.3) *Busy_Line/Trunk* – внутреннее событие, возникающее при отсутствии доступного канала. Событие вызывает передачу в SSF операции *ResetTimer* с подходящей к случаю выдержкой времени для таймера, регламентирующего допустимую длительность ожидания, и последующий переход в состояние 2.2.2 – «Queueing».

В состоянии 2.2.2 – «Queueing» SCSM ожидает индикацию доступности ресурса для маршрутизации вызова к свободному каналу. Кроме того, в этом состоянии SCSM поддерживается передача речевых оповещений. По своей внутренней структуре состояние 2.2.2 не отличается от состояний 3 и 4 SCSM, которые будут рассмотрены ниже. Отличие состоит в том, что если речевое оповещение закончится прежде, чем будет возобновлено обслуживание вызова и SSF-FSM перейдет в состояние «Waiting for Instructions», то должна быть передана операция *ResetTimer* с соответствующим значением выдержки времени для $T_{SCF-SSF}$. При переходе SCSM в это состояние запускается таймер ожидания и перезапускается таймер $T_{SCF-SSF}$. Назначение таймеров следующее:

- таймер ожидания ограничивает время пребывания вызова в очереди; определяемая им выдержка времени может быть изменена поставщиком услуги;
- $T_{SCF-SSF}$ определяет, когда следует передать операцию *ResetTimer* в SSF-CCF, чтобы последний не отбросил вызов, т.е. определяемая этим таймером выдержка времени должна быть в соответствии с выдержкой, определяемой таймером T_{SSF} .

В это время к SSF может быть передана операция *FurnishChargingInformation*, чтобы сообщить момент начала ожидания для внесения в запись информации о вызове. В этом состоянии существенными являются следующие события:

- (e2.2.4) *Refresh_Timer_Expired* – внутреннее событие, которое приводит к передаче в SSF-CCF операции *ResetTimer* и к возврату в то же самое состояние;
- (e2.2.5) *Idle_Line/Trunk* – внутреннее событие, соответствует событию (e2.10) в состоянии 2;
- (e2.2.6) *Queueing_Timer_Expired* – внутреннее событие, результатом которого является отказ в выделении ресурса и переход в состояние 2.1 – «Preparing SSF Instructions» [переход (e2.10)];
- (E2.2.7) *Abort_from_SSF* – внешнее событие, состоящее в приеме от SSF сообщения *Abort* (при отказе от ожидания); вызывает переход, который соответствует событию (E2.11) в состоянии 2.

В состоянии 2.3 – «Waiting for Notification or Request» (ожидание уведомления или запроса) SCSM ждет уведомление или запрос от SSF. При переходе в это состояние останавливается таймер $T_{SCF-SSF}$. Существенными здесь являются события вида:

- (E2.7) *EDP-R* – внешнее событие, заключающееся в приеме операции *EventReportBCSM* (для *EDP_R*); вызывает переход в состояние 2.1 – «Preparing SSF Instructions».
- (E2.8) *Not_Last_EDP-N* – внешнее событие, состоящее в приеме одной из следующих операций: *ApplyChargingReport*; *CallInformationReport*; *EventReportBCSM* (для *EDP_N*); *EventNotificationCharging*.

Перечисленные операции несут информацию о том, что в SSF еще имеется назначенная точка EDP или ждущая обработки операция вида *CallInformationReport* или *ApplyChargingReport*. Событие приводит к возврату в состояние 2.3 – «Waiting for Notification or Request».

- (E2.9) *Last_EDP-N* – внешнее событие, заключающееся в приеме одной из следующих операций: *ApplyChargingReport*; *CallInformationReport*; *EventReportBCSM* (для *EDP_N*).

В этом случае отсутствуют назначенные точки EDP и не ожидаются операции вида *CallInformationReport* или *ApplyChargingReport*. Событие (E2.9) соответствует событию (e4) SCSM.

3.3.4.3 Состояние 3: «Routing to Resource» (подключение к ресурсу)

Под ресурсом в данном случае понимаются средства, входящие в состав SRF. Переходы из состояния 3 вызывают следующие события:

- (e7) *Resource_Attached* – SRF подключен; событие вызывает переход в состояние 4 «User Interaction»;
- (e8) *Handoff_Needed* – требуется передача управления. Событие состоит в том, что с началом процедуры передачи управления обслуживанием вызова другому SSF, SCSCM заканчивает взаимодействие с SSF, инициировавшим эту процедуру. Происходит переход в состояние 1 – «Idle». Получив операцию *AssistRequestInstructions* от того SSF, которому передается управление, SCME-Control создает новую SCSCM. SCF должен обладать информацией, достаточной для того, чтобы связать операцию *AssistRequestInstructions* с активной программой логики услуг;
- (E9) *Failure_from_SSF* – событие состоит в том, что SSF не смог установить соединение с требующимися ресурсами. Это событие вызывает переход в состояние 2 – «Preparing SSF Instructions»;
- (e10) *Timer_Expired* – это событие возникает при срабатывании таймера $T_{\text{ASSIST/HANDOFF}}$ и вызывает переход в состояние 2 – «Preparing SSF Instructions».

Состояние 3 удобно разделить на два состояния (3.1 и 3.2), как это показано на рис.3.3.7.

В состоянии 3.1 – «Determine Mode» (определить режим) SCSCM определяет режим взаимодействия с пользователем при связи с SRF. Существенными в этом состоянии являются следующие события:

- (e3.1) *Instruction_Ready* – внутреннее событие, которое может возникнуть только при взаимодействии с SSF-инициатором связи, когда он транслирует операции, которые направляются от SCF к SRF. SCSCM передает операцию *ConnectToResource*, сопровождаемую операцией *PlayAnnouncement* или *PromptAndCollectUserInformation*, к SSF-инициатору связи и переходит в состояние 4 «User Interaction». Этот переход соответствует событию (e7);
- (e3.2) *Assist_Needed* – внутреннее событие, которое имеет место, когда требуется взаимодействие с ассистирующим SSF или непосредственная связь SCF-SRF. В этом случае SCSCM передает к SSF-инициатору операцию *EstablishTemporaryConnection* с адресом ассистирующего SSF или с адресом SRF и переходит в состояние 3.2 – «WaitingForAssistRequestInstructions»;
- (e3.3) *Handoff_Needed* – внутреннее событие, которое имеет место только в случае передачи управления обслуживанием вызова. SCSCM передает к инициирующему SSF операцию *Connect* с адресом SSF, которому будет передано управление, запускает таймер $T_{\text{ASSIST/HANDOFF}}$ и переходит в состояние 1 – «Idle». Этот переход соответствует событию (e8). SCF должен обладать информацией, достаточной для того, чтобы связать с активной програм-

мой логики услуг операцию *AssistRequestInstructions*, которая будет передана впоследствии от ассистирующего SSF или от SRF,

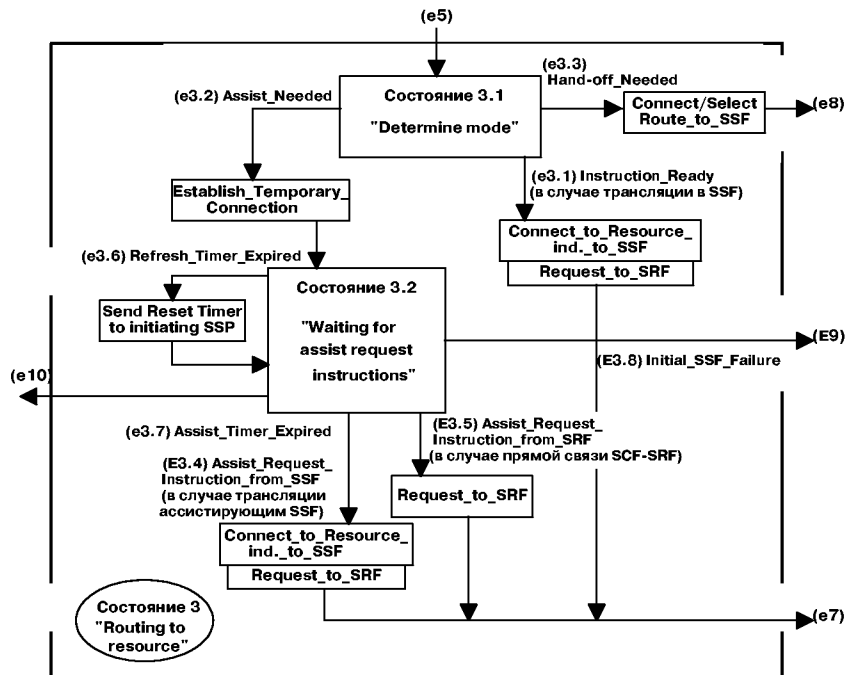


Рис. 3.3.7 Диаграмма, детализирующая FSM состояния 3

В состоянии 3.2 – «Waiting for Assist Request Instructions» SCSM ожидает от ассистирующего SSF операцию *AssistRequestInstructions* (в случае трансляции к SRF) или от SRF (в варианте прямого взаимодействия SCF-SRF). При переходе в это состояние SCSM запускает таймер $T_{ASSIST/HANDOFF}$ и перезапускает таймер $T_{SCF-SSF}$ (если он используется). Существенными в этом состоянии являются следующие события:

- (E3.4) *Assist_Request_Instructions_from_SSF* – внешнее событие, обусловленное получением от ассистирующего SSF операции *AssistRequestInstructions* (в случае, когда он ее транслирует). SCSM передает к ассистирующему SSF операцию *ConnectToResource*, сопровождаемую операцией *PlayAnnouncement* или *PromptAndCollectUserInformation*, и переходит в состояние 4 – «User Interaction». Этот переход соответствует событию (e7);
- (E3.5) *Assist_Request_Instructions_from_SRF* – внешнее событие, обусловленное получением от SRF операции *AssistRequestInstructions* (в случае прямого взаимодействия SCF-SRF). SCSM передает к SRF операцию *PlayAnnouncement* или *PromptAndCollect-*

UserInformation и переходит в состояние 4 «User Interaction». Этот переход соответствует событию (e7);

- (e3.6) *Refresh_Timer_Expired* – внутреннее событие, которое имеет место после срабатывания таймера $T_{SCF-SSF}$. SCSM передает к SSF-инициатору операцию *ResetTimer* и возвращается в прежнее состояние;
- (e3.7) *Assist_Timer_Expired* – внутреннее событие, которое имеет место после срабатывания таймера $T_{ASSIST/HANDOFF}$. SCSM информирует об этом SCME и SLP и, после передачи к SSF операции *DisconnectForwardConnection*, переходит в состояние 2 – «Preparing SSF Instructions». Это событие соответствует событию (e10);
- (E3.8) *Initial_SSF_Failure* – внешнее событие, обусловленное получением отказа со стороны SSF. Это событие вызывает переход, который соответствует событию (E9) SCSM.

3.3.4.4 Состояние 4: «User Interaction» (взаимодействие с пользователем)

В этом состоянии SCF дает указание SRF обеспечить взаимодействие с пользователем (принять от него дополнительную информацию и/или передать ему речевое сообщение). По окончании взаимодействия SCF может передать к SSF команду нарушить связь между SSF и SRF. В другом варианте SCF может передать к SRF операцию с индикатором, разрешающим разъединение по инициативе SRF. При входе в это состояние SCSM перезапускает таймер $T_{SCF-SSF}$ (если он используется). Выход из состояния 4 вызывает одно событие:

- (e11) *Continue_SCF_Processing* – событие состоит в том, что SCF получил от SRF всю информацию, которая нужна, чтобы сформировать для SSF инструкции по завершению обслуживания вызова. Это событие вызывает переход в состояние 2 – «Preparing SSF Instructions».

FSM состояния 4 представлена на рисунке 3.3.8.

В состоянии 4.1 – «Waiting for Response from the SRF» (ожидание инструкций от SRF) SCF ожидает от SRF ответа на предварительно переданную операцию и, получив ответ, производит его анализ. В этом состоянии существенными являются следующие события:

- (e4.1) *More_Information_Needed* – это событие приводит к передаче в SRF еще одной операции и вызывает переход обратно в состояние 4.1;
- (e.4.2) *Response_from_SRF* – внешнее событие, обусловленное приемом операции *SpecializedResourceReport* или компонента *Return result* с результатом выполнения действий, которые были предписаны переданной ранее операцией *PromptAndCollect-UserInformation*. По окончании приема SCSM возвращается в прежнее состояние;

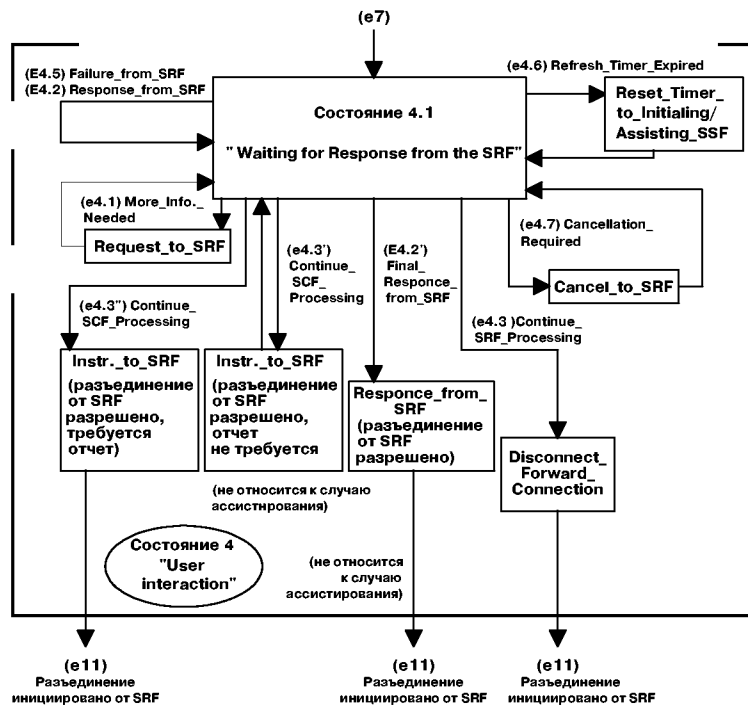


Рис. 3.3.8 Диаграмма, детализирующая FSM состояния 4

- (e4.2') Final_Response_from_SRF – внешнее событие, обусловленное приемом операции *SpecializedResourceReport* в ответ на переданную к SRF операцию *PlayAnnouncement* или приемом компонента Return result с результатом выполнения действий, которые были предписаны переданной ранее операцией *PromptAndCollectUserInformation* с разрешением разъединения по инициативе SRF. В случаях с трансляцией SSF-инициатором и с прямой связью SCF-SRF это событие вызывает переход SCSM в состояние 2 «Preparing SSF Instructions». Это событие соответствует событию (e11);
- (e4.3) Continue_SCF_Processing – внутреннее событие, которое происходит, когда SCSM заканчивает взаимодействие с пользователем и дает указание SCF нарушить соединение между SSF-инициатором и SRF. SCSM передает к SSF-инициатору операцию *DisconnectForwardConnection* и переходит в состояние 2 – «Preparing SSF Instructions». Это событие соответствует событию (e11);
- (e4.3') Continue_SCF_Processing – внутреннее событие, которое происходит, когда SCSM заканчивает взаимодействие с пользователем и дает указание SRF нарушить соединение между SSF-инициатором и SRF. SCSM передает к SRF операцию *PlayAn-*

nouncement (с запросом операции *SpecializedResourceReport*) или операцию *PromptAndCollectUserInfo* с разрешением разъединения по инициативе SRF. В случае с ассистирующим SSF разъединение со стороны SRF невозможно. SCSM возвращается в прежнее состояние.

- (e4.3") *Continue_SCF_Processing* – внутреннее событие, которое имеет место, когда SCSM заканчивает взаимодействие с пользователем и дает указание SRF нарушить соединение между SSF-инициатором и SRF. SCSM передает операцию *PlayAnnouncement* (без запроса ответной операции *SpecializedResourceReport*) с разрешением разъединения со стороны SRF. В случае с ассистирующим SSF разъединение со стороны SRF невозможно. SCSM переходит в состояние 2 – «Preparing SSF Instructions». Это событие соответствует событию (e11);
- (E4.5) *Failure_from_SRF* – внешнее событие, обусловленное приемом компонента Return Error в ответ на операцию *PlayAnnouncement* или на операцию *PromptAndCollectUserInfo*. SCSM не изменяет своего состояния.
- (e4.6) *Refresh_Timer_Expired* – внутреннее событие, которое имеет место при срабатывании таймера $T_{SCF-SSF}$. SCSM передает операцию *ResetTimer* к SSF-инициатору или к ассистирующему SSF и возвращается в прежнее состояние;
- (e4.7) *Cancellation_Required* – внутреннее событие, которое имеет место в случае, когда SCSM отменяет ранее переданную операцию *PlayAnnouncement* или *PromptAndCollectUserInfo*. SCSM передает операцию *Cancel* к ассистирующему SSF (в случае трансляции) или к SRF (при прямой связи SCF-SRF) и возвращается в прежнее состояние.

Соединение между SSF и SRF нарушается при выходе SCSM из состояния 4.1.

В таблицах 3.3.1 и 3.3.2 приведен перечень состояний и событий, вызывающих переходы, для SCF-FSM. В таблицах предпринята попытка подобрать эквивалентные названия состояний и событий на русском языке.

Таблица 3.3.1

	Состояние SCME		События, вызывающие выход из состояния
M3	Service Filtering Idle (Просеивание обращений к услуге; исходное)	(em5)	Filtering_Request_to_SSF (Запрос_просеивания_к_SSF)
M4	Waiting for SSF Service Filtering Response (Ожидание ответа SSF на запрос просеивания)	(em7)	End_of_Service_Filtering (Окончание_просеивания_обращений_к_услуге)
		(em8)	Filtering_Response_from_SSF (Ответ_SSF_на_запрос_просеивания)

Таблица 3.3.2

Состояния SCSM			События, вызывающие выход из состояния		
1	Idle (Исходное)		(e1)	SL_Invocations (Обращение_от_логики_услуг)	
			(E2)	Query_from_SSF (Запрос_от_SSF)	
2	Preparing SSF Instructions (Подготовка инструкций для SSF)		(e4)	Processing_Completed (Обработка_завершена)	
			(e5)	SR_Facilities_Needed (Требуется_ресурсы_SRF)	
			(e6)	Processing_Failure (Сбой_при_обработке)	
	2.1	Preparing SSF Instructions (Подготовка инструкций для SSF)	(e2.1)	Non_Call_Processing_Instructions (Инструкции, не связанные с обработкой вызова)	
			(e2.2)	SR_Facilities_Needed (Требуется_ресурсы_SRF)	
			(e2.3)	Call_Processing_Instructions_Ready (MRQ) Инструкции для обработки вызова готовы (требуется мониторинг)	
			(e2.4)	Call_Processing_Instructions_Ready (MNRQ) Инструкции для обработки вызова готовы (мониторинг не требуется)	
			(e2.5)	Ready_for_Queueing_Processing (готовность к работе с очередью)	
			(e2.6)	Processing_Failure (сбой_при_обработке)	
	2.2	Queueing FSM (Ожидание)	(e2.10)	Queueing_Processing_Finished (работа с очередью завершена)	
			(E2.11)	Abort_from_SSF (отказ от ожидания со стороны SSF)	
		2.2.1	Preparing SSF Instructions (Подготовка инструкций для SSF)	(e2.2.1)	Instructions_Ready (инструкции готовы)
				(e2.2.2)	Non_Call_Processing_Instructions (инструкции, не связанные с обработкой вызова)
				(e2.2.3)	Busy_Line/Trunk (занятость линий/каналов)
		2.2.2	Queueing (Ожидание)	(e2.2.4)	Refresh_Timer_Expired (перезапустить сработавший таймер)
				(e2.2.5)	Idle_Line/Trunk (линии/каналы свободны)
				(e2.2.6)	Queueing_Timer_Expired (сработал таймер ожидания)
				(e2.2.7)	Abort_from_SSF (отказ от ожидания со стороны SSF)
		2.3	Waiting for Notification or Request (Ожидание уведомления или запроса)		(E2.7)

Окончание таблицы 3.3.2

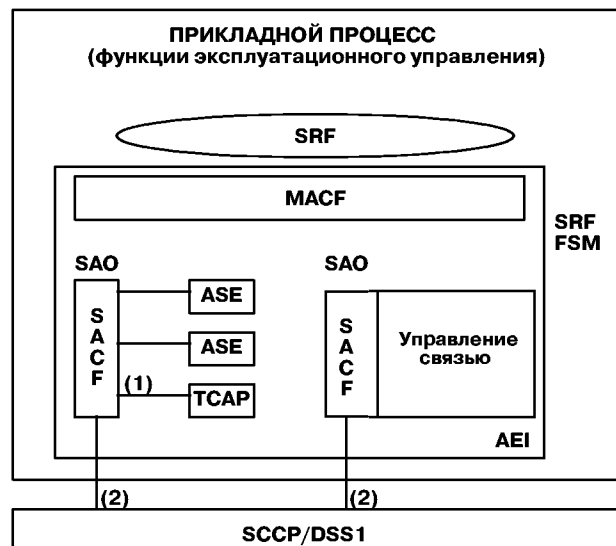
Состояния SCSM				События, вызывающие выход из состояния
			(E2.8)	Not_Last_EDP-N (не_последняя_точка_EDP_с_уведомлением)
			(E2.9)	Last_EDP-N (последняя_точка_EDP_с_уведомлением)
3	Routing to Resource (Подключение к ресурсу)		(e7)	Resource_Attached (ресурс_подключен)
			(e8)	Hand-off_Needed (требуется_передача_управления_обслуживанием_вызова)
			(E9)	Failure_from_SSF (отказ_со_стороны_SSP)
			(e10)	Timer_Expired (срабатывание_таймера)
	3.1	Determine Mode (Определить режим)	(e3.1)	Instructions_Ready (инструкции_готовы)
			(e3.2)	Assist_Needed (требуется_ассистирование)
			(e3.3)	Hand-off_Needed (требуется_передача_управления_обслуживанием_вызова)
	3.2	Waiting for Assist Request Instructions (Ожидание запроса инструкций при ассистировании)	(E3.4)	Assist_Request_Instructions_from_SSF (запрос_инструкций_от_ассистирующего_SSF)
			(E3.5)	Assist_Request_Instructions_from_SRF (запрос_инструкций_от_SRF_при_ассистировании)
			(e3.6)	Refresh_Timer_Expired (перезапустить_сработавший_таймер)
(E3.8)			Initial_SSP_Failure (отказ_со_стороны_SSF-инициатора)	
4	User Interaction (Взаимодействие с пользователем)		(e11)	Continue_SCF_Processing (продолжить_обработку_в_SCF)
	4.1	Waiting for Responce from the SRF (Ожидание ответа от SRF)	(e4.1)	More_Informations_Needed (требуется_дополнительная_информация)
			(E4.2)	Response_from_SRF (ответ_от_SRF)
			(E4.2')	Final_Response_from_SRF (заключительный_ответ_от_SRF)
			(e4.3) (e4.3') (e4.3'')	Continue_SCF_Processing (продолжить_обработку_в_SCF)
			(E4.5)	Failure_from_SRF (отказ_со_стороны_SRF)
			(e4.6)	Refresh_Timer_Expired (перезапустить_сработавший_таймер)
			(e4.7)	Cancellation_Required (требуется_сброс)

Глава 3.4

Процедуры прикладного объекта SRF

3.4.1 Модель и интерфейсы прикладного объекта SRF-AE

Приводимые ниже процедуры прикладного объекта SRF-AE определены для интерфейса между функциональными объектами SRF и SCF. Процедуры основаны на использовании системы сигнализации OKC №7, однако, как и для уже рассмотренных прикладных объектов SSF-AE и SCF-AE, могут использоваться и другие системы сигнализации, поддерживающие структуры прикладного уровня, например DSS1.



(1) Прimitives TC - пользователей

(2) Прimitives сетевого уровня

Рис. 3.4.1 Функциональная модель прикладного объекта SRF-AE

На рисунке 3.4.1. показана модель прикладного объекта, соответствующая функциональному объекту SRF, и интерфейс с TCAP (для связи с SCF). Здесь же показан интерфейс с функциями эксплуатационного управления, который является внутренним и в CS-1 не стандартизируется. Предметом рассмотрения этой главы является затемненная область рисунка 3.4.1.

3.4.2 Связь между SRF-FSM и функциями эксплуатационного управления/управления связью

Всякий раз, когда SSF инициирует вызов, создается машина конечных состояний SRF-FSM. Созданная машина SRF-FSM поддерживает взаимодействие с SCF-FSM и SSF-FSM. Функции административного управления, которые относятся к исполнению операций, принятых от SCF, выполняет объект административного управления SRME (SRF Management Entity). SRME имеет интерфейсы с разными машинами SRSM, а также интерфейс с администратором доступа FEAM (Functional Entity Access Manager). Внутренняя структура SRF-FSM показана на рисунке 3.4.2.

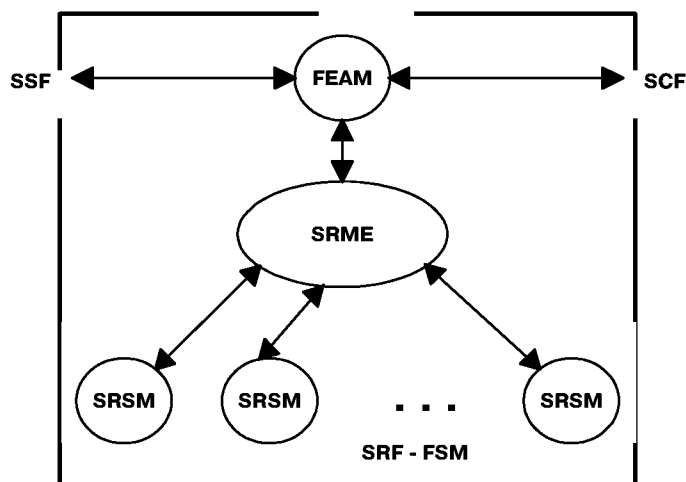


Рис. 3.4.2 Структура SRF-FSM

Тот факт, что запросы от SCF к SRF поступают асинхронно, требует, чтобы существовал отдельный объект SRME (SRF Management Entity), ответственный за выполнение задач создания, привлечения и техобслуживания объектов SRSM. Помимо этого, SRME поддерживает диалоги с SCF и SSF от имени всех машин состояний SRSM. В частности, SRME:

- интерпретирует сообщения, которые поступают от других функциональных объектов, и преобразует их в соответствующие события SRSM;
- преобразует выходные данные SRSM в сообщения, подлежащие передаче к другим функциональным объектам;
- поддерживает тестирование связи между SCF и SRF.

Администратор доступа FEAM освобождает SRME от функций нижних уровней в интерфейсах. Функции FEAM включают в себя:

- организацию и поддержку интерфейсов с SSF и SCF;
- контроль прохождения (и, если это требуется, – установку в очередь) сообщений, поступающих от SCF и SSF;
- форматирование, установку в очередь (если требуется) и передачу к SCF и SRF сообщений, принятых от SRME.

3.4.3 Диаграмма состояний SRSМ

Диаграмма состояний SRSМ представлена на рисунке 3.4.3.

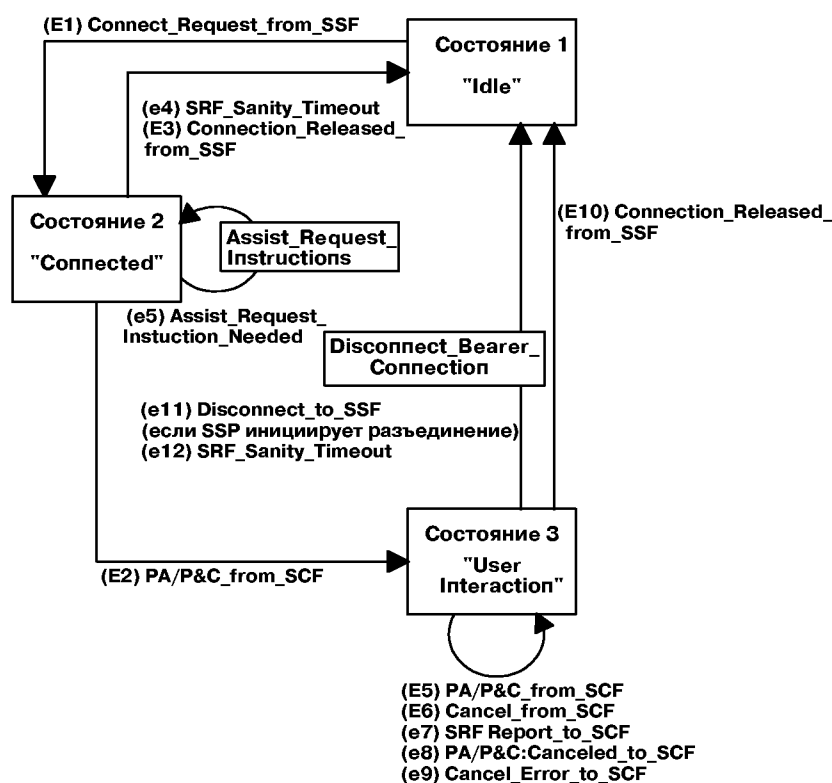


Рис. 3.4.3 Диаграмма состояний SRSМ

Рассмотрим некоторые общие правила, относящиеся к описываемым ниже состояниям SRSМ. Один или более компонентов, принятых в одном или нескольких сообщениях TCAP, могут включать в себя одну операцию или последовательность нескольких операций и должны обрабатываться следующим образом.

- SRSМ обрабатывает операции в том порядке, в котором они были приняты.

- SRSM последовательно анализирует принятые операции. Если в состоянии «User Interaction» в последовательности операций встретилась операция *Cancel* (для *PlayAnnouncement* или *PromptAndCollectUserInformation*), то SRSM выполняет ее немедленно. Во всех остальных случаях SRSM устанавливает операции в очередь и ждет определенного события (таким событием может быть завершение выполняемой операции или прием внешнего события).
- Если при обработке одной из операций обнаружена ошибка, то SRF-FSM обрабатывает эту ошибку (см. ниже) и сбрасывает все оставшиеся операции последовательности.
- Если операция не распознана или нарушает определенные для SRSM правила, то SRF-FSM обрабатывает ошибку, используя TC U-REJECT или ошибку типа *UnexpectedComponentSequence*.

В любом состоянии информация об ошибке, обнаруженной в принятой операции, передается функциям техобслуживания. Как правило, SRSM остается в том же состоянии, хотя в определенных случаях возможны другие алгоритмы обработки ошибки. В зависимости от класса операции SRF может информировать об ошибке SCF, используя для этого соответствующий компонент TCAP.

При наличии между SCF и SRF прямой связи, когда диалог с SCF завершен, SRSM возвращается из любого состояния в состояние «Idle» лишь после освобождения всех использовавшихся для данного диалога ресурсов. Соединение между SRF и SSF должно сохраняться до тех пор, пока в SRF имеются активные или буферизированные операции *PlayAnnouncement*. Ресурсы, выделенные для обслуживания вызова, освобождаются либо после завершения всех речевых сообщений, либо после разъединения со стороны SSF вследствие отбоя абонента.

В любом состоянии (кроме «Idle»), если SSF нарушает соединение с SRF до того, как SRF закончит взаимодействие с пользователем, SRSM освобождает ресурсы SRF, выделенные для обслуживания вызова, и, убедившись, что они освобождены, переходит в состояние «Idle».

В SRSM имеется таймер T_{SRF} , предотвращающий чрезмерно продолжительную приостановку обслуживания вызова. Этот таймер запускается, когда SRF передает к SSF сообщение, подтверждающее исполнение запроса соединения (Setup Response) (при связи между SCF и SRF через SSF), или операцию *AssistRequestInstructions* (при прямой связи SCF-SRF). Таймер останавливается после приема ответных инструкций от SCF. Если нет ожидающей операции *UserInteraction*, SRF может перезапускать таймер T_{SRF} при передаче операции *SpecializedResourceReport* или при передаче результата действий, предписанных операцией *PromptAndCollectUserInformation*. После

срабатывания таймера T_{SRF} SRSM переходит в состояние «Idle», убедившись, что все ресурсы SRF, выделенные для обслуживания вызова, освобождены.

3.4.3.1 Состояние 1 – «Idle»

В состоянии «Idle» SRSM-FSM находится либо до начала взаимодействия с пользователем, либо после его окончания. В это состояние SRSM-FSM переходит в результате событий E3, e4, E10, e11 и e12. Выход из состояния «Idle» происходит в результате события E1.

- (E1) *Connect_Request_from_SSF*: это событие является результатом приема от SSF сообщения запроса сигнального соединения. SRSM переходит в состояние «Connected»;
- (E3) *Connection_Released_from_SSF*: данное событие имеет место, когда SRSM в состоянии «Connected» принимает от SSF сообщение освобождения. SRSM переходит в состояние «Idle»;
- (e4) *SRF_Sanity_Timeout* – данное событие имеет место, когда SRSM находилась в состоянии «Connected» в течение времени, определенного таймером T_{SRF} , не получая от SCF операций *PlayAnnouncement/PromptAndCollectUserInformation*. SRF инициирует разъединение тракта связи с SSF, используя соответствующую систему сигнализации по этому тракту. SRSM переходит в состояние «Idle»;
- (E10) *Connection_Released_from_SSF*: событие имеет место, когда в состоянии «User Interaction» SRSM принимает сообщение RELEASE со стороны SSF. SRSM переходит в состояние «Idle»;
- (e11) *Disconnect_to_SSF*: событие имеет место, когда SCF передал в SRF последнюю операцию *PlayAnnouncement/PromptAndCollectUser Information* (E2) или (E5) с параметром, разрешающим SRF инициировать разъединение. Передав к SCF последнюю операцию *SpecializedResource Report* (e7), SRSM инициирует разъединение тракта связи с SSF, используя соответствующую систему сигнализации по этому тракту. SRSM переходит в состояние «Idle»;
- (e12) *SRF_Sanity_Timeout*: событие имеет место, когда SRSM находилась в состоянии «User Interaction» в течение времени, определенного таймером T_{SRF} , не получая от SCF операций *PlayAnnouncement/PromptAndCollect UserInformation*. SRF инициирует разъединение тракта связи с SSF, используя соответствующую систему сигнализации по этому тракту. SRSM переходит в состояние «Idle».

3.4.3.2 Состояние 2 – «Connected»

В состоянии «Connected» SRSM находится, когда между пользователем и SRF установлен тракт связи, но начальная операция *PlayAnnouncement/ PromptAndCollect UserInformation* еще не получена (например, когда используются процедуры *EstablishTemporaryConnection*).

- (E1) *Connect_Request_from_SSF*: событие является результатом приема от SSF в состоянии «Idle» сообщения запроса сигнального соединения. SRSM переходит в состояние «Connected».
- (E2) *PlayAnnouncement/PromptAndCollectUserInformation_from_SCF*: это событие имеет место в случае, когда от SCF получена первая операция *PlayAnnouncement* или *PromptAndCollectUserInformation*. SRSM переходит в состояние «User Interaction».
- (E3) *Connection_Released_from_SSF*: это событие имеет место, когда SRF принимает от SSF сообщение освобождения. SRSM переходит в состояние «Idle».
- (e4) *SRF_Sanity_Timeout*: данное событие имеет место, когда SRSM находилась в состоянии «Connected» в течение времени, определенного таймером T_{SRF} , не получая от SCF операций *PlayAnnouncement/PromptAndCollectUserInformation*. SRF инициирует разъединение тракта связи с SSF, используя соответствующую систему сигнализации по этому тракту. SRSM переходит в состояние «Idle».
- (e5) *Assist_Request_Instructions_Needed*: это событие имеет место, когда от SRSM к SCF передается операция *AssistRequestInstructions* при отсутствии события (E2), что бывает обусловлено наличием операции *PlayAnnouncement/PromptAndCollectUserInformation*, связанной с запросом соединения от SSF (E1) в случае прямой связи SCF-SRF. В результате события (e5) состояние SRSM не изменяется.

3.4.3.3 Состояние 3 – «User interaction»

В состоянии «User interaction» между пользователем и SRF существует тракт связи, предварительно созданный в состоянии «Connected». Переход в состояние 3 происходит в результате события E2. События E10, e11 и e12 приводят к выходу из этого состояния, а события E5, E6, e7, e8 и e9 не вызывают изменения состояния. Событие E5 состоит в приеме не первой и в наличии хотя бы одной буферизированной операции *PlayAnnouncement/PromptAndCollectUserInformation*.

- (E2) и (E5) *PlayAnnouncement/PromptAndCollectUserInformation_from_SCF*: событие имеет место, когда от SCF принята первая (E2) или не первая (E5) операция *PlayAnnouncement* или *PromptAnd-*

CollectUserInformation. SRSM переходит в состояние «User interaction» при событии (E2) и остается в этом состоянии при последующих событиях (E5).

- (E6) *Cancel_from_SCF* (для операций *PlayAnnouncement/PromptAndCollectUserInformation*): это событие имеет место в случае, когда от SCF была принята операция *PlayAnnouncement* или *PromptAndCollectUserInformation*. Если соответствующие действия уже выполняются, то они прекращаются, в противном случае операция удаляется из буфера. SRSM остается в состоянии «User interaction».
- (e7) *SRF_Report_to_SCF*: событие имеет место, когда к SCF передается операция *SpecializedResourceReport* или возвращается результат выполнения операции *PromptAndCollectUserInformation*. SRSM остается в состоянии «User interaction».
- (e8) *PlayAnnouncement/PromptAndCollectUserInformation_Canceled_to_SCF*: это событие имеет место, когда в SCF передается ошибка выполнения операции *PlayAnnouncement/PromptAndCollectUserInformation*, вызванная приемом операции *Cancel* (для операции *PlayAnnouncement* или *PromptAndCollectUserInformation*). Событие соответствует успешной отмене выполняемой (но еще не выполненной) или буферизированной операции *PlayAnnouncement/PromptAndCollectUserInformation*. SRSM остается в состоянии «User interaction».
- (e9) *Cancel_Error_to_SCF*: это событие имеет место, когда в SCF передается ошибка в выполнении операции *Cancel* (для *PlayAnnouncement* или *PromptAndCollectUserInformation*). Событие соответствует неуспешной отмене операции *PlayAnnouncement/PromptAndCollectUserInformation*. SRSM остается в том же состоянии.
- (E10) *Connection_Released_from_SSF*: это событие имеет место, когда SRSM принимает от SSF сообщение разъединения. SRSM переходит в состояние «Idle».
- (e11) *Disconnect_to_SSF*: это событие имеет место, когда SCF разрешает SRF инициировать разъединение после приема от SCF последней операции *PlayAnnouncement/PromptAndCollectUserInformation* (E2) или (E5). В этом случае после передачи к SCF последней операции *SpecializedResourceReport* или результата выполнения операции *PromptAndCollectUserInformation* SRSM инициирует разъединение тракта связи с SSF, используя соответствующую систему сигнализации по этому тракту. SRSM переходит в состояние «Idle».

- (e12) SRF_Sanity_Timeout: событие имеет место, когда SRSM находилась в состоянии «User Interaction» в течение времени, определенного таймером T_{SRF} , не получая от SCF операций *PlayAnnouncement/PromptAndCollectUserInformation*. SRF инициирует разъединение тракта связи с SSF, используя соответствующую систему сигнализации по этому тракту. SRSM переходит в состояние «Idle».

В дополнение к отмеченным переходам, сбой при создании тракта связи между пользователем и SRF приводит к переходу SRSM в состояние «Idle» из любого состояния. С целью упрощения диаграммы такого рода переходы на рисунке 3.4.3 не показаны.

3.4.4 Примеры процедур управления SRF

Различие в процедурах управления SRF связано с разными вариантами размещения SRF в соответствии с физическими сценариями архитектуры протокола. Показанные на диаграммах сигнальные сообщения управления трактом связи имеют только функциональное значение, поскольку стандартизации не подлежат.

3.4.4.1 Процедуры подключения SRF

Для разных физических сценариев требуются разные процедуры. Ниже описываются рассматриваемые нами случаи, которые иллюстрирует рисунок 3.4.4:

- a) IP непосредственно подключена к SSP, который взаимодействует с SCP, или интегрирована в этот SSP, но операции от SCP транслируются к IP через SSP, выполняющий в этом случае все необходимые преобразования протоколов;
- b) IP непосредственно подключена к SSP, который взаимодействует с SCP, но операции от SCP передаются к IP без переприема в SSP;
- c) IP непосредственно подключена ко второму (ассистирующему) SSP или интегрирована в этот SSP. Операции от SCP транслируются к IP через второй SSP (метод ассистирования – «Assist»), а по завершении взаимодействия с пользователем управление возвращается первому SSP;
- d) IP непосредственно подключена к коммутационной системе без функций SSP, при этом операции от SCP передаются к IP без переприема в SSP («Assist»-метод, но с некоторым изменением физической конфигурации). По завершении взаимодействия с пользователем управление возвращается к SSP;
- e) IP подключена ко второму SSP, и по завершении взаимодействия с пользователем управление обслуживанием вызова инициирую-

щему SSP не возвращается, а сохраняется за этим вторым SSP (метод передачи управления – «Handoff»).

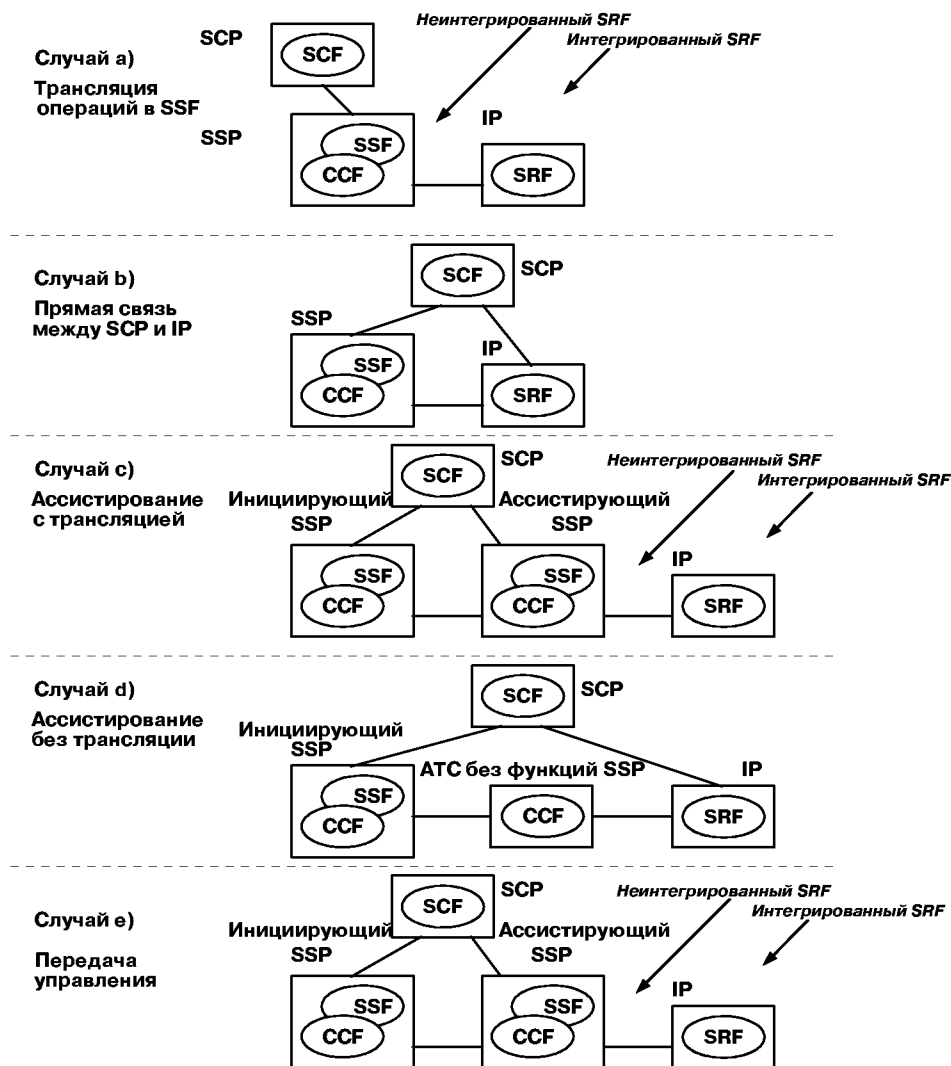


Рис. 3.4.4 Физические сценарии процедур соединения с SRF

В каждом из этих случаев для передачи операций между SCP и SSP может использоваться протокол TCAP системы OKC №7; обмен операциями между SSP и IP, при трансляции через SSP, может осуществляться через систему сигнализации DSS1 с использованием информационного элемента FACILITY (в этом случае SSP должен осуществлять преобразование протоколов для операций и ответов

INAP, которые передаются между SCP и IP). В случае непосредственной связи между SCP и IP для обмена сообщениями может использоваться протокол TCAP, для передачи сигналов управления трактом связи – любая подходящая система сигнализации (например, подсистема ISUP ОКС-7 или уровень 3 DSS1).

Случай а) показан на рисунке 3.4.5. Для интегрированного IP/SSP, внутренние действия узла могут моделироваться таким же образом, как на рисунке 3.4.5, но детали реализации оставлены на усмотрение разработчика. Такой подход приводит к тому, что SCP не должен различать, является ли IP интегрированным или внешним, но непосредственно подключенным к SSP. Установление связи между SCP и SRF в этом случае является опосредованным.

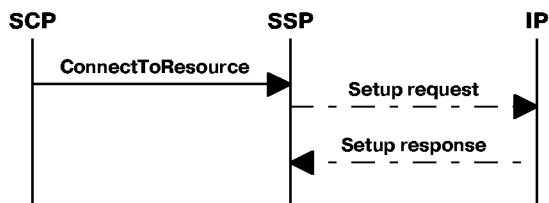


Рис. 3.4.5 Соединение с интегрированной или внешней IP с трансляцией операций через SSP

Случай b) требует от IP индикации готовности принять операции от SCP (см. рис. 3.4.6). В данном случае SCP и SRF связаны непосредственно. Обратите внимание, что необходимо передать связывающий идентификатор (*Correlation ID*), чтобы гарантировать, что транзакция, установленная между SCP и IP, может быть соотнесена с заданным трактом связи, созданным в результате выполнения операции, ранее переданной от SCP к SSP.

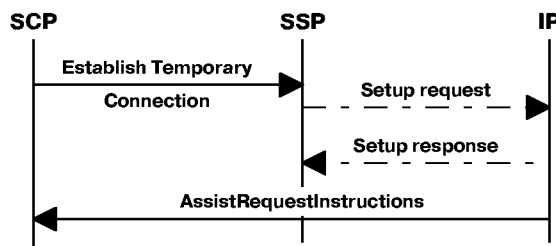


Рис. 3.4.6 Соединение IP с SCP по прямому каналу (IP инициирует взаимодействие с SCP)

Случай с) требует организации транзакции с ассистирующим SSP для того, чтобы он мог ретранслировать операции от SCP к интегрированной или внешней IP. Как только сигналы управления трактом связи достигнут ассистирующего SSP, он регистрирует идентификатор вызываемого ресурса и инициирует взаимодействие с SCP, запросившим процедуру ассистирования. Имеется возможность регистрировать и другие информационные элементы, например, адрес-

ную информацию. Сигналы управления трактом связи должны содержать информацию для идентификации SCP, запросившего ассистирующий SSP, и связывающий идентификатор. После того как SCP получит операцию *AssistRequestInstructions*, используются такие же процедуры, как и в случае а). Преамбулу данного случая иллюстрирует рисунок 3.4.7.

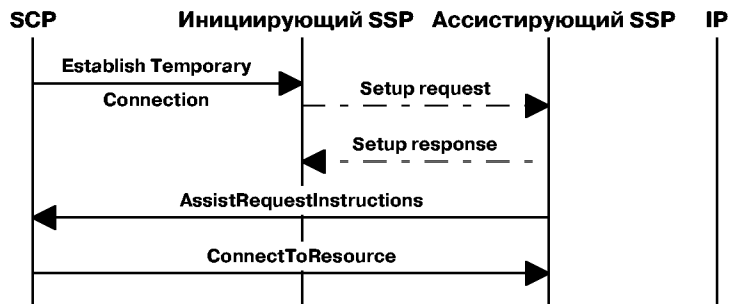


Рис. 3.4.7 Преамбула для случая ассистирования с интегрированной или внешней IP (сообщения SCP-IP транслируются через SSP)

Случай d) не требует организации второй транзакции от ассистирующего узла, так как не требуется, чтобы он обладал функциями SSP (см. рис. 3.4.8.).

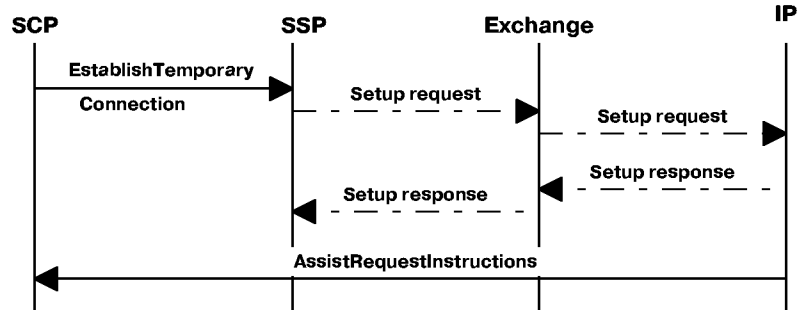


Рис. 3.4.8 Преамбула для случая ассистирования с внешней IP (прямая связь между SCP и IP)

Случай е) требует только передачи к первому SSP операции для маршрутизации соединения к тому SSP, которому будет передано управление, то есть рис. 3.4.5 применим и к данному случаю. Эта ситуация показана на рисунке 3.4.9. Следует обратить внимание на то, что действия SSP, принявшего на себя управление, порождают новое взаимодействие с SCP, в связи с которым используется операция *AssistRequestInstructions*. Как только сигналы управления трактом связи достигнут ассистирующего SSP, он идентифицирует вызываемый ресурс и инициирует взаимодействие с SCP, запросившим ассистирование. Сигналы управления трактом связи должны содержать

жать информацию для идентификации SCP, запросившего ассистирующий SSP, и связывающий идентификатор.

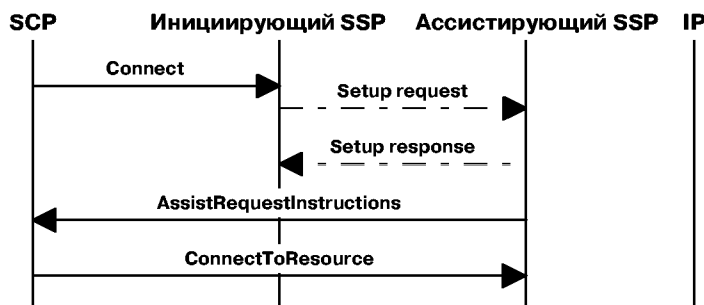


Рис. 3.4.9 Преамбула для случая ассистирования с внешней IP при непосредственной связи SCP-IP

3.4.4.2 Процедуры взаимодействия с пользователем

Процедуры взаимодействия с пользователем позволяют:

- передавать ему одно или несколько речевых сообщений, используя операции *PlayAnnouncement*;
- вести с ним диалог, используя одну или несколько последовательных операций *PromptAndCollectUserInformation*;
- комбинировать вышеупомянутые возможности;
- отменять операции *PlayAnnouncement* или *PromptAndCollectUserInformation*, используя операцию *Cancel*.

Имеются только два сценария взаимодействия с пользователем:

- первый сценарий описан на рисунке 3.4.10 и относится к случаю, когда SSP транслирует операции от SCP к IP и ответы от IP к SCP;

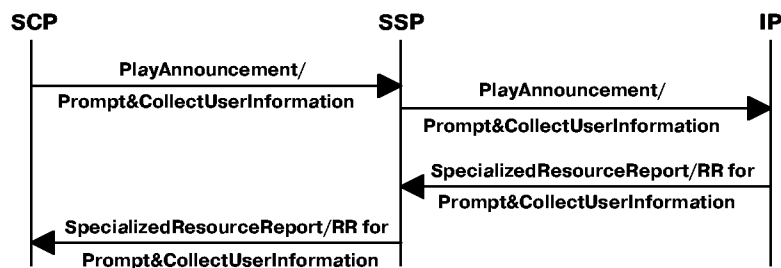


Рис. 3.4.10 Трансляция операций SCP-IP через SSP при взаимодействии с пользователем

- второй сценарий описан на рисунке 3.4.11 и относится к случаю, когда операции от SCP к IP и ответы от IP к SCP передаются без участия SSP (по каналу прямой связи).

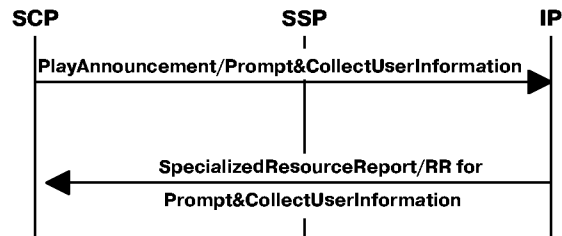


Рис. 3.4.11 Непосредственный обмен операциями SCP-IP при взаимодействии с пользователем

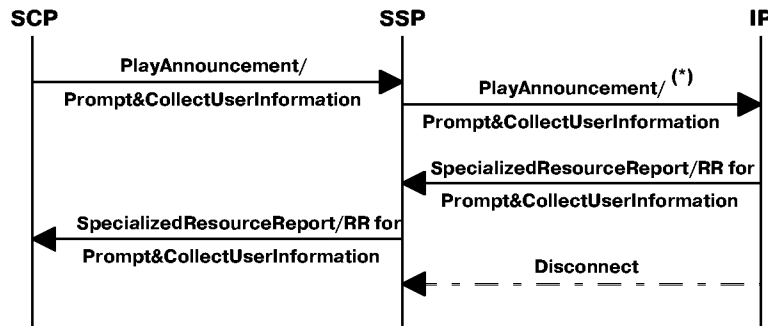
Целесообразно использовать способность TCAP переносить несколько операций в одном сообщении. Это позволяет в сценарии, показанном на рисунке 3.4.5, передавать в одном сообщении TCAP операцию *ConnectToResource* и первую операцию *PlayAnnouncement/Prompt AndCollectUserInfoInformation*, что сокращает общее количество сообщений.

3.4.4.3 Процедуры отключения SRF

Процедуры отключения контролируются SCF, а вид используемой процедуры выбирается с учетом требований предоставляемой услуги. Для разъединения тракта связи SCP выбирает либо процедуру, разрешающую разъединение со стороны SRF при завершении взаимодействия с пользователем, либо процедуру, предусматривающую, что SSF должен разъединить тракт, только получив прямое указание от SCF.

Процедуру разъединения, инициированную SRF, иллюстрирует рисунок 3.4.12. Разъединение со стороны SRF разрешается SCF при помощи операций *PlayAnnouncement/PromptAndCollectUserInfoInformation*. Когда SRF принимает такую операцию с разрешением разъединения по собственной инициативе, диалог заканчивается в соответствии с инструкциями, входящими в эту операцию, и SRF инициирует разъединение, используя соответствующие сигналы управления трактом связи. SSF/CCF определяет, что разъединение производит SRF, и сохраняет соединение с конечным пользователем. SSF возвращается в состояние «Waiting for instructions» и выполняет операции, буферизированные за время взаимодействия. В случае передачи управления, SSP, показанный на рисунке 3.4.12, является тем SSP, который принимает управление.

В случае с ассистирующим SSF процедуры разъединения, инициируемые SRF, не используются, так как ассистирующий SSF остается в состоянии «Waiting for instructions» и не допускает разъединения тракта связи с инициирующим SSF. В этом случае используются процедуры разъединения по инициативе SCF.



(*) Разъединение со стороны SRF запрещено

Рис. 3.4.12 Процедура разъединения по инициативе SRF

При наличии непосредственной связи между SCF и SRF процедуры работают таким же образом. Разъединение по инициативе SRF разрешается SCF при помощи операции *PlayAnnouncement/PromptAndCollectUserInfo*. Когда SRF принимает такую операцию с разрешением разъединения по собственной инициативе, диалог заканчивается в соответствии с инструкциями, содержащимися в этой операции, и SRF инициирует разъединение, используя соответствующие сигналы управления трактом связи. SSF/CCF определяет, что разъединение производит SRF, и сохраняет соединение с конечным пользователем. SSF возвращается в состояние «Waiting for instructions» и выполняет операции, буферизированные за время взаимодействия.

Процедура разъединения по инициативе SCF показана на рисунке 3.4.13. Сообщения управления трактом связи показаны пунктирными линиями. Чтобы инициировать отключение SRF, SCF должен передать последнюю операцию *PlayAnnouncement/PromptAndCollectUserInfo* и принять ответ на нее. Операция *SpecializedResourceReport* содержит параметр, указывающий на окончание передачи речевого сообщения (*Announcement complete*), а возвращаемым результатом операции *PromptAndCollectUser Information* является накопленная информация (*Collected information*).

Для разъединения по инициативе SCF используется передаваемая SCF операция *DisconnectForwardConnection*. Приняв ее, SSF производит разъединение тракта связи между SSF и SRF. Так как SCF (инициирующий разъединение), SSF (передающий сигналы управления разъединением) и SRF (принимающий сигналы) осведомлены о факте разъединения, их действия синхронизированы. Поэтому прекращение транзакция может быть предопределено в неявном виде. Это, однако, не запрещает использовать для прекращения транзакции явные указания с помощью специальных сообщений.

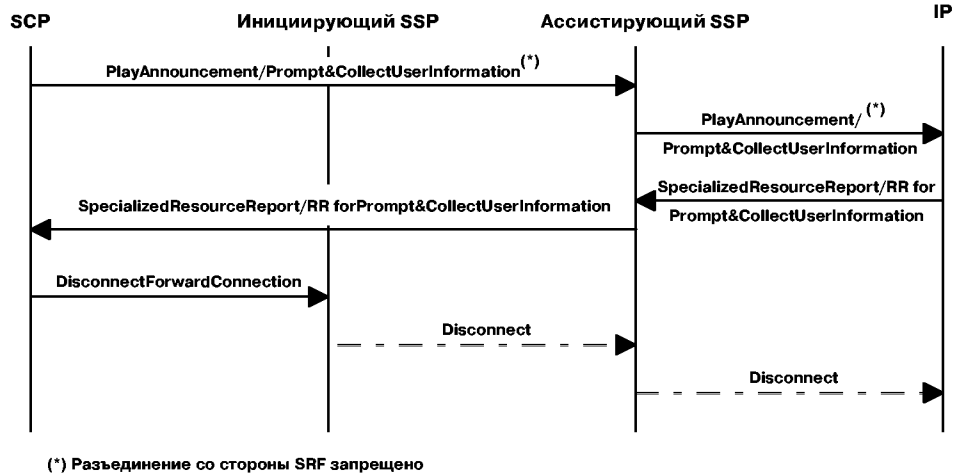


Рис. 3.4.13 Разъединение по инициативе SCF в случае с ассистированием

В случае с ассистирующим SSF SSP-инициатор, приняв от SCP операцию *DisconnectForwardConnection*, нарушает соединение с ассистирующим SSP, который, в свою очередь, разъединяет тракт связи с IP. SSP-инициатор, зная, что соединение с ассистирующим SSP было установлено в соответствии с предписанием операции *EstablishTemporaryConnection*, сохраняет соединение с пользователем и возвращается в состояние «Waiting for Instructions».

Часть 4

От теории к практике

Глава 4.1

Применение концепции IN для спецификации услуг

4.1.1 Услуга «FREEPHONE» (Бесплатный вызов)

4.1.1.1 Описание и перечень атрибутов

Услуга FREEPHONE (FPH) предоставляет пользователю телефонной сети возможность бесплатной связи с одним из нескольких терминалов абонента этой услуги путем набора единого логического номера. Терминалы абонента услуги могут размещаться в разных пунктах территории, обслуживаемой телефонной сетью, быть включенными в разные АТС этой сети и иметь разные физические (списочные) номера, а правила пересчета логического номера в тот или иной физический номер определяются критериями, которые назначает абонент услуги. Вызов по номеру FPH из той или иной точки ТфОП может маршрутизироваться в зависимости от времени суток, дня недели (рабочий, выходной, праздничный), номера вызывающего абонента, специфических требований абонента – владельца номера, поступающей нагрузки и т. п. В случае занятости или при отсутствии ответа маршрутизация может отличаться от маршрутизации при наличии ответа. Кроме того, логика услуги может предусматривать подачу вызывающему пользователю речевых сообщений и акустических сигналов. Схема предоставления услуги FREEPHONE показана на рисунке 4.1.1.

В соответствии с рекомендацией ITU-T Q.1211 для реализации услуги FPH обязательными являются атрибуты: *One Number (ONE)* – атрибут, обеспечивающий доступ к нескольким физическим номерам по единому логическому номеру, и *Reverse Charging (REVC)* – атрибут, предусматривающий отнесение платы за связь не на счет вызывающего пользователя, а на счет абонента услуги. В таблице 4.1.1 приводится полный перечень обязательных и разрешенных для услуги FPH дополнительных атрибутов.

При обращении к услуге FPH пользователь, в соответствии с используемым в его сети планом нумерации, набирает код доступа к IN, код доступа к услуге и, далее, логический номер абонента услуги. Логический номер не входит в план нумерации ТФОП и может состоять из легко запоминающихся цифр. Абонентом услуги FPH может быть агентство, магазин, ресторан или иное предприятие, желающее иметь единый логический номер для всех своих филиалов, находящихся в разных географических регионах.

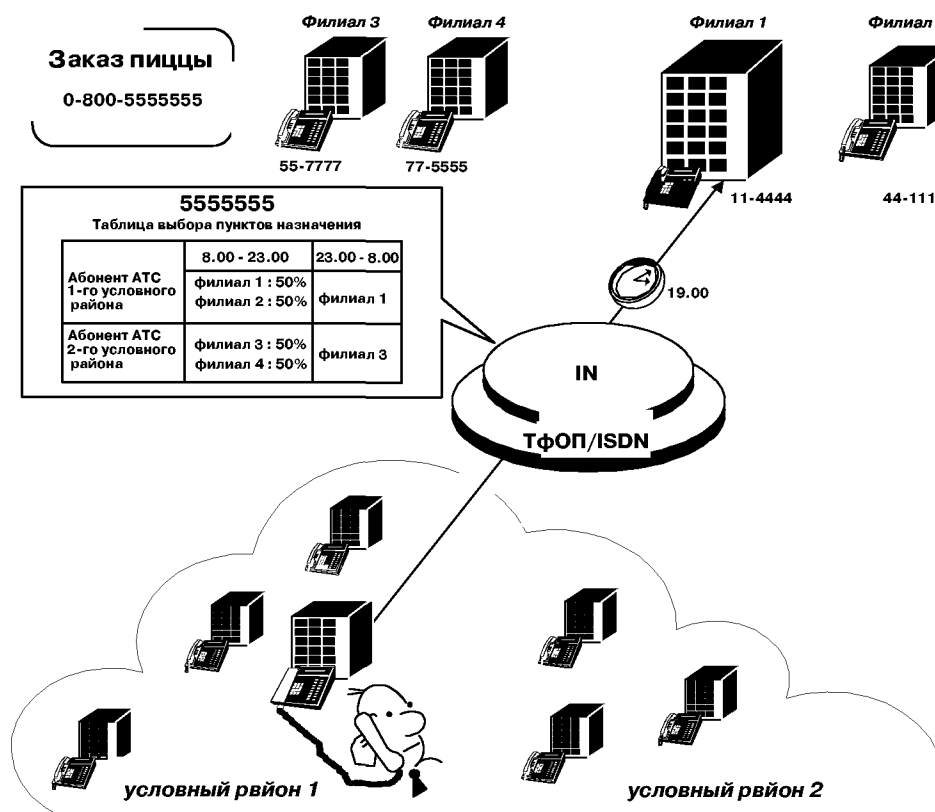


Рис. 4.1.1 Схема предоставления услуги FPN

В телефонных сетях России обращение к услуге FPN осуществляется набором номера вида $P_n - 800-X_1X_2\dots X_n$, где:

P_n – префикс, равный:

8 – если выход к платформе IN осуществляется через АМТС,

0 – если выход к платформе IN осуществляется через УСС;

800 – код доступа к услуге FPN;

$X_1X_2\dots X_n$ – логический номер абонента услуги FPN.

Обоснование целесообразности использования АМТС или УСС в качестве точки доступа на первоначальном этапе построения IN и используемые при этом правила нумерации услуг в России читатель найдет в главе 4.3.

Таблица 4.1.1

Обязательные атрибуты:		Дополнительные атрибуты:	
ONE - One Number (Единый номер)		AUTC - Authentication (Проверка прав)	
REVC - Reverse Charging (Начисление платы вызываемому абоненту)		CD - Call Distribution (Распределение вызовов)	
		CFC - Call Forwarding on Busy/Don't answer (Переадресация при занятости или при отсутствии ответа)	
		GAP - Call Gapping (Прореживание потока вызовов)	
		LIM - Call Limiter (Ограничение количества вызовов)	
		LOG - Call Logging (Регистрация вызовов)	
		QUE - Call Queueing (Установка вызова на ожидание)	
		CPM - Customer Profile Management (Управление профилем обслуживания)	
		CRA - Customised Recorder Annoucement (Воспроизведение записанного сообщения).	
		CRG - Customised Ringing (Назначение типов вызывного сигнала)	
		DUP - Destinating User Prompter (Подсказка вызываемому абоненту)	
		MAS - Mass Calling (Массовые вызовы)	
		ODR - Origin Dependent Routing (Маршрутизация в зависимости от местоположения вызывающего пользователя)	
		OCS - Originating Call Screening (Просеивание вызовов по признаку места их возникновения)	
		OUP - Originating User Prompter (Подсказка вызывающему абоненту)	
		TDR - Time Dependent Routing (Маршрутизация в зависимости от времени)	

4.1.1.2 Алгоритм работы

Рассмотрим работу логики услуги в случае, когда имеется интегрированный SSP/IP, а доступ к ресурсам IN осуществляется через УСС (Пн=0). Алгоритм работы будем рассматривать на примере, когда в городе имеется компания по доставке пиццы на дом клиенту в течение 30-60 мин с момента заказа. Компания является абонентом услуги FPN и связаться с ней можно, набрав номер 0-800-555-5555, который доводится до сведения потенциальных клиентов с помощью рекламы в средствах массовой информации (телевидение, радио, печать).

Компания имеет 4 филиала, размещенных в разных частях города, при этом территория города делится на два условных района, обслуживаемых двумя разными группами АТС. Каждый условный район обслуживается двумя филиалами компании. Телефонные номера филиалов, обслуживающих первый условный район, – 11-4444 и 44-1111. Телефонные номера филиалов, обслуживающих второй условный район, – 55-7777 и 77-5555. С 8.00 до 23.00 работают все четыре филиала. В ночное время в каждом районе остаются работать по одному филиалу, например, с телефонными номерами 11-4444 и 55-7777. Номер расчетного счета компании, на который будет относиться плата за все входящие вызовы, – 000XXX1111XXX. Главный менеджер компании по доставке пиццы имеет возможность посредством доступа к SMP изменять данные абонированной услуги.

Итак, пусть клиент с телефонным номером 22-2211 заказывает на дом пиццу, набирая объявленный в рекламе номер услуги FPH, принадлежащий компании по доставке пиццы, – 0-800-555-5555. Будем считать, что телефонная линия пользователя 22-2211 включена в АТС, обслуживающую первый условный район, и что вызов поступил в 19.00. Для самого клиента не имеет значения, откуда и в котором часу он звонит, и какой именно филиал будет выполнять его заказ.

При приеме вызова от пользователя сеть, анализируя номер, откуда поступил вызов (22-2211), определяет, что клиент находится в первом условном районе. Затем сеть сравнивает текущее время с расписанием работы филиалов и маршрутизирует вызов к одному из двух филиалов данного условного района с учетом заданного процентного распределения вызовов между этими филиалами.

Алгоритм предоставления услуги содержит следующие шаги.

1. На основании набранных пользователем цифр 0-800-555-5555 вызов направляется средствами ТФОП по прямым пучкам соединительных линий от оконечной АТС к УСС с функциями SSP. На этом участке может использоваться как система сигнализации по выделенным сигнальным каналам (2ВСК), так и общеканальная система ОКС-7 (МТР, ISUP-R).
2. Обнаружив, что вызов связан с обращением к услуге Интеллектуальной Сети, SSP приостанавливает процесс его обслуживания и передает в SCP соответствующий запрос посредством операции INAP *InitialDP*. В случае услуги FPH эта операция обязательно содержит параметр *Service Key* со сведениями о типе услуги, информацию о набранных пользователем цифрах номера, а также о номере самого пользователя.

3. При наличии в сети IN нескольких SCP операция *InitialDP* может проходить через транзитные пункты сигнализации OKC-7 (STP). Чтобы определить, к какому SCP должна быть маршрутизирована данная операция, STP производит преобразование поля *Global Title* параметра *Calling Party Number* подсистемы SCCP. Тип трансляции, проводимой подсистемой SCCP, определяется STP на основании набранного пользователем логического номера.
4. Приняв операцию INAP *InitialDP*, SCP анализирует параметр *Service Key*, в результате чего операция направляется к приложению, обслуживающему услугу FPH. Если такое приложение не найдено, операция *InitialDP* игнорируется, а процесс обслуживания вызова, приостановленный в SSP, после срабатывания таймера операции *InitialDP* направляет соединение к автоинформатору.
5. Логический номер используется для доступа к данным о профиле обслуживания абонента услуги FPH, содержащим сведения о физических номерах и о критериях, в соответствии с которыми должен производиться пересчет логического номера в тот или иной физический номер. Определяется статус абонентского оборудования, и если оно находится в неактивном состоянии (например, выключено по какой-либо причине администратором услуги), то вызывающей стороне передается сообщение о прекращении обслуживания вызова.
6. Используя логический номер абонента услуги FPH, SCP определяет номер его счета и информирует SSP об организации начисления оплаты и о физическом вызываемом номере ТФОП с помощью операций INAP *FurnishChargingInformation* и *Connect*. Приостановленный процесс обслуживания вызова в SSP возобновляется в соответствии с данными, полученными от SCP.

4.1.1.3 Конструктивные блоки

На рисунке 4.1.2 показана цепочка блоков SIB, образуемая для организации услуги FPH. Некоторые детали описанного выше алгоритма опущены с сохранением, однако, общности подхода, принятого при спецификации услуг IN. В рассматриваемом нами примере использованы блоки SIB, реализующие обязательные и некоторые из дополнительных атрибутов. Применительно к этому примеру продемонстрируем спецификацию параметров блоков SIB в соответствии с приведенным выше описанием услуги. Логика услуги начинается с блока базового процесса обслуживания вызова BCP в точке POI, далее проходит по цепочке блоков SCREEN-COMPARE-DISTRIBUTION-TRANSLATE-CHARGE-CONNECT и затем снова возвращается в блок BCP в точке POR.

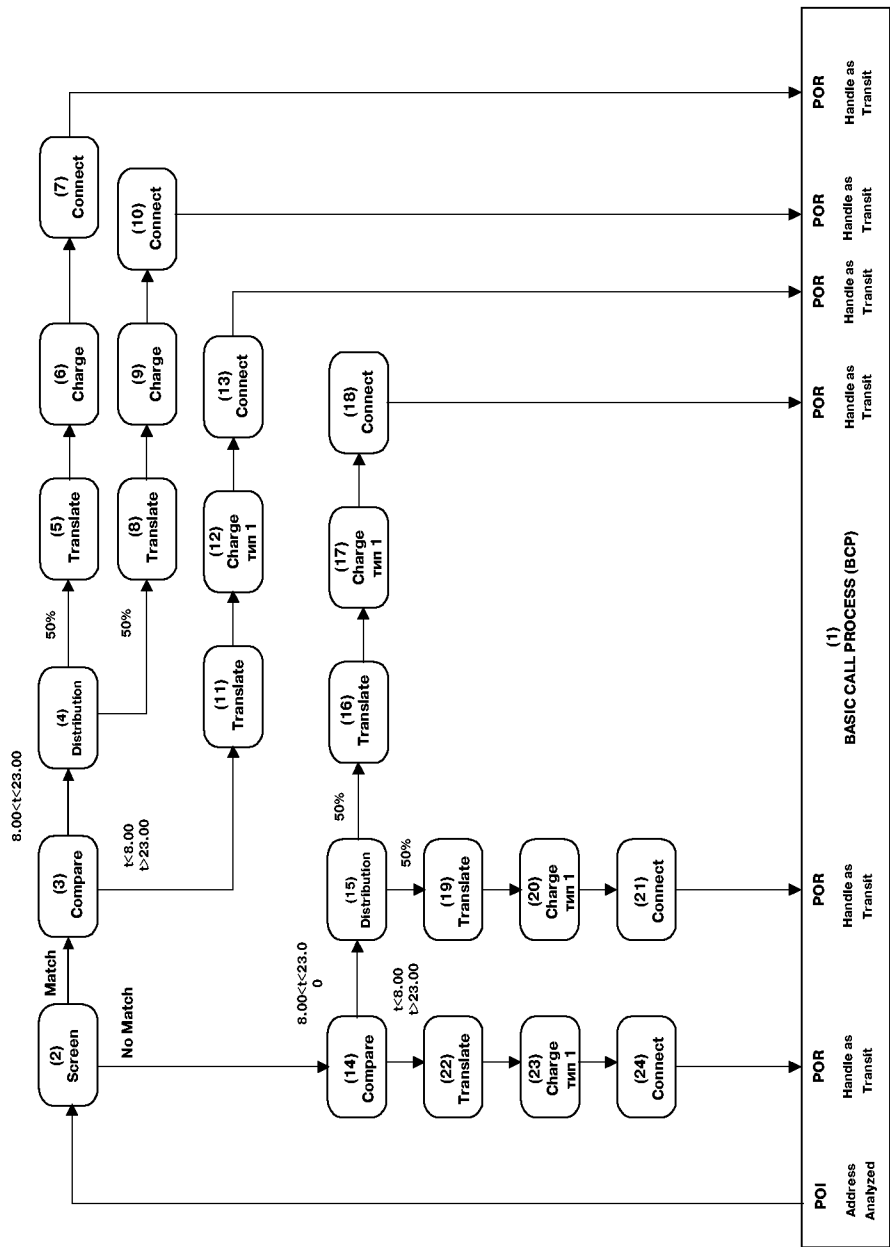


Рис. 4.1.1.2 Вариант организации цепочки блоков SIB для реализации услуги FPN

Блок (1) BCP

После приема и анализа переданной пользователем адресной информации базовый процесс обслуживания вызова BCP приостанавливается в точке POI Address Analysed, происходит выход из BCP и начинается последовательное выполнение цепочки блоков SIB, реализующих логику услуги. Динамическими данными CID в данном случае являются номер вызывающего пользователя (клиента компании по доставке пиццы) 22-2211 и набранные им цифры 800-555-5555.

Выход (из BCP в GSL):		
Логическое завершение	= POI Address Analysed (номер проанализирован)	Переход к блоку (2) SCREEN
CID Dialed Number (набранный номер)	= код доступа к услуге FPH, логический номер	800-555-5555
CID Calling Line Identity (идентификация вызывающего пользователя)	= номер вызывающего пользователя	22-2211
Вход (в BCP из GSL):		
Логический старт	= POR Handle as Transit (обработать как транзитный)	Возобновление BCP с новыми данными для маршрутизации

Возврат в блок BCP происходит после завершения выполнения цепи SIB в одной из точек возврата POR.

Блок (2) SCREEN

Блок (2) SCREEN служит здесь для того, чтобы определить, из какого условного района поступил вызов. Поскольку в нашем примере вся территория города разделена всего на два условных района, то номер вызывающего пользователя сопоставляется с находящимся в SDF списком номеров, определяющих принадлежность к какому-то одному (для определенности – к первому) условному району. На входе блока параметр SSD List Indicator определяет список телефонных номеров первого условного района. Параметр CID Identifier содержит номер вызывающего пользователя.

Вход:		
SSD Screen List Indicator (Индикатор списка)	= имя списка с телефонными номерами первого условного района	«Список телефонных номеров первого условного района»
CID Identifier (Идентификатор)	= номер вызывающего пользователя	22-2211
Выход:		
CID Identifier (Идентификатор)	= номер вызывающего пользователя	22-2211
Логическое завершение 1	= номер вызывающего пользователя соответствует первому условному району	Положительный результат - переход к блоку (3) COMPARE
Логическое завершение 2	= номер вызывающего пользователя не соответствует первому условному району	Отрицательный результат - переход к блоку (18) COMPARE

В рассматриваемом примере нас будет интересовать только положительный результат: Match – номер вызывающего пользователя есть в списке и соответствует первому условному району. Логика услуги переходит к выполнению блока (3) COMPARE.

Блок (3) COMPARE

Блок (3) COMPARE сопоставляет время поступления вызова (Time of Day) с двумя пороговыми значениями (8.00 ч и 23.00 ч). Блок реализуется как комбинация двух блоков COMPARE, один из которых сравнивает текущее время с 8.00, а другой – с 23.00. Дальнейшая работа логики услуги зависит от того, в какое время суток поступил вызов. Напомним, что вызов от клиента поступил в 19.00 ч, то есть в период действия основного режима работы компании.

Вход:		
SSD CIDFP-Compare (Указатель поля CID Compare)	= указатель поля CID, в которое помещены данные о времени поступления вызова	
SSD Reference Value (Контрольное значение)	= заданный временной интервал	8.00<t<23.00
CID Identifier (Идентификатор)	= время поступления вызова	19.00
Выход:		
Логическое завершение 1	= внутри интервала	Положительный результат - переход к блоку (4) DISTRIBUTION
Логическое завершение 2	= вне интервала	Отрицательный результат - переход к блоку (11) DISTRIBUTION

На входе блока с помощью параметра SSD CIDFP-Compare определяется поле, в котором помещается величина, подлежащая сравнению (то есть время поступления вызова). С помощью параметра SSD Reference Value определяется контрольное значение (т.е. интервал времени от 8.00 до 23.00), с которым эта величина должна сравниваться. Поскольку наш пример соответствует положительному результату (время поступления вызова находится внутри заданного интервала), то логика услуги переходит к выполнению блока (4) DISTRIBUTION.

Блок (4) DISTRIBUTION

Блок (4) DISTRIBUTION служит для распределения вызовов по двум разным направлениям в соответствии с заданным процентным соотношением: 50% вызовов, требующих связи с логическим

номером 800-555-5555, маршрутизируется в направлении 1 (Logical End – 1), соответствующем физическому номеру одного филиала первого условного района (11-4444), а другие 50% – в направлении 2 (Logical End -2), соответствующем физическому номеру другого филиала того же района (44-1111).

На входе блока параметр SSD Number of Logical Ends определяет число возможных направлений. Параметр SSD Algorithm Parameters задает процентное соотношение, в котором вызовы распределяются по этим направлениям. Поскольку в нашем примере вызовы распределяются между двумя направлениями поровну, то логика услуги может равновероятно выбрать как логическое завершение 1, так и логическое завершение 2, и, соответственно, перейти к выполнению блока (5) TRANSLATE или блока (8) TRANSLATE. Для определенности будем считать, что логика услуга выбрала переход к блоку (5) TRANSLATE.

Вход:		
SSD Number of Logical Ends (Число логических завершений)	= 2	Распределение поступающих вызовов по двум направлениям
SSD Algorithm Parameters (Способ распределения поступивших вызовов)	= процентное соотношение	50% - логическое завершение 1 50% - логическое завершение 2
Выход:		
Логическое завершение 1	= направление 1	Переход к блоку (5) TRANSLATE
Логическое завершение 2	= направление 2	Переход к блоку (8) TRANSLATE

Блок (5) TRANSLATE

Блок 5 TRANSLATE выполняет пересчет логического номера, набранного пользователем, в физический номер, используемый для маршрутизации. На входе блока TRANSLATE определяется тип выполняемой операции. В рассматриваемом примере параметр SSD Type определяет, что в блоке (5) TRANSLATE один номер (логический 800-555-5555) пересчитывается в другой (физический 11-4444).

Параметр CID Information на входе блока содержит данные, которые подлежат пересчету, т. е. набранный пользователем номер 800-555-5555. На выходе параметр CID Translated Data содержит данные, полученные в результате пересчета, т.е. физический номер 11-4444, необходимый для маршрутизации вызова.

Вход:		
SSD Type (Тип пересчета)	= один номер в один номер	Пересчет номера FPH в номер ТФОП
SSD File Indicator (Индикатор файла)	= тип базы данных	База данных для пересчета номеров FPH
SSD CIDFP-Info	= указатель поля CID, в котором помещен набранный номер	CID Information
SSD CIDFP-Translated	= указатель поля CID, в котором помещен пересчитанный номер	CID Translated data
CID Information (Информация)	= набранный номер	800-555-5555
Выход:		
CID Translated data (Пересчитанные данные)	= номер назначения	11-4444
Логическое завершение	= успешное	Переход к блоку (6) CHARGE

Блок (6) CHARGE

Блок (6) CHARGE используется для формирования информации о том, куда должна быть отнесена плата за связь. На входе блока статический параметр SSD Number of Accounts определяет, что для этой цели предусматривается всего один счет. Параметр SSD Account состоит из двух частей, одна из которых содержит указатель поля CID, в котором помещается номер счета абонента услуги FPH, а вторая определяет, что на этот счет следует относить 100% начисленной платы.

Параметр CID Account содержит номер счета, на который начисляется плата. Этот номер является функцией от набранного пользователем логического номера.

Вход:		
SSD Number of Accounts (количество счетов)	= 1	Плата всегда относится на один счет
SSD Account		
CIDFP-Number	= указатель поля CID, в котором помещен номер счета абонента	CID-Account
Account % (процент распределения платы между счетами)	=100	Плата полностью относится на счет абонента услуги FPH
CID Account (Номер счета)	=функция от набранного номера	f(555-5555)=000xxx1111xxx
Выход:		
Логическое завершение	= успешное	Переход к блоку (7) CONNECT

Блок (7) CONNECT

Блок (7) CONNECT служит для установления соединения между пользователем и вызываемой стороной. На входе блока параметр SSD CIDFP-Destination содержит указатель поля CID, в котором помещен физический номер филиала-адресата. Параметр SSD CIDFP-CLI указывает то поле CID, в котором помещен номер вызывающего пользователя.

Вход:		
SSD CIDFP-Destination	= указатель поля CID, где помещен физический номер назначения	CID Destination Number
SSD CIDFP-CLI	= указатель поля CID с номером вызывающего пользователя	CID CLI
CID Destination Number (Номер назначения)	= номер назначения	11-4444
CID- CLI (Идентификатор вызывающей стороны)	= номер вызывающего пользователя	22-2211
Выход:		
Логическое завершение	= успешное	Возврат в блок (1) BCP

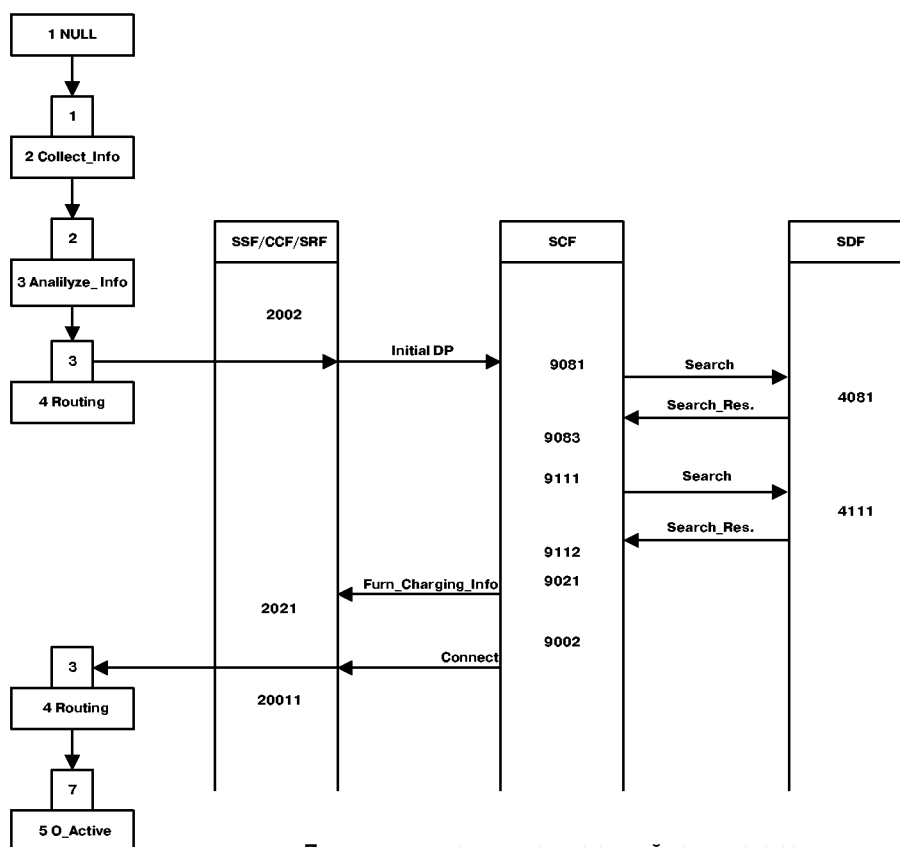
Блок (1) BCP

Выполнение цепочки блоков (2) SCREEN – (3) COMPARE – (4) DISTRIBUTION – (5) TRANSLATE – (6) CHARGE – (7) CONNECT завершается возвратом в блок (1) BCP. Процесс обслуживания вызова возобновляется и далее производится уже на основании данных, полученных через точку возврата POR Handle as Transit. Возобновленный в такой точке процесс, вплоть до разъединения, выполняется блоком BCP как процесс обслуживания только что поступившего вызова, но с параметрами маршрутизации, полученными от блока (7) CONNECT.

4.1.1.4 Информационные потоки и действия функциональных объектов

На рисунке 4.1.3 изображена диаграмма информационных потоков, имеющих место при реализации услуги FPH посредством выбранной нами цепочки блоков SIB. Ниже описываются действия функциональных объектов в той последовательности, в которой они выполняются при предоставлении услуги.

O_BCSM приостанавливает обслуживание вызова после обнаружения точки TDP Analysed_Info и получения положительного результата проверки того, что критерий, назначенный для этой точки, удовлетворяется; в нашем случае критерием служат цифры 800.



Примечание - показан только случай успешного соединения

Рис. 4.1.3 Диаграмма информационных потоков

Стартовое событие	Выполняемые действия
2002 (SSF, (1) BCP) Обнаружена триггерная точка - типа «запрос»	- Прерывание базового процесса обслуживания вызова - посылка информационного потока <i>InitialDP</i>
9081 (SCF, (2) SCREEN) Запрос проверки того, что номер вызывающего пользователя принадлежит «Списку телефонных номеров первого условного района»	- прием запроса проверки принадлежности списку - создание и передача к SDF информационного потока <i>Search</i>
4081 (SDF, (2) SCREEN) Прием информационного потока <i>Search</i>	- определение принадлежности номера вызывающего пользователя «Списку телефонных номеров первого условного района» - передача в SCF информационного потока <i>Search Result</i>
9083 (SCF, (2) SCREEN) Прием информационного потока <i>Search Result</i>	- предоставление результата сопоставления со списком (Match)

Стартовое событие	Выполняемые действия
9031 (SCF, (3) COMPARE) Запрос сопоставить время поступления вызова с заданным временным интервалом	- сопоставление времени поступления вызова с заданным временным интервалом

Стартовое событие	Выполняемые действия
9041 (SCF, (4) DISTRIBUTION) Запрос распределения вызовов в соответствии с заданным процентным соотношением	- выполнение алгоритма распределения вызовов

Стартовое событие	Выполняемые действия
9111 (SCF, (5) TRANSLATE) Запрос пересчета набранного пользователем логического номера в соответствующий физический номер	- начало процесса пересчета номера - создание и передача в SDF информационного потока <i>Search</i>
4111 (SDF, (5) TRANSLATE) Прием информационного потока <i>Search</i>	- пересчет номера - передача в SCF информационного потока <i>Search Result</i>
9112 (SCF, (5) TRANSLATE) Прием информационного потока <i>Search Result</i>	- предоставление результата пересчета номера

Стартовое событие	Выполняемые действия
9021 (SCF, (6) CHARGE тип 1) Запрос создания записи о начисляемой плате	- передача информационного потока <i>Furnish Charging Information</i>
2021 (SSF, (6) CHARGE тип 1) Обработка запроса начать начисление платы	- прием и анализ информационного потока <i>Furnish Charging Information</i> - создание записи о начисляемой плате

Стартовое событие	Выполняемые действия
9002 (SCF, (7) CONNECT) Запрос отправить инструкции, необходимые для установления соединения	- передача к SSP информационного потока <i>Connect</i>
20011 (SSF, (7) CONNECT) Запрос установления соединения	- установление соединения в соответствии с полученными из SCP инструкциями

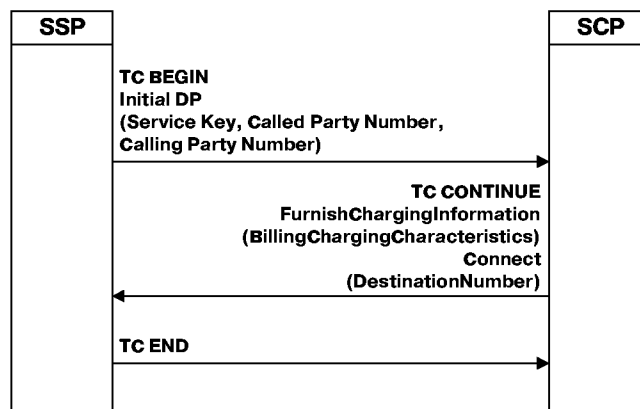


Рис. 4.1.4 Диаграмма обмена сообщениями TCAP/INAP для услуги FPH

На рисунке 4.1.4 показана последовательность сообщений подсистемы TCAP OKC-7 и переносимых ими операций INAP между SSP/IP и SCP при реализации услуги FPH.

4.1.2 Услуга «Account Card Calling» (Вызов по телефонной карте)

4.1.2.1 Описание и перечень атрибутов

Услуга связи с отнесением платы за нее на счет, указанный в (виртуальной) расчетной телефонной карте, дает пользователю возможность производить вызовы как с телефонного аппарата, так и с карточного таксофона. Начисляемая за связь плата автоматически относится на счет, номер которого соответствует идентификатору карты. Пользователь набирает код доступа, система считывает с карты необходимые данные, или просит пользователя ввести эти данные и свой персональный идентификационный номер PIN, после чего проверяет данные и информирует пользователя о возможности ввести номер вызываемого абонента. При этом плата за связь не относится на ту телефонную линию, с которой произведен вызов. На рисунке 4.1.5 показана схема предоставления услуги Account Card Calling (ACC).

Разновидностью услуги ACC является услуга с оплатой связи по кредитной карте (CREDIT CARD CALLING, CCC), позволяющая использовать обычные кредитные карты, например, для связи с карточных таксофонов. В таблице 4.1.2 приводится полный перечень обязатель-

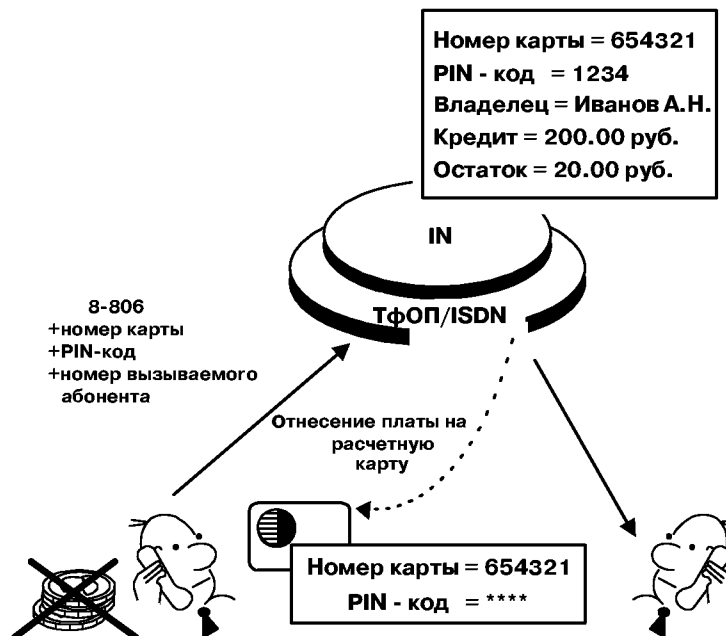


Рис. 4.1.5 Схема предоставления услуги АСС

ных и разрешенных для услуги АСС дополнительных атрибутов.

Основными атрибутами услуги являются сокращенный набор номера (ABD), проверка прав доступа (AUTZ) и передача речевого приглашения вызывающему пользователю (OUP). В качестве дополнительной возможности пользователю может быть разрешено производить последующие вызовы, не набирая заново номер карты и PIN-код. Система способна отслеживать в реальном времени допустимый лимит и имеющийся на карте депозит.

Таблица 4.1.2

Обязательные атрибуты:	Дополнительные атрибуты:
ABD - Abbreviated Dialling (Сокращенный набор)	LOG - Call Logging (Регистрация вызовов)
AUTC - Authentication (Проверка прав)	CRA - Customised Recorder Annoucement (Воспроизведение записанного сообщения)
AUTZ - Authorization Code (Разрешающий код)	OUP - Originating User Prompter (Подсказка вызывающему абоненту)
	TCS - Terminating Call Screening (Просеивание входящих вызовов)

Чтобы обратиться к услуге АСС, пользователь вводит код доступа, назначенный оператором сети для поставщика этой услуги. Если один поставщик обслуживает несколько абонентов АСС (организаций, выпускающих телефонные карты), то, чтобы идентифицировать абонента, пользователь должен ввести дополнительную цифру. Пе-

ред кодом доступа к услуге может потребоваться набор префикса выхода к сети IN.

Пользователь услуги идентифицируется номером кредитной карты и PIN-кодом. На случай ошибочного набора номер карты и PIN-код могут вводиться несколько раз. Максимальное количество неудачных попыток доступа может быть ограничено, после превышения допустимого числа попыток система отмечает кредитную карту как недействительную.

В соответствии с международными стандартами формат номера карты и PIN-кода имеет следующий вид:

[8 9 A A B B B] X X X . . . X X L P I N N, где:

- 89 – идентификатор экономической деятельности, присвоенный отрасли «Электросвязь»;
- AA – код страны (от 1 до 3 цифр);
- BBB – идентификатор производителя (до 4 цифр, но внутри одной страны количество цифр фиксировано);
- XXX...XX – индивидуальный идентификатор карты, где X – число от 0 до 9 и количество цифр не фиксировано;
- L – проверочная цифра (от 0 до 9);
- PINN – персональный идентификационный номер (обычно 4 цифры от 0 до 9).

При обращении к услуге АСС пользователь, в соответствии с принятым в его сети планом нумерации, набирает код выхода к IN и код доступа к услуге. В телефонных сетях России обращение к услуге АСС осуществляется набором номера вида Пн – 806.

4.1.2.2 Алгоритм работы

Рассмотрим работу логики услуги. Для этого предположим, что используется интегрированный SSP/IP, а доступ к ресурсам IN осуществляется через АМТС (Пн=8). В качестве примера рассмотрим предоставление услуги АСС пользователю, которому нужна связь с абонентом телефонной сети, имеющим номер 11-22-33. Номер расчетной карты вызывающего пользователя – 654321, PIN-код – 1234. Вызывающий является пользователем той же местной сети, что и вызываемый, и имеет номер 33-22-11. Алгоритм предоставления услуги содержит следующие шаги.

1. На основании набранных пользователем цифр 8-806 вызов направляется средствами ТФОП от оконечной АТС по прямым пучкам соединительных линий к АМТС с функциями SSP. На этом участке может использоваться система сигнализации по двум выделенным сигнальным каналам (2BCK) или общеканальная система сигнализации ОКС-7 (MTP, ISUP).

2. SSP анализирует коды доступа к IN и к услуге. Обнаружив, что вызов связан с обращением к услуге IN, SSP приостанавливает процесс его обслуживания.
3. SSP формирует и передает к SCP операцию INAP *InitialDP*, содержащую, по крайней мере, параметр *Service Key*, который определяет тип услуги, а также информацию о номере вызывающей стороны и набранные пользователем цифры.
4. Приняв операцию *InitialDP*, SCP анализирует параметр *Service Key*, в результате чего принятая операция направляется к приложению, обслуживающему услугу ACC. Если такое приложение не найдено, то операция игнорируется, а процесс обслуживания вызова, приостановленный в SSP, после срабатывания таймера операции *InitialDP* направляет соединение к автоинформатору.
5. Передавая операцию INAP *Prompt&CollectUserInfo*, SCP информирует SSP/IP о необходимости организации диалога между функциями SRF и пользователем для проверки его прав пользования услугой. Получив речевое сообщение, запрашивающее номер карты и PIN-код, пользователь передает эту информацию с помощью средств многочастотного набора DTMF. Функции SRF в IP принимают цифры от пользователя и передают их к SCP посредством операции INAP *CollectedUserInfo*.
6. Если обмен информационными потоками *Search* и *Search Result* между SCF и SDF не дал положительного результата (то есть номер карты не найден в SDF или PIN код не соответствует номеру данной карты), то пользователю предлагается еще раз ввести номер расчетной карты и PIN-код (повторение п. 5). Если количество повторов превысит допустимое, расчетная карта отмечается как недействительная, а приостановленный в SSP процесс направляет вызов к автоинформатору и инициирует процедуру разъединения.
7. В случае успешного завершения описанных процедур SCP формирует и передает к SSP/IP операцию INAP *Prompt&CollectUserInfo*, в результате чего пользователю передается речевое сообщение, предлагающее ему набрать номер вызываемой стороны. Функции SRF в IP принимают этот номер от пользователя и передают его к SCP посредством операции INAP *CollectedUserInfo*. После формирования записей, нужных для начисления платы, от SCP к SSP/IP передается операция INAP *Connect*, содержащая номер назначения. Приостановленный процесс обслуживания вызова возобновляется с использованием полученных от SCP данных.
8. Для организации начисления платы за связь и для контроля текущей начисленной суммы SCP передает к SSP операции INAP *Fur-*

nishChargingInformation и *ApplyCharging*, содержащие формат записи и данные, нужные для ее формирования, а также сведения о допустимом для данного вызова лимите стоимости (обычно – в тарифных единицах). Во время связи SSP начисляет плату за нее в реальном времени, периодически передавая текущие данные к SCP в операции *INAP ApplyChargingReport*. После окончания связи (или когда достигается лимит) сведения о начисленной плате передаются в операции *INAP ApplyChargingReport* к SCP, что позволяет обновить информацию об остатке на счете данной телефонной карты. В записи, произведенной SSP, отмечается, что плата за данную связь не должна быть отнесена к линии, по которой поступил запрос этой связи.

4.1.2.3 Конструктивные блоки

На рисунке 4.1.6 показана цепочка блоков SIB, образуемая для организации услуги АСС. При этом, как и это было сделано и для услуги FPH, некоторые детали описанного выше алгоритма опущены. В данном примере приведен минимальный набор блоков, который, в комбинации с глобальной логикой услуги GSL, реализует только обязательные атрибуты.

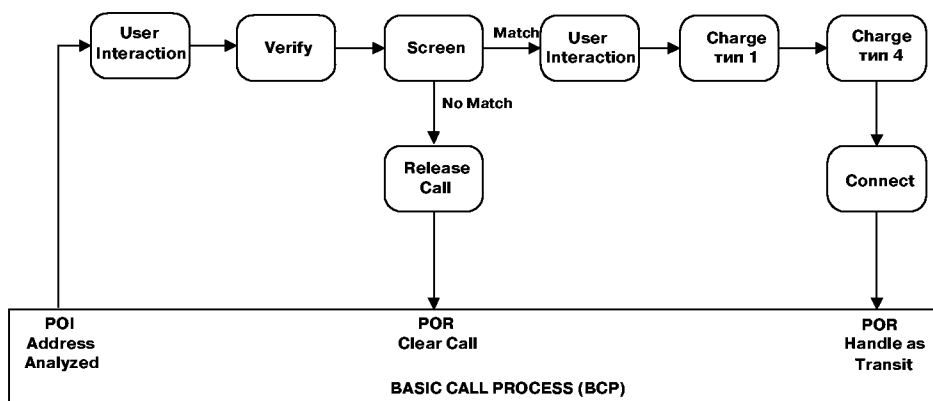


Рис. 4.1.6 Взаимосвязь блоков SIB, используемых услугой АСС

Блок BCP

После приема и анализа набранной пользователем адресной информации базовый процесс обслуживания вызова BCP приостанавливается в точке *POI Address Analysed*. Происходит выход из BCP и начинается последовательное выполнение цепи блоков SIB, реализующих логику услуги. Динамическими данными CID в данном случае будут служить обеспечивающий доступ к услуге код 806 и номер вызывающего пользователя 33-22-11.

Выход (из BCP в GSL):		
Логическое завершение	= POI Address Analysed (номер проанализирован)	Переход к User Interaction
CID Dialed Number (набранный номер)	= код доступа к услуге ACC	806
CID Calling Line Identity (идентификация линии вызывающего пользователя)	= номер вызывающего пользователя	33-22-11
Вход (в BCP из GSL):		
Логический старт	= POR1 Handle as Transit (обработать как транзитный)	Возобновление BCP с новыми данными для маршрутизации
	= POR2 Release Call (разъединить)	Возобновление BCP с целью выполнить разъединение

Возврат в блок BCP происходит после выполнения цепи SIB в одной из точек возврата – POR1 или POR2.

Блок USER INTERACTION

Блок USER INTERACTION, обеспечивающий обмен информацией между сетью и пользователем, в данном случае применяется для запроса и приема номера карты и PIN-кода. На входе блока с помощью статического параметра SSD Announcement Parameters определяется вид передаваемого пользователю речевого сообщения; в нашем случае это будет приглашение к набору цифр. Параметр SSD Type устанавливает форму представления передаваемой пользователем информации; в данном примере параметр указывает, что цифры должны передаваться многочастотным кодом DTMF. Параметр CID Call Party Identifier идентифицирует пользователя, с которым должен идти обмен информацией (в данном случае это – вызывающая сторона).

Вход:		
SSD Announcement Parameters (Параметры речевого сообщения)	= идентификатор речевого сообщения	«Введите номер карты и PIN-код»
SSD Type (Параметры принимаемой информации)	= тип информации	DTMF
CID Call Party Identifier (Идентификатор участника связи)	= вызывающая сторона	33-22-11
Выход:		
CID Collected data (Принятые данные)	= номер карты, PIN-код	654321, 1234
Логическое завершение	= успешное	Переход к Verify

Блок VERIFY

Блок VERIFY проверяет полученную от пользователя информацию (PIN-код и номер карты) на предмет синтаксического соответствия установленному формату. Предполагается, что пользователь вводит оба параметра (4-значный PIN-код и 6-значный номер карты) по одной подсказке со стороны SRF.

Вход:		
SSD Max Number of Characters (Максимальное количество символов)	= 10	Число цифр номера карты и PIN-кода
SSD Min Number of Characters (Минимальное количество символов)	= 10	Число цифр номера карты и PIN-кода
SSD Format (Формат)	NNNNNN NNNN	10 десятичных цифр, от 0 до 9 каждая
CID Call Party Identifier (Идентификатор участника связи)	= вызывающая сторона	33-22-11
Выход:		
CID Identifier (Идентификатор)	= номер карты, PIN-код	654321, 1234
Логическое завершение	= проверка произведена	Переход к SCREEN

Блок SCREEN

Блок SCREEN проверяет право пользователя на доступ к услуге, то есть сопоставляет набранные номер карты и PIN-код со списком, находящимся в SDF. На входе блока параметр SSD List Indicator идентифицирует список, с которым должно производиться сопоставление. Параметр CID Identifier определяет тип идентифицируемых данных (номер карты, PIN-код).

Вход:		
SSD Screen List Indicator (Индикатор списка)	= имя списка с номерами карт и PIN-кодами	«Номера карт и PIN-коды»
CID Identifier (Идентификатор)	= номер карты, PIN-код	654321, 1234
Выход:		
CID Identifier (Идентификатор)	= номер карты, PIN-код	654321, 1234
Логическое завершение 1	= соответствие списку	Положительный результат - переход к блоку USER INTERACTION
Логическое завершение 2	= несоответствие списку	Отрицательный результат - переход к блоку RELEASE CALL

При положительном результате сопоставления (Match – номер карты найден и введен соответствующий ему PIN-код) обслуживание вызова будет продолжено в соответствии с логикой услуги ACC. Если же номер карты не обнаружен в соответствующем списке услуги ACC, или введенный PIN-код не соответствует данной расчетной карте (No Match), то происходит переход к блоку RELEASE CALL.

Блок RELEASE CALL

В случае несоответствия номера карты и PIN-кода, введенных пользователем, списку, имеющемуся в базе данных SDF, логика услуги переходит к выполнению блока RELEASE CALL. На входе этого блока параметр SSD Cause несет сведения о причине, по которой обслуживание вызова невозможно, и BCP должен прекратить работу.

Вход:		
SSD Cause (Причина)	= причина отказа в обслуживании	
Выход:		
Логическое завершение	= успешное	Возврат к BCP

После выполнения блока RELEASE CALL происходит возврат в блок BCP в точке POR Clear Call. Приостановленный базовый процесс обслуживания вызова возобновляется лишь для того, чтобы выполнить процедуру освобождения.

Блок USER INTERACTION

В случае совпадения набранных пользователем номера карты и PIN-кода со списком, имеющимся в SDF, логика услуги переходит к выполнению блока USER INTERACTION. В этой части алгоритма блок обеспечивает передачу пользователю речевого сообщения с предложением набрать номер адресата, а также прием этого номера.

Вход:		
SSD Announcement Parameters (Параметры речевого сообщения)	= идентификатор сообщения	«Введите номер вызываемого абонента»
SSD Collect Information Parameters (Параметры принимаемой информации)	= тип информации	DTMF
CID Call Party Identifier (Идентификатор участника связи)	= номер вызывающего пользователя	33-22-11
Выход:		
CID Collected data (Накопленные данные)	= номер адресата	11-22-33
Логическое завершение	= успешное	переход к CHARGE

Блоки CHARGE

Последовательно соединенные блоки CHARGE (тип 1 и тип 4) определяют особенности, связанные с начислением платы за связь с использованием расчетной телефонной карты. Блок применяется дважды: один раз – для передачи необходимой информации о параметрах записи о начисленной плате, второй раз – для начисления платы в реальном времени. На входе каждого блока параметр SSD Account состоит из двух частей, одна из которых (Number) содержит указатель поля CID с номером счета, на который должна быть отнесена начисленная плата, а вторая (Account%) определяет способ разделения этой платы между участниками связи. В нашем случае плата должна быть полностью отнесена на счет, указываемый параметром CID Account и соответствующий карте, с использованием которой производился вызов.

Вход:		
SSD Number of Accounts (количество счетов)	= 1	Плата всегда относится на один счет
SSD Account		
Number (номер счета)	= указатель поля данных CIDFP-Account	CID-Account
Account % (процент распределения платы между счетами)	= 100	Вся начисленная плата относится на счет, соответствующий карте
CID Account (номер счета)	= номер карты	654321
Выход:		
Логическое завершение	= успешное	Переход к CONNECT

Блок CONNECT

Блок CONNECT служит для передачи к ВСР информации, которая необходима, чтобы установить соединение между вызывающей и вызываемой сторонами. На входе этого блока параметр SSD CIDFP-Destination содержит указатель поля CID, где помещен физический номер вызываемой стороны.

Вход:		
SSD CIDFP-Destination	= указатель поля CID, где помещен физический номер адресата	CID Destination Number
SSD CIDFP-CLI	= указатель поля CID, где помещен номер вызывающей стороны	CID CLI
CID Destination Number (Номер адресата)	= номер вызываемой стороны	11-22-33
CID CLI (Идентификатор вызывающей стороны)	= номер вызывающей стороны	33-22-11
Выход:		
Логическое завершение (Success)	= успешное	Возврат в ВСР

Параметр SSD CIDFP-CLI содержит указатель поля CID, где помещен номер вызывающей стороны, то есть физический номер телефонной линии, от которой поступил обслуживаемый вызов.

Выполнение цепочки SIB (USER INTERACTION, VERIFY, SCREEN, TRANSLATE, CHARGE, CONNECT) завершается возвратом в блок BCP в точке POR Handle as Transit. Процесс обслуживания вызова возобновляется и производится далее уже на основании данных, полученных через эту точку POR. Вызов обрабатывается блоком BCP как только что поступивший, но с параметрами маршрутизации, полученными от блока CONNECT.

4.1.2.4 Информационные потоки и действия функциональных объектов

На рисунке 4.1.7 приведена диаграмма информационных потоков, возникающих при реализации услуги АСС посредством рассмотренной нами цепочки блоков SIB. Ниже описаны действия функциональных объектов в той последовательности, в которой они выполняются при предоставлении услуги.

О_BCSM приостанавливает обслуживание вызова после того, как будет встречена точка TDP Analysed_Info и проверка покажет, что назначенный для этой точки критерий удовлетворяется (в нашем случае критерием служат цифры 806).

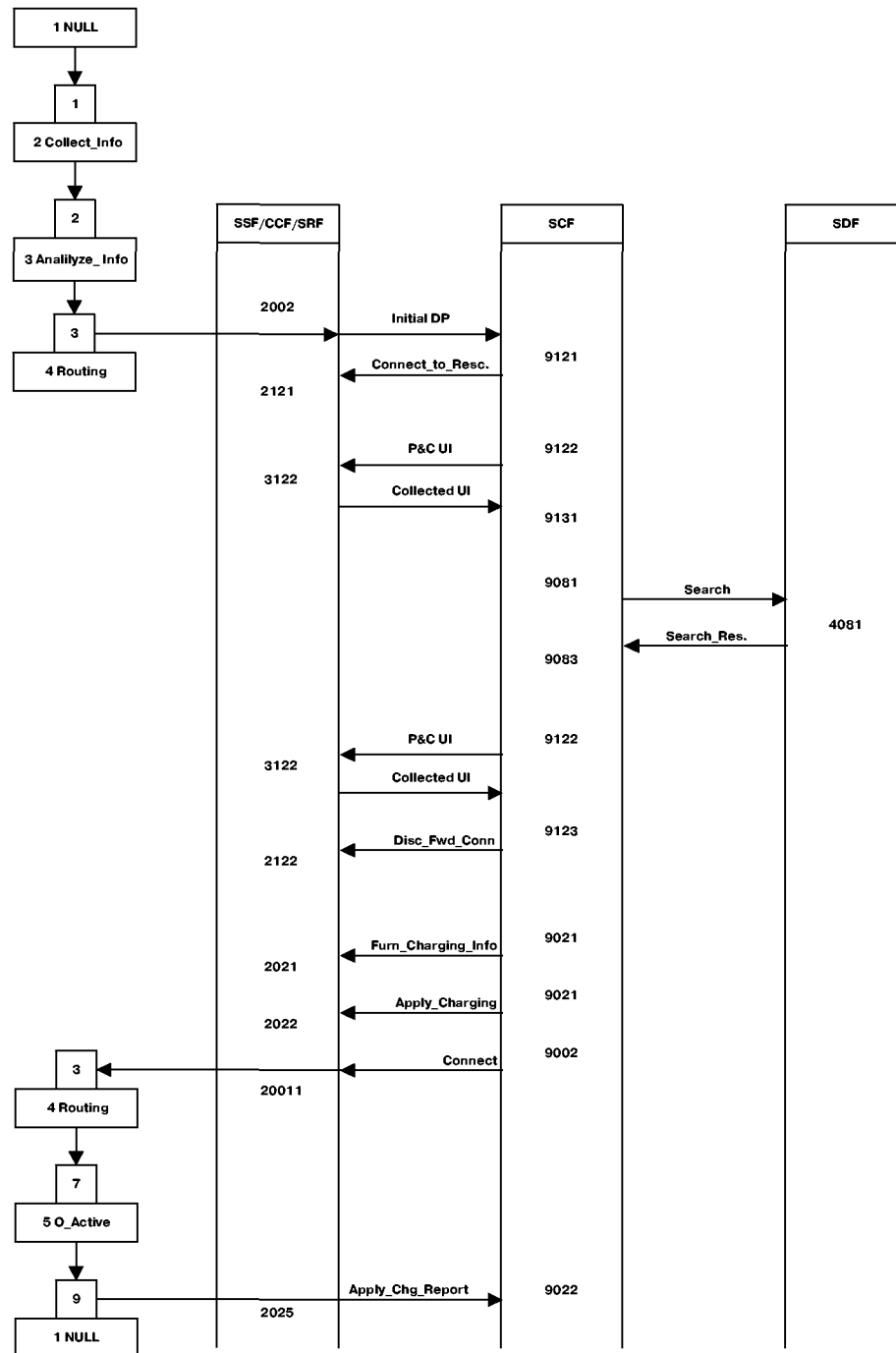
Стартовое событие	Выполняемые действия
2002 (SSF, BCP) Встречена триггерная точка обнаружения типа "запрос"	- прерывание базового процесса обслуживания вызова - передача информационного потока <i>InitialDP</i>
9121 (SCF, USER INTERACTION) Запрос инициировать процесс подключения SRF	- формирование и передача информационного потока <i>Connect To Resource</i>
2121 (SSF, USER INTERACTION) Получен запрос подключения SRF	- обработка информационного потока <i>Connect To Resource</i> - анализ полученной информации (идентификатор вызова, адрес пользователя)
Запрос установления соединения	- Установление соединения между пользователем и SRF
9122 (SCF, USER INTERACTION) Запрос передать приглашение к набору и принять информацию	- передача к SRF информационного потока <i>Prompt&Collect User Information</i>
3122 (SRF, USER INTERACTION) Запрос передать приглашение к набору цифр	- анализ операции <i>Prompt&CollectUserInformation</i> - передача пользователю приглашения к набору номера карты и PIN-кода
Запрос принять информацию	- прием информации от пользователя (номер карты, PIN-код) - формирование и передача к SCF операции <i>Collected User Information</i>

Стартовое событие	Выполняемые действия
9131 (SCF, VERIFY)	
Запрос проверки	- проверка номера карты и PIN-кода

Стартовое событие	Выполняемые действия
9081 (SCF, SCREEN)	
Запрос обработки принятых данных	- прием запроса обработки данных - создание и передача к SDF информационного потока <i>Search</i>
4081 (SDF, SCREEN)	
Прием информационного потока <i>Search</i>	- сопоставление номера карты и PIN-кода со списком - передача в SCF информационного потока <i>Search Result</i>
9083 (SCF, SCREEN)	
Прием информационного потока <i>Search Result</i>	- предоставление результата сопоставления (Match)

Стартовое событие	Выполняемые действия
9122 (SCF, USER INTERACTION)	
Запрос передать приглашение к набору и принять информацию	- передача к SRF информационного потока <i>Prompt&Collect User Information</i>
3122 (SRF, USER INTERACTION)	
Запрос передать приглашение к набору цифр	- анализ информационного потока <i>Prompt&Collect User Information</i> - передача пользователю приглашения к набору номера адресата
Запрос принять информацию	- прием от пользователя информации о вызываемом номере - формирование и передача к SCF информационного потока <i>Collected User Information</i>
9123 (SCF, USER INTERACTION)	
Запрос инициировать отключение SRF	- формирование и передача к SRF информационного потока <i>Disconnect Forward Connection</i>

Стартовое событие	Выполняемые действия
9021 (SCF, CHARGE тип 1)	
Запрос создать запись о начисляемой плате	- передача информационного потока <i>Furnish Charging Information</i>
2021 (SSF, CHARGE тип 1)	
Обработка запроса инициировать процесс начисления платы	- прием и анализ информационного потока <i>Furnish Charging Information</i> - создание записи о начисляемой плате



Примечание - показан только случай успешного соединения

Рис. 4.1.7 Диаграмма информационных потоков

Стартовое событие	Выполняемые действия
9002 (SCF, CONNECT) Запрос передать инструкции, необходимые для установления соединения	- передача к SSF информационного потока <i>Connect</i>
20011 (SSF, CONNECT) Запрос установления соединения	- установление соединения в соответствии с полученными от SCF инструкциями

Стартовое событие	Выполняемые действия
9021 (SCF, CHARGE тип 4) Запрос начисления платы в реальном времени	- передача к SCF информационного потока <i>Apply Charging</i> , содержащего допустимый лимит стоимости
2022 (SSF, CHARGE тип 4) Прием информационного потока <i>Apply Charging</i>	- начисление платы в реальном времени
2025 (SSF, CHARGE тип 4) Окончание связи или достижение допустимого лимита стоимости	- передача к SCF данных о текущей стоимости связи в информационном потоке <i>Apply Charging Report</i>
9022 (SCF, CHARGE тип 4) Прием информационного потока <i>Apply Charging Report</i>	- динамический контроль остатка на счете, соответствующем номеру карты

На рисунке 4.1.8 показана последовательность сообщений подсистемы TCAP ОКС-7 и переносимых в этих сообщениях операций INAP между SSP/IP и SCP при реализации услуги ACC.

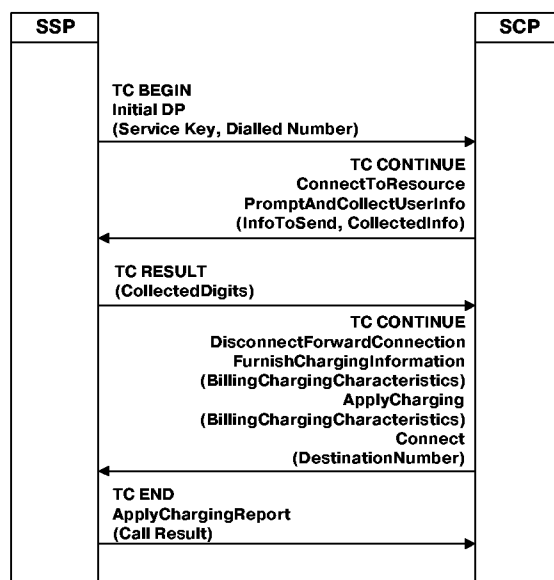


Рис. 4.1.8 Диаграмма обмена сообщениями TCAP/INAP для услуги ACC

4.1.3 Услуга «Premium Rate» (Информационная услуга за дополнительную плату)

Информационная услуга за дополнительную плату PREMIUM RATE позволяет организовать для абонентов ТФОП предоставление информации по интересующему их вопросу с доступом к ней по единому логическому номеру. Абоненту начисляется плата как за связь (междугородную или местную при наличии учета стоимости на местной сети), так и за информацию, предоставляемую поставщиком информации. Поставщик услуги PRM распределяет начисленную плату между оператором телефонной сети и поставщиком информационной услуги. В случае, когда информацией владеет поставщик услуги PRM, сумма делится между ним самим и оператором ТФОП.

Для организации услуги владелец информации (юридическая фирма, медицинское учреждение, справочное бюро и т.п.) заключает договор на приобретение у поставщика услуги логического номера, по которому с ним можно связаться, и на получение части платы за предоставленную им во время связи информацию. Вся сумма взимается с пользователя услугой (получателя информации) оператором телефонной сети, абонентом которой является получатель информации. Расчет с конечным пользователем осуществляется по получении оператором телефонной сети сводки счетов от поставщика услуги PRM, содержащей две составляющие: плату за связь через телефонную сеть по обычному тарифу и доплату за информационную услугу. Схема предоставления услуги PRM приведена на рисунке 4.1.9.

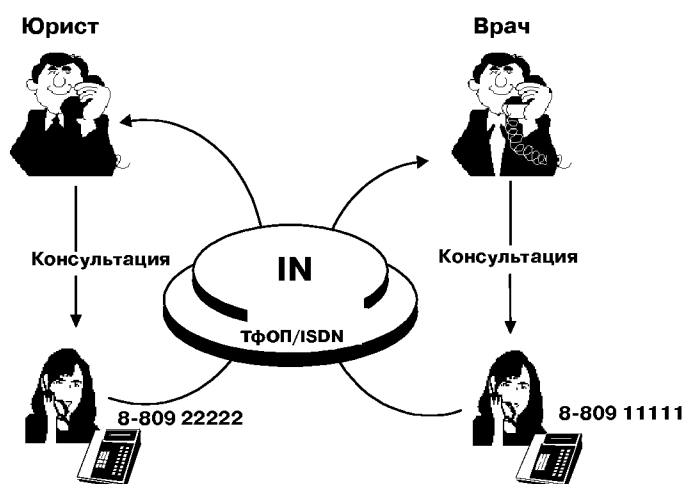


Рис. 4.1.9 Схема предоставления услуги PRM

В телефонных сетях России вызов услуги PRM осуществляется набором номера Пн-809- $X_1X_2\dots X_n$. Несмотря на принципиальное отличие (с точки зрения пользователя) услуги PRM от услуги FPH, спецификации услуг PRM и FPH практически аналогичны, за исключением принципа начисления платы.

Для реализации услуги обязательными являются атрибуты One Number (ONE) и Premium Charging (PRMC) – атрибут, позволяющий причислить к стоимости связи добавку за полученную информацию. Перечень дополнительных атрибутов практически совпадает с перечнем, определенным для услуги FPH, за исключением того, что для PRM отсутствуют в качестве разрешенных атрибуты Authentication и Mass Calling.

4.1.4 Услуга «Televoting» (Телеголосование)

Услуга TELEVOTING позволяет проводить опрос общественного мнения с использованием ресурсов телефонной сети. В соответствии с числом вариантов предлагаемых ответов назначаются логические номера, а участники голосования набирают тот номер, который соответствует их мнению. Один из каждых N вызовов может быть направлен в студию, откуда ведется трансляция теле- (или радио-) программы, для проведения диалога голосующего пользователя с ведущим. Средствами IN производится подсчет числа вызовов, поступивших на каждый номер, и предоставление администратору и абоненту услуги доступа к результатам опроса. Голосование может быть премиальным. В этом случае один или несколько абонентов, первыми правильно ответивших на вопрос, получают приз или денежное вознаграждение. В телефонных сетях России вызов услуги VOT осуществляется набором номера Пн-803- $X_1X_2\dots X_n$. Схема предоставления услуги VOT приведена на рисунке 4.1.10.

Для реализации услуги TELEVOTING обязательным является атрибут Mass Calling (MAS) – обработка большого числа сконцентрированных во времени вызовов, возникающих в процессе проведения игр, викторин или изучения общественного мнения с помощью телефонной сети. Атрибут реализуется посредством процедуры, подсчитывающей количество вызовов на каждый номер, просеянных в SSP. Основой для спецификации услуги служит блок LIMIT и поддерживающие его операции INAP *ActivateServiceFiltering* и *ServiceFilteringResponse*. В таблице 4.1.3 приводится полный перечень обязательных и разрешенных для услуги VOT дополнительных атрибутов.

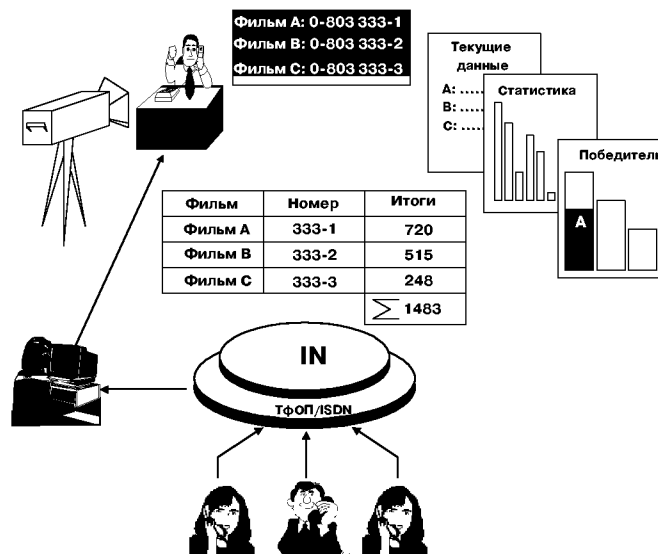


Рис. 4.1.10 Схема предоставления услуги VOT

Таблица 4.1.3

Обязательные атрибуты:	Дополнительные атрибуты:
MAS - Mass Calling (Массовые вызовы)	CD - Call Distribution (Распределение вызовов)
	CFC - Call Forwarding on Busy/Don't answer (Переадресация при занятости или при отсутствии ответа)
	GAP - Call Gapping (Прореживание потока вызовов)
	LIM - Call Limiter (Ограничение количества вызовов)
	LOG - Call Logging (Регистрация вызовов)
	QUE - Call Queueing (Установка вызова на ожидание)
	CPM - Customer Profile Management (Управление профилем обслуживания)
	CRA - Customised Recorder Announcement (Воспроизведение записанного сообщения).
	ODR - Origin Dependent Routing (Маршрутизация в зависимости от местоположения вызывающего пользователя)
	OCS - Originating Call Screening (Просеивание вызовов по признаку места их возникновения)
	OUP - Originating User Prompter (Подсказка вызывающему абоненту)
	PRMC - Premium Charging (Дополнительная плата)
	REVC - Reverse Charging (Начисление платы вызываемому абоненту)
	SPLC - Split Charging (Разделение платы)
	TDR - Time Dependent Routing (Маршрутизация в зависимости от времени)

Перед началом процедуры SCP передает ко всем SSP операцию *INAP ActivateServiceFiltering*, которая определяет время начала и окончания опроса, критерий просеивания и тип обслуживания, применяемого к отсеянным и просеянным вызовам. Во время действия операции фильтрации соединения, соответствующие вызовам на определенные для голосования номера, устанавливаются только до SSP, где производится подсчет вызовов на каждый из номеров. Через заданные промежутки времени SSP посылают текущие результаты опроса в SCP при помощи операций *INAP ServiceFilteringResponse*. SCP собирает и накапливает данные от SSP и посылает их к SMP, куда заказавший опрос абонент может обратиться за получением отчета о результатах голосования.

Услуга VOT предоставляется абоненту на определенный период времени по договоренности с поставщиком услуги. Для проведения голосования администратор услуги определяет необходимые параметры в соответствии с ее профилем. Среди основных параметров услуги VOT, определяемых администратором, можно выделить число логических номеров, доля вызовов, подлежащих просеиванию, периодичность передачи данных в SCP и временной интервал между сообщениями, посылаемыми из SCP в SMS.

Глава 4.2

Рынок услуг и оборудования

4.2.1 Рыночные аспекты внедрения IN

После отделения логики услуг от функций коммутации, произошедшего с внедрением IN, прогнозировалось, что поставщики коммутационного оборудования потеряют лидирующее положение на рынке. Прогнозы основывались на том факте, что за последние годы определенную долю телекоммуникационного рынка завоевали новые его участники – поставщики продукции информационной промышленности (программного и аппаратного обеспечения).

Однако анализ показывает, что никаких существенных изменений на телекоммуникационном рынке в связи с внедрением IN не наблюдается. Поставщики коммутационного оборудования сохраняют лидирующее положение на рынке IN-продукции, как и на телекоммуникационном рынке в целом. Фирмы, занимающиеся изготовлением компьютерных систем, и системные интеграторы начинают поставку независимых IN-платформ, но делают это, главным образом, через альянсы, а не посредством прямой конкуренции. Несмотря на то, что операторы сетей связи часто призывают поставщиков информационного оборудования и программного обеспечения участвовать в тендерах, контракты, как правило, достаются поставщикам коммутационного оборудования.

Связано это, прежде всего, с тем, что основное влияние на процесс стандартизации сетевых элементов и протоколов связи оказывают именно поставщики коммутационного оборудования. Во-вторых, до сих пор недостаточно стандартизированы системы эксплуатационного управления услугами и их генерации; поставляемые сегодня системы этого назначения могут совмещаться только с платформой того же поставщика. Данное обстоятельство серьезно препятствует поставке приложений третьими фирмами. Кроме того, техническая реализация IN-платформ может оказаться достаточно сложной. Операторы предпочитают пользоваться услугами одного и того же поставщика при закупке коммутационного оборудования, платформы IN, систем эксплуатационного управления услугами и систем их создания, поскольку это избавляет от проблем, связанных со стыковкой.

Хотя ситуация меняется медленно, тем не менее, на рынке появляется продукция независимых поставщиков, поддерживающая стандарты CS-1, AIN0.1 и AIN0.2, которые позволяют обеспечить совместимость оборудования различных поставщиков. В целом же,

для операторских компаний существуют три основных критерия выбора поставщика IN: поддержка необходимых стандартов, наличие требуемой функциональности и наличие механизма создания новых услуг.

По данным консалтинговой фирмы Frost & Sullivan, в 1996 г. доход от реализации оборудования и услуг IN в США составил 26,4 млрд. долларов. Предполагалось, что к концу 2000 г. этот показатель должен достигнуть 41,6 млрд. долл. Таким образом, среднегодовой прирост доходов в 1996-2003 гг. должен составить 11%. Европейский рынок оборудования IN в 1999 г. составил 1,42 млрд. долларов. Среднегодовые темпы роста объема рынка в 1994-1999 гг. варьируются в пределах 20-30%.

Компания Ovum опубликовала прогноз рынка трех основных доходообразующих услуг IN – FPH, PRM и VPN в тех регионах, где рынок этих услуг наиболее развит (см. рис. 4.2.1). К 2003 г. суммарный объем рынка составит свыше 30 млрд. долл. США будут оставаться лидером в развитии услуг IN, на их долю придется больше доходов от этих услуг, чем на все остальные страны вместе. Однако доля США в общем объеме рынка в 2003 г., по сравнению с 1997 г., будет существенно ниже.

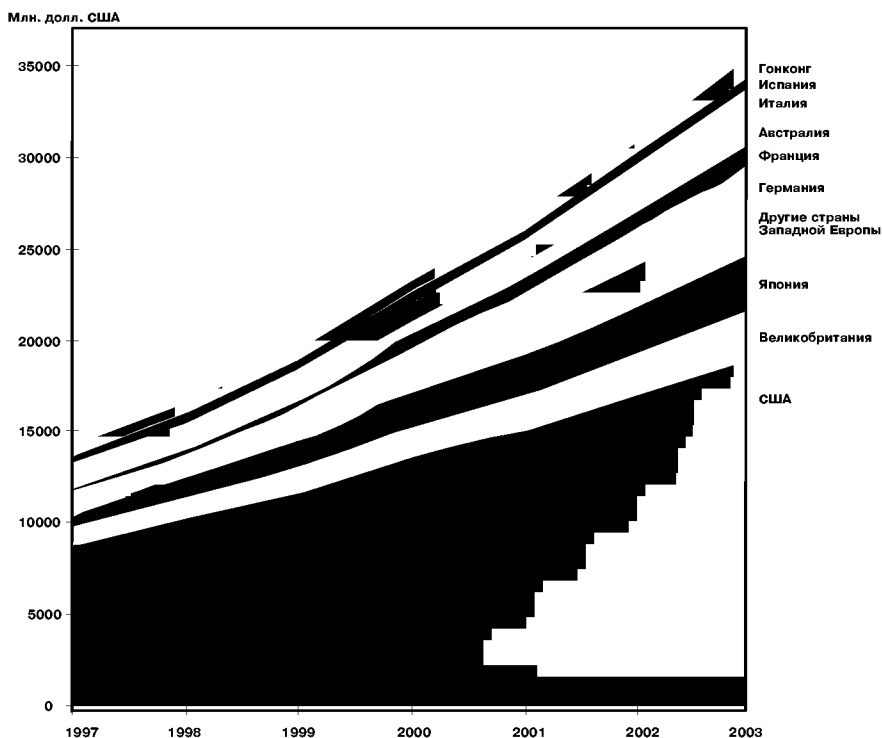


Рис. 4.2.1 Прогноз доходов от отдельных услуг IN

В целом темп роста услуг будет выше в тех странах, где сегодня эти услуги менее развиты. В Азиатско-Тихоокеанском регионе темпы роста будут ниже, чем в странах Западной Европы.

Что касается отдельных услуг, то темпы роста рынка услуги FPN будут очень высокими во всех странах, за исключением США, где объем рынка, по-видимому, уже приближается к пределу. Наивысшими темпами развития будет характеризоваться рынок услуг VPN. В наиболее развитых странах в 2003 г. на виртуальные сети придется около 15% всех вызовов, исходящих от абонентов делового сектора.

Среди факторов, способствующих развитию сетей VPN, следует отметить в первую очередь дерегулирование национальных телекоммуникационных рынков и стремление операторов полностью использовать открывающиеся возможности продвижения на эти рынки. VPN предоставляет им идеальную возможность предложить пользователям новые услуги и завоевать плацдарм на уже либерализованных рынках.

Кроме того, продолжается разрушение барьеров международной торговли, что ведет к глобализации деятельности крупных корпораций, которые, в свою очередь, для сохранения конкурентоспособности нуждаются в высокоскоростной, надежной и эффективной связи. В ведении бизнеса наблюдается переход от традиционной модели «продажа-заказ-покупка» к таким методам как «телепродажа» и «online-закупки», что тоже способствует внедрению виртуальных частных сетей.

С 1999 г. темпы роста доходов от услуг IN увеличились за счет значительного роста трафика, создаваемого мобильными абонентами. Рынок услуги PRM сохраняет известную стабильность, поэтому прирост доходов от абонентов квартирного сектора (которые, в основном, пользуются именно этой услугой) будет невелик. В то же время, прирост рынка услуг VPN приведет к росту доходов, получаемых от абонентов делового сектора.

Таким образом, поставщики IN-продукции в Европе, где рынок начал формироваться с более низкой стадии развития IN, находятся в лучшем положении, чем в США, поскольку американские фирмы-поставщики и операторы сфокусировали свою деятельность на получении доходов от инвестиций в продукцию стандарта AIN0.1.

В следующих разделах описано оборудование для реализации IN на примере продукции двух компаний – американской Lucent Technologies и европейской Alcatel. Основанием для выбора именно этих компаний послужило следующее. Обе компании являются гигантами телекоммуникационного рынка и поставляют полный спектр автономного оборудования для построения IN. Lucent Technologies

(бывшая AT&T), будучи первой в мире компанией, претворившей в жизнь идею IN, до сих пор сохраняет лидирующие позиции на рынке оборудования IN в США. Компания же Alcatel занимает первое место по количеству линий, установленных в России.

4.2.2 Платформа IN компании Lucent Technologies

Компания Lucent Technologies поставляет полный спектр оборудования, определенного рекомендациями ITU-T и стандартами ETSI для создания сетей IN – SSP, SCP, IP, SMP, SCEP. Связи между элементами платформы IN показаны на рисунке 4.2.2. Оборудование Lucent Technologies использовано для построения сетей IN в Северной Америке, Великобритании, Испании, Италии и др. странах.

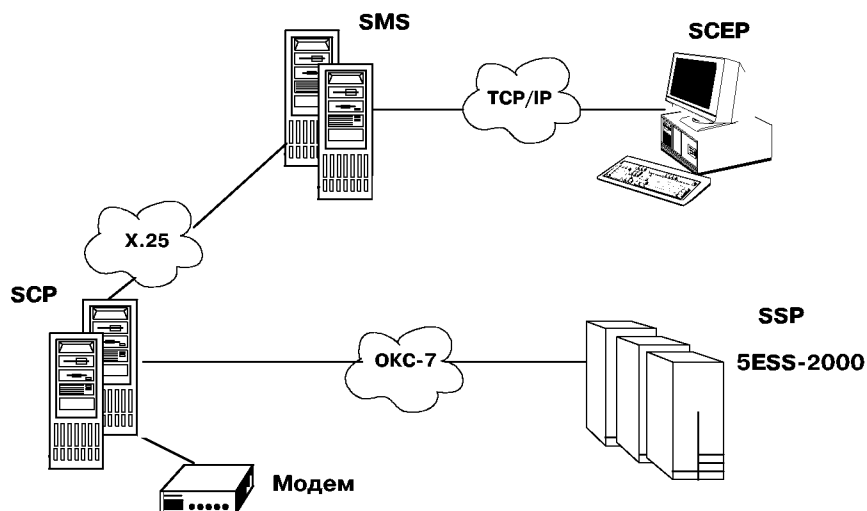


Рис. 4.2.2 Платформа IN компании Lucent Technologies

4.2.2.1 SSP на базе 5ESS-2000

Узел коммутации услуг SSP компании Lucent Technologies строится на базе коммутационной системы 5ESS-2000 и обеспечивает функции управления связью пользователя CCF и функции коммутации услуг SSF, а также поддержку интерфейса с автономными SCP в соответствии с протоколом INAP. Как опция, 5ESS-2000 может работать и в качестве узла доступа NAP, лишь выявляя вызовы, требующие обращения к услугам IN, чтобы направлять их к SSP.

Возможность работы 5ESS-2000 в качестве SSP реализована как программное приложение, работающее на стандартной программно-аппаратной платформе коммутационной системы, и не требует оснащения ее дополнительными аппаратными средствами. Систе-

ма 5ESS-2000 состоит из одного или нескольких концентраторов SM-2000, модуля связи CM и административного модуля AM (см. рис. 4.2.3).

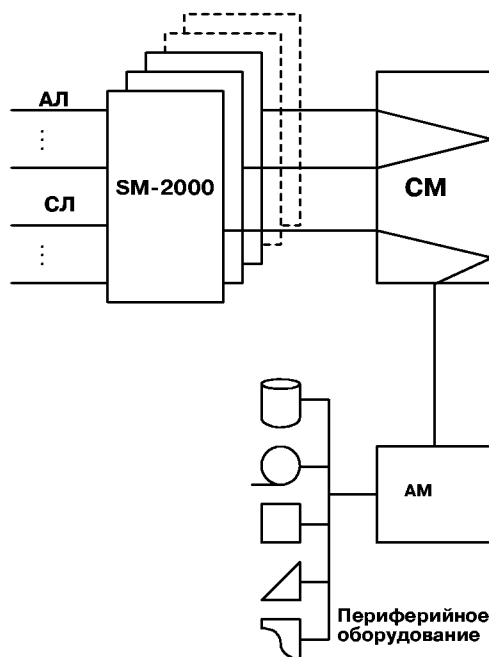


Рис. 4.2.3 Архитектура станции 5ESS-2000

Стандартные функции управления связью пользователя CCF расширены функциями SSF, что позволяет SSP 5ESS-2000 выявлять вызовы, требующие услуг IN, и приостанавливать процесс их обработки в специфицированных ITU-T точках обнаружения до получения инструкций со стороны SCP.

4.2.2.2 SCP платформы IN Lucent Technologies

Первая версия узла управления услугами, разработанная в 1981 г. компанией Lucent Technologies (в то время AT&T), называлась 1NCP. Следующая версия – SCP-II, как, впрочем, и 1NCP, до сих пор используется во многих странах мира в качестве централизованной базы данных для сетевых услуг. Версия узла управления услугами поставлялась для сетей AIN в Северной Америке под названием A-I-Net SCP-2000 с 1992 г. Предлагаемый сегодня вариант называется просто SCP; мы будем далее называть это оборудование SCP-2000, по аналогии с названием коммутационной системы 5ESS-2000.

SCP-2000 позволяет объединять услуги в индивидуальные пакеты услуг – SP (Service Package). Обычно такой пакет содержит логику

и набор данных, необходимых для реализации одной услуги, однако SP позволяет объединять пакеты нескольких взаимодействующих услуг в один программный модуль, который может исполняться в SCP-2000 независимо от других пакетов услуг. Таким образом, добавление новых SP не затрагивает ранее установленных и, более того, позволяет поддерживать несколько вариантов реализации одной и той же услуги в одном SCP-2000.

SCP-2000 (см. рис. 4.2.4) состоит из двух управляющих серверов CC (Control computer) – рабочего и резервного – и из четырех серверов связи. С целью повышения надежности используется специальный модуль – RCC, производящий наблюдение за конфигурацией аппаратных средств, пуск системы и переключение между рабочим и резервным серверами. CC выполнен на базе промышленного мультипроцессорного варианта Pentium Pro, оснащенного оперативной памятью объемом от 512 Мбайт в минимальной конфигурации до 2 Гбайт в максимальной.

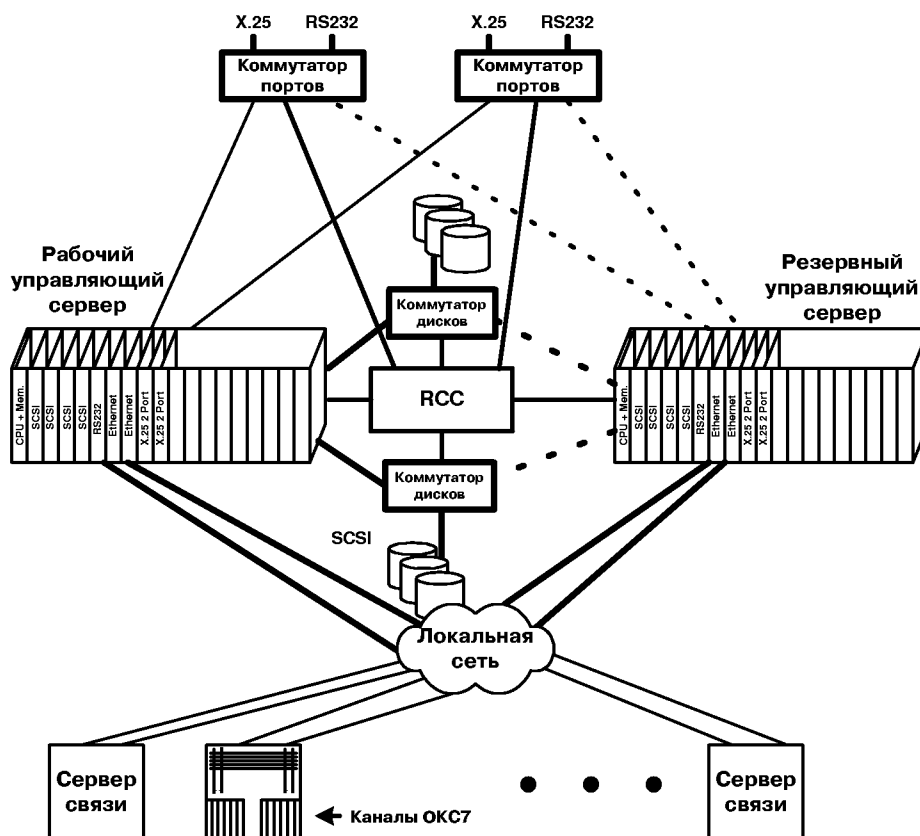


Рис. 4.2.4 Архитектура SCP Lucent Technologies

Высоконадежная дисковая подсистема управляющих серверов состоит из двух отдельных шин стандарта SCSI (Small computer system interface), подсоединенных к рабочему серверу через специализированный коммутатор 2X2. Одна шина поддерживает пять дисковых накопителей емкостью по 4 Гбайта, при этом каждый накопитель обеспечен резервным на другой шине. Две шины SCSI, подключенные к резервному управляющему серверу, поддерживают шестую дисковую пару. Общий доступный объем дискового пространства – от 48 до 96 Гбайт.

Взаимодействие между серверами СС и сетью ОКС-7 входит в функции серверов связи, каждый из которых также реализован на базе Pentium Pro, оснащенного памятью 128 Мбайт. Один сервер связи обслуживает до 8 звеньев ОКС-7.

Программное обеспечение SCP-2000 построено по модульному принципу (см. рис. 4.2.5). Модули прикладных программ пакетов услуг SPA (Service package applications) представляют собой программные процессы, исполняемые в среде операционной системы UNIX. Базой для работы SPA служит программное обеспечение обработки вызова. Прикладные программы пакетов услуг реализуют специфическую для услуги логику, а также эксплуатационные функции управления выделяемыми для каждого вызова ресурсами, сбора статистических данных о функционировании и т.п.

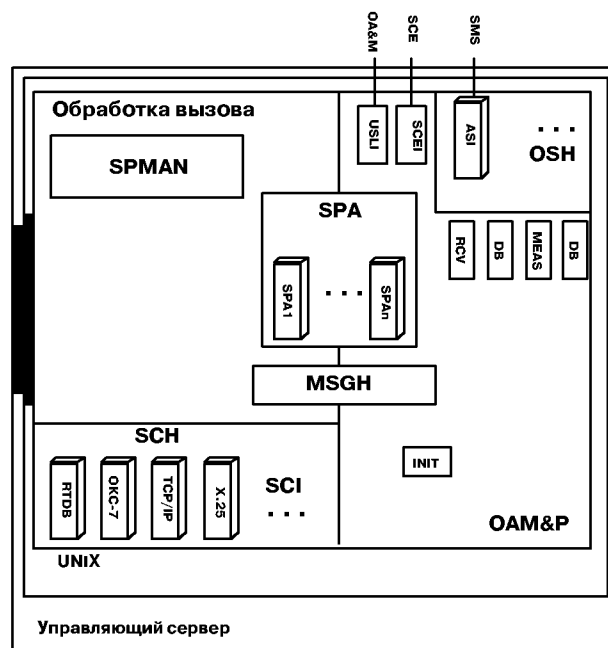


Рис. 4.2.5 Архитектура программного обеспечения SCP Lucent Technologies

Подсистема менеджера пакетов услуг SPMAN управляет запуском, остановкой и проверкой SPA; направляет вызовы к нужному SPA; реализует механизмы контроля перегрузки системы. Подсистема реализует также разделяемый доступ прикладных программ к таблицам, содержащим данные об абонентах.

Межпроцессорная связь обеспечивается модулем обработки сообщений MSGH. Интерфейсы с внешними устройствами поддерживаются драйверами операционной системы. Прикладные программы получают доступ к драйверам через контроллеры внешних устройств SCH, которые управляют разными устройствами через общий интерфейс SCI. Интерфейс состоит из резервированных каналов Ethernet и асинхронных каналов эксплуатационного управления. Контроллеры поддерживают внешние устройства, реализующие связь с внешним миром по протоколам OKC-7, X.25, TCP/IP и RTDB (для взаимодействия с базами данных). Единая структура контроллеров обеспечивается стандартным программным интерфейсом с каждым из них.

Программные модули эксплуатационного управления OAM&P необходимы для реконфигурации прикладных программ пакетов услуг, а также для администрирования всего SCP-2000. Ниже приведено описание функций основных модулей подсистемы OAM&P.

В функции менеджера баз данных DB входит работа с данными, относящимися к процессу обслуживания вызова, с конфигурационной информацией, с результатами изменений. Эти действия производятся при помощи системы управления реляционной базой данных ORACLE. Отчеты о функционировании генерируются посредством стандартного языка обращения к базам данных SQL.

Поступающая со стороны процессов обслуживания вызова информация о событиях, происходящих внутри SPA, обрабатывается модулем графа решений DG.

Иницирующий модуль INIT наблюдает за системными ресурсами и, в случае сбоя, производит необходимый выбор и переключение на резерв. В качестве ресурсов выступают дисковая подсистема, коммутаторы портов и каналы Ethernet. Подсистема измерений MEAS через заданные промежутки времени производит сбор данных о функционировании со стороны контроллеров внешних устройств и пакетов SP. Связь SCP-2000 с внешними системами эксплуатационного управления обеспечивается соответствующими программными модулями OSH.

Пользователь обновляет информацию в базе данных путем заполнения экранных форм. Ответственный за это модуль RCV скрывает от пользователя внутреннюю структуру данных и следит при этом за

сохранением целостности базы. Подсистема базы данных реального времени RTDB содержит миллионы записей и обеспечивает синхронизацию (также в реальном времени) между основной и резервной частями.

SCP-2000 имеет два уровня программируемости со стороны пользователя, реализуемых в опционально поставляемом самостоятельном оборудовании SCE: программируемость услуг и программируемость платформы.

Интерфейсные модули USLI, ASI, SCEI поддерживают связи SCP-2000, необходимые для вывода информации о текущих состояниях и ввода команд управления аппаратными средствами, для взаимодействия обслуживающего персонала с системой эксплуатационного управления, а также связь с внешними SMS и SCE.

Для взаимодействия с сетью OKC-7 серверы связи поддерживают прикладные протоколы в соответствии с несколькими стандартами, в том числе, AIN и ETSI CS-1 INAP, причем один SCP-2000 может одновременно поддерживать несколько разных протоколов. Связь между SCP-2000 и SMS реализована по протоколу X.25.

Производительность SCP-2000 зависит от сложности поддерживаемых им услуг и от параметров обслуживаемой нагрузки. Основными ограничивающими ресурсами, разделяемыми всеми услугами, являются процессорное время, оперативная и дисковая память, звенья OKC-7. В ряде случаев ограничивающим фактором может быть также количество пользователей.

4.2.2.3 SMP платформы IN Lucent Technologies

Для эксплуатационного управления услугами, развернутыми в SCP-2000, компания Lucent Technologies предлагает специализированный программно-аппаратный комплекс SMS. Прикладное программное обеспечение SMS обеспечивает платформу IN функциями введения, проверки, распределения и слежения в отношении глобальных данных и данных, специфических для абонентов. Эти функции одинаково применимы к разным услугам и к их версиям. Графический интерфейс пользователя основан на операционной системе Windows95, а доступ к внутренним базам данных SMS осуществляется через прикладной программный интерфейс API.

Программное обеспечение SMS устанавливается на стандартной аппаратной платформе Hewlett Packard (HP), оснащенной процессорами типа K-Class или T-Class. Для повышения надежности может быть использована дублированная конфигурация с рабочей и резервной частями (см. рис. 4.2.6) Соединение SMS с SCP-2000 осуществляется по дублированным каналам, работающим по протоколу X.25, через интерфейсные платы платформы HP.

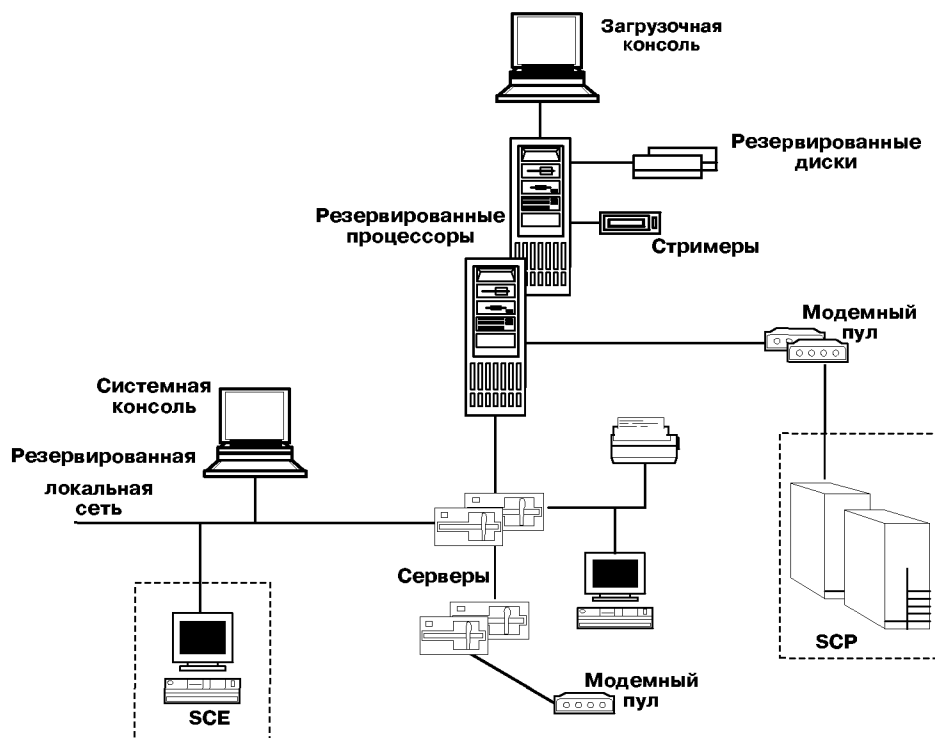


Рис. 4.2.6 Архитектура SMS Lucent Technologies

Высокоскоростной доступ к системе со стороны терминалов пользователя осуществляется через локальную сеть Ethernet или через маршрутизаторы. Возможен также дистанционный доступ через телефонную сеть, для этой цели предназначен модемный пул.

Объем оперативной и дисковой памяти зависит от числа пользователей системы и количества сетевых элементов. В случае необходимости может быть установлена дополнительная память. Резервная копия последней версии программного обеспечения записывается на накопитель на магнитной ленте и может быть использована для восстановления текущих данных при сбое системы.

Программное обеспечение SMS делится на четыре части. На нижнем уровне находится операционная система UNIX System V. Выше расположен уровень платформы реального времени, содержащий программные средства, общие для многих приложений. Этот уровень управляет реляционной базой данных, поддерживает протоколы связи с терминалами пользователей системы и с другими внешними системами, запускает и останавливает отдельные приложения и программные процессы, сохраняет резервную копию и восстанавливает текущие данные, управляет конфигурационной базой данных и распределяет запросы пользователей.

Над платформой реального времени расположен уровень поддержки прикладных программ, который включает в себя функции защиты доступа к системе, назначения привилегий доступа к услугам и абонентским данным, создания новых форм отчетов для абонентов, приспособления логики услуг к индивидуальным требованиям абонентов, распределения данных по сетевым элементам, проверки совместимости данных перед их загрузкой в SCP, инсталляции получаемых от SCE программ логики новых услуг.

Высшим уровнем программного обеспечения является прикладной уровень, обеспечивающий администрирование пользователей системы, доступ пользователей к системе посредством графического интерфейса, а также интерфейс API.

4.2.2.4 SCE платформы IN Lucent Technologies

Оборудование для создания услуг SCE является самостоятельной, но опциональной частью платформы IN компании Lucent Technologies. Отличительными его особенностями являются возможность использования при создании услуг нескольких специализированных языков, ориентированных на поддержку прикладных программ, наличие нескольких уровней программирования сетевых элементов, минимизация взаимодействия услуг за счет инкапсуляции совместно используемых услуг в индивидуальные пакеты SP, автоматическое создание файлов эксплуатационного управления, необходимых для новой услуги, простота создания новых блоков SIB.

Компоненты программного обеспечения созданной в SCE услуги загружаются в SMS и в SCP, причем в SMS направляются схемы организации баз данных, а в SCP – еще и программы логики исполнения. Загрузка может быть произведена как с ленты, так и дистанционно по протоколу TCP/IP. Полученные данные используются в SMS для создания таблиц подстройки индивидуальных параметров под абонентов услуги. В SCP услуги инсталлируются либо с терминала технической эксплуатации, либо дистанционно из SMS по протоколу X.25.

Аппаратная часть SCE (см. рис. 4.2.7) состоит из файлового сервера SUN Solaris Sparc, рабочих станций разработчиков услуг Sun Sparc и компилирующих серверов SUN Solaris Pentium, ресурсы которых в разделяемом режиме доступны любой рабочей станции через локальную сеть Ethernet.

Файловый сервер содержит диск емкостью 5 Гбайт и представляет собой основную часть SCE, включающую в себя интерфейс пользователя, базу данных ORACLE, пароли пользователей и пакеты услуг SP. Компилирующий сервер содержит средства кросс-компиляции и компилятор SUN C++, библиотеки и специализированную среду разработки приложений. Загрузка программного обеспечения в серверы производится с накопителя на магнитной ленте.

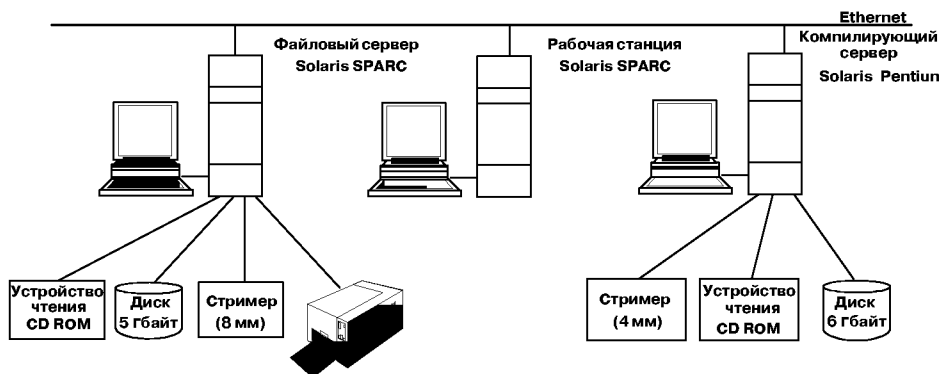


Рис. 4.2.7 Архитектура SCE Lucent Technologies

Программное обеспечение SCE используется для создания SPA в виде логики услуги и сопутствующих данных. Часть логики услуги создается с использованием языка программирования логики услуг SLL, часть – посредством языка графов решений DGL. Язык SLL похож на языки программирования PASCAL или BASIC, отличаясь от них такими дополнительными свойствами как встроенные автоматы конечных состояний, асинхронные вызовы функций и типы данных, специфические для телекоммуникационных протоколов.

В отличие от текстового языка SLL, DGL является языком графического программирования. Программы на нем конструируются из уже готовых для использования блоков SIB. Каждый SIB реализует операцию, которая может иметь несколько результатов. Входные параметры блоков, доступных разработчику, могут быть описаны на языке SLL, который, кстати, может использоваться и для создания новых SIB.

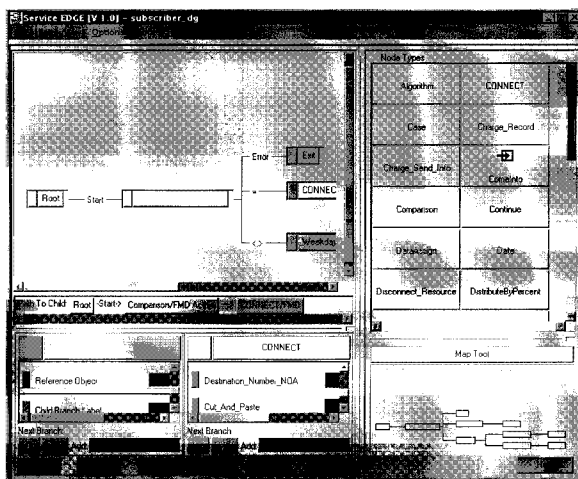


Рис. 4.2.8 Интерфейс графического редактора Lucent Technologies

Большинство функций SCE доступно пользователю через графический интерфейс вида OPEN LOOK, используемый на рабочих станциях SUN. Так, менеджер графов решений EDGE позволяет создавать и редактировать DG. Система EDGE работает в двух режимах – в режиме редактора и в режиме симулятора. При редактировании разработчик выбирает нужные SIB, соединяет их определенным образом, задает входные параметры. Экранная форма режима редактирования приведена на рисунке 4.2.8. При редактировании интерактивно происходит проверка корректности вводимых параметров. В режиме симулятора предоставляется возможность исполнения цепочки SIB как с ее начала, так и с любого другого места.

4.2.3 Платформа IN компании Alcatel

Конфигурация программно-аппаратных комплексов платформы IN компании Alcatel приведена на рисунке 4.2.9. Основными элементами архитектуры IN Alcatel, поставляемыми в настоящее время, являются SSP Alcatel 1000 S12, SCP Alcatel 1425 или 1420, SMP Alcatel 1435 или 1430, SCE Alcatel 1452. Оборудование Alcatel в разное время (начиная с 1990 г.) было использовано при построении сетей IN в Бельгии, Бразилии, Германии, Франции и других странах.

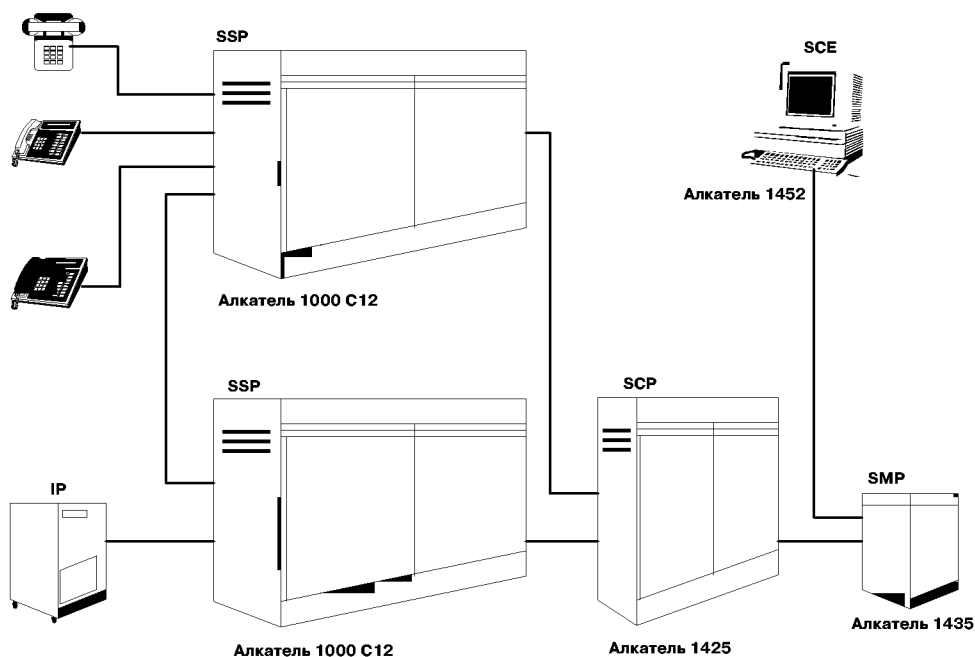


Рис. 4.2.9 Платформа IN компании Alcatel

4.2.3.1 SSP на базе Alcatel 1000 S12

Узел коммутации услуг в платформе IN компании Alcatel реализован на базе коммутационной системы Alcatel 1000 S12 посредством дооснащения ее дополнительным аппаратным модулем с сопутствующим программным обеспечением (см. рис. 4.2.10). Дооснащение осуществляется довольно просто и без перерыва в обслуживании, что является следствием открытой архитектуры коммутационной системы и полностью распределенного управления.

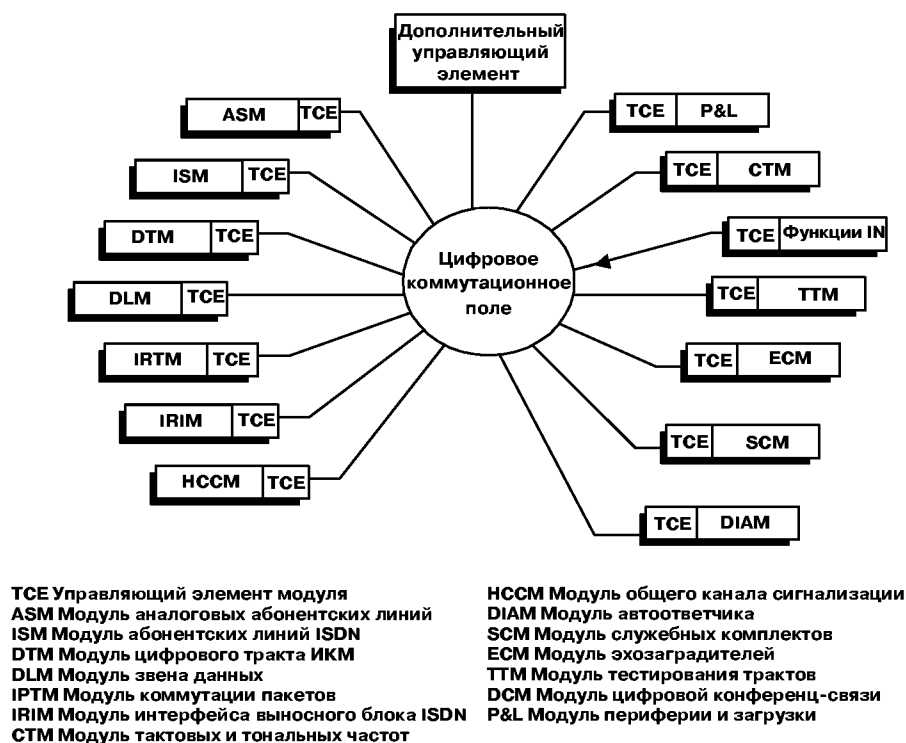


Рис. 4.2.10 Архитектура Alcatel 1000 S12

4.2.3.2 SCP платформы IN Alcatel

Компания Alcatel поставляет оборудование SCP двух типов – Alcatel 1420 и Alcatel 1425. Оборудование первого типа предназначено для корпоративных сетей и имеет специфические программно-аппаратные интерфейсы, в то время как Alcatel 1425 применяется для работы в сетях общего пользования в качестве узла управления услугами в составе открытой платформы и имеет стандартные интерфейсы.

SCP Alcatel 1425 представляет собой распределенную многопроцессорную программно-аппаратную платформу (см. рис. 4.2.11), состоящую из процессоров переднего плана FEP и процессоров зад-

него плана ВЕР, объединенных локальной сетью Ethernet. Системные данные хранятся на накопителях на жестких дисках, подключенных к шине стандарта SCSI. Для загрузки системы используется CD-ROM. Объем памяти и число модулей FEP и ВЕР определяются обслуживаемой нагрузкой.

К процессорам переднего плана подключаются первичные тракты ИКМ, в которых организованы каналы ОКС-7. Процессоры FEP реализуют подсистемы MTP, SCCP и TCAP. Процессоры ВЕР работают в режиме разделения нагрузки, содержат функциональные объекты SCF и SDF и реализуют протокол INAP.

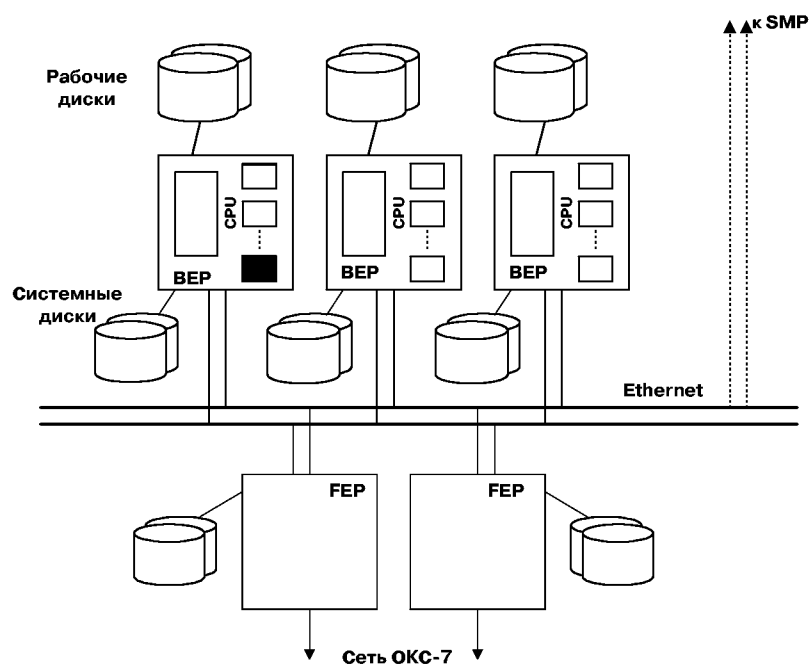


Рис. 4.2.11 Структура SCP Alcatel

Процессорный блок ВЕР выполнен на базе сервера Alfa 4100 компании DEC и содержит до 4 процессоров типа Alfa Processor с общим объемом оперативной памяти 8 Гбайт, объединенных 128-битной системной шиной со скоростью передачи 1.1 Гбайт/с. Отдельный модуль преобразует системную шину в 64-битовую шину стандарта PCI, к которой подключены внешние устройства (накопители, порты и контроллеры шин SCSI и EISA, а также контроллеры локальной сети).

Структура программного обеспечения SCP Alcatel 1425 приведена на рисунке 4.2.12. В его основе лежит многозадачная операционная система UNIX. Ядро программного обеспечения образуют операционная система, пользовательский интерфейс GUI, система управле-

ния файлами FMS и стеки протоколов ОКС-7 (для связи с SSP), X.25 и TCP/IP (для связи с SMS). Над ядром находятся система управления реляционной базой данных ORACLE, которая содержит административные, постоянные и динамические данные, а также интерпретатор логики услуг SLI. Самый верхний уровень программного обеспечения образуют программы реализации логики услуг SLP.

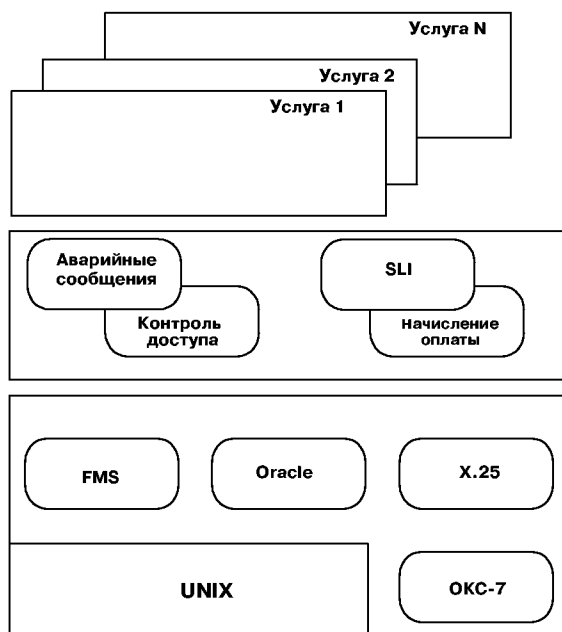


Рис. 4.2.12 Структура программного обеспечения SCP Alcatel

4.2.3.3 SMP платформы IN Alcatel

Система SMP Alcatel 1435 (или 1430 – для корпоративного варианта) предназначена для эксплуатационного управления услугами, предоставляемыми платформой IN. Программно-аппаратная реализация системы (как и SCP Alcatel 1425) построена на базе серверов Alfa 4100, работающих в режиме рабочий/резервный (см. рис. 4.2.13), к которым подключены внешние накопители на жестких дисках. Для подключения рабочих станций операторов используется либо локальная сеть Ethernet, объединяющая все компоненты системы, либо сервер X.25 или модемный пул в случае дистанционного доступа персонала или абонентов. Опционально поставляемое оборудование SCE подключается к системе по протоколу TCP/IP.

Прикладное программное обеспечение системы предоставляет операторам сети и поставщикам услуг ряд развитых средств адаптации услуг к потребностям абонентов с использованием интуитивного графического интерфейса, средств подготовки отчетов о стоимости связи по индивидуальным требованиям, средств

сбора и предоставления статистических данных поставщикам и абонентам услуг.

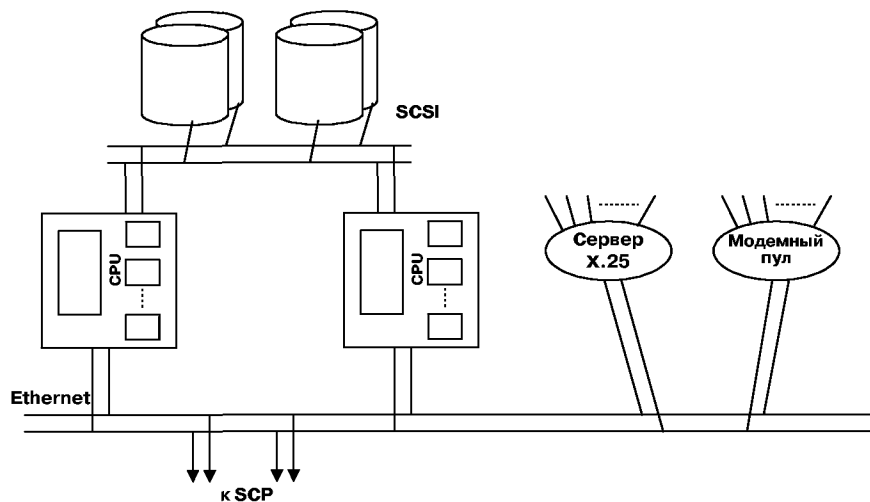


Рис. 4.2.13 Структура SMP Alcatel

4.2.3.4 SCE платформы IN Alcatel

SCE Alcatel 1452 реализована в виде специализированной программной среды, работающей в качестве прикладной программы под управлением операционной системы Windows. Среда обеспечивает возможность описания услуги с использованием палитры библиотечных блоков SIB, проверку алгоритма функционирования услуги посредством моделирования, описание графического интерфейса пользователя услуги для SMP, конфигурирование конечных автоматов, прогнозирование показателей качества обслуживания и развития сети.

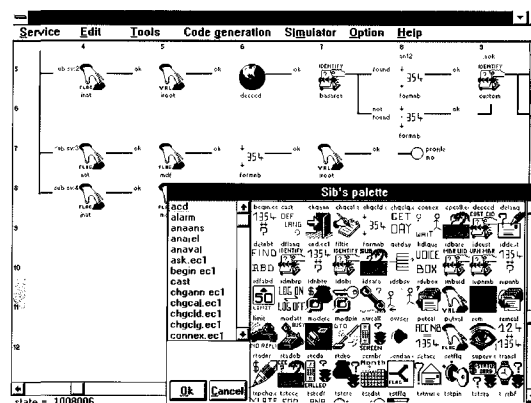


Рис. 4.2.14 Интерфейс графического редактора SCE Alcatel

В каждом из блоков SIB, используемых при работе с редактором услуг, инкапсулировано несколько объектов и методов доступа к ним с целью их конфигурации (используются при создании части программного обеспечения услуги, необходимой для SMP) и с целью их применения (используются при создании части программного обеспечения услуги, необходимой для SCP). На рисунке 4.2.14 показан графический интерфейс редактора услуг SCE Alcatel 1452.

В отдельном программном модуле содержится описание структуры сети – количество SCP, SMP и их аппаратная конфигурация. Это обеспечивает возможность генерировать на этапе создания услуги конфигурационные файлы для каждого элемента сети. Другой модуль предназначен для выбора параметров собираемой статистики, определения обслуживаемой нагрузки и ресурсов платформы, отводимых для предоставления услуги.

Глава 4.3

Аспекты внедрения

4.3.1 Варианты доступа к IN

Декларируемый в стандартах для IN принцип независимости ее архитектуры от типа сети связи справедлив лишь в том смысле, что международными стандартами однозначно определены функциональные модули платформы IN и взаимосвязи между компонентами (узлами) IN. Однако ответ на вопрос о том, какой части абонентов сети общего пользования будут доступны услуги IN (что, собственно, и определяет в каждом отдельном случае целесообразность применения концепции IN), зависит от выбранного принципа доступа к платформе IN и от таких характеристик телефонной сети как доля цифровых коммутационных станций, используемые способы маршрутизации, план нумерации, системы сигнализации и т.п.

Анализ мирового опыта показывает, что из-за большой стоимости программно-аппаратных средств, нужных для построения платформы, и из-за неопределенности спроса на услуги IN ее внедрение даже в сетях крупных операторов начинается с введения функций SSP в одной-двух коммутационных станциях. Кроме того, ограничения, обусловленные наличием в сетях коммутационного оборудования координатной и декадно-шаговой систем, приводят к тому, что на первом этапе реализации интеллектуальной сети экономически целесообразным и технически возможным оказывается введение ограниченного набора услуг, доступных любому абоненту сети, в которую вводится платформа IN.

Возможны разные варианты организации доступа абонентов сети связи к оборудованию платформы IN. Основными критериями выбора точки и способа доступа являются возможность выхода к платформе IN любого абонента данной зоны путем набора единого номера, возможность получения в точке доступа информации о номере вызывающего абонента, возможность передачи от абонента АТС любого типа к точке доступа переменного числа цифр. Если на всех участках соединения при связи оконечных АТС с коммутационной станцией, через которую обеспечивается доступ к платформе, не используется подсистема ISUP OKC-7, то перечисленные требования технически выполнимы только при наличии прямых пучков соединительных линий.

Сети, претендующие на предоставление услуг абонентам нескольких междугородных зон или (в перспективе) всей ТФОП, удобно называть Интеллектуальными сетями федерального статуса. Если же оператор ориентируется на предоставление услуг только абонентам своей зоны, то Интеллектуальная сеть будет иметь региональный статус. Сети обоих видов могут предоставлять функционально анало-

гичные услуги разным группам абонентов, а один и тот же пользователь услугами IN, набирая разные префиксы, имеет возможность выхода к IN, имеющим разный статус.

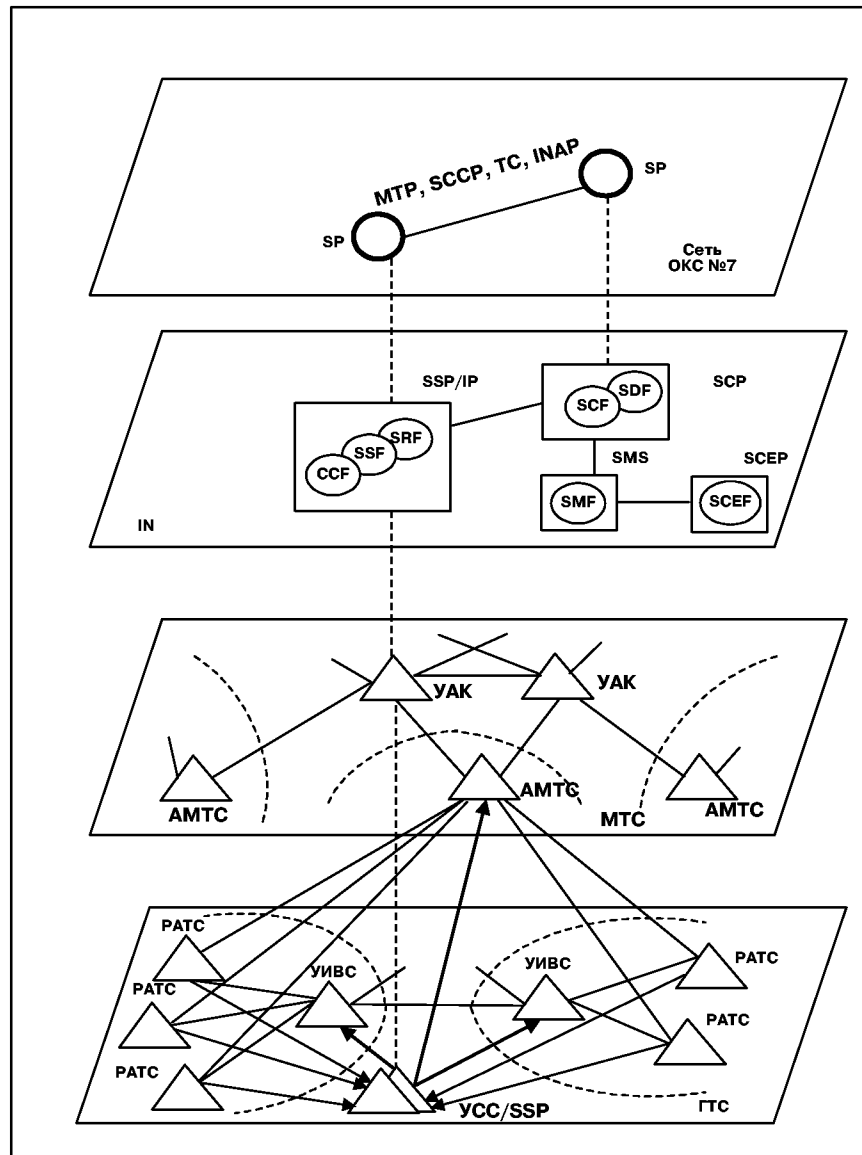
В зависимости от степени развития телефонной сети и потребностей ее абонентов в услугах IN администрацией этой сети может быть принят один из описываемых ниже вариантов первоначального развертывания Интеллектуальной сети и организации доступа к ней. С учетом перечисленных выше критериев выбора, существующих принципов построения и плана нумерации ТФОП в России, а также возможностей АТС электромеханических систем, возможны два основных способа организации доступа к платформе IN:

- 1) доступ путем набора префикса «8» через АМТС (или введение функций SSP в АМТС) с использованием свободных кодов ABC, для которых принято обозначение «DEF»;
- 2) доступ путем набора префикса «0» через узел спецслужб (или введение функций SSP в УСС) с использованием свободных индексов YZ.

Первый вариант связан с необходимостью увеличения емкости пучков заказно-соединительных и соединительных линий (ЗСЛ и СЛМ), посредством которых осуществляется связь местной сети с АМТС. Для доступа к услуге IN в данном случае требуется набор префикса «8», прослушивание акустического сигнала «Ответ АМТС», набор кода DEF, определенного для запрашиваемой услуги, и набор оставшихся цифр номера. Основной недостаток такого решения – использование ресурсов АМТС (а также линий ЗСЛ и СЛМ) для обслуживания трафика IN, замыкающегося в пределах местной сети и объективно не нуждающегося в ресурсах, изначально предназначенных для междугородной связи.

Второй вариант доступа связан с изменением системы нумерации узла спецслужб (в ряде случаев, – только с введением новых индексов Y и Z, обеспечивающих доступ к ресурсам IN). При введении функций SSP непосредственно в цифровой УСС, возникает новое для УСС требование – организация исходящей связи в сторону ГТС и АМТС. В качестве недостатка такого решения можно упомянуть смешивание на уровне местных станций трафика, который создают вызовы, требующие услуг IN, и вызовы, направленные к экстренным службам.

Диаграммы, иллюстрирующие взаимосвязь между конфигурациями телефонной сети, Интеллектуальной сети и сети ОКС-7 при реализации описанных способов доступа, приведены на рисунках 4.3.1 и 4.3.2. На рисунках показаны, в основном, только те элементы взаимосвязи, которые существенны с точки зрения вызовов, требующих услуг Интеллектуальной сети, и не показана организация сети ОКС-7 (если таковая имеется в сети оператора), обеспечивающей сигнализацию между коммутационными узлами ТФОП.



ГТС - городская телефонная сеть
МТС - междугородная телефонная сеть

Рис. 4.3.1 Взаимосвязь между конфигурациями телефонной сети, Интеллектуальной сети и сети ОКС-7 (доступ к ИН через УСС)

Основные преимущества и недостатки каждого из способов доступа при первоначальном развертывании ИН сведены в таблицу 4.3.1. Анализируя их, можно заключить, что если ресурсы местной и междугородной сетей в одном крупном регионе принадлежат разным операторам, то в нем могут быть организованы сети ИН двух типов: с доступом путем набора цифры «8» и с доступом путем набора циф-

ры «0». «Местный» оператор может предоставлять абонентам ГТС услуги IN регионального статуса с возможностью (после пересчета логического номера) исходящей местной, междугородной и международной связи. «Междугородный» оператор может предоставлять абонентам своей зоны услуги IN, имеющие как федеральный, так и региональный статус, с возможностью обеспечения всех видов исходящей и входящей (в том числе от абонентов других зон) связи. Следует заметить, что номер вызывающего абонента может быть передан по междугородной сети только средствами подсистемы ISUP ОКС-7.

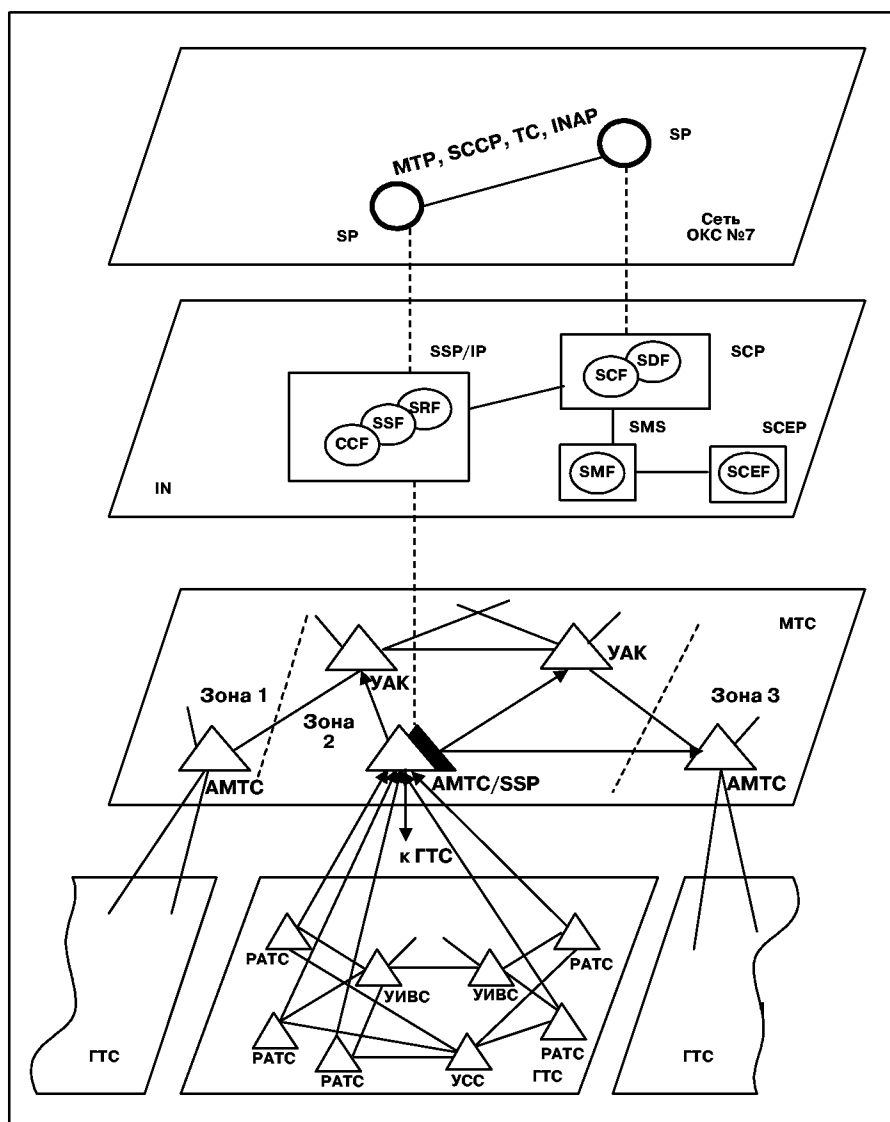


Рис. 4.3.2 Взаимосвязь между конфигурациями телефонной сети, Интеллектуальной сети и сети ОКС-7 (доступ к IN через AMTC)

При организации интеллектуальной сети «местным» оператором для предоставления значительного числа услуг IN (обслуживание компаний, имеющих несколько отделений в одном городе, проведение телеголосования, предоставление информационных услуг по вопросам, имеющим локальное значение, и многое другое) будут использоваться только ресурсы местной сети, а занимать ресурсы, предназначенные для междугородной связи, не понадобится.

Возможны и другие (менее очевидные) способы доступа, однако, по нашему мнению, для первоначального развертывания IN они вряд ли подойдут. Так, начало построения IN с введения функций SSP в транзитный (а тем более – в оконечный) коммутационный узел связано с выделением для доступа к нему от АТС электромеханических систем (составляющих до 70%-90% общего количества местных АТС) индекса «а» («ab»), что значительно ограничит номерную емкость местной сети. Кроме того, в крупных сетях, где введение услуг IN будет происходить в первую очередь, просто может не оказаться свободных индексов. Компромиссным решением при введении функций SSP, например, в цифровой узел местной сети (обычно совмещающий в себе функции узла исходящей, входящей и междугородной связи), может стать организация доступа к нему через цифровую АМТС посредством ISUP ОКС-7, тогда как доступ оконечных АТС к АМТС осуществляется по ЗСЛ обычным способом.

Дальнейшее развитие IN за счет введения новых SSP в цифровые оконечные АТС, в транзитные узлы и в АМТС других зон позволит распределить между ними трафик, ранее проходивший через один SSP.

Таблица 4.3.1

Доступ к платформе IN через АМТС	Доступ к платформе IN через УСС
Преимущества	Преимущества
1. Возможность доступа абонентов всех местных сетей данной зоны и абонентов АМТС других зон (при выделении на этих АМТС специального направления для услуг IN).	1. Доступ абонентов ГТС к услугам IN без привлечения ресурсов АМТС создает основу для снижения затрат на развитие сетевой инфраструктуры (следствие - меньший срок окупаемости инвестиций) и стоимости предоставления услуг местного уровня (следствие - большее число клиентов).
2. Эффективность организации услуг федерального уровня.	2. Независимость предоставления услуг местного уровня от оператора АМТС
	3. Простота развития сети путем установки дополнительных SSP на цифровых опорно-транзитных АТС.
Недостатки	Недостатки
1. Необходимость использования коммутационных ресурсов АМТС, а также ЗСЛ и СЛМ для обслуживания вызовов IN внутри ГТС.	1. Необходимость использования ресурсов УСС и СЛ для предоставления услуг, требующих маршрутизации вызовов по междугородной сети (после пересчета логического номера).
2. Затруднено развитие сети в случае распределения вызовов IN по нескольким SSP внутри ГТС.	2. Невозможность доступа к услугам со стороны абонентов других зон.

4.3.2 Нумерация услуг IN

Для любой услуги IN последовательность цифр, набираемых абонентом, содержит:

- префикс (индекс) выхода к платформе IN;
- кода доступа к услуге или к группе услуг IN;
- код доступа к определенному атрибуту услуги (не обязательно);
- цифры, используемые логикой услуги.

Логика некоторых услуг требует, чтобы после установления соединения абонента с SSP и организации логической связи между SSP и SCP последний давал команду оборудованию интеллектуальной периферии IP передать абоненту приглашение к набору дополнительной номерной информации многочастотным способом DTMF. В ряде случаев это позволяет существенно расширить функциональные возможности услуги (набор PIN-кода, номера счета для начисления платы и т.п.). Современные телефонные аппараты предусматривают переход к режиму DTMF в процессе набора нажатием одной клавиши.

В таблице 4.3.2 приводится пример соответствия между планом нумерации и услугами IN для телефонных сетей РФ. При решении вопросов, связанных с нумерацией услуг, необходимо различать, какой статус (федеральный или региональный) имеет сеть и какой тип доступа используется; последнее однозначно определяет последовательность цифр, набираемых при обращении к услуге IN.

Выделение кодов/кода «DEF» из числа резервных кодов «ABC» для доступа к IN, имеющим федеральный статус (как и распределение логических номеров), производится на централизованной основе. В этом случае значность номера может соответствовать принятой в ТФОП для междугородных номеров: 8-DEF- $X_1X_2\dots X_n$, где 8 – индекс выхода на АМТС; DEF – код услуги, который назначается из 8-ой сотни; $X_1X_2\dots X_n$ – цифры, используемые логикой услуги.

Таблица 4.3.2

Услуга	Декадный набор			DTMF
	Пн	Код доступа к услуге	Цифры номера	Дополнительная информация
FPN	8 или 0	800	Логический номер	-
PRM	8 или 0	809	Логический номер	-
ААВ	8 или 0	801	-	номер счета, PIN, номер вызываемого абонента
ССС		802		
АСС		806		
VOT	8 или 0	803	Номера для голосования	-

Для доступа к IN местного уровня целесообразно использовать ресурсы плана нумерации местной сети. В этом случае значность номера может (но не обязательно должна) совпадать с принятой в местных сетях ТФОП России. При организации доступа к IN регионального уровня через УСС (или введении функций SSP в УСС) структура номера для сети с семизначной нумерацией выглядит следующим образом: 0-YZ-X₁X₂...X_n, где YZ – номера, выделяемые на УСС из имеющейся резервной емкости узла.

Назначение кодов доступа к разным услугам, а также отведение логических номеров покупателям услуг, осуществляется операторами местных сетей. Целесообразно в этом случае иметь YZX₁=DEF с тем, чтобы «облегчить жизнь» абонентам, которые пользуются одной и той же услугой IN при переезде из одного города в другой.

Набираемые пользователем при обращении к услуге IN цифры X₁X₂...X_n фактически уже не входят в план нумерации местной сети (т.е. составляют не физический, а логический номер), и их количество может быть разным для разных услуг (быть больше или меньше значности, используемой в местной сети). Только после взаимодействия между SSP и SCP коммутационный узел с функциями SSP начинает установление соединения, используя ресурсы имеющейся сигнализации, стандартную схему маршрутизации и план нумерации ТФОП.

4.3.3 Использование сигнализации ОКС-7

На участке от оконечной АТС до SSP сигнализация по ОКС-7 предпочтительна, но не обязательна. Использование же подсистем MTP, SCCP, TC и протокола INAP ОКС-7 для обмена информацией между узлами IN (SSP, SCP и IP) совершенно необходимо; последние в этом случае становятся пунктами сигнализации сети ОКС. Принципы адресации и маршрутизации зависят от принадлежности взаимодействующих пунктов сети ОКС-7 (ассоциируемых с каким-либо из элементов IN) к одному из уровней ТФОП – междугородному или местному.

Узлы Интеллектуальной сети могут находиться как на местном, так и на междугородном уровне телефонной сети. При взаимодействии по сети ОКС-7 узлов IN, размещенных на одном и том же уровне ТФОП, для маршрутизации и адресации сообщений SCCP целесообразно использовать поле «код пункта назначения» (DPC) этикетки и поле «номер подсистемы» (SSN) параметра «адрес вызываемой (вызывающей) стороны». Этой информации достаточно, чтобы (1) осуществлять, при необходимости, транзит сообщений SCCP средствами подсистемы MTP в транзитных пунктах сигнализации STP и (2) направлять сообщение к требуемой подсистеме внутри элемента сети.

При размещении взаимодействующих узлов IN на разных уровнях иерархии ТФОП возникает необходимость в использовании для адресации сообщений SCCP поля «глобальный заголовок» (GT) параметра «адрес вызываемой (вызывающей) стороны» и пересчета содержащейся в нем адресной информации в «код пункта назначения» (DPC). В этом случае пункт сигнализации, выполняющий функции шлюза между междугородной и местными сетями ОКС-7, производит транзит сигнальных сообщений вида MTP-SCCP-MTP с пересчетом адресной информации подсистемой SCCP, то есть является транзитным пунктом на уровне SCCP (SPR – Signalling point relay). Транзитный пункт сигнализации может быть как автономным, так и совмещенным с одним из узлов IN (например, с AMTC/SSP). Далее, в пределах одного уровня иерархии сети, транзит сообщений SCCP может осуществляться транзитными пунктами сигнализации на уровне MTP.

Параметр GT содержит последовательность цифр, набранных абонентом при обращении к услуге IN, то есть, как было сказано выше, код услуги и логический номер. Транзитный пункт SCCP выполняет функции пересчета кода услуги и логического номера в код пункта сигнализации сети ОКС-7, соответствующего тому узлу управления услугами SCP, который отвечает за предоставление этой услуги и за обслуживание данного логического номера. При наличии в одной сети IN нескольких SCP адресация на основе GT позволяет избавить SSP от необходимости знать адрес в сети ОКС-7 того SCP, который содержит логику затребованной услуги.

Прикладной протокол INAP поддерживает гибкое распределение функциональных объектов Интеллектуальной сети (CCF/SSF, SCF, SRF) по физическим элементам IN и, следовательно, – по узлам сети ОКС-7. Совмещением нескольких объектов в одном узле может быть достигнут компромисс между стоимостью оборудования и предоставляемыми оператору сети возможностями. INAP обеспечивает максимальную свободу распределения функциональных объектов между узлами (вплоть до одного функционального объекта на узел).

Для поддержки в России услуг IN разработана версия протокола INAP-R, основанная на стандарте ETSI ETS 300 374. Основные отличия INAP-R от стандарта ETSI вызваны тем, что использование режима, в котором при отсутствии IP или свободных ресурсов IP на исходящем SSP последний может установить соединение к IP другого SSP, признано необязательным. По этой причине в INAP-R не используются прикладные контексты, определяющие группы ASE для поддержки процедур ассистирования и передачи управления. Кроме того, определены необходимые параметры для операций начисления платы, оставленные в стандарте на усмотрение оператора сети.

4.3.4 План действий

Российским операторам вряд ли подойдет единое для всех решение вопроса о способе доступа, поскольку и в структуре, и в потенциальных возможностях развития многих местных сетей существуют определенные различия. В большинстве сетей отсутствует практика предоставления дополнительных услуг, неизвестен спрос и непонятна номенклатура услуг, которые будут пользоваться спросом. Кроме того, уже сегодня становится очевидным, что в крупных городах предоставлять услуги IN смогут несколько операторов, для которых предпочтительными окажутся разные способы доступа.

Это дает основание попытаться сформулировать не зависящие от способа доступа аспекты, требующие системно-технического анализа, и определить последовательность этапов модернизации сети связи с целью введения IN. Примерный перечень задач выглядит следующим образом:

- определение номенклатуры услуг первого этапа развертывания IN, перечня атрибутов для каждой услуги и алгоритмов предоставления услуг, включая необходимое для каждой услуги количество цифр и способ их передачи от абонента до SSP или до того узла сети общего пользования, через который предполагается осуществлять доступ к сети IN;
- выбор принципа доступа, нумерации услуг, маршрутизации вызовов в направлении IN с учетом доступных ресурсов первичной сети, установленного коммутационного оборудования и используемых систем сигнализации;
- согласование алгоритма предоставления услуг в сети оператора с единым планом нумерации услуг IN, имеющих федеральный статус, и резервирование номерной емкости для услуг местного уровня;
- определение технических и финансовых принципов взаимодействия с другими операторами;
- модификация алгоритмов приема набираемых абонентом цифр в тех цифровых коммутационных узлах сети общего пользования, через которые предполагается осуществлять доступ к IN, с тем, чтобы имелась возможность приема от абонента нестандартного (в общем случае – переменного) количества цифр;
- установка таксофонов, способных работать с кредитными картами и/или оснащенных номеронабирателем с многочастотным набором;
- определение числа и типов коммутационных узлов, планируемых к оснащению функциями узла коммутации услуг и интеллектуальной периферии SSP/IP, расчет емкости и производительности IP;

- выбор поставщика и развертывание узла управления услугами SCP, оснащенного программным обеспечением для определенного ранее перечня услуг и атрибутов;
- модификация услуг IN под требования заказчика с использованием технических средств среды создания услуг SCEP и организация эксплуатационного управления услугами IN средствами SMP;
- планомерное введение средств ОКС-7 для облегчения доступа к ресурсам IN и последующая оптимизация структуры сети ОКС с целью повышения эффективности ее работы, включая определение целесообразности использования автономных транзитных пунктов сигнализации на уровне SCCP (STP/SPR).

На организационно-административных аспектах хотелось бы остановиться отдельно, поскольку они незаслуженно обделены вниманием со стороны технических специалистов. Организационные мероприятия, связанные с реализацией дополнительных услуг, должны проводиться при тесном взаимодействии операторов сетей связи (городских и междугородных, стационарных и подвижной связи) и операторов IN с организациями, планирующими развитие сетей, с разработчиками оборудования, с потенциальными поставщиками и покупателями услуг. Из основных организационно-административных аспектов введения дополнительных услуг можно выделить следующие:

- определение потенциальных покупателей услуг IN и выяснение их пожеланий в отношении конкретной реализации услуг, подготовка и проведение рекламной компании, предшествующей введению какой-либо услуги, выяснение мнения абонентов о том, какого рода услуги они хотели бы иметь и сколько готовы за них платить;
- согласование и выделение кодов, префиксов и номеров для идентификации, регистрации и непосредственного использования абонентами дополнительных услуг IN. Желательно, чтобы такие номера состояли из ограниченного количества цифр, образующих хорошо запоминающиеся комбинации;
- решение вопросов, касающихся взаиморасчетов между операторами разных сетей (городских, сотовых, корпоративных, междугородных и IN).

Перед введением услуг IN следует тщательно продумать процедуры начисления платы и выставления счетов, которые должны существенно отличаться от процедуры начисления платы за предоставление основных телекоммуникационных услуг, реализуемой, как правило, только на АМТС.

Процедура начисления платы за пользование услугами IN предполагает специальные процедуры взаимодействия между SSP и SCP.

При этом для каждого запроса услуги IN в SSP под управлением SCP формируется запись о тех параметрах запроса, которые требуются для начисления платы. Применительно к каждой услуге IN запись может содержать поля, в которые заносится специфическая для данной услуги информация.

Рассмотренные принципы доступа к IN – через AMTC и через УСС – практически равноценны. Например, невозможность доступа к услугам IN через УСС со стороны абонентов других зон уравнивается возможностью предлагать услуги по более низким тарифам благодаря тому, что для обслуживания местного трафика не требуется использовать ресурсы междугородной сети.

Однако выбранный для данной сети принцип доступа накладывает определенные ограничения на маршрутизацию вызовов и на нумерацию услуг IN. Наличие услуг, имеющих федеральный статус, влечет за собой необходимость согласования нумерации с учетом различий в правилах маршрутизации.

4.3.5 Добраться до интеллекта

Исторически сложилось так, что городские телефонные сети предоставляли, в основном, услуги телефонной связи и справочную информацию, и развитие этих сетей осуществлялось в направлении удовлетворения такими услугами всех секторов рынка. Вследствие особенностей технических характеристик используемого оборудования, телефонные сети крупных городов построены по радиально-узловому принципу с применением опорных станций емкостью до 10 тыс. номеров и узлов исходящего/входящего сообщения, обслуживающих до 100 тыс. номеров.

Сегодня большинство операторов уже осознали, что, используя достижения в области цифровых систем передачи и коммутации, можно развивать сети по новым принципам, отойдя от классических методов улообразования. Основная идея большинства проектов модернизации телефонных сетей заключается в создании мощных опорно-транзитных узлов (ОПТУ), обслуживающих транзитный трафик конечных АТС.

В качестве одной из главных составляющих таких проектов многие операторы видят создание сети сокращенной нумерации (ССН), предусматривающей, кроме развития и модернизации существующих экстренных, справочно-информационных и заказных служб, обеспечение необходимых функций для доступа к платформе IN. Однако приходится иметь в виду, что конечные АТС координатной и декадно-шаговой систем будут использоваться до выработки их ресурса.

Решение об использовании для создания сети IN определенного оборудования принимается оператором на основе анализа и сравнения нескольких технико-экономических вариантов повышения конкурентоспособности и увеличения доходов.

Рассмотрим вариант, когда администрация сети выбрала в качестве способа доступа к платформе IN доступ через УСС. В этом варианте построение Интеллектуальной сети начинается с введения функций коммутации услуг SSF и функций интеллектуальной периферии IP в коммутационные системы, являющиеся частью комплекса оборудования цифрового УСС, дооснащенного программно-аппаратными средствами организации исходящей местной, междугородной и международной связи. Для доступа к IN используются ресурсы плана номеров сети сокращенной нумерации. Планируемая структура номера при обращении к услугам IN выглядит следующим образом: 0-80- $X_1X_2...X_n$, где 80 – цифры, выделяемые из емкости узла спецслужб для доступа к IN.

Взаимодействие между SSP/IP и SCP происходит по прикладному протоколу INAP, а между УСС/SSP и ОПТУ – по протоколу ISUP системы сигнализации ОКС-7. На рисунках 4.3.3 и 4.3.4 приведены схемы прохождения вызовов от оконечных районных АТС разных типов до УСС/SSP с учетом структуры телефонной сети и принятой нумерации услуг. Первый из рисунков разъясняет некоторые технические аспекты прохождения номерной информации от абонента до узла коммутации услуг. Отдельно показаны цифры, анализируемые каждым узлом, и цифры, передаваемые между ними.

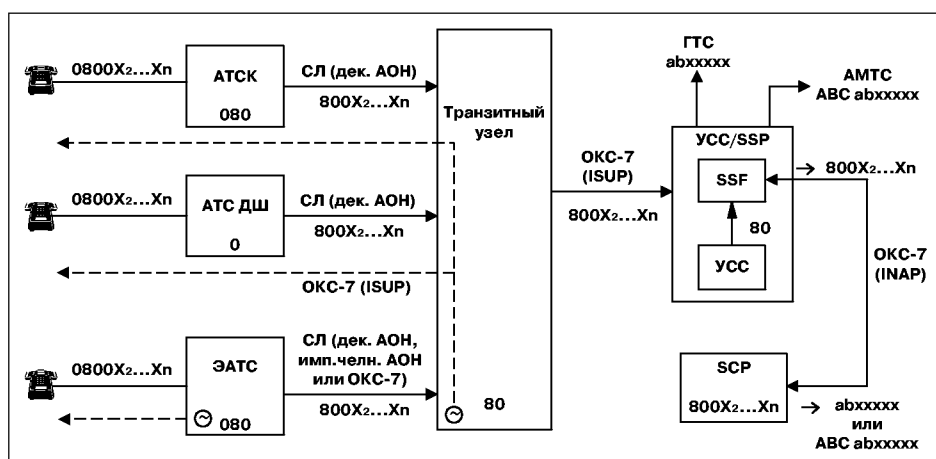


Рис. 4.3.3 Доступ к IN от АТС разных типов

На рисунке 4.3.4 показана схема организации доступа к IN при наличии функций SSP на уровне окончных АТС и развитой сети ОКС-7. Взаимодействие между опорными АТС электромеханических систем и ОПТУ происходит по прямым пучкам СЛ с использованием линейной сигнализации 2ВСК. Что касается электронных опорных АТС, то они принимают все цифры номера, необходимые для обращения к услуге IN, и транслируют их до ОПТУ.

Транзитный узел сразу же после фиксации первых двух цифр производит анализ номера на предмет распознавания вызовов, направленных к платформе IN УСС/SSP. Если вызов направлен к IN, ОПТУ подает абонентам электромеханических АТС акустический сигнал приглашения к дальнейшему набору, фиксирует все цифры (переменное количество) и передает их к УСС/SSP по протоколу ISUP ОКС-7.

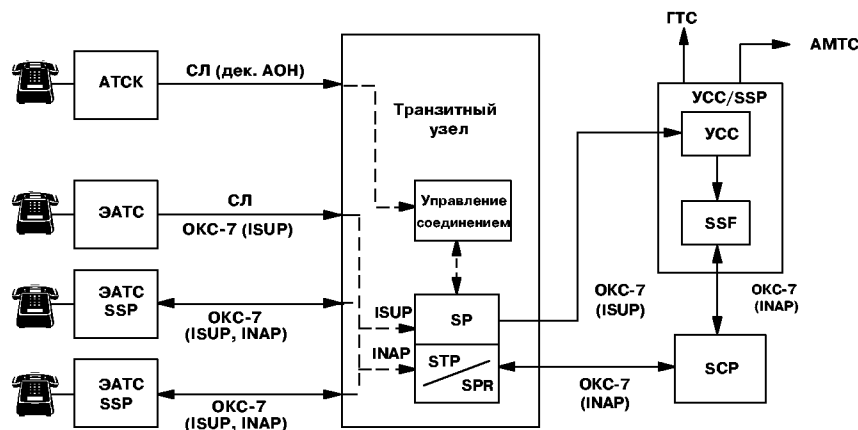


Рис. 4.3.4 Развитие сети IN

После завершения начального этапа развития IN, который характеризуется наличием функций SSP только в УСС, последуют другие этапы. На втором этапе, с ростом трафика IN, целесообразно ввести функции SSP в транзитные узлы, что даст возможность распределить этот трафик между УСС/SSP и несколькими ОПТУ/SSP.

Третий этап развития IN предполагает оснащение функциями SSP цифровых окончных АТС, что даст возможность максимально расширить перечень услуг IN, предоставляемых абонентам этих АТС. К этому этапу оператор переходит постепенно, оснащая функциями SSP новые АТС сети по мере возврата начальных инвестиций. Сравнительная характеристика этапов и стратегии внедрения приведена в таблице 4.3.3.

Таблица 4.3.3

ПРЕИМУЩЕСТВА	НЕДОСТАТКИ	УСЛУГИ
I. SSP НА ВЕРХНЕМ УРОВНЕ (УСС)		
Наиболее экономичный способ, позволяющий построить IN при минимальных инвестициях. Для предоставления услуг не требуется изменений в существующей сети. Быстрый ввод в эксплуатацию.	Не все услуги и атрибуты могут быть использованы. Вся нагрузка сконцентрирована в УСС.	FRN, PRM, VOT, ACC/CCC/AAB.
II. SSP НА ТРАНЗИТНОМ УРОВНЕ (ОПТУ)		
Частичное устранение возможной перегрузки направлений между ОПТУ и УСС/SSP.	Остается возможность перегрузки на участке PATC ОПТУ/SSP.	Те же.
III. SSP НА ЛОКАЛЬНОМ УРОВНЕ (ОКОНЕЧНЫЕ АТС)		
Абонентам АТС/SSP доступны все услуги и атрибуты. Отсутствие перегрузки направлений в сторону SSP.	Высокие затраты.	Весь спектр услуг CS-1.

4.3.6 Подходы и альтернативы

Используя тот факт, что телекоммуникационная сеть превратилась в крупнейшую и наиболее сложную систему в мире, в 80-90-х годах телефонные сети стали брать на себя новые функции, которые ранее выполнялись иными средствами (печатью, почтой, радио, телевидением). За короткое время операторы сетей и производители оборудования связи создали и опробовали разные подходы к решению проблем, вызванных новыми функциями сетей. В данном разделе проведен анализ отдельных подходов к внедрению IN и некоторых «узких мест», связанных с созданием услуг и с реализацией сетей IN.

4.3.6.1 Различия в подходах

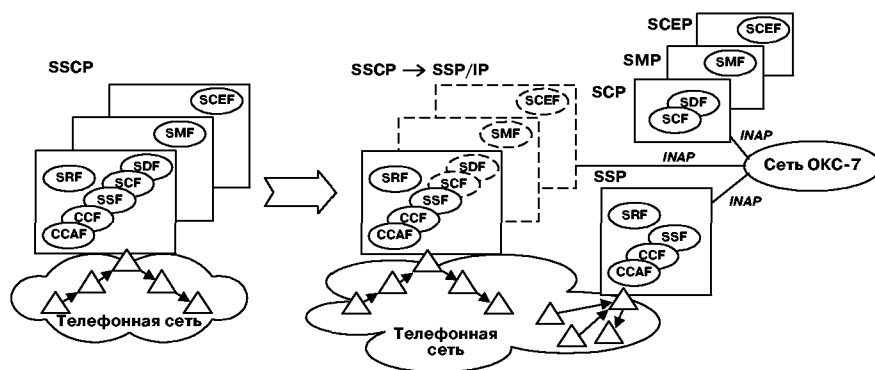
Концепция IN представляет собой совокупность функциональных требований, интерфейсов и протоколов для поэтапного продвижения к долговременной целевой архитектуре. Резюмируя сказанное во второй части книги, можно сказать, что первая фаза реализации IN, ассоциируемая с набором CS-1, представляет собой, скорее, новый способ реализации давно известных услуг или структурирования существующей сети связи, чем создание новой сети или принципиально новых услуг. Основные функции предоставления услуг в сетях IN переносятся из коммутационных станций в небольшое число узлов управления услугами (SCP). Для взаимодействия с SCP существующие коммутационные станции дооборудуются функциями

коммутации услуг и становятся после этого узлами коммутации услуг (SSP). Таким образом, на первом этапе введения услуг IN существующая сеть связи лишь структурируется в соответствии с концепцией IN с целью более эффективного предоставления, в основном, уже зарекомендовавших себя услуг (пересчета номерной информации, разнообразных способов начисления платы и т.д.).

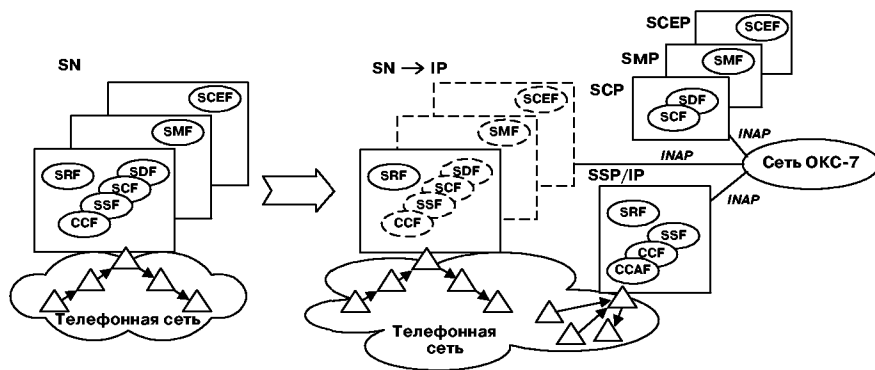
Следует отметить, что получает достаточное распространение и альтернативный способ предоставления тех же услуг (пересчета номера, использования телефонных кредитных карт и т.д.), базирующийся на применении компьютерной телефонии. Внешний компьютер, связанный с коммутационной станцией, подобно SCP в сетях IN, берет на себя часть функций реализации логики услуг и, в этом смысле, функционально аналогичен ему. Однако для способа компьютерной телефонии пока еще отсутствуют общепринятые стандарты взаимодействия с коммутационной станцией на прикладном уровне. Кроме того, отсутствуют решения, позволяющие использовать один и тот же компьютер для обслуживания нескольких коммутаторов, и, как следствие, отсутствует возможность централизованного развертывания и модификации услуг во всей сети.

История развития услуг IN помнит несколько способов поддержки взаимодействия с централизованными программно-аппаратными средствами реализации логики услуг, позднее получивших общее название SCP. Один из них был основан на применении протокола X.25, другой – на использовании модифицированной версии подсистемы TUP (или ISUP) системы ОКС-7, третий и четвертый – на использовании совмещенного узла коммутации и управления SSCP и узла услуг SN. Разница между двумя последними способами состоит в том, что SSCP строится на базе существующего узла коммутации, а SN представляет собой интегральный элемент сети, включающий в себя основные функциональные объекты IN, который подключается к существующему узлу ТФОП по соединительным линиям (см. рис. 4.3.5).

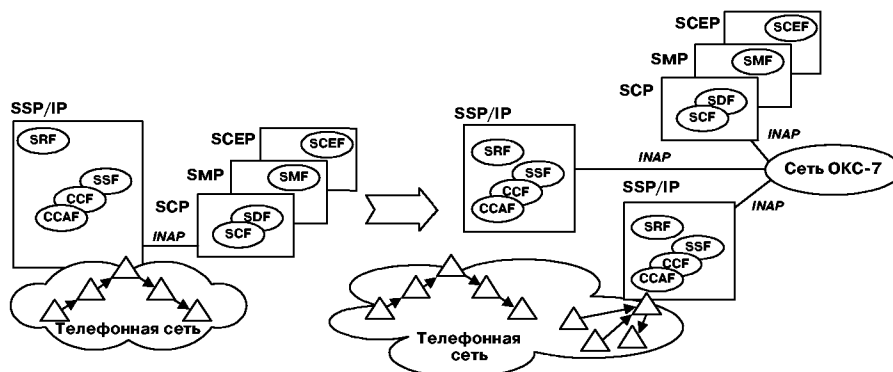
Представленный на рисунке 4.3.5 SSCP предполагает нестандартный интерфейс между функциями SSF и SCF с сохранением, однако, всех стандартных интерфейсов SSP. Следствием этого является возможность предоставления услуг IN даже при полном отсутствии сети ОКС-7, с одной стороны, и неоптимальное развитие сети при увеличении числа SSP, с другой. Ведь после ввода нового SSP прежний SSCP сможет функционировать уже только как SSP; следовательно, в сети потребуются иметь другой SCP, а также и SMP. И что делать со ставшей ненужной инфраструктурой управления и создания услуг, оставшейся в прежнем SSCP?



а) развитие платформы IN в случае SSCP



б) развитие платформы IN в случае SN



в) развитие платформы IN в случае автономных SSP и SCP

△ - коммутационная станция
 — - прохождение вызова IN

Рис. 4.3.5 Перспективы развития платформ IN

Различие между SSCP и SN – в том, что последний не имеет своих абонентов и должен подключаться к коммутационному узлу разговорными и сигнальными каналами, «пропуская через себя» все разговорные соединения. Кроме того, SN не может управляться со стороны внешнего SCP (т.е. не применяется в качестве автономного SSP). Правда, он способен поддерживать взаимодействие с несколькими SSP, и при установке новых SSP мог бы быть оставлен в качестве SCP. Проблема, однако, в том, что SN вряд ли обеспечит требуемую для возросшего трафика IN производительность, и, следовательно, ему придется искать иное применение (вероятнее всего оставить в качестве IP), а всю сеть IN строить практически заново. В таблице 4.3.4 приведен сравнительный анализ вариантов реализации платформ IN.

Таблица 4.3.4

Преимущества при первоначальном внедрении	Недостатки (в т.ч., при развитии сети)
SSCP (CCAF, CCF, SSF+SCF, SDF, SRF)	
Быстрота развертывания платформы.	Невозможность взаимодействия с другими SSP в качестве SCP (т.е. при развитии сети SSCP всегда трансформируется в SSP).
Отсутствие необходимости в сети ОКС-7.	Необходимо новое оборудование с функциями SCF, SMF и SCEF.
Минимум начальных инвестиций.	Функции SCF, SMF и SCEF, имеющиеся в SSCP, остаются невостребованными. Надежность недостаточна, т.к. нет резервирования.
SN (CCF, SSF, SCF+SRF, SDF)	
Быстрота развертывания платформы.	Подключается только по СЛ, поэтому все разговорные соединения проходят через него.
Возможность взаимодействия с несколькими внешними SSP.	При взаимодействии в качестве SCP с внешним SSP речевые каналы должны быть доведены до SN, так как функции SRF интегрированы.
Минимум начальных инвестиций.	При работе в качестве SCP в сети с несколькими SSP обладает недостаточной производительностью. Надежность недостаточна, т.к. нет резервирования.
SSP/IP (CCAF, CCF, SSF, SRF) и SCP (SCF, SDF)	
Гибкость при наращивании производительности и функциональности	Значительные первоначальные вложения
Простота развития	
Надежность, необходимая для крупных сетей общего пользования	

4.3.6.2 Что имеем?

Очевидно, что чем совершеннее сеть связи оператора, тем более эффективную платформу для предоставления услуг интеллектуальной сети можно создать на ее основе. Телефонная сеть как основа

для предоставления услуг IN – это, фактически, не то, что оператор только пытается создать, а то, что он, в большей или меньшей степени, уже имеет. Строить или менять телефонную сеть – сложное и дорогостоящее мероприятие, требующее немало времени, а потому оператор вынужден использовать то, что у него есть, по крайней мере, в ближайшем будущем, которое как раз и соответствует времени, когда предстоит сделать первые шаги на пути создания платформы IN.

Сказанное не означает, что нет нужды производить какие бы то ни было изменения базовой телефонной сети, однако такие, например, шаги, как введение во всей сети связи системы сигнализации ОКС-7, в силу объективных причин требуют длительного времени. Следовательно, необходимо искать решения, которые помогут наилучшим образом использовать существующие у каждого оператора сетевые возможности. Вопрос этот непростой и мы остановимся на нем несколько подробнее. Создание платформы IN облегчается наличием в ТФОП:

- системы сигнализации ОКС-7,
- систем подробного учета стоимости,
- цифровых узлов коммутации на всех уровнях,
- цифровых систем передачи,
- распространенностью телефонных аппаратов с многочастотным набором,
- организаций, осуществляющих маркетинг и обслуживание услуг связи.

В принципе, оператор может начать создание платформы IN даже если в его сети не выполняется ни одно из приведенных условий, однако, чем больше позиций этого перечня удовлетворяется, тем быстрее и дешевле обойдется создание платформы. В большинстве применений для поддержки взаимодействия между узлами SCP и SSP обязательно требуется система сигнализации ОКС-7. Но наиболее важный аспект строительства платформы – идентичный по функциям интерфейс SCP-SSP.

Система подробного учета стоимости дает возможность вынести действия, связанные с начислением платы, в биллинг-центр, благодаря чему обеспечивается гибкость в назначении тарифов и отнесении платы на разные счета в зависимости от вида услуги и ее поставщика, что практически неосуществимо в системе, основанной на применении индивидуальных абонентских счетчиков.

Преимущества цифровых систем коммутации и передачи настолько очевидны, что иногда мы забываем упомянуть о них. Однако наличие в сетях большого количества аналоговых систем коммутации,

в основном, на уровне окончательных АТС, приводит к тому, что обслуживаемые ими абоненты лишаются многих услуг IN из-за устаревших протоколов сигнализации. В России ситуация несколько проще благодаря тому, что даже на АТС электромеханических систем предусмотрена процедура автоматического определения номера вызывающего абонента, доступная, впрочем, только на уровне местных сетей.

4.3.6.3 Реализация услуг в сетях IN

С точки зрения анализируемых нами аспектов различают услуги, реализация которых привязана к конкретной АТС, и услуги, которые реализуются согласно концепции IN. Услуги первого вида как бы «фиксированы» и их трудно модифицировать, а вариант с IN опирается на независимые от услуг программные модули – блоки SIB. И хотя SIB довольно тесно привязаны к технической платформе (см. рис. 4.3.6), тем не менее, путем комбинации этих блоков может быть создан широкий спектр услуг. Перечень SIB определяется отдельно для каждого набора CS. Время, требуемое для создания, тестирования и введения новой услуги, в варианте IN снижается, по сравнению с первым вариантом, с нескольких лет до нескольких месяцев.

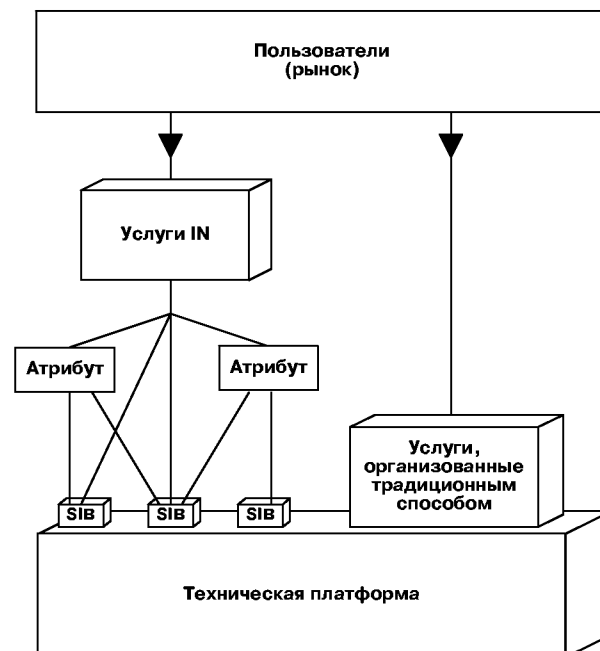


Рис. 4.3.6 Сравнение способов создания услуг

Услуга – это то, что получает пользователь в виде законченного коммерческого продукта. Услуги компонуются из атрибутов оператором сети, если, конечно, он имеет необходимые для этого средства (SCEP). Ни одна сеть в мире не поддерживает все 25 услуг и 38 их атрибутов, специфицированных для набора CS-1, однако реализация платформой IN всех SIB дает оператору возможность создавать недостающие услуги и атрибуты позднее, ничего не меняя в своих SSP, а начать можно и с ограниченного перечня. Ведь не все услуги одинаково доходны и не все они могут быть доступны сразу всем абонентам телефонной сети.

В случае IN, когда логика услуг находится в SCP и подконтрольна поставщику, ему значительно проще создавать услуги и обеспечивать их предоставление. Оператор сети IN может оперативно дать поставщику возможность проведения опытной эксплуатации услуги, определив в одном или в нескольких своих SSP соответствующие триггерные точки и загрузив логику услуги в SCP. На основе анализа результатов опытной эксплуатации поставщик может принять решение о целесообразности предоставления услуги и о необходимой территории охвата.



Рис. 4.3.7 Создание структур данных для новой базовой услуги

Когда услуга IN скомпонована и может предоставляться абонентам, то говорят, что создана новая *базовая услуга* (рис. 4.3.7). Однако разные абоненты предъявляют к одной и той же услуге разные требования. Например, в отношении услуги Freephone требования

компании по доставке пиццы, имеющей 15 ресторанов, могут существенно отличаться от требований банка с 15 отделениями. В процессе совместной работы оператора сети и конкретного заказчика услуги производится индивидуальная подстройка базовой услуги к требованиям этого заказчика.

В варианте с компанией по доставке пиццы, желающей иметь один логический номер для всех своих отделений, на этапе индивидуальной подстройки услуги должны быть приняты решения по таким вопросам как разделение территории на зоны, обслуживаемые каждым отделением, и как часы работы отделений. Нужны ответы и на следующие вопросы. Что делать, если отделение не работает, – передать клиенту речевое сообщение? Что делать при чрезмерно большом трафике? Кто должен изменять вышеназванные параметры? Какие действия разрешены со стороны интерфейса управления, доступного заказчику, и кому они разрешены?

После создания и подстройки к требованиям заказчика услуга готова к *вводу в эксплуатацию* (рис. 4.3.8). На этапе ввода должны быть приведены в соответствие все системы администрирования и начисления платы. При этом следует разделять ввод новой базовой услуги и индивидуальную подстройку ранее введенной базовой услуги к требованиям нового абонента.

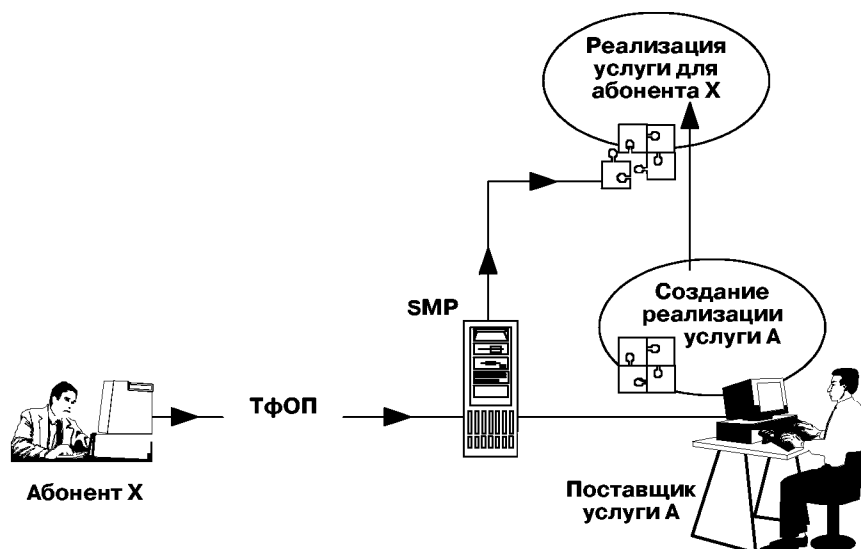


Рис. 4.3.8 Введение в эксплуатацию новой базовой услуги

При вводе новой базовой услуги нужно протестировать ее в SMP, загрузить необходимое программное обеспечение в SCP, выделить уникальный номер для доступа к услуге. Последнее требует, чтобы оконечные и транзитные АТС могли бы маршрутизировать вызовы,

связанные с обращением к новой услуге, в направлении SSP, а SSP – имел бы возможность распознавать ее номер для маршрутизации к нужному SCP.

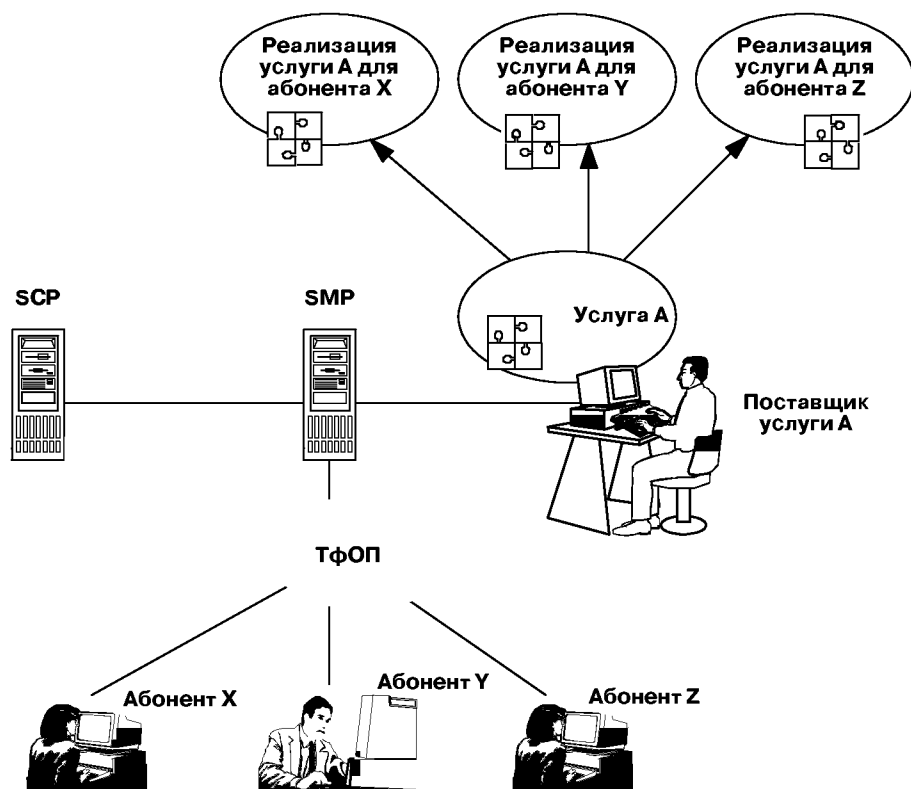


Рис. 4.3.9 Подстройка базовой услуги под требования абонентов

При подстройке базовой услуги к требованиям нового абонента (рис. 4.3.9) необходимо выделить ему логический номер, ввести необходимые численные параметры (временные – для каждого маршрута, процентные – для распределения вызовов и т.д.), задать пароли и PIN-коды, ввести информацию об абоненте в соответствующие базы данных, определить географическую область доступности услуги.

4.3.7 Не все так просто

4.3.7.1 Новые услуги с позиций телетрафика

Расчет числа соединительных линий и производительности систем коммутации, как и оценка их поведения в условиях перегрузки, сегодня производится с привлечением методов теории телетрафи-

ка. Эти методы позволяют также с достаточной для практики точностью предсказать ожидаемую нагрузку. При расчете коммутационных и линейных ресурсов обычно используется концепция часа наибольшей нагрузки (ЧНН).

При введении IN появляется новый тип услуг, например, Televoting и Premium Rate, отличительная особенность которых состоит в том, что они создают нагрузку, параметры которой отличаются от привычных. Трафик характеризуется ярко выраженными выбросами, амплитуду которых не всегда можно предсказать.

При нагрузке 0,1 Эрл на абонентскую линию в ЧНН и средней длительности обслуживания вызова 120 секунд АТС емкостью 10000 номеров должна иметь производительность порядка 8-ми вызовов в секунду. Предположим, что АТС спроектирована с запасом и способна обрабатывать 30 вызовов в секунду. Посмотрим, хватит ли такого четырехкратного запаса и что произойдет, если субботним вечером будет объявлен номер для телеголосования во время популярной телевизионной программы.

Итак, в программе, идущей в прямом эфире, объявляется, что первые 100 человек, дозвонившиеся по номеру, показанному на экране, получают в качестве приза автомобиль. Предположим, что эту передачу смотрят 20% абонентов АТС и все они, естественно, захотят выиграть автомобиль. Следовательно, АТС должна будет принять на обслуживание 2000 вызовов. Вызовы эти поступят к АТС практически одновременно, избежать перегрузки не удастся, так что часть желающих дозвониться получит отказ сразу после снятия трубки.

Если предположить, что все 2000 вызовов поступят к АТС в течение 10 секунд, то интенсивность потока этих вызовов составит 200 вызовов в секунду. Интенсивность потока вызовов, поступающих к АТС от остальных 80% абонентов, пренебрежимо мала, однако вряд ли кому-нибудь из них удастся получить связь. Ясно, что возникновение перегрузки АТС во время телеголосования практически неизбежно, и это не зависит ни от типа доступа (через АМТС или УСС), ни от типа АТС, ни от количества соединительных линий.

Те из вызовов, запрашивающих услугу телеголосования, которые все-таки были обработаны рассматриваемой АТС, маршрутизируются к ближайшему SSP, после чего запросы услуги направляются в SCP. Внутри самой Интеллектуальной сети предусмотрены процедуры обработки массовых вызовов (Service Filtering), посредством которых производится их просеивание на предмет возможности обслуживания каждого из этих вызовов в SCP. Если процедура Service Filtering активирована, то отправляемое в сторону SCP сообщение содержит результат обработки целой группы вызовов.

Таким образом, видно, что все вызовы создаются абонентами оконечной АТС и маршрутизируются к SSP по соединительным линиям, причем ни АТС, ни линии не рассчитаны на «взрывной» характер нагрузки. Если в узлах IN средства борьбы с перегрузками, вызванными «взрывным» характером трафика, предусмотрены, то предотвратить описанную выше ситуацию на участке от оконечной АТС до SSP можно лишь с помощью процедур предотвращения перегрузок в АТС и правильно выбранной платой за вызов. Влияние на перегрузки могут оказать и увеличение общего числа телевизионных каналов (как в США), и разнообразие телеопросов, и продумывание тем опросов на каждом из них.

4.3.7.2 Все яйца в одной корзине

При традиционном способе реализации услуг каждая АТС выполняет функции предоставления услуг только своим абонентам. IN дает возможность централизованного управления услугами. При нарушении связи SSP с SCP (или выходе SCP из строя) абонентам соответствующей части сети (или всей сети связи) будет отказано во всех услугах IN. Для обеспечения постоянной доступности услуг приобретает важнейшее значение проблема надежности сети.

Чтобы решить эту проблему, совершенно необходимо обеспечить резервные сигнальные маршруты между SSP и SCP, а сами SCP должны быть сгруппированы в связанную(ые) пару(ы) так, чтобы каждый из них мог взять на себя обслуживание абонентов всей сети при выходе из строя другого. Целесообразно иметь в сети, как минимум, два SSP и по одному тракту от каждой оконечной АТС для связи с каждым из них.

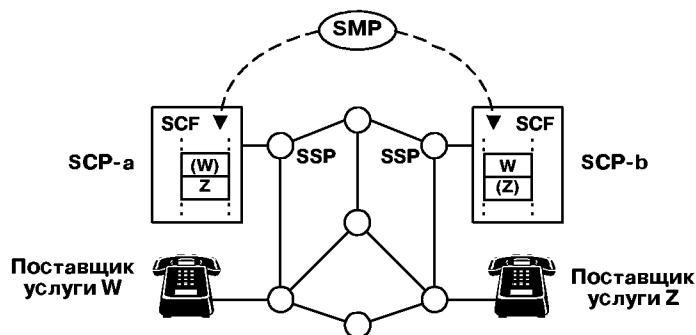


Рис. 4.3.10 Повышение надежности

При объединении SCP в связанные пары оба члена одной пары содержат идентичные данные, которые одновременно обновляются из SMP. Если в сети предоставляются, например, две услуги, то один из SCP может использоваться в качестве основного для первой ус-

луги и в качестве резервного для второй, а другой – наоборот. Транзитный пункт сигнализации STP сети ОКС обеспечивает необходимые функции маршрутизации, а при отсутствии STP организуются прямые звенья сигнализации для каждой связи SSP-SCP. Таким образом, наряду с обеспечением надежности достигается равномерное разделение нагрузки между SCP (см. рис. 4.3.10).

Одно из преимуществ IN – возможность уменьшения сроков и значительное упрощение самого процесса создания услуг. Однако все, что можно сделать быстро и просто, сопряжено с определенным риском. Можно ожидать большого количества разных вариантов реализации в разных сетях одних и тех же услуг (большого числа диалектов), которые, кроме того, могут быть определенным образом модифицированы под требования разных абонентов.

Рост количества услуг при отсутствии стандартных принципов их взаимодействия (не говоря уже об отдельных диалектах) приводит к необходимости спецификации правил разрешения конфликтных ситуаций. Классическим примером нежелательного взаимодействия между услугами является шутка абонента с заказом на свой номер услуги автоматической побудки и с одновременным заказом безусловной переадресации вызова на номер своего «приятеля». Другой пример – запрос услуги с дополнительной оплатой во время пользования услугой бесплатной связи.

4.3.8 Начисление платы за услуги и взаиморасчеты

4.3.8.1 Сценарии начисления платы за пользование услугами

Процесс начисления платы за услуги связи очень трудно поддается стандартизации. Не явилась исключением и ситуация с услугами IN. ETSI предложено несколько вариантов поддержки в INAP для CS-1 функций начисления платы, описание которых вынесено в приложение В к стандарту ETS 300 374-1. Хотя перечень сообщений и параметров для поддержки вынесенных в приложение сценариев начисления платы определен в основной части стандарта, детали кодирования параметров оставлены на усмотрение администрации каждой сети IN.

В стандарте рассмотрены следующие процессы, относящиеся к системе начисления платы за услуги IN:

- процесс определения условий (DET) отвечает за определение плательщика (вызывающая сторона, абонент услуги, или оба), базовой цены (тарифа), услуг и функций, подлежащих оплате;
- процесс генерации (GEN) осуществляет генерацию импульсов учета, сигналов с данными о начисляемой плате для процессов,

работающих в реальном времени, или учетной информации для процессов, не работающих в реальном времени;

- процесс регистрации (REG) обеспечивает создание учетных записей о вызовах или управление счетчиками тарифных импульсов.

Выбор сценария начисления платы предоставлен оператору сети. Стандартом ETSI выделены четыре следующих сценария (в таблице 4.3.5 приведены те из них, которые поддерживаются протоколом INAP-R):

Сценарий 1: начисление платы, полностью осуществляемое ТФОП с использованием собственных механизмов учета (например, подсчет тарифных импульсов с определением тарифа по коду доступа к услуге).

Сценарий 2: начисление платы, полностью проводимое средствами сети IN. Коммутационные станции ТФОП определяют, например, по коду услуги, что активизировать функции начисления платы не нужно, поскольку оно выполняется средствами IN. Управление начислением платы всегда ведется в SCP, но регистрация учетных записей о вызовах может проводиться, по разным сценариям, либо в SSP (сценарий 2.2), либо в SCP (сценарий 2.3), либо и в том, и в другом (сценарий 2.4).

Сценарий 3: начисление платы совместно средствами IN и ТФОП. В этом случае SCP сообщает SSP о необходимости передачи информации об оплате (сценарий 3.2). SSP, используя возможности сигнализации, передает команду к оконечной АТС, которая, в свою очередь, запускает тарифный счетчик или создает стандартную запись учета стоимости.

Сценарий 4: начисление платы в SCP при поддержке SSP. SCP имеет всю необходимую учетную информацию и инструктирует SSP, каким образом следует произвести расчет стоимости для конкретного вызова. Инструкции содержат сведения об условиях, в отношении которых SSP должен запросить от SCP дополнительную информацию (например, о превышении заданного лимита). Регистрация начисленной платы ведется либо в SCP (сценарий 4.1), либо в SSP (сценарий 4.2).

Дополнительные сценарии, помеченные в таблице 4.3.5 как INAP-R, необходимы для случая, когда SSP и SCP одной сети IN находятся в собственности разных операторов. В этом случае необходимо иметь учетную информацию как в SSP, так и в SCP. Сценарии INAP-R при расчете по картам, с одной стороны, позволяют независимо начислять плату по тарифам, заложенным в SSP каждого оператора, и, с другой стороны, вести в SCP контроль кредита в базе данных.

Таблица 4.3.5

Сценарии	DET	GEN	REG	Операции INAP
2.3	SCF	SSF	SSF	<i>FurnishChargingInformation</i>
3.2	SCF	SSF	ТФОП	<i>SendChargingInformation (Charge/No-Charge)</i>
4.2	SCF	SSF	SSF	<i>FurnishChargingInformation, ApplyCharging/ApplyChargingReport</i>
INAP-R	SSF+SCF	SSF	SSF	<i>FurnishChargingInformation, ApplyCharging/ApplyChargingReport</i>
INAP-R	SSF	SSF	SSF+SCF	<i>FurnishChargingInformation, ApplyCharging/ApplyChargingReport</i>
INAP-R	SCF	SSF	SSF+SCF	<i>FurnishChargingInformation, ApplyCharging/ApplyChargingReport</i>

Пример сценария с начислением платы, проводимым в SSP под управлением SCP, приведен на рисунке 4.3.11. Параметры начисления платы в операциях INAP-R отражают специфическую для каждой из услуг IN информацию и передаются в операциях *FurnishChargingInformation (FCI)*, *ApplyCharging/ApplyChargingReport (AC/ACR)*, *ActivateServiceFiltering (ASF)*. Передаются следующие параметры:

- Идентификатор услуги (*InServiceIdentity*);
- Сторона, оплачивающая пользование услугой (*ChargedParty*);
- Добавка к стоимости (*Surcharge*);
- Класс тарифа (*TariffRegimeCode*);
- Модулятор частоты тарифных единиц (*RateModulator*);
- Информация, специфическая для услуги (*ServiceSpecificInfo*).

Система расчетов за услуги IN должна быть близка к существующей системе расчетов и учитывать требования, возникающие при внедрении услуг IN. В записях AMA, служащих основой для начисления платы и для выписки счетов, целесообразно при регистрации услуг IN в SSP фиксировать следующую информацию:

- продолжительность пользования услугой;
- дату;
- номер тарифной зоны;
- сторону, оплачивающую пользование услугой;
- логический номер (набранный номер);
- физический номер вызываемого абонента, используемый в SSP для установления соединения;
- номер вызывающего абонента;
- изменение тарифа;
- тип и размер дополнительной платы за услугу PRM;

- специальную информацию (например, номер счета для оплаты услуг расчета по виртуальным картам ACC/CCC).

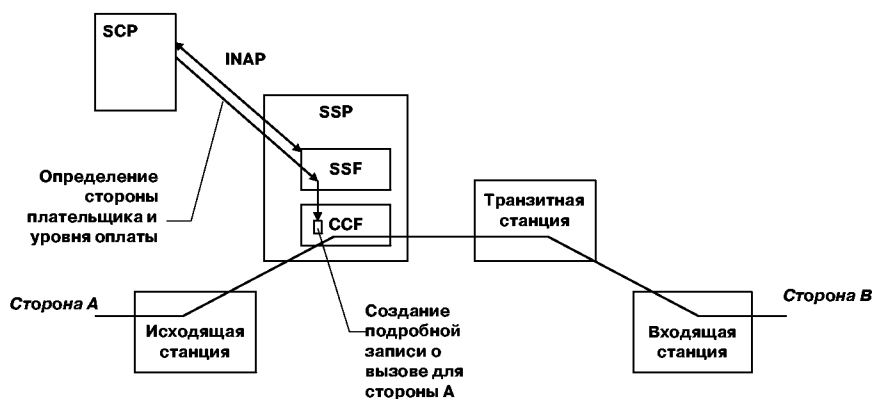


Рис. 4.3.11 Начисление платы в SSP

Плата за услугу складывается из двух частей: стоимость вызова IN, учитывающая установление соединения в ТФОП, и стоимость самой услуги. После вычислений, проведенных биллинг-центром на основании обработанных учетных записей, производится выписка счетов всем участникам связи.

4.3.8.2 Принципы взаиморасчетов

В биллинг-центр передаются учетные записи, различающиеся по функциям. На рисунке 4.3.12 показан пример процедуры расчета, когда в биллинг-центр поступают три файла с разными данными.

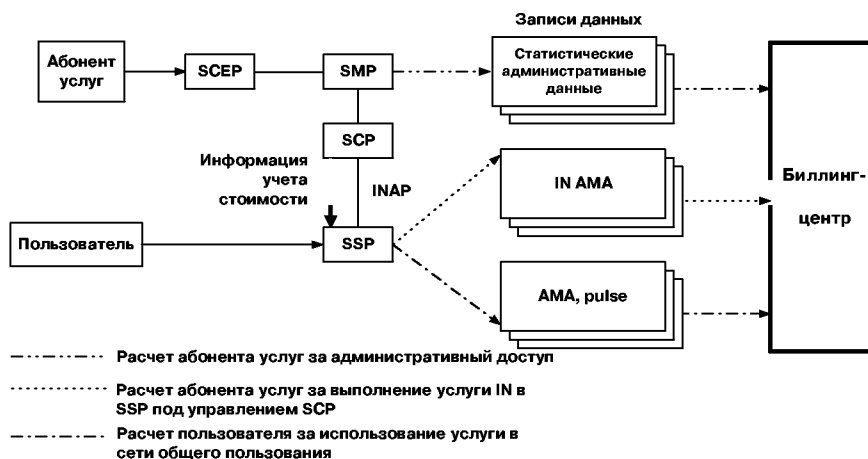


Рис. 4.3.12 Процедура расчета за услуги IN

Первый файл содержит данные с подробными учетными записями АМА или данные счетчиков, накапливаемыми на станции. Второй файл содержит записи АМА, предназначенные для поставщика услуг IN. В отличие от обычных записей они содержат дополнительные реквизиты (например, код услуги, плательщик, дополнительные характеристики и данные, которые используются при последующей обработке). Третий файл содержит статистические и административные данные для расчета с абонентом услуг за административный доступ. Например, абоненту нужно изменить требования к услуге, скажем, изменить план нумерации или расписание предоставления услуг. Для этого абонент услуги инициирует запрос к SMP, выбирая соответствующий компонент уровня услуг. Результат выполнения каждого запроса предоставляется клиенту.

Система взаиморасчетов между субъектами, вовлеченными в процесс предоставления услуг IN, опирается на гибкую стратегию учета стоимости, поддерживаемую протоколом INAP-R. Для необходимых ему расчетов поставщик услуги может заключить прямые договоры со всеми операторами и пользователями услугами. Однако возможен и более сложный случай, когда поставщик услуги заключает договор с одним оператором, а пользователи услугами находятся в разных регионах. Тогда оператор, который заключил договор с поставщиком услуги, предъявляет счета за пользование услугой административным связям других регионов. В свою очередь, этому оператору должен выставить счет поставщик услуги.

Взаиморасчеты между всеми участниками сильно зависят от организации взаимодействия между операторами. Следует стремиться к тому, чтобы взаиморасчеты за предоставление услуг IN максимально следовали существующим принципам. Тогда новые участники процесса взаимодействия будут лишь добавлять новые звенья в установившуюся цепочку взаиморасчетов с учетом конкретных услуг. Так, например, в эту цепочку могут включаться взаиморасчеты между пользователями или абонентами услуги и операторами IN или операторами сети связи.

При разработке системы взаиморасчетов за основу целесообразно принять принцип долевого участия в процессе предоставления услуг. Плата за пользование платформой может взиматься с абонента услуги или пользователя услугой либо за каждое обращение к ней, либо одновременно в момент инсталляции услуги для абонента оператором IN, либо в виде абонентной платы. Сумма, полученная за оплату пользования базой данных, составляет доход оператора IN.

Возможен вариант оплаты услуг IN как обычных вызовов, что особенно привлекательно для абонентов услуг и для пользователей услугами. В этом случае оператор сети увеличивает свой доход за счет обслуживания трафика IN и, по договоренности с оператором IN, расплачивается с ним пропорционально обслуженному трафику. Основанием для взаимных расчетов могут служить подробные учетные записи и статистические данные с определенным набором параметров для каждой услуги.

Глава 4.4

Тестирование протокола INAP

4.4.1 Принципы и архитектура аттестационного тестирования

Одним из наиболее значительных отличий современных систем связи, к которым, безусловно, относятся и программно-аппаратные комплексы Интеллектуальных сетей, от систем предыдущего поколения является применение сложных коммуникационных протоколов, в число которых входит стек протоколов общеканальной системы сигнализации ОКС-7 и сопутствующие прикладные протоколы.

В условиях, когда на рынке средств связи присутствует несколько поставщиков, решающим фактором, определяющим возможность совместной работы коммутационного оборудования разных производителей, является совместимость коммуникационных протоколов ОКС-7. В связи с этим особо актуальной становится задача стандартизации и создания специализированных инструментальных средств для тестирования разных реализаций протокола с целью определить, отвечает ли та или иная реализация требованиям национальных и/или международных спецификаций.

Сложность этой задачи обусловлена и сложностью самих задаваемых стандартами протоколов, и допускаемой стандартами многовариантностью реализаций протоколов, и рядом ограничений, которые накладывают на возможность и способы тестирования особенности тестируемых реализаций. Корректность взаимодействия коммутационных систем одного и того же производителя означает всего лишь то, что протоколы, реализованные в оборудовании этого производителя, работоспособны. Но для корректного взаимодействия систем разных производителей нужно, чтобы реализация протоколов в каждой из них соответствовала эталонным спецификациям, определенным международными или национальными стандартами.

Система сигнализации ОКС-7 представляет собой структурированный по уровням протокол. Распределение функций по уровням системы, в основном, соответствует семиуровневой модели Взаимодействия открытых систем (ВОС). Для проверки пригодности разных реализаций протоколов ОКС-7 к совместной работе в одной сети сигнализации требуется их аттестационное тестирование (conformance testing). Общие принципы аттестационного тестирования протоколов ВОС сформулированы в рекомендации ITU-T X.290. Рекомендация делит все требования к соответствию

реализации протокола ВОО его спецификации на две группы – требования к статическому соответствию и требования к динамическому соответствию.

В качестве документа, содержащего информацию производителя о том, каким именно требованиям к статическому соответствию удовлетворяет представляемая этим производителем реализация протокола, используется *ведомость соответствия протоколу (PICS)*. Требования к динамическому соответствию, составляющие основную часть рекомендаций по протоколам ВОО, задают набор вариантов допустимого поведения реализации протокола. Аттестационное тестирование предусматривает проверку того, что тестируемая реализация отвечает требованиям как к статическому, так и к динамическому соответствию. При положительном результате проверки система считается аттестованной.

Тесты поведения предусматривают проверку того, как реагирует тестируемая реализация на разрешенные (предусмотренные протоколом) и на не разрешенные (синтаксически неправильные или не предусмотренные протоколом) воздействия со стороны тестирующей системы. Тесты поведения определяются в форме *абстрактных тестовых комплектов*, представляющих собой некоторые наборы *тестовых сценариев*. Каждый тестовый сценарий предназначен для проверки отдельного четко сформулированного свойства, конкретной способности или определенного характера поведения тестируемой реализации в строго оговоренных условиях.

Модель ВОО для применений ITU-T предполагает, что стандартизации подлежит только внешнее поведение реальных открытых систем. Поэтому спецификации протоколов ВОО описывают поведение любого логического объекта протокола в терминах абстрактных сервисных примитивов (ASP – Abstract service primitive) на границах с уровнями, расположенными выше и ниже этого объекта. Сказанное определяет и концептуальную архитектуру тестирования, основанную на представлении тестируемой реализации в виде «черного ящика», а тестирования – в виде процесса активных воздействий тестера на этот «ящик» с проверкой его реакции на каждое воздействие. Воздействия и проверка реакции осуществляются в точках контроля и наблюдения PCO (Point of control and observation) на границах с уровнями, расположенными под тестируемой реализацией и над ней. Концептуальная архитектура тестирования протоколов ВОО, применимая и к стеку протоколов ОКС-7, приведена на рисунке 4.4.1.

Тестовая архитектура, состоящая из тестера, тестируемой реализации IUT (Implementation under test), тестового контекста, точек контроля и наблюдения PCO и точек доступа к реализации, представляет собой описание среды, в которой проводится проверка реализации протокола. Тестер выполняет эксперименты над системой и наблюдает за результатами.

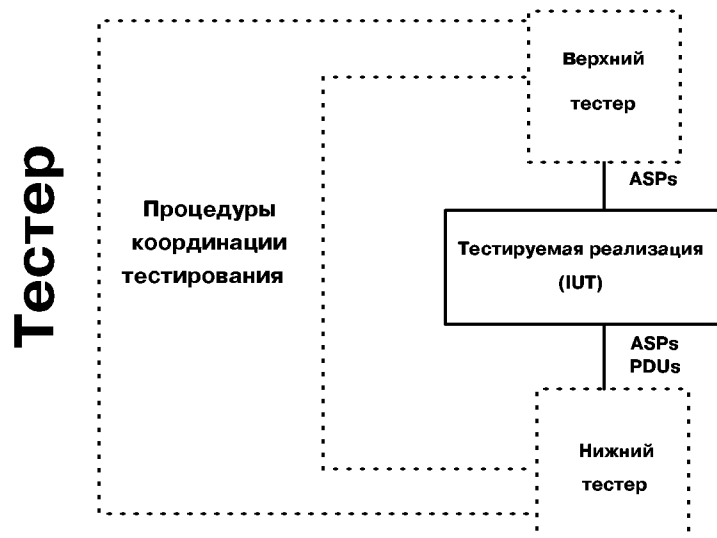


Рис. 4.4.1 Концептуальная архитектура тестирования

В тестере выделяются две части, именуемые нижним и верхним тестером. *Нижний тестер* (LT) абстрактно определяется как средство, которое в процессе тестирования воздействует на тестируемую реализацию и наблюдает за ней в PCO, находящейся на нижней границе этой реализации. Определяемый подобным образом *верхний тестер* (UT) взаимодействует с тестируемой реализацией в PCO на верхней границе этой реализации. При необходимости верхний и нижний тестеры взаимодействуют между собой с помощью процедур координации тестирования (TCP – Test coordination procedures).

Нижний тестер сложнее верхнего, поскольку он выполняет функции контроля и наблюдения за протокольными блоками данных (PDU – Protocol data units). Эти блоки являются частями абстрактных примитивов ASP, которые нижний тестер передает и принимает во время выполнения теста.

Для тестирования конкретной системы специфицируются последовательности тестовых событий, воздействующих на эту систему и ожидаемых в качестве ее реакции на воздействия. Последовательность таких событий, исчерпывающая определенную задачу тестирования, представляет собой отдельный *тест* (test case). Полный набор тестов для тестирования определенного протокола называется *комплексом тестов* (test suite).

Воздействие на тестируемую реализацию протокола и наблюдение за ее реакцией могут производиться либо локально, когда PCO, доступные тестеру, находятся внутри тестируемой системы на верхней и нижней границах (в смысле модели BOC) тестируемой реали-

зации, либо извне, например, через звено ОКС-7. Для тестирования реализаций открытых систем определены следующие четыре категории методов:

- a) методы локального (local) тестирования;
- b) методы внешнего распределенного (external distributed) тестирования;
- c) методы внешнего координированного (coordinated) тестирования;
- d) методы внешнего дистанционного (remote) тестирования.

Методы локального тестирования характеризуются, как уже было сказано, тем, что РСО, к которым подключаются верхний и нижний тестеры, располагаются внутри системы на границах тестируемой реализации. Все методы внешнего тестирования ориентированы на то, что связь между нижним тестером и тестируемой системой поддерживается услугами нижних уровней, предоставляемыми другой системой, причем подразумевается, что названные услуги достаточно надежны. Методы различаются возможностями доступа к верхней границе тестируемой реализации, требованиями к верхнему тестеру и требованиями к процедурам координации действий верхнего и нижнего тестеров. Методы b) и c) требуют, чтобы функции верхнего тестера были вполне определенными, методы d) этого не требуют. Методы c) требуют координации действий верхнего и нижнего тестеров с использованием протокола административного управления тестированием, методы b) и d) не предусматривают процедур координации. Методы b) предполагают наличие доступа к верхней границе тестируемой реализации, методы c) и d) такого доступа не требуют.

4.4.2 Архитектура и методы тестирования INAP

Стандартом ETS 300 374-9 для тестирования INAP предлагаются метод внешнего распределенного и метод дистанционного тестирования. Первый метод основан на использовании в верхнем тестере таких тестовых сценариев, которые предусматривают считывание из базы данных определенных параметров, интерпретируемых затем в действиях протокола. База данных может быть частью SCP. Преимущество этого метода состоит в том, что он не требует использования логики услуг и, следовательно, оборудования SCEP, когда нужно модифицировать тестовый сценарий. Недостатком метода является использование специального протокола передачи информации, необходимой для конфигурирования базы данных в соответствии с тестовым сценарием.

При использовании второго метода тестовые сценарии состоят из поставляемых в составе SCP элементарных программ SLP, выполняю-

щих заданную последовательность операций INAP. Активизация и выбор SLP производится либо вручную командами MML с консоли оператора, либо на основе значения параметра Called Party Number в принятой операции *InitialDP*. Метод дистанционного тестирования представляется более предпочтительным по той причине, что специальный протокол обмена информацией для координации тестирования оказывается ненужным, так как в качестве такого протокола выступает сам INAP (совместно с TC/SCCP/MTP), связывая нижний и верхний тестеры. Кроме того, этот метод обеспечивает полную автоматизацию процесса тестирования. На рисунке 4.4.2 приведена конфигурация включения средств тестирования для проверки SCP, представляющая собой модификацию концептуальной архитектуры тестирования применительно к протоколу INAP.

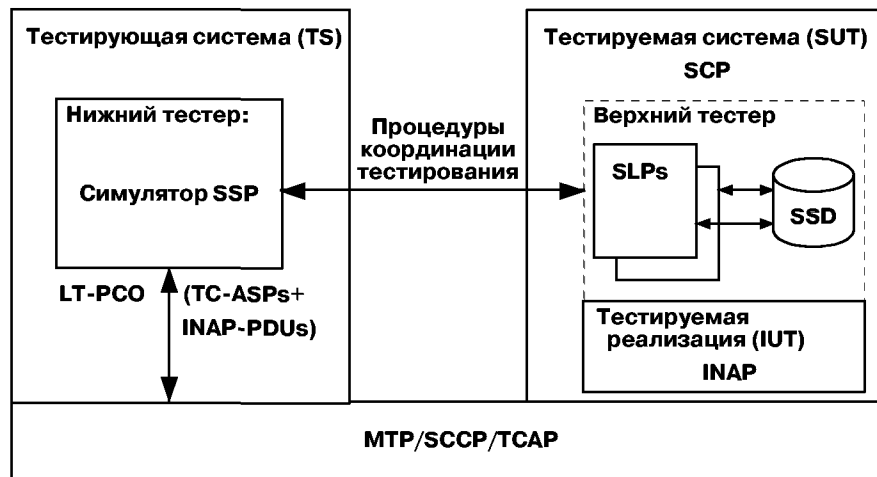


Рис. 4.4.2 Архитектура тестирования протокола INAP в SCP

Задачи тестирования протокола INAP в интерфейсе SCF-SSF/SRF требуют наличия в SCP дополнительных функций, определенных в виде программ логики услуг, которые предназначены специально для тестирования. В качестве тестирующей системы может использоваться либо реальный SSP/IP, либо симулятор. Использование реального SSP/IP позволяет тестировать обе стороны интерфейса одновременно. Однако с помощью реального SSP/IP не всегда можно выполнить все тесты, например, из-за того, что отсутствует возможность передавать сообщения, содержащие синтаксические ошибки. Преимущество использования симулятора состоит еще и в том, что он дает возможность полностью автоматизировать рутинную процедуру выполнения сотен тестовых сценариев.

В зависимости от того, насколько детально проверяется с помощью некоторого набора тестов соответствие реализации протокола ВОО его спецификациям, различаются четыре типа тестирования:

- базовые тесты взаимодействия, являющиеся средством ограниченной (как правило, предварительной) проверки соответствия;
- тесты функциональных возможностей, предусматривающие проверку того, что реализация действительно удовлетворяет требованиям к статическому соответствию, как это объявлено в PICS;
- тесты поведения, предусматривающие проверку того, удовлетворяет ли реализация всем требованиям к динамическому соответствию;
- аттестационные тесты, предусматривающие тщательную проверку соответствия реализации каждому конкретному требованию (с ответом «да»/«нет» и с соответствующей диагностической информацией); эти тесты стандартизации не подлежат.

Любой отдельный тест представляет собой определенную последовательность *тестовых шагов*, состоящих каждый из упорядоченного множества тестовых событий. Тестовые шаги, существенные для достижения цели теста и получения его результата, представляют собой *рабочую часть теста*. Шаги, необходимые для перехода из начального устойчивого состояния теста в начальное состояние его рабочей части, образуют *преамбулу*, а шаги, приводящие из конечного состояния рабочей части к конечному устойчивому состоянию теста, – *постамбулу*. Среди устойчивых состояний протокола, интерпретируемого как конечный автомат, всегда есть исходное (Idle) состояние. Для проведения любого абстрактного теста в автономном режиме преамбула должна начинаться с исходного состояния, а постамбула – оканчиваться этим состоянием. В зависимости от результата проведения каждого абстрактного теста выносится вердикт в виде заключения типа «удовлетворительно» (pass), «неудовлетворительно» (fail) или «тест не завершен» (inconclusive).

Структуры тестовых комплектов для тестирования реализаций протокола INAP в интерфейсе между SSF/SRF и SCF определены стандартами ETS 300 374-3 и ETS 300 374-9. Например, для SCF определены четыре группы функций, подлежащие тестированию: группа базовых функций, группа функций поддержки в SSF трансляции в направлении к SRF, группа функций поддержки ассистирования и группа функций поддержки прямой связи с SRF. Структура тестового комплекта для каждой из этих групп функций включает в себя следующие основные группы тестов:

- выполняемые в первую очередь базовые тесты взаимодействия;
- тесты соответствия фактических функциональных возможностей тем возможностям, которые заявлены в PICS;
- тесты реакции автоматов SCF-FSM и SCME-FSM на корректные воздействия;

- тесты реакции на синтаксически некорректные сообщения;
- тесты реакции на возникновение нештатных ситуаций.

Большинство рекомендованных для интерфейса SCF-SSF/SRF тестов предполагает вызвать такое поведение SCF, которое приведет к передаче с его стороны соответствующего сообщения. Например, проверка способности SCF генерировать операцию *RequestReportBSCMEvent* после приема операции *InitialDP* требует наличия в нем специальной логики, могущей вызвать такое событие, которое заставит протокол INAP передать ответную операцию. Чтобы запустить логику поддержки того или иного сценария тестирования, стандартом ETS 300 374-9 определено использование специальных символических значений параметра Called Party Number в операции *InitialDP*.

4.4.3 Особенности тестирования INAP-R

Внедрение IN в России осуществляется на базе российских спецификаций протокола INAP-R. Многие тесты ETSI могут использоваться для тестирования российской версии протокола INAP без каких-либо модификаций, некоторые из них не применимы для тестирования INAP-R, а для тестирования отдельных специфических для России функций, например, операций, связанных с начислением платы, требуется создание новых групп тестов.

Необходимость создания таких групп тестов для INAP-R вызвана тем, что содержимое операций, связанных с начислением платы, стандартом ETSI не специфицировано – аргументы этих операций описаны в виде октетной строки, а их содержание определяется национальными спецификациями. В INAP-R для названных операций специфицирован ряд обязательных и необязательных параметров. В таблицах 4.4.1 и 4.4.2. приведены два примера тестовых сценариев для операции *ApplyCharging*.

Стандартный набор тестов не предусматривает проверку этих параметров, а ограничивается проверкой типа аргумента. Для тестирования реализаций протокола на соответствие спецификациям INAP-R должны быть разработаны специальные тестовые сценарии, предусматривающие проверку содержимого аргумента в операциях, которые связаны с начислением платы. Новые группы тестов должны обеспечивать проверку параметров операций: *ApplyCharging*, *ApplyChargingReport*, *ActivateServiceFiltering*, *ServiceFilteringResponse*, *FurnishChargingInformation*, *SendChargingInformation*.

Нетрудно видеть, что любая версия реального средства тестирования протокола сигнализации будет реализацией некоторого нового протокола, представляющего собой расширение протоко-

ла, для тестирования которого это средство предназначено. Указанное расширение протокола, реализуемого средством тестирования, следует понимать так, что это средство, кроме нормальных протокольных процедур (режим эмулятора), должно еще распознавать и идентифицировать определенные нештатные для тестируемого протокола ситуации (режим монитора), а также иметь возможность создавать их (режим симулятора). Такого рода расширение, фактически являющееся спецификацией протокол-тестера, к настоящему моменту никакими международными стандартами не определено.

Таблица 4.4.1

ТЕСТ 1:	
Предварительное условие:	Со стороны SSP к SCP передана операция <i>InitialDP</i>
Условие окончания:	Прием операции <i>ReleaseCall</i> со стороны SCP и завершение диалога
Сценарий:	Убедиться, что тестируемый объект SSP, находящийся в состоянии "Ожидание инструкций": 1. Принимает операцию <i>ApplyCharging</i>, содержащую только обязательные параметры: • <i>AchBillingChargingCharacteristics</i>, содержащий параметр <i>callSupervision</i> с входящим в него параметром <i>supervisionMethod</i> с параметром <i>unitsGranted</i> с максимальным значением в соответствии с INAP-R; • <i>SendCalculationToSCFIndication</i> со значением <i>TRUE</i>. 2. Принимает операцию <i>Connect</i>, содержащую только обязательные параметры. 3. Не сообщает ни о каких ошибках, отказах в приеме запросов или прерываниях диалога в течение времени, отведенного на операцию.

Данное обстоятельство порождает следующую проблему: одна и та же реализация протокола, тестируемая с помощью одного и того же тестового комплекта, но разными тестовыми средствами, может получить разные оценки соответствия. Это неизбежно ставит вопрос о необходимости стандартизации инструментальных средств тестирования, что проще всего обеспечить, если создание таких средств идет одновременно с разработкой соответствующих спецификаций протокола. Данное условие удалось соблюсти при создании протокол-тестера STA-7, коммерчески доступного средства для анализа и тестирования российской версии подсистем и прикладных протоколов ОКС-7, разработка которого начата еще во время обсуждения национальных спецификаций INAP-R в Минсвязи РФ.

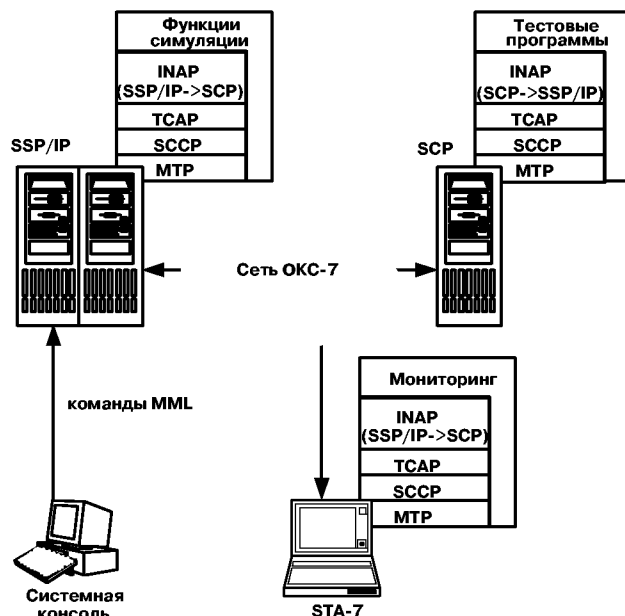
Таблица 4.4.2

ТЕСТ 2:	
Предварительное условие:	Со стороны SSP к SCP передана операция <i>InitialDP</i>
Условие окончания:	Прием операции <i>ReleaseCall</i> со стороны SCP и завершение диалога
Сценарий:	Убедиться, что тестируемый объект SSP, находящийся в состоянии "Ожидание инструкций": 1. Принимает операцию <i>ApplyCharging</i>, содержащую обязательные и необязательные параметры: • Обязательный параметр <i>aChBillingChargingCharacteristics</i>, содержащий: • обязательный параметр <i>callSupervision</i> с входящими в него параметром <i>supervisionMethod</i> с параметром <i>unitsGranted</i> с максимальным значением в соответствии с INAP-R и необязательным параметром <i>heartBeat</i> с минимальным значением в соответствии с INAP-R; • необязательные параметры: <i>TariffRegimeCode</i>, <i>ChargeRateModulator</i>. • Обязательный параметр <i>sendCalculationToSCFIndication</i> со значением TRUE. • Необязательный параметр <i>partyToCharge</i> с <i>legID</i>, содержащий параметр <i>sendingSideID</i> со значением <i>leg1</i> (1). 2. Принимает операцию <i>Continue</i>. 3. Не сообщает ни о каких ошибках, отказах в приеме запросов или прерываниях диалога в течение времени, отведенного на операцию.

На этапе системного проектирования прибора были определены наиболее характерные для средств тестирования протоколов ОКС-7 функциональные возможности:

- наличие аппаратных средств поддержки стандартных интерфейсов в виде сменных модулей;
- возможность одновременного мониторинга нескольких протоколов в разных интерфейсах;
- наличие пакетов стандартизированного тестового программного обеспечения;
- наличие средств ввода и редактирования тестовых сценариев через графический интерфейс пользователя и средств трансляции сценариев в выполнимые тесты;
- удобный пользовательский интерфейс и возможность сохранения данных тестирования на дисках и вывода их на принтер;

- возможность дистанционного управления тестером, обеспечивающая объединение нескольких тестеров в одном тестовом стенде.



**Рис. 4.4.3 Использование STA-7
в качестве анализатора протокола INAP**

Именно по таким требованиям разработана система тестирования и анализа протоколов сигнализации STA-7 (System for Testing and Analyzing of CCS-7 protocols). Прибор хорошо зарекомендовал себя при разработке программного обеспечения ОКС-7 в целом ряде отечественных и импортных коммутационных станций, у эксплуатационного персонала цифровых систем коммутации и операторов сетей при проведении приемо-сдаточных и сертификационных испытаний оборудования, имеющего функции общеканальной системы сигнализации.

Предусмотрена возможность подключения STA-7 параллельно тракту ИКМ для работы в режиме мониторинга (рис. 4.4.3) и возможность подключения его в режиме эмуляции и симуляции SSP или SCP (рис. 4.4.4). Программные опции монитора подсистем MTP, ISUP, SCCP, TC, INAP, эмулятора подсистемы MTP, симуляторов подсистемы ISUP и прикладного протокола INAP реализованы как в соответствии с требованиями российских спецификаций, так и в соответствии с рекомендациями ITU-T и со стандартами ETSI. Дополнительно прибор оснащается опциями мониторинга/симуляции системы сигнализации DSS1 и интерфейса V5.1/V5.2; при этом обеспечивается возможность одновременного наблюдения за корректностью взаи-

модействия нескольких систем сигнализации в произвольном их сочетании.

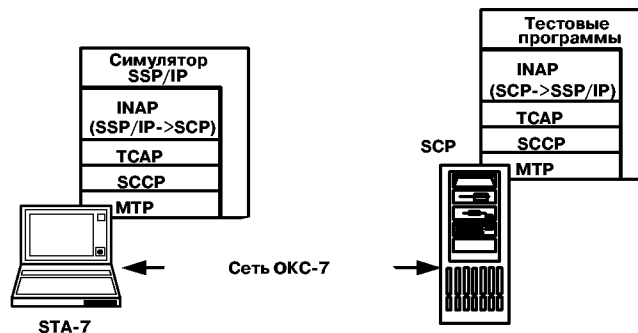


Рис. 4.4.4 Использование STA-7 в качестве симулятора SSP/IP

В режиме анализатора STA-7 производит мониторинг и запись блоков данных соответствующего протокола, передаваемых через межстанционный интерфейс, и отображает их на экране прибора в расшифрованном виде с возможностью гибкого конфигурирования режимов тестирования разных подсистем. Пользователь имеет возможность с помощью меню включать необходимые фильтры и выбирать характеристики вывода информации для каждого протокола: в шестнадцатеричных кодах, в сокращенной или в подробной форме представления.

В режиме симуляции с помощью готовых наборов тестовых сценариев можно моделировать последовательности сообщений для проверки корректности реализации протокола в тестируемой системе. При обнаружении некорректной реакции тестируемой системы на экран прибора выводятся соответствующие сообщения. Пользователю предоставляется возможность создания собственных тестовых сценариев. Результаты тестирования могут сохраняться в текстовых файлах или выводиться на принтер.

Пользователь может контролировать процесс тестирования, производить необходимые настройки (выбор сигнального канала, редактирование тестовых сценариев и т.д.), получать контекстную помощь как в отношении специфики тестируемых протоколов, так и в отношении самой системы тестирования, сохранять и распечатывать результаты тестов. Программное обеспечение разработано с учетом современных требований к интерфейсу пользователя и может быть установлено на компьютеры с операционными системами Windows/NT и Windows'95.

Многоуровневое пользовательское меню обеспечивает введение управляющих команд с помощью манипулятора «мышь» или с клавиатуры. Большинство команд доступно также через «горячие клавиши», справка о которых находится рядом с соответствующим пунк-

том меню. По поводу любого пункта меню можно получить быструю справку. В интерфейс встроены контекстные подсказки, касающиеся всех сообщений и параметров подсистем ОКС-7 и прикладных протоколов, а также программа, обучающая пользователя работе с системой STA-7 на английском или русском языках.

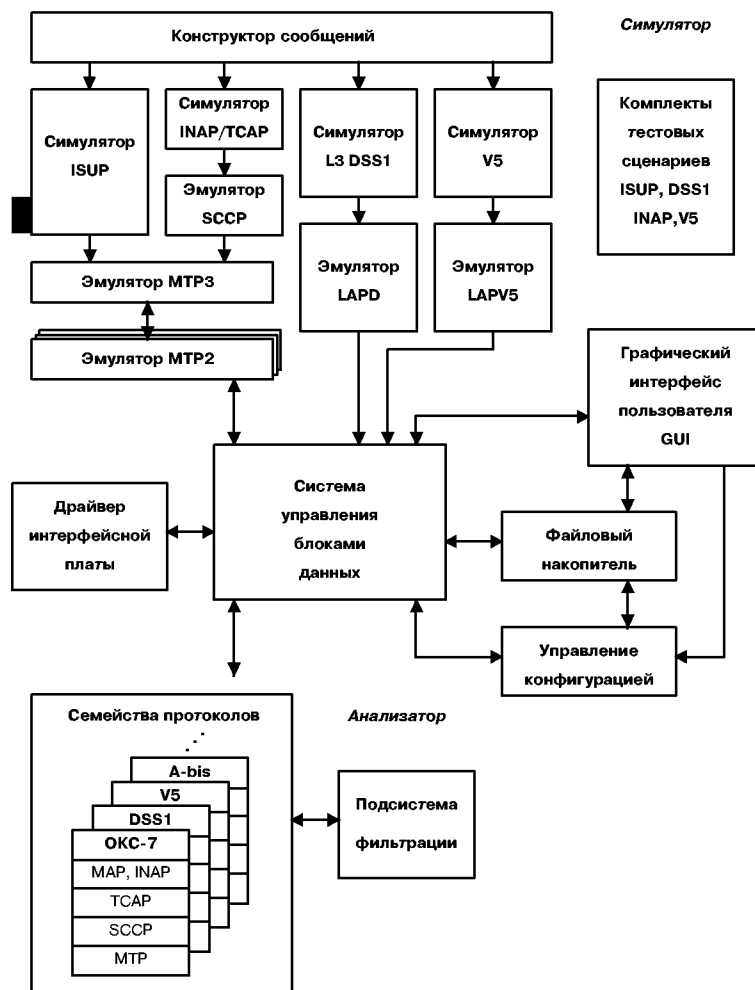


Рис. 4.4.5 Структура программного обеспечения STA-7

Структура программного обеспечения STA-7 представлена на рисунке 4.4.5. Драйвер интерфейсных плат работает под одной из операционных систем Windows95, WindowsNT, OS/2 или Linux. Система управления файлами, в соответствии с заданной пользователем через графический интерфейс конфигурацией прибора, пропускает

принятые драйвером по линии протокольные блоки данных в файло-
вый накопитель, где организован кольцевой буфер, достаточный для
хранения и последующей выборки данных, принятых по линии в те-
чение 3-х суток. Принимаемые в текущий момент данные динамиче-
ски выводятся на экран прибора.

Программный блок анализатора содержит средства расшифров-
ки и анализа принимаемых данных в соответствии с выбранными
протоколами. Подсистема фильтрации выбирает из всех хранящих-
ся в файловом накопителе данных те, которые требуется вывести че-
рез графический интерфейс для представления пользователю. Со-
общения и параметры, не соответствующие требованиям анализи-
руемого протокола, выделяются на экране другим цветом. Графиче-
ский интерфейс пользователя при работе прибора в качестве анали-
затора протокола INAP приведен на рисунке 4.4.6.

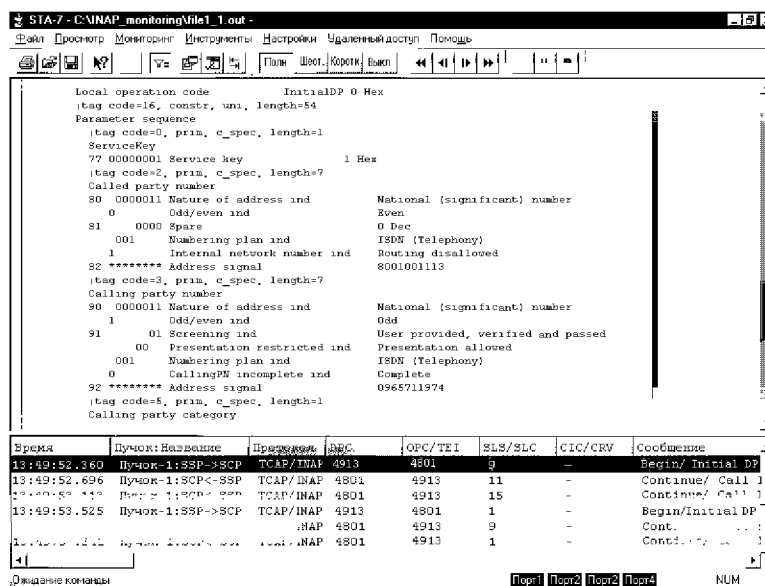


Рис. 4.4.6 Графический интерфейс пользователя STA-7
в режиме мониторинга протокола INAP

Программный блок симулятора содержит эмуляторы нижних уров-
ней протоколов разных семейств, необходимые для запуска комплек-
тов тестовых сценариев имитируемых протоколов, поставляемых с
прибором. С помощью конструктора сообщений пользователю пре-
доставляется возможность создавать собственные сообщения и но-
вые тестовые сценарии. Результаты автоматически прогоняемых
комплектов тестовых сценариев сохраняются в файле в виде стан-
дартного отчета.

Часть 5

Перспективы

Глава 5.1

Направления эволюции концепции IN

5.1.1 Чего нет в CS-1?

Предыдущие главы были посвящены тому, что дает IN и набор возможностей CS-1, как первый этап продвижения к долговременной целевой архитектуре IN, которая обеспечит реализацию всех свойств, декларируемых ее концепцией, – независимость платформы от услуг, независимость реализации услуг от архитектуры сети и независимость от производителя оборудования. Отметим, чего же нет в CS-1, и как это отражается на практическом внедрении платформы у операторов сетей связи.

В наборе CS-1 не стандартизованы интерфейсы между SCEP и SMP, а также интерфейсы SMP-SSP и SMP-SCP. Следствием этого является то, что оборудование для создания услуг, для их загрузки в SCP, как и сам SCP, должны быть изготовлены одним производителем.

Не стандартизовано взаимодействие между SCP разных сетей. Данное обстоятельство не позволяет использовать услуги, абоненты которых и/или пользователи которыми находятся в разных сетях IN. Не поддерживаются услуги, которые требуют управления несколькими участниками связи, из-за чего отсутствует возможность динамически подключать и отключать участников многосторонней конференцсвязи.

Отсутствует поддержка функций мобильности терминала. Это связано с тем, что в CS-1 не обеспечена поддержка «роуминга» услуг, совершенно необходимая для абонентов сетей подвижной связи. Работы в этом направлении ведутся ETSI, однако исчерпывающее решение проблемы ожидается только в CS-4.

Решение задач поддержки широкополосных услуг и услуг мультимедиа также ожидается только в CS-4. Модель базового процесса обслуживания вызова (ядро концепции IN) для широкополосных сетей заметно отличается от модели, использованной в CS-1 и CS-2 и ориентированной на возможности существующих узкополосных коммутационных систем стационарных сетей связи. Кроме того, спецификации систем сигнализации для широкополосных сетей ISDN (B-ISDN – Broadband ISDN) лишь сравнительно недавно стали обретать ту стабильность, которая необходима для использования в качестве основы новой модели процесса обслуживания вызова.

5.1.2 IN и современные технологии

Следующие за CS-1 наборы возможностей призваны заполнить некоторые из отмеченных выше пробелов. В утвержденном в 1997 году пакете рекомендаций по набору CS-2 учтен основной недостаток его предшественника – невозможность взаимодействия между разными сетями IN – и специфицирован интерфейс между SCP разных сетей, что позволяет реализовать функциональные возможности, обеспечивающие предоставление услуг CS-1 на международном уровне.

При переходе к международному уровню значительно усложняются вопросы стандартизации протокола INAP. Предоставление международных услуг предполагает совместную деятельность нескольких операторов, что требует организации между ними альянсов. В каждой стране должна быть организована регистрация «окончаний» международных услуг. Регистрация услуги международной бесплатной связи IFS (International freephone service) в ITU-T началась с 1 февраля 1997 года.

Перечень блоков SIB в CS-2 был расширен и их описание (по крайней мере, его первая стадия) претерпело сдвиг в сторону объектно-ориентированного подхода, который теперь широко используется во всех телекоммуникационных технологиях. Для интерфейса с базами данных, наряду с протоколом INAP, в CS-2 предлагается использовать протокол доступа к директории (DAP – Directory access protocol), определенный рекомендацией X.500. Протокол DAP X.500 призван обеспечить не поддерживаемую INAP CS-1 независимость реализации SDF от параметров операций, связанных с конкретной услугой.

Для интерфейсов SSF-SCF и SCF-SCF протокол INAP был расширен средствами поддержки мобильности пользователей (но не терминалов!), обмена информацией о начислении платы и управления несколькими участниками связи и/или сегментами одного соединения. В конечном итоге, пакет рекомендаций по CS-2 фактически охватил все возможности, которые можно, используя заложенные в концепции IN принципы, реализовать на базе стационарной телефонной сети.

Бурное развитие в последние годы альтернативных (по отношению к стационарной телефонной сети) систем связи (сетей подвижной связи, сетей IP, широкополосных сетей) привлекло внимание к вопросу о применимости к ним концепции IN и/или о возможности интеграции их в глобальную сеть IN, обладающую свойствами всех конкурирующих технологий. Международные организации по стандартизации пока еще не готовы предложить единый подход, поэтому обсуждаемые в настоящее время проекты рекомендаций по CS-3

и CS-4 лишь декларируют необходимость иметь определенные функциональные средства (см. таблицу 5.1.1), но не доведены до уровня спецификаций протокола.

Таблица 5.1.1

	CS-2	CS-3	CS-4
Основные функции	Распределенная логика услуги (выполнение одной SLP несколькими SCP). Управление несколькими сегментами одного соединения. Управление несколькими участниками связи. Запрос услуги во время связи.	Несколько точек управления (при обслуживании вызова на один CCF/SSF могут влиять несколько программ SLP в разных SCP). Взаимодействие с дополнительными услугами ISDN. Управляемое взаимодействие атрибутов в CCF/SSF.	
Интерфейсы между сетями IN	SCF-SCF, SDF-SDF, SCF-SDF.	SCF-SCF, SCF-SDF, SDF-SDF и SMF-SMF	
Мобильность пользователя и терминала (динамическая активизация, деактивизация и загрузка триггерных точек)	Поддержка мобильности пользователя для услуги UPT в узкополосных сетях.	Услуги мобильности терминала для узкополосных сетей. Поддержка мобильности терминала для услуги UPT в узкополосных сетях.	Услуги мобильности для широкополосных сетей. Поддержка мобильности для услуги UPT в широкополосных сетях. Виртуальная "домашняя" среда (VHE). Управление мобильностью для IMT-2000.
Взаимодействие с IP-сетями	-	Предоставление услуги IN в телефонной сети по запросу, инициированному в сети IP.	Установление соединения между сетями IP. Запрос предоставления услуги IN в сети IP. (Поддержка пересчета адресов IP/ТФОП).
Поддержка услуг B-ISDN и систем сигнализации DSS2 и B-ISUP	-	Интерактивные. Выборки информации. (Соединения типа "точка-точка").	Интерактивные. Выборки информации. Распределительные. (Однонаправленные соединения "точка - группа точек" и двунаправленные многоточечные).
Интерфейсы эксплуатационного управления	-	SMF-CUSEF, SMF-SCF, SMF-SCEF, SMF-SDF, SMF-SMAF, SMF-SRF и SMF-SSF (CMIP/Q.812).	

Такое положение вовсе не свидетельствует о том, что выработать единый подход принципиально невозможно, а объясняется тем, как стремительно развивается отрасль электросвязи. Предложено сразу несколько (пока не совместимых) решений, каждому из которых

предстоит доказать на практике свое право считаться лучшим и послужить основой международного стандарта. Отработка принципов, которые будут заложены в спецификациях CS-3 и CS-4, ведется в рамках международных проектов одновременно в нескольких опытных сетях с участием основных производителей оборудования связи и крупнейших сетевых операторов.

Рассмотрим основные направления эволюции концепции IN.

СЕТИ ПОДВИЖНОЙ СВЯЗИ с самого начала опирались на принципы, сходные с используемыми в концепции IN, что позволило реализовать в сетях GSM стандартный перечень дополнительных услуг сетевого уровня («следуй-за-мной», «речевая почта» и др.). Произшедшая в последние годы либерализация рынка услуг связи в Западной Европе породила множество новых операторов, особенно в области мобильной связи, каждый из которых борется теперь за свою нишу в этой доходной отрасли. Многие из них небезуспешно пробуют использовать для этой цели концепцию IN.

Для обеспечения услуг IN в инфраструктуре мобильных сетей требуется ввести в модель процесса обслуживания вызова средства взаимодействия со специфическими для таких сетей компонентами, обеспечивающими отслеживание местоположения абонента. Однако предоставление услуг мобильным абонентам требует динамической загрузки, активизации и деактивизации триггерных точек в SSP той «чужой» сети, в которую переместился абонент, на основе запроса информации о текущей подписке, постоянно хранящейся в «своей» сети этого абонента.

ВСЕМИРНАЯ СЕТЬ INTERNET. В связи с ошеломляющим ростом популярности Internet, а также иных сетей, использующих протокол IP (Internet Protocol), телефония все больше и больше прибегает к их услугам. Что касается Интеллектуальной сети, то, в первую очередь, сказанное относится к услуге VPN. Возможности предоставления, наравне с телефонными услугами, услуг Internet, связанных с передачей речи, данных и факсимильных сообщений, видеоконференций и т. д., а также единая система начисления платы, весьма привлекательны для корпоративных клиентов, которые получают единую сеть с прекрасной возможностью контролировать расходы.

Внедрение оборудования, поддерживающего интерфейс с SCP по ОКС-7, позволит добиться интеграции на базе концепции IN и прозрачности функций (например, переадресации, перевода в режим ожидания) между коммутируемыми телефонными сетями и сетями IP. Кроме интерфейса с SCF оборудование будет поддерживать новые протоколы Интернет-телефонии, такие как MGCP (Media gateway control protocol), и гарантированное качество обслуживания.

ШИРОКОПОЛОСНЫЕ УСЛУГИ МУЛЬТИМЕДИА. Вместе с увеличением скоростей, доступных пользовательским терминалам (персональным компьютерам, видеотелефонам) при подключении их к аналоговой телефонной линии через модем, а также по цифровым абонентским линиям ISDN или ADSL (Asymmetric digital subscriber line), изменилось представление людей о доступных им услугах. Растет популярность и расширяется перечень услуг, связанных с передачей видеоинформации, таких как покупки по телефону, электронная коммерция, видео по требованию и видеоконференции. В борьбе с сетями Internet за рынок широкополосных услуг мультимедиа операторы и производители коммутационного оборудования рассматривают использование принципов IN в сетях B-ISDN. Организация широкополосных сетей IN требует разработки принципов создания адекватной архитектуры.

B-SSP (Broadband SSP) представляет собой широкополосный коммутатор ATM с функциями распознавания вызовов, требующих услуг IN, и взаимодействия с B-SCP (Broadband SCP), ответственным за выполнение логики услуг. Элементы широкополосной IN должны поддерживать необходимые для реализации услуг мультимедиа протоколы сигнализации. К сожалению, в ITU-T существует значительный временной разрыв между процессом стандартизации возможностей (услуг) B-ISDN и разработкой необходимых для их поддержки стандартов сигнализации. Утвержденные к настоящему времени стандарты не поддерживают манипулирование соединениями и участниками связи в сложных конфигурациях («точка – группа точек» и многоточечных). Совместное использование возможностей B-ISDN и IN способно быстро заполнить существующий пробел.

5.1.3 Гармонизация компьютерных и телекоммуникационных технологий

Работы, направленные на гармонизацию компьютерных и телекоммуникационных технологий на базе новой архитектуры программного обеспечения, с 1993 года проводятся международным консорциумом TINA-C, объединяющим более 40 производителей оборудования. Консорциум предлагает модульную объектно-ориентированную архитектуру управления, позволяющую конструировать услуги, минимально заботясь о физической конфигурации, в которой они будут развернуты. Такая гибкость способствует «свободной конкуренции» при размещении функций управления услугами между терминалом пользователя и сетевыми элементами. Наибольшее прогресс достигнут в областях управления соединениями и среды распределенной обработки (DPE – Distributed processing environment).

С самого начала концепция TINA опиралась на новейшую методологию проектирования систем связи и на технические решения, полученные в результате разработки технологий B-ISDN, TMN, IN. От Интеллектуальной сети TINA заимствовала идею разделения функций предоставления услуг и функций коммутации и принципы функционального моделирования, от концепции TMN был взят принцип деления на уровни эксплуатационного (административного) управления, от B-ISDN – принцип разделения задач управления связью пользователя и управления соединением.

Концепция управления соединениями в архитектуре TINA, в принципе, не зависит от технологии, используемой в транспортной (первичной) сети, однако нацелена, преимущественно, на стандарты сигнализации в сетях ATM (B-ISDN, фаза 3), предполагающие полное разделение процессов управления связью пользователя и соединением. DPE, предлагаемое архитектурой TINA, расширяет концепцию CORBA (Common object request broker architecture) новыми объектными интерфейсами для работы с сообщениями и потоками, а также с видами связи, ориентированными и не ориентированными на соединения.

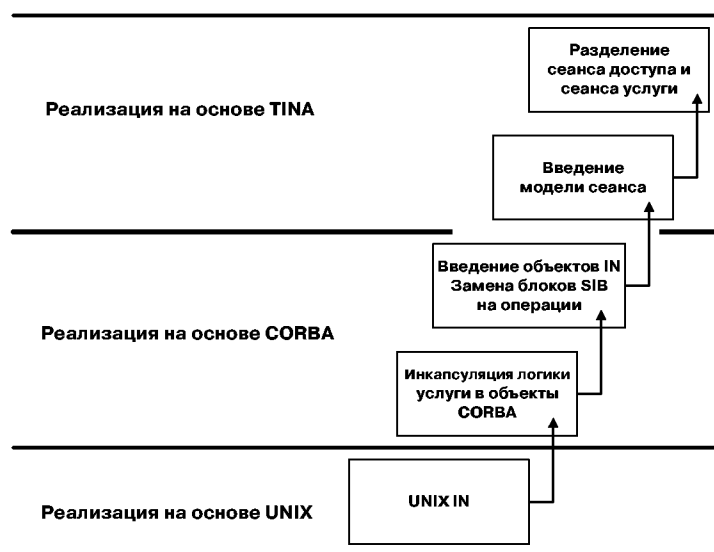


Рис. 5.1.1 Эволюция IN

Аналогично принципам IN, в соответствии с которыми логика управления услугами (SCF) и соответствующие базы данных (SDF) вынесены из коммутационных систем, принципы TINA позволяют вынести из них еще и функции коммутации услуг (SSF) вместе с триггерными таблицами. В архитектуре TINA коммутационные системы извне представляются как набор простых функций коммутации,

управляемых объектами, которые размещены в совместимых с DPE терминалах и/или на серверах.

Распределение функций управления услугой между терминалами пользователей и сетевыми элементами нарушает принцип управления из одной точки, применяемый в CS-1 и CS-2. Воздействие на процесс управления обслуживанием вызова со стороны нескольких точек (SCP) выдвигает такие новые задачи, как разрешение конфликтов при взаимодействии атрибутов, обеспечение сохранности данных, обеспечение надежности, начисление платы и другие серьезные проблемы.

Можно сформулировать несколько естественных требований к сетям IN следующих поколений, реализация которых позволит решить новые задачи:

- поддержка стандартов открытых систем;
- сложность системы, вызванная разнородностью ее компонентов, должна быть скрыта от пользователей и разработчиков приложений;
- поддержка как традиционных услуг телефонной связи, так и услуг передачи изображений;
- наличие средств дистанционного управления сетевыми ресурсами и абонентскими данными.

Новые подходы, нацеленные на удовлетворение перечисленных требований, позволят объединить существующие разнородные сети и реализовать в них принципиально новые услуги, а также упростят разработку универсальных приложений. Современные средства и методы, такие как объектно-ориентированный подход, открытая распределенная обработка информации (ODP – Open distributed processing), распределенная информационная среда (DCE – Distributed communication environment), концепция TMN, методы управления объектами и языки описания интерфейсов IDL группы OMG (Object management group) и технология ATM, образуют основу новой архитектуры.

Основная идея организации программного обеспечения – объектно-ориентированный подход – обеспечивает высокую степень инкапсуляции данных и функций. Каждое приложение состоит из набора объектов (возможно, расположенных в разных узлах сети), взаимодействующих через объектные интерфейсы.

Система, работающая в среде DCE, настроена на взаимодействие приложений, которые функционируют на удаленных компьютерах, связанных сетью. Для реализации взаимодействия используются *агенты* – программные компоненты, действующие «по поручению» *клиента* для выполнения его требований. Они изолируют клиента от

услуги и от системного взаимодействия сетевых ресурсов. Основные этапы эволюции базы для реализации концепции IN показаны на рисунке 5.1.1.

Следующие главы посвящены более подробному рассмотрению обозначенных здесь проблем. По материалам утвержденного в 1997 году пакета рекомендаций ITU-T серии Q.122X проведен анализ набора CS-2, при этом акцент сделан на его главных отличиях от набора CS-1, подробно рассмотренного в предыдущих частях книги. Основой остальных глав послужили проекты рекомендаций исследовательских комиссий ITU-T, стандартов ETSI, рабочие материалы международных проектов INSIGNIA и ReTINA, проводимых институтом EURESCOM в рамках исследовательской паневропейской программы ACTS, а также здравый смысл и построенные на его основе прогнозы и планы. Насколько это удалось, оценит читатель, к которому в любом случае эти строки попадут несколько позже времени их написания, когда уже могут быть заметны неизбежные погрешности в оценках и прогнозах. Тем не менее, представляется целесообразным обратить самое пристальное внимание на вопросы, изложенные ниже в этой последней и самой проблемной части данной книги.

Глава 5.2

Набор возможностей CS-2

5.2.1 Нормативная база

Пакет рекомендаций, определяющий набор возможностей CS-2, был одобрен ITU-T сравнительно недавно – в конце 1997 г. Так же, как и для CS-1, пакет состоит из 6 рекомендаций:

Q.1221: Введение в CS-2;

Q.1222: Плоскость услуг IN для CS-2

Q.1223: Глобальная функциональная плоскость IN CS-2

Q.1224: Распределенная функциональная плоскость IN CS-2

Q.1225: Физическая плоскость IN для CS-2

Q.1228: Интерфейсы IN для CS-2

В следующих разделах рассматриваются основные аспекты CS-2. Заметим, что делается это не так детально, как для CS-1. Внимание уделяется, во-первых, главным отличиям CS-2 от CS-1, выражающимся в значительном расширении функциональных возможностей, и, во-вторых, сохранению преемственности, позволяющей использовать в CS-2 уже готовые решения.

5.2.2 Сравнительный анализ CS-1 и CS-2

5.2.2.1 Определяющие критерии CS-2

В процессе введения услуг, реализуемых в рамках инфраструктуры CS-1, операторы осознали преимущества нового подхода и проявили заинтересованность в предоставлении более сложных услуг IN. Это – услуги, которые дают пользователю возможность добавлять к уже установленной связи новых участников или, наоборот, удалять некоторых из них, услуги поддержки мобильности абонента и терминала (как внутри одной сети, так и на территории, обслуживаемой несколькими сетями), а также стандартные средства создания услуг и конфигурирования их параметров. Сетевые ресурсы, которые необходимы для поддержки таких возможностей, определили инфраструктуру набора возможностей CS-2.

В отличие от CS-1, основной целью которого была спецификация процесса реализации (введения и предоставления) уже известных дополнительных услуг связи, CS-2 обеспечивает инфраструктуру для создания услуг, ранее не предоставлявшихся, и для эксплуатацион-

ного управления услугами. В связи с этим CS-2 имеет ряд характерных особенностей в части предоставляемых услуг и сетевых возможностей, а также ряд отличий в части моделирования на плоскостях концептуальной модели.

В соответствии с определением, CS-2 представляет собой следующий за CS-1 шаг в развитии архитектурной концепции IN. В рамках CS-2 определены следующие три класса услуг:

- услуги связи (telecommunication services);
- услуги эксплуатационного управления услугами (service management services);
- услуги создания услуг (service creation services).

Несмотря на то, что IN является архитектурной концепцией, независимой от набора реализуемых в ее рамках услуг, можно выделить наиболее общие сетевые возможности, характеризующие набор CS-2 и определяемые особенностями услуг, которые предполагается реализовать в этом наборе.

Кроме определенных для CS-1 сетевых возможностей, обеспечивающих гибкую маршрутизацию, гибкую систему начисления платы и гибкое взаимодействие с пользователем, в CS-2 предполагается расширение сетевых функций, поддерживающих взаимодействие между пользователем и сетью, а также введение новых возможностей управления участниками многосторонней связи и взаимодействия между сетями IN при предоставлении услуг.

Что касается услуг и их атрибутов, обеспечивающих поддержку мобильности терминала (например, аутентификацию и передачу управления связью в другую сеть при перемещении в нее терминала пользователя), то CS-2 детально специфицирует только часть требуемой для этого функциональной архитектуры, да и та вынесена в приложение к рекомендации Q.1224. Такое положение возникло в связи с тем, что, несмотря на первоначальное намерение производителей достигнуть между собой согласия, сделать это к моменту, определенному для завершения работы над CS-2, не удалось. По этой причине мы исключим из дальнейшего рассмотрения подробности поддержки функций мобильности.

5.2.2.2 Новые сетевые аспекты CS-2

Новые сетевые аспекты, определяющие инфраструктуру CS-2, могут быть сгруппированы следующим образом.

Расширенный спектр и использование специализированных ресурсов – мостов видеоконференцсвязи, синтезаторов речи, конвертеров протоколов, приемников речевых сообщений, размещаемой внутри SRF законченной части логики услуг (например, сценариев взаимодействия пользователя с услугой).

Возможности управления участниками многосторонней связи, позволяющие создавать услуги и атрибуты, которые обеспечивают связь трех или более участников. Для этого используется специальный подход, основанный на том, что логика услуг SCF управляет обслуживанием в CCF/SSF одновременно нескольких «ветвей» такой связи. При предоставлении услуг многосторонней связи используются функции моста конференцсвязи в SRF. Кроме того, должно быть обеспечено взаимодействие с вызывающим пользователем в разговорном состоянии и прием от него дополнительной информации, например, после того как он кратковременно нажмет на рычажный переключатель телефонного аппарата.

Возможности взаимодействия между сетями при предоставлении услуг связаны с новыми аспектами, требующими технического и коммерческого согласования между сетевыми операторами. При этом возможна совместная поддержка услуги несколькими сетями в части эксплуатационного управления и совместная работа разных сетей в процессе предоставления услуги (IN с обычными сетями общего пользования и IN с обычными ведомственными и частными сетями).

При сохранении принципа управления услугой из одной точки в CS-2 добавлена возможность *влияния на два сегмента соединения* со стороны SCP, но опять, как и в CS-1, рассматриваются только односегментные услуги.

Взаимодействие между пользователем и логикой услуги вне не-сущего канала требует «прозрачного» переноса информации между пользователем и логикой услуги. Функционально такой перенос реализуется средствами системы сигнализации с помощью двух информационных элементов: UTSI (User to service interaction) – в направлении от пользователя к логике услуги и STUI (Service to user interaction) – в обратном направлении. Взаимодействие возможно в двух вариантах – *в контексте управления услугой* и *вне контекста управления услугой* (например, для индикации наличия ожидающего вызова или при регистрации местоположения мобильного терминала пользователя).

5.2.3 Дополнения в INCM для CS-2

Особенности каждого этапа развития архитектуры IN, формулируемые в виде наборов CS, должны быть соответствующим образом отражены на каждой из плоскостей INCM. Далее приводится сравнительный анализ плоскостей INCM для наборов CS-1 и CS-2.

5.2.3.1 Плоскость услуг

Для набора CS-1 плоскость услуг практически не используется. Группа атрибутов, предполагаемых к реализации в CS-1 с целью компоновки из них услуг, определена, но в плоскости услуг не реализована. В CS-2 сделан небольшой шаг вперед за счет разделения всех атрибутов услуг на три типа (см. рис. 5.2.1).

Атрибуты услуг связи

Аспекты услуг связи, которые могут быть использованы индивидуально или совместно с другими услугами или атрибутами услуг связи как часть коммерческого предложения. Как и в CS-1, они могут быть основной или дополнительной частью коммерческого предложения. В дополнение к CS-1 вводятся новые атрибуты услуг, необходимые для создания услуг связи, среди которых можно выделить следующие группы:

- ⇒ предоставляющие возможность мобильности пользователя и терминала;
- ⇒ обеспечивающие санкционирование доступа к сети;
- ⇒ связанные с услугами, которые могли частично поддерживаться в CS-1.

Атрибуты услуг эксплуатационного управления

Аспекты услуг, относящиеся к деятельности операторов сетей или поставщиков услуг. Включают в себя контроль и техническое обслуживание услуг связи, сбор данных и управление нагрузкой, начисление платы.

Атрибуты услуг эксплуатационного управления, включенные в CS-2, сгруппированы следующим образом:

- ⇒ атрибуты услуг адаптации к требованиям пользователя;
- ⇒ атрибуты услуг управления;
- ⇒ атрибуты услуг контроля.

Атрибуты услуг создания

Аспекты услуг, относящиеся к деятельности операторов сетей и/или поставщиков услуг для спецификации, разработки, развертывания и проверки новых услуг. Атрибуты услуг создания, включенные в CS-2, сгруппированы следующим образом:

- ⇒ атрибуты услуг спецификации;
- ⇒ атрибуты услуг разработки;
- ⇒ атрибуты услуг проверки;
- ⇒ атрибуты услуг развертывания;
- ⇒ атрибуты услуг эксплуатационного управления услугами создания.

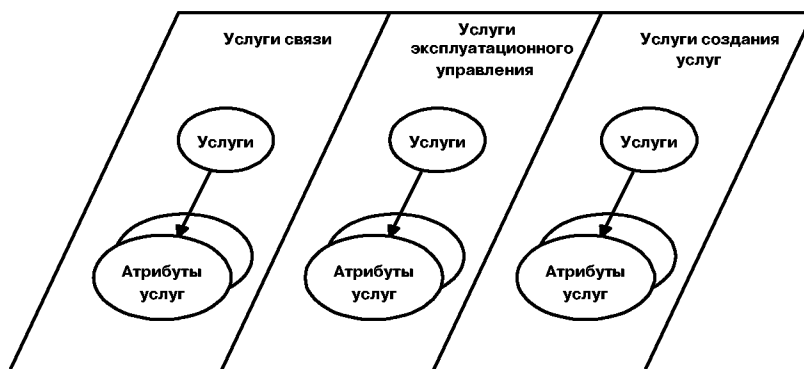


Рис. 5.2.1 Плоскость услуг для CS-2

Для управления взаимодействием между атрибутами услуг в CS-2 определены два механизма: статический и динамический. *Статический механизм* основан на выявлении возможных вариантов взаи-

модействия на этапе создания и требует от поставщика услуг предварительной подготовки специальных правил их разрешения. *Динамический механизм* направлен на выработку решений, которые должны приниматься во время обслуживания реальных вызовов. Эти решения подразделяются на те, которые относятся к «ожидаемым» вариантам взаимодействия (к вариантам, возможность возникновения которых разработчик услуги мог предвидеть), и на те, которые относятся к «непредвиденным» вариантам (например, к вариантам, которые возникают вследствие модификации услуги после того, как правила взаимодействия были специфицированы).

5.1.3.2 Глобальная функциональная плоскость

Для моделирования услуг в рамках архитектуры CS-2 определен ряд новых блоков SIB, а также определены дополнительные требования, касающиеся глобальной функциональной плоскости, такие как:

- поддержка взаимодействия между BCP и GSL;
- возможность дробления SIB;
- возможность рекурсивного использования SIB;
- возможность формального описания структур данных для SIB при помощи абстрактного синтаксиса ASN.1;
- поддержка параллельного исполнения нескольких SIB;
- совместная работа сетей должна быть «видима» со стороны GSL.

Как и в CS-1, глобальная функциональная плоскость в CS-2 используется для представления основных сетевых возможностей в виде блоков SIB и описания способа комбинирования этих SIB при моделировании услуг, введенных на плоскости услуг. Для решения этих задач определены два разных аспекта представления услуги на глобальной функциональной плоскости. Графическая иллюстрация этих представлений приведена на рисунке 5.2.2.

Первый аспект – описание отдельных SIB в виде присущих каждому из них операций (представление с точки зрения возможностей). Операции комбинируются таким образом, чтобы было удобнее задавать соответствие (mapping) между SIB и функциональными объектами и информационными потоками на распределенной функциональной плоскости.

Второй аспект – описание конкретной услуги в виде последовательности SIB или отдельных операций (представление с точки зрения услуги). В этом случае набор операций и необходимые структуры данных являются специфическими для каждой услуги.

Здесь под новым для глобальной функциональной плоскости термином *операция* (на рисунке обозначаемая как SIB op) понимается

непрерываемая и неделимая функция, выполняемая внутри блока SIB в соответствии с его возможностями. Новое определение SIB позволяет сделать представление его внутренних возможностей более удобным с точки зрения программиста. Таким образом, теперь термин SIB определяет, скорее, целую библиотеку доступных программисту (разработчику услуги) инструкций, чем одну сложную инструкцию.

Плоскость услуг

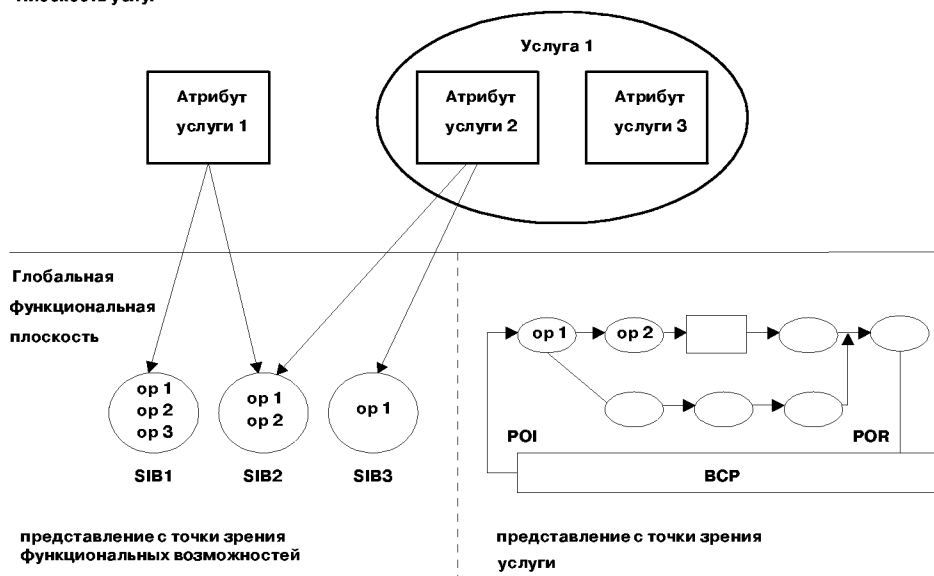


Рис. 5.2.2 Аспекты глобальной функциональной плоскости для CS-2

В связи с введением операций несколько изменился шаблон описания SIB. По этой причине как новые SIB, так и блоки, введенные ранее рекомендациями для CS-1, приведены в Q.1224 в соответствии с новыми правилами (см рис. 5.2.3). Основное отличие заключается в том, что шаблон, похожий на описание SIB для CS-1, теперь приводится для каждой составляющей этот блок операции. При этом данные SSD и CID определены отдельно для каждой операции и делятся на входные и выходные.



Рис. 5.2.3 Описание SIB CS-2

В качестве примера рассмотрим SIB SERVICE FILTER (бывший LIM-IT, переименованный в CS-2). В соответствии с функциональным назначением, блок составляют две операции – Service filter activating

и Service filter reporting, первая из которых предназначена для активизации процедуры просеивания, вторая – для выдачи отчета о результатах просеивания. Как видно из рисунков 5.2.4 и 5.2.5, на которых изображены шаблоны операций, составляющих блок, практически все данные SSD и CID, ранее определенные для блока LIMIT рекомендациями по CS-1, сохранены.

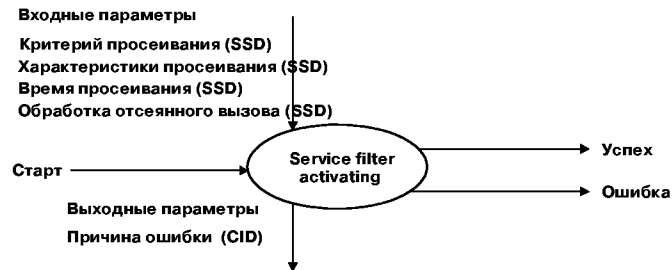


Рис. 5.2.4 Описание операции Service filter activating



Рис. 5.2.5 Описание операции Service filter reporting

Соответствие между услугами и возможностями глобальной функциональной плоскости в CS-2 задается с использованием техники композиции и декомпозиции «обычных» SIB в/из SIB высокого уровня HLSIB (High level SIB), а также параллельного исполнения SIB. Иллюстрация таких возможностей приведена на рисунках 5.2.6 и 5.2.7. Для моделирования параллельно исполняемых процессов введены точки синхронизации (POS – Points of synchronization), представляющие те состояния процесса, в которых возможна передача сообщений другому процессу.

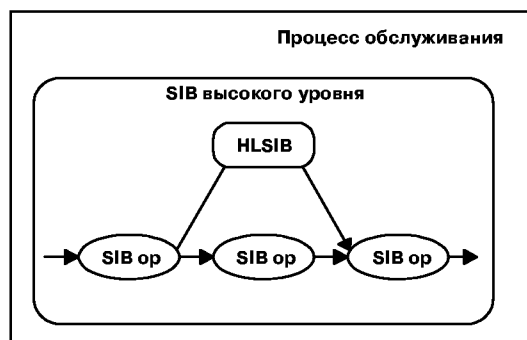


Рис. 5.2.6 Композиция в/из SIB высокого уровня

SIB высокого уровня отличаются от обычных тем, что могут komponоваться из отдельных операций и из других HLSIB, исполняемых последовательно. HLSIB представляет собой абстрактное описание, скрывающее определенную часть логики услуг и структуры данных, которая может рассматриваться как локальная для данного HLSIB.

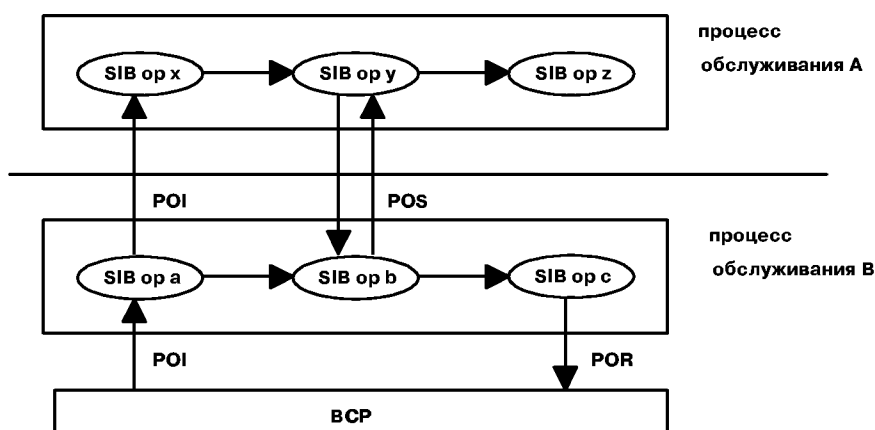


Рис. 5.2.7 Параллельное выполнение независимых конструктивных блоков

То, каким образом SIB должны быть соединены друг с другом, описывается с помощью глобальной логики услуги GSL. В конце цепи SIB логика услуги задает точку возврата в базовый процесс обслуживания вызова. Рисунок 5.2.8 демонстрирует пример того, как логика услуги связывает BCP и цепь SIB произвольного вида.

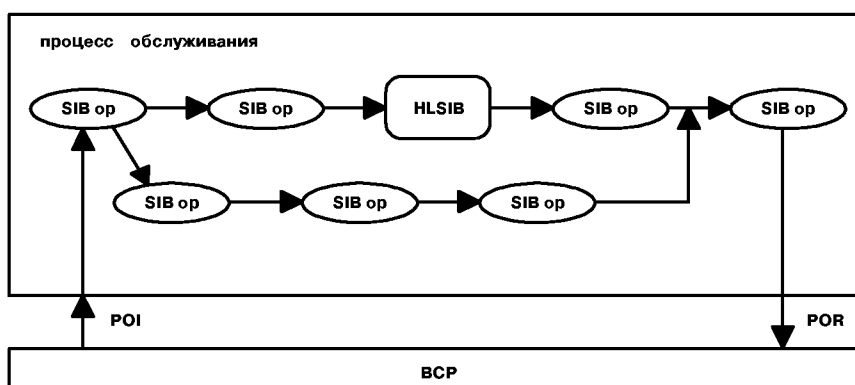


Рис. 5.2.8 Связь логики услуги и цепи SIB для CS-2

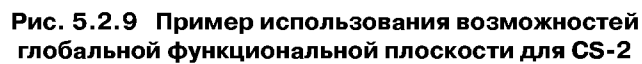
Для связи между собой нескольких фрагментов IN, обладающих разными сетевыми возможностями (на фазах взаимодействия сетей и услуг IN), требуется разделение логики услуги на несколько процессов обслуживания. Для поддержки этих возможностей в CS-2 предусмотрены механизмы создания процесса обслуживания и обмена данными между процессами обслуживания.

Для поддержки атрибутов услуг, предполагаемых к реализации в CS-2, определено еще шесть SIB (в дополнение к 14-ти, определенным в CS-1):

- ПРИСОЕДИНИТЬ (JOIN) – подключает к обслуживаемой связи определенного участника или определенную группу участников;
- ОТДЕЛИТЬ (SPLIT) – отключает от данной связи одного участника или группу участников и подключает его (их) к другой существующей или создаваемой вновь связи;
- ИНИЦИИРОВАТЬ ПРОЦЕСС ОБСЛУЖИВАНИЯ (INITIATE SERVICE PROCESS) – создает новый процесс обслуживания, работающий параллельно;
- КОНТРОЛЛЕР СООБЩЕНИЙ (MESSAGE HANDLER) – обеспечивает передачу параллельно работающему процессу сообщения с необходимыми данными и обработку в нем этого сообщения;
- ОКОНЧАНИЕ (END) – указывает на нормальное завершение исполняемого процесса;
- БАЗОВЫЙ ВНЕКОНТЕКСТНЫЙ ПРОЦЕСС (BASIC CALL UNRELATED PROCESS), – обеспечивает доступ к таким услугам или атрибутам услуг IN, обращение к которым инициируется (либо пользователем, либо сетью) вне контекста управления связью пользователя.

Определенный в CS-1 блок USER INTERACTION дополнен в CS-2 новыми функциями поддержки исполнения делегированной в SRF части логики услуги, обеспечивающей ведение диалога с пользователем, а блок AUTHENTICATION дополнен новыми функциями в части санкционирования доступа пользователя к сети и одного функционального объекта к другому.

На рисунке 5.2.9 приведен пример использования возможностей глобальной функциональной плоскости для моделирования услуг IN в рамках CS-2.



5.2.3.3.1 Функциональные объекты CS-2

На рисунке 5.2.10 приведена модель распределенной функциональной плоскости для CS-2, показывающая функциональные объ-

екты и возможные связи между ними. Следует отметить, что логические связи между SMF и остальными функциональными объектами в действующих рекомендациях определены лишь на концептуальном уровне, однако постулировано, что эти связи будут базироваться на принципах TMN.

Функции доступа к сети IN (IAF – Intelligent network access function) обеспечивают доступ к IN со стороны сетей, не структурированных в соответствии с концепцией IN. IAF физически находятся вне IN, а единственная разрешенная связь с ними осуществляется через SCF.

Перечень функциональных объектов CS-2, относящихся к управлению связью пользователя, остается таким же, как и для CS-1, но возможности CCF, SSF и SRF расширены с тем, чтобы реализовать новые ключевые свойства – приостановка связи в активном ее состоянии, управление несколькими участниками связи, поддержка взаимодействия с пользователем вне разговорного канала и т.д.

На рисунке 5.2.10 не показаны объекты, введенные для обеспечения мобильности терминала, так как они вынесены в приложение к стандарту. Приведем лишь их перечень: функции управления радиодоступом, непосредственно относящиеся к связи пользователя (CRACF – Call related radio access control functions), функции управления радиодоступом, не имеющие отношения к связи пользователя (CURACF – Call unrelated radio access control functions) и функции управления радиосредствами (RCF – Radio control functions).

Возможности функциональных объектов, относящихся к управлению услугами, расширены дополнительными функциями поддержки взаимодействия услуг в пределах нескольких сетей.

Для моделирования таких видов взаимодействия логики услуг с пользователем, которые не имеют прямого отношения к связи этого пользователя, введена новая группа функциональных объектов. В нее входят объекты, которые предназначены для приема и обработки информации, не относящейся к связи пользователя CUSF (Call unrelated service functions), и объекты, обеспечивающие передачу информации управления услугой от пользователя SCUAF (Service control user agent functions).

Группа функций, относящиеся к эксплуатационному управлению услугами и к созданию услуг (их использование в CS-1 не оговаривалось), состоит из функций среды создания услуг (SCEF), которые используются для спецификации, конструирования и тестирования услуг; функций доступа к эксплуатационному управлению услугами (SMAF), обеспечивающих интерфейс с SMF, и собственно функций эксплуатационного управления услугами (SMF).

ции реконфигурированного соединения. Существенным является сохранение принципа прямого воздействия на услугу со стороны только одного SCF.

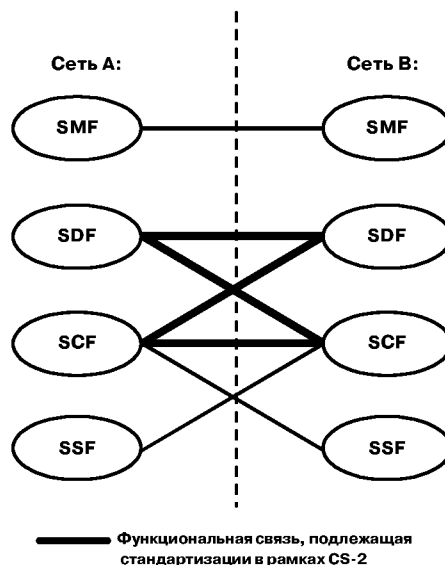


Рис. 5.2.11 Функциональные связи при взаимодействии сетей IN

Интерфейс SDF-SDF имеет два основных назначения. Первое – обеспечить механизм копирования данных между сетями, второе – обеспечить «прозрачный» доступ к структурам данных. Логические связи вида SMF-SMF для CS-2 определены, но информационные потоки для них не специфицированы.

5.2.3.3.2 Модель CUSF

Модель функционального объекта CUSF, содержащего функции поддержки услуг, которые не имеют прямого отношения к связи пользователя (для краткости их удобно называть внеконтекстными услугами), приведена на рис. 5.2.12. С целью облегчить понимание структуры CUSF на рисунке показана также внутренняя структура CCF/SSF с уже известными нам функциональными компонентами модели их внутренних ресурсов, имеющих непосредственное отношение к связи пользователя. Функции каждого из компонентов модели «внеконтекстного» объекта CUSF аналогичны функциям компонента модели CCF/SSF, имеющего схожее название, но с буквой U или N (что значит «не») в середине: BCUSM <-> BCSM, BNCM <-> BCM, IN-USM <-> IN-SM, FIM/NCM <-> FIM/CM.

BNCM является абстрактным представлением связи между пользователем и сетью при их взаимодействии вне контекста вызова. Она обнаруживает такие события, происходящие вне этого контекста,

которые могут потребовать активизации новой программы логики услуг или обращения к уже активной программе.

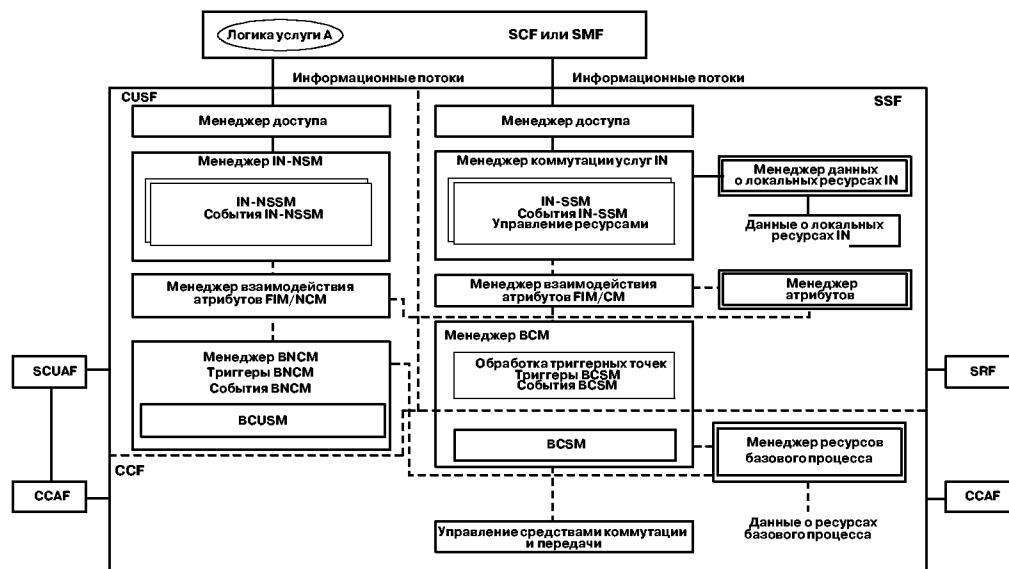


Рис. 5.2.12 Модель CUSF

IN-NSM взаимодействует с SCF во время предоставления услуги пользователю с целью сделать события, происходящие в CUSF, видимыми со стороны SCF. FIM/NCM обеспечивает механизм, который предотвращает одновременное обращение к нескольким услугам/атрибутам, относящимся к одной логической связи.

5.2.3.4 Физическая плоскость

Применительно к CS-2 определены протоколы для следующих интерфейсов:

Тип интерфейса	Протокол
<i>Управление соединением</i>	
CCAF-CCF	DSS1/Q.931
CCF-CCF	OKC-7/ISUP
CCF-SRF	DSS1/Q.931; OKC-7/ISUP
<i>Управление вне сети IN</i>	
CCAF-CCF	DSS1/Q.931
CCF-CCF	OKC-7/ISUP
CCF-SRF	DSS1/Q.931; OKC-7/ISUP
<i>Управление услугами IN</i>	
SCF-CUSF	OKC-7/TCAP/INAP
SCF-SDF	OKC-7/TCAP/INAP
SCF-SSF	OKC-7/TCAP/INAP
SCF-SCF	OKC-7/TCAP/INAP
SCF-SRF	OKC-7/TCAP/INAP
SDF-SDF	OKC-7/TCAP/INAP

Функции CUSF содержатся в терминальном оборудовании ISDN, что отражено на рисунке 5.2.13. Для поддержки функционирования SIB, определенных для CS-2, а также связанных с этими SIB операций и информационных потоков, в интерфейсе управления услугами определены соответствующие процедуры и структуры данных протокола прикладного уровня IN (INAP).

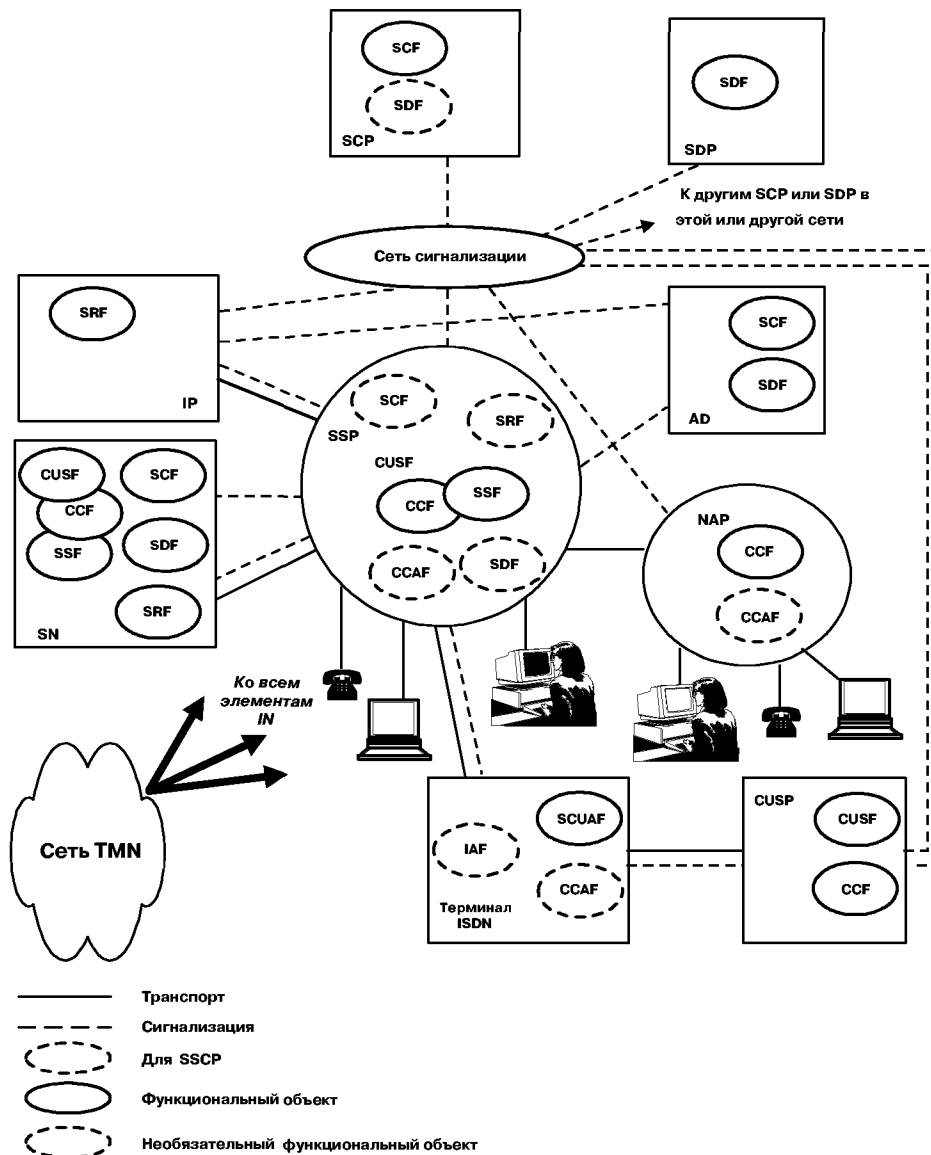


Рис. 5.2.13 Физическая плоскость CS-2

5.2.4 Особенности базового процесса обслуживания вызова для CS-2

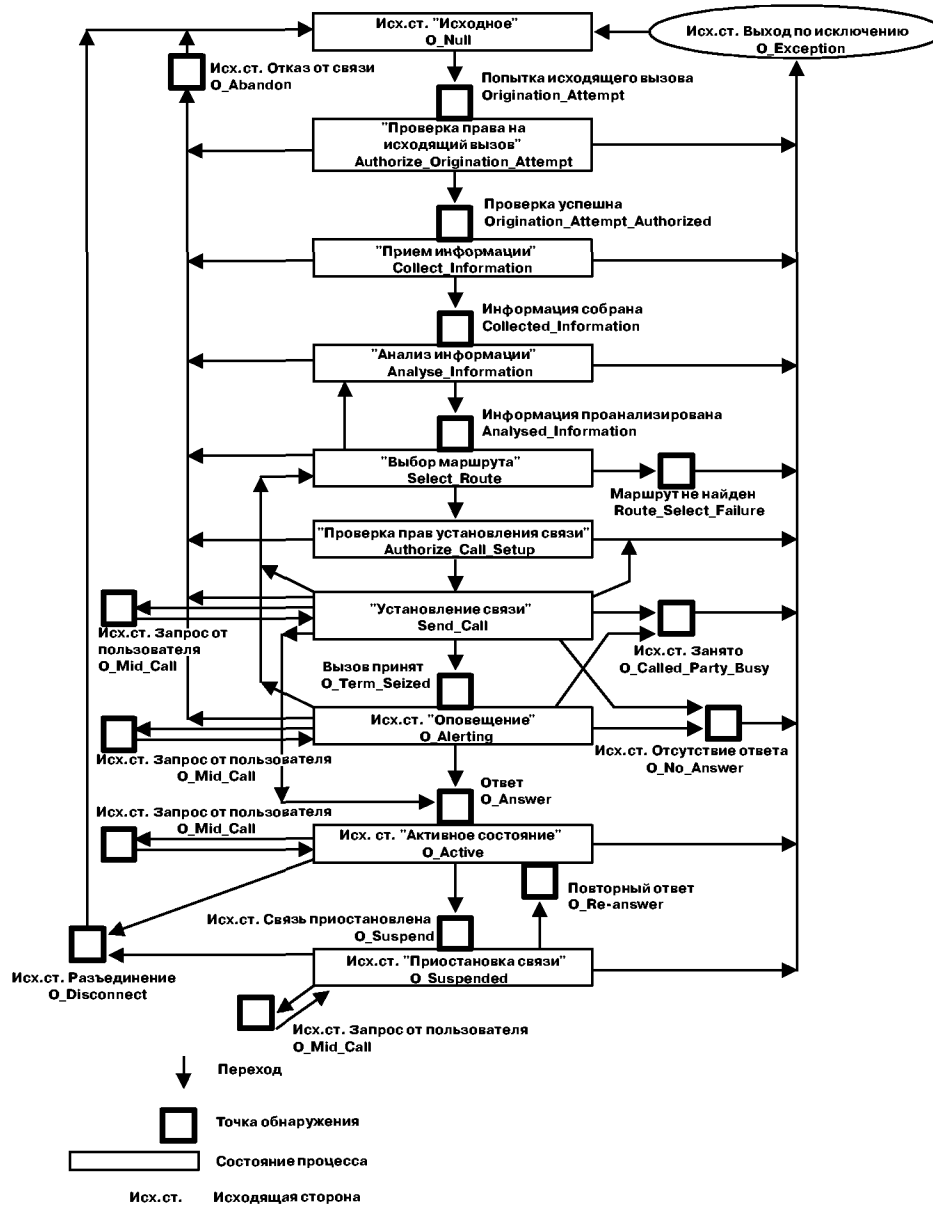
5.2.4.1 Предпосылки

В соответствии с концепцией IN модель состояний базового процесса обслуживания вызова BCSM для любого из наборов CS является подмножеством общей модели, отражающей сегодняшнее видение целевой (завтрашней) архитектуры IN. За основу такой модели для CS-2 взята модель BCSM, определенная в Q.1214 для CS-1. На рисунках 5.2.14 и 5.2.15 представлены исходящая и входящая части BCSM для набора CS-2, управляемые функционально разделенными менеджерами BCM в SSF/CCF.

5.2.4.2 Исходящая часть BCSM для CS-2

Исходящая часть BCSM (рис. 5.2.14) соответствует той части процесса обслуживания вызова, которая относится к исходящей стороне. Ниже приводится описание состояний процесса в точках PIC этой части процесса:

O_Null:	исходное состояние.
Authorize_Origination_Attempt:	индикация исходящего вызова (сообщение Setup Q.931, сообщение IAM ISUP), проверка права пользователя на связь с запрошенными характеристиками.
Collect_Information:	прием адресной информации (коды услуг, префиксы, цифры набора номера) от исходящей стороны. Информация проверяется с учетом плана нумерации на предмет ее полноты.
Analyze_Information:	анализ и/или персчет адресной информации в соответствии с планом нумерации для определения физического адреса назначения и вида связи.
Select_Route:	выбор маршрута в соответствии с адресом назначения и видом связи.
Authorize_Call_Setup:	проверка права вызывающей стороны на данную связь.
Send_Call:	установление соединения с вызываемой стороной.
O_Alerting:	ожидание ответа от вызываемой стороны.
O_Active:	соединение между исходящей и входящей стороной установлено; начисление платы.
O_Suspended:	ожидание возобновления обслуживания вызова или немедленного разъединения.
O_Exception:	обработка исключительных ситуаций (по умолчанию).



5.2.4.3 Входящая часть BCSM для CS-2

Входящая часть BCSM (рис. 5.2.15) соответствует части процесса, относящейся к входящей стороне. Описание состояний процесса (PIC) для этой части модели приведены ниже:

T_Null:	исходное состояние.
Authorize_Terminating_Attempt:	индикация входящего вызова, проверка правомочности его приема.
Select_Facility:	выбор ресурса на входящей стороне.
Present_Call:	передача сигнала о входящем вызове (занятие линии, сообщение Q.931 Setup, сообщение ISUP IAM).
T_Alerting:	исходящая часть BCSM извещается о передаче вызывного сигнала входящей стороне, ожидается ответ входящей стороны.
T_Active:	исходящая часть BCSM извещается о том, что получен ответ, установлено соединение между исходящей и входящей стороной.
T_Suspended:	физические ресурсы, предоставленные для связи, удерживаются.
T_Exception:	индикация возникновения исключительной ситуации при обработке входящего вызова.

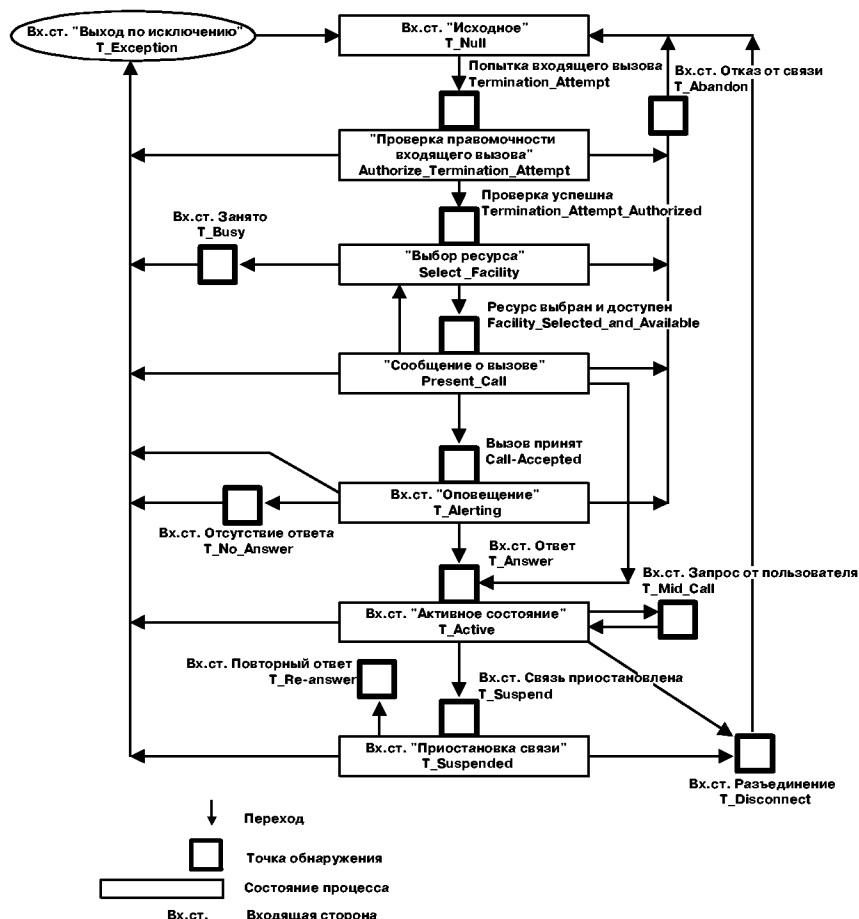


Рис. 5.2.15 Входящая часть BCSM для CS-2

5.2.4.4 Отличия от BCSM для CS-1

В BCSM для CS-1 некоторые функции обработки вызова сгруппированы вместе и связаны с одной точкой PIC (состоянием процесса), в то время как в модели для CS-2 состояние процесса распадается на несколько отдельных состояний. Кроме того, добавлены новые точки обнаружения DP (Origination_Attempt DP, Termination_Attempt DP, O_Re-Answer DP, T_Re-Answer DP, O_Suspended DP, T_Suspended DP, Facility_Selected_&_Available DP), в результате чего в модели BCSM для CS-2 потребовались следующие изменения:

В исходящей части:

- Состояние процесса (PIC) O_Authorize_Origination_Attempt разделено на два состояния:
 - O_Null;
 - Authorize_Origination_Attempt.
- Состояние процесса Routing & Alerting разделено на четыре состояния:
 - Select_Route;
 - Authorize_Call_Setup;
 - Send_Call;
 - O_Alerting.
- В результате добавления новой точки обнаружения O_Suspend DP появилось новое состояние:
 - O_Suspended.

Во входящей части:

- Состояние процесса T_Null & Authorize_Termination_Attempt разделено на два состояния:
 - T_Null;
 - Authorize_Termination_Attempt.
- Состояние процесса Select_Facility & Present_Call разделено на два состояния:
 - Select_Facility;
 - Present_Call.
- В результате добавления новой точки обнаружения T_Suspend DP появилось новое состояние:
 - T_Suspended.

Изменение модели для CS-2 привело к появлению новых переходов из состояния в состояние. Перечень новых переходов представлен в таблице 5.2.1.

Таблица 5.2.1

ОТ	К
И с х о д я щ а я ч а с т ь В С С М	
Orig_Attempt DP	Authorize_Orig_Attempt PIC Collect_Information PIC Analyze_Information PIC Select_Route PIC
O_Suspended DP	O_Suspended PIC
O_Re-Answer DP	O_Active PIC
O_Mid_Call DP	O_Suspended PIC Collect_Information PIC Analyze_Information PIC Select_Route PIC
O_Disconnect DP	Collect_Information PIC Analyze_Information PIC Select_Route PIC
O_Null PIC	Origination_Attempt DP
Auth_Orig_Attempt PIC	O_Abandon DP O_Exception
O_Active PIC	O_Suspend DP
O_Suspended PIC	O_Re-Answer DP O_Mid_Call DP O_Disconnect DP
В х о д я щ а я ч а с т ь В С С М	
Termination_Attempt DP	Auth_Termination_Attempt PIC Select_Facility PIC
T_Called_Party_Busy DP	Present_Call PIC
T_Suspend DP	T_Suspended PIC
T_Re-Answer DP	T_Active PIC
T_Null PIC	Termination_Attempt DP
Auth_Termination_Attempt PIC	T_Abandon DP T_Exception DP
T_Active PIC	T_Suspend DP
T_Suspended PIC	T_Re-Answer DP T_Disconnect DP

5.2.5 Услуги CS-2

5.2.5.1 Услуги связи

Все услуги связи, определенные инфраструктурой CS-1, должны поддерживаться в CS-2. Приведем сведения о некоторых из услуг, при реализации которых требуются средства и функциональные возможности, специфические для CS-2:

Межсетевой бесплатный вызов (IFPN - Internetwork freephone)	Обеспечивает доступ к пользователю, оборудование которого размещено в одном или нескольких пунктах одной сети, со стороны другой сети по номеру FreePhone.
Межсетевая информационная услуга за дополнительную плату (IPRM - Internetwork premium rate)	Обеспечивает двустороннюю диалоговую связь между абонентом и поставщиком услуг, когда они находятся в разных сетях. Вызывающий абонент оплачивает стоимость услуги связи и, дополнительно, стоимость информационных услуг. Дополнительная сумма делится между оператором сети и поставщиком информационных услуг.
Межсетевые массовые вызовы (IMAS - Internetwork mass calling)	Позволяет направлять большое число одновременных вызовов от абонентов одной сети на один номер, входящий в план нумерации другой сети. Обеспечивается односторонняя связь с каждым вызывающим абонентом.
Межсетевое телеголосование (IVOT - Internetwork televoting)	Позволяет поставщику услуги в одной сети обеспечить участие в телеголосовании абонентов этой и/или других сетей.
Глобальная виртуальная сеть (GVNS - Global virtual network service)	Виртуальная частная сеть, организованная с использованием ресурсов нескольких разных сетей связи.
Конференцсвязь (CONF - Conference calling)	Позволяет организовать конференцсвязь.
Удержание соединения (CH - Call hold)	Позволяет пользователю перевести установленную связь в режим удержания, передав уведомление об этом остальным ее участникам, и установить новое соединение. Нарушив новое соединение, пользователь может возобновить прежнюю связь, переведя ее из режима удержания в активный режим.
Переключение связи (TRA - Call transfer)	Позволяет пользователю перевести установленное соединение в режим удержания, установить новое соединение и отключиться. В результате устанавливается связь между участниками удержанного и вновь установленного соединений.
Уведомление об ожидающем вызове (CW - Call waiting)	Позволяет уведомлять абонента о входящем вызове во время связи.
Прием и пересылка сообщений (MSF - Message store and forward)	Позволяет запоминать принимаемые сообщения с целью их последующей передачи другим абонентам.
Международная Расчетная Карта (ITCC - International telecommunication charge card)	Позволяют владельцу расчетной карты, выпущенной оператором одной сети, пользоваться услугами связи, предоставляемыми сетью другого оператора.

5.2.5.2 Услуги технической эксплуатации услуг

Для поддержки надлежащего действия услуг IN администратору необходимы функциональные средства эксплуатационного управления услугами на каждой из трех фаз их жизненного цикла – развертывания, ввода в эксплуатацию и использования. Эти средства содержатся в функциональном блоке SMF.

Администратор получает доступ к SMF посредством SMAF. Администратором до сих пор являлся оператор сети, в исключительном ведении которого находилось эксплуатационное управление. Теперь же администратор – это тандем поставщика услуги и оператора сети. Более того, в IN некоторые функции эксплуатационного управления

услугами может выполнять абонент. Такие функции различны для разных услуг, но, в целом, они дают абоненту возможность получать доступ к заранее оговоренным с поставщиком параметрам услуг, на которые он подписан, изменять эти параметры, а также запрашивать и получать отчеты.

Эта возможность обеспечивается доступом к SDF посредством SMAF со специального терминала или доступом через телефонную сеть посредством CCAF. Способ зависит от услуги – например, регистрация местоположения абонента UPT производится с обычного телефонного аппарата, тогда как модификация параметров абонентов VPN осуществляется со специального терминала.

Функциональные связи эксплуатационного управления определены в CS-2 только до распределенной функциональной плоскости. Никаких модификаций протокола INAP в целях поддержки этих связей между элементами сети IN не предусматривается, а предполагаемый протокол для использования протокол – это определенной для сетей TMN в рекомендации ITU-T Q.812 протокол SMIP (Simple management interface protocol).

Услуги эксплуатационного управления услугами IN сгруппированы по критерию предоставляемых ими возможностей. Детализация технических аспектов такого рода функций в CS-2 не приводится, говорится лишь о том, что принципы эксплуатационного управления должны соответствовать принципам, которые специфицируются в настоящее время для сетей TMN.

- Группа услуг адаптации к требованиям пользователя:
 - адаптация услуг связи;
 - адаптация услуг управления;
 - адаптация услуг контроля.
- Группа услуг управления:
 - активизация/деактивизация абонентом заказа услуги связи;
 - активизация/деактивизация абонентом заказа услуги контроля;
 - эксплуатационное управление данными (позволяет абоненту заказывать или отменять создание приписанной ему специальной структуры данных, содержащей информацию относительно используемых им услуг связи и контроля);
 - ограничение (со стороны абонента) использования услуги;
 - вызов абонентом услуги (для ввода или изменения ее параметров).
- Группа услуг контроля:
 - отчет об услуге (позволяет абоненту получить отчет об использовании заказанной услуги);

- отчет о начисленной плате (позволяет абоненту получить отчет о счетах за пользование заказанной услугой);
- отчет о состоянии заказанной услуги (позволяет абоненту получить информацию о таких характеристиках услуги как производительность, перегрузка, сбои и т.д.);
- наблюдение за трафиком (позволяет абоненту проводить наблюдение в реальном времени за функционированием заказанной услуги);
- отчет об использовании возможностей эксплуатационного управления (обеспечивает получение отчета о пользовании услугами из групп управления и контроля).

5.2.5.3 Услуги создания услуг

С точки зрения оператора процесс создания услуги содержит три фазы – спецификации, разработки и проверки, – на выходе каждой из которых ожидается соответствующая составляющая логики услуги: часть, необходимая для исполнения услуги, часть, нужная для эксплуатационного управления ею, и часть, служащая для модификации абонентских данных. При спецификации услуги определяется и разделение функций между SMF и SCEF, то есть устанавливается, какие аспекты услуги модифицируются разработчиком в SCEF, а какие – администрацией или абонентом в SMF.

Услуги создания тоже сгруппированы по критерию предоставляемых ими возможностей. В CS-2 процесс спецификации услуги создания лишь идентифицирован без какой-либо детализации соответствующих технических аспектов.

- Группа услуг спецификации:
 - обнаружение возможных вариантов взаимодействия между атрибутами услуг;
 - генерация правил взаимодействия атрибутов услуг;
 - каталогизация услуг и SIB.
- Группа услуг разработки:
 - выбор интерфейса;
 - инициирование процесса создания;
 - редактирование;
 - комбинирование;
 - генерация правил размещения данных;
 - проверка синтаксиса и данных;
 - архивация услуг и SIB;

- управление конфигурацией услуги.
- Группа услуг проверки:
 - моделирование создаваемой услуги;
 - тестирование создаваемой услуги.
- Группа услуг развертывания:
 - распределение услуги;
 - распределение конструктивных блоков;
 - распределение правил размещения данных;
 - распределение правил взаимодействия.
- Группа услуг эксплуатационного управления услугами создания:
 - контроль доступа к функциям создания услуг (SCEF);
 - восстановление SCEF после сбоя;
 - расширение возможностей SCEF;
 - преобразование услуг, созданных на оборудовании с разными SCEF;
 - взаимодействие с услугами эксплуатационного управления услугами.

5.2.6 Аспекты предоставления международных услуг IN

5.2.6.1 Услуга бесплатной международной связи IFS

Рекомендация E. 152 дает описание услуги IFS с точки зрения ее внедрения, т.е. дает информацию о процессе заказа услуги и других аспектах ее эксплуатации. При этом используются следующие определения:

- поставщик услуги IFS – зарегистрированный телефонный оператор (ROA – Recognized operating agency), который предоставляет услугу IFS;
- абонент услуги IFS – физическое или юридическое лицо, заказавшее услугу IFS у поставщика услуги IFS и ответственное за оплату всех связанных с этим расходов;
- пользователь услугой IFS – лицо или устройство, производящее вызов, который требует обращения к услуге IFS.

Под поставщиком А понимается оператор ROA, обеспечивающий связь с заказчиком услуги IFS и несущий ответственность за все аспекты предоставления услуги IFS. Поставщик В является оператором ROA в стране, где запрашивается услуга IFS, и несет ответственность за обеспечение доступа к услуге IFS в этой стране. Поставщик А, кроме того, несет ответственность за качество предоставляе-

мой услуги. Для этого он назначает лицо, ответственное за заказ услуги, ее тестирование, контроль неисправностей. Эта информация должна быть доступна всем поставщикам, вовлеченным в предоставление услуги IFS.

Услуга IFS обеспечивается двухсторонним соглашением между операторами услуги. Участвующие операторы могут выбрать любой или все из перечисленных ниже трех методов доступа к услуге IFS в стране, из которой она запрашивается.

Доступ обеспечивается набором национального номера услуги FPH. Пользователь услугой IFS набирает национальный логический номер услуги FPH, который пересчитывается в физический номер назначения, после чего вызов направляется в страну назначения. Из-за допустимых вариаций в структуре номера желательно, чтобы присваиваемый в данной стране номер абонента IFS был разным в разных странах.

Доступ осуществляется через международную сеть набором внутреннего номера услуги FPH чужой страны. Единственный внутренний номер услуги FPH, присвоенный заказчику услуги IFS, используется для приема вызовов из других стран. Пользователь услугой IFS набирает префикс и код страны, а затем упомянутый внутренний номер услуги FPH, который пересчитывается в номер пути и передается в сеть страны назначения.

Доступ обеспечивается набором универсального номера международной услуги IFS (UIFN – Universal IFS number,). Единый UIFN, одинаковый для всех пользователей в мире, присваивается заказчику услуги IFS. Пользователь услугой IFS набирает международный префикс, а затем – номер UIFN, который пересчитывается в физический номер и направляется в страну назначения.

В рекомендации ITU-T Q.736 рассмотрены четыре случая отнесения платы за услугу на счет вызываемого абонента.

1. Начисление платы на счет абонента услуги по запросу обратившегося к ней пользователя на этапе установления соединения. Запрос пользователя абонент может либо подтвердить, либо отвергнуть. Если было получено подтверждение, то абоненту посылается сообщение о начале начисления платы на его счет.

2. Изменение во время связи адреса для начисления платы за остаток разговора:

- отнесение платы на счет абонента услуги по запросу пользователя этой услугой, произведенному после установления соединения; если, однако, абонент отказывается оплачивать разговор, плата продолжает начисляться на счет пользователя;
- отнесение платы на счет абонента услуги по запросу самого абонента.

3. Безусловное отнесение платы на счет абонента услуги по запросу пользователя после установления соединения.
4. Отнесение на счет абонента платы за все входящие вызовы. Пользователь не посылает специального запроса, а получает уведомление о том, что разговор оплачивается абонентом.

5.2.6.2 Услуга оплаты связи по международной телефонной карте ITCC

Услуга ITCC позволяет держателю кредитной карты получать телекоммуникационные услуги от акцептора карты (оператора, принимающего карту), в то время как расчет за оказанные услуги ведет эмитент карты (организация, выпустившая карту). Услуга ITCC регламентируется рекомендацией Q.736 и рядом других рекомендаций ITU-T, например E.113, E.116, E.118 и т.д.

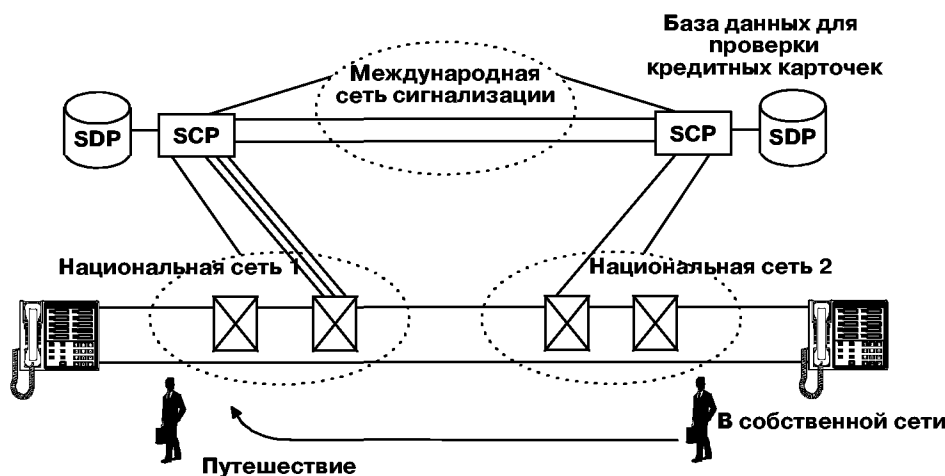


Рис. 5.2.16 Схема организации услуги ITCC

Согласно этим рекомендациям оператор, желающий выпустить карты ITCC, должен заключить необходимые соглашения с другими операторами и эмитентами других телефонных карт, чтобы выпускаемые этими организациями карты могли использоваться совместно. Указывается также на необходимость пройти регистрацию в Генеральном Секретариате ITU-T и внимательно отобрать карты других эмитентов, учитывая при этом их техническую совместимость, возможность подтверждения подлинности карты и проверки пользователя, возможность взимания телефонной платы, существование адекватных процедур на случай потери или кражи карты.

Глава 5.3

Поддержка мобильности

5.3.1 Подходы к созданию и специфика беспроводных IN

Одной из областей, к которым в последнее время заметно возрос интерес сетевых операторов, является конвергенция (взаимное проникновение) Интеллектуальных сетей, создаваемых на базе стационарных сетей связи, и беспроводных сетей подвижной связи. Рассмотрим тот аспект этой проблемы, который наиболее тесно связан с тематикой книги, – конвергенцию услуг.

Как и в случае с телефонными сетями, естественным начальным побуждением операторов подвижной связи было расширение обслуживаемой территории и насыщение рынка. Как только эта цель была достигнута, внимание операторов сфокусировалось на дополнительных услугах, что немедленно вызвало интерес к концепции IN.

Даже на первый взгляд, архитектура IN и архитектура сетей подвижной связи очень сходны (см. рис. 5.3.1). При определении местоположения мобильного абонента между элементами сетей подвижной связи используется сигнализация, основанная на принципах транзакций, похожая на ту, которая используется при запросе услуги IN. Центр коммутации сети подвижной связи (MSC – Mobile switching center), к которому попадает вызов, направленный к абоненту обслуживаемой этим MSC сети, передает в регистр местоположения «своих» абонентов (HLR – Home location register) запрос о том, где находится в данный момент этот абонент. HLR постоянно обновляет информацию о местоположении абонента на основе данных, получаемых из последней «чужой» сети, в которой тот оказался, и по запросу MSC передает ему информацию, необходимую для маршрутизации.

Однако ни стационарные IN, ни сети подвижной связи не обладают теми возможностями, какие могла бы иметь сеть, соединившая в себе свойства и тех, и других. Стационарные IN-сети (как с набором CS-1, так и с набором CS-2) не владеют в полной мере механизмами поддержки мобильности, а сети подвижной связи не способны адекватно обеспечивать принцип независимости от услуг, присущий концепции IN. Естественно, что операторы сетей подвижной связи стремятся овладеть преимуществами, предлагаемыми концепцией IN, а операторы стационарных сетей IN заинтересованы в услугах, поддерживающих мобильных абонентов.

Независимо от того, какой использован подход к формированию беспроводной Интеллектуальной сети, она приобретает черты, при-

сущие сетям подвижной связи, такие как необходимость контроля передвижения мобильного абонента, специфика радиодоступа и проблемы роуминга услуг.

КОНТРОЛЬ ПЕРЕДВИЖЕНИЯ МОБИЛЬНОГО АБОНЕНТА. В сетях подвижной связи, в отличие от стационарных сетей, местоположение абонента заранее не известно и динамически определяется с помощью специальных процедур сигнализации, когда к абоненту нужно доставить входящий вызов. При перемещении абонента из зоны, которая обслуживается «своей» сетью, в другую зону, обслуживаемую «чужой» сетью, его терминал регистрируется этой «чужой» сетью, и она передает в сеть, которая является для этого абонента «своей», обновленные данные о его местоположении. «Своя» сеть подтверждает прием этих данных и передает в «чужую» сеть, зарегистрировавшую терминал, информацию о профиле абонентских данных.

События, связанные с контролем передвижения мобильного абонента (например, регистрация терминала в «чужой» сети), как не зависящие от событий в процессе обслуживания вызова, так и зависящие от них, могут представлять интерес в качестве точек обнаружения DP. Это предполагает разработку соответствующих моделей, что связано с решением проблемы формального представления связей между процессом обслуживания вызова, контролем передвижения и логикой услуг. Задача создания моделей для состояний процесса контроля передвижения мобильного терминала вне связи с процессом обслуживания вызова не имеет аналога в стационарных IN-сетях (кроме осуществляемой по инициативе абонента регистрации его в качестве пользователя услугой, как, например, в случае услуги UPT).

СПЕЦИФИКА РАДИОДОСТУПА. В некоторых случаях запрос услуги необходимо связать с событиями, происходящими при выполнении таких специфических для радиодоступа процедур как аутентификация, постоянный контроль качества канала связи и эстафетная передача управления, которые также не имеют аналогов в стационарных сетях связи.

РОУМИНГ УСЛУГ. Одним из главных преимуществ, предлагаемых сетями сотовой подвижной связи, является возможность свободного перемещения абонента из своей зоны в другие зоны, обслуживаемые сетями подвижной связи других операторов. При этом абонент хотел бы сохранить свой персональный набор (профиль) услуг и в других сетях, иными словами, иметь возможность роуминга услуг. При применении концепции IN к сетям подвижной связи обеспечение сохранности профиля услуг сопряжено с рядом трудностей.

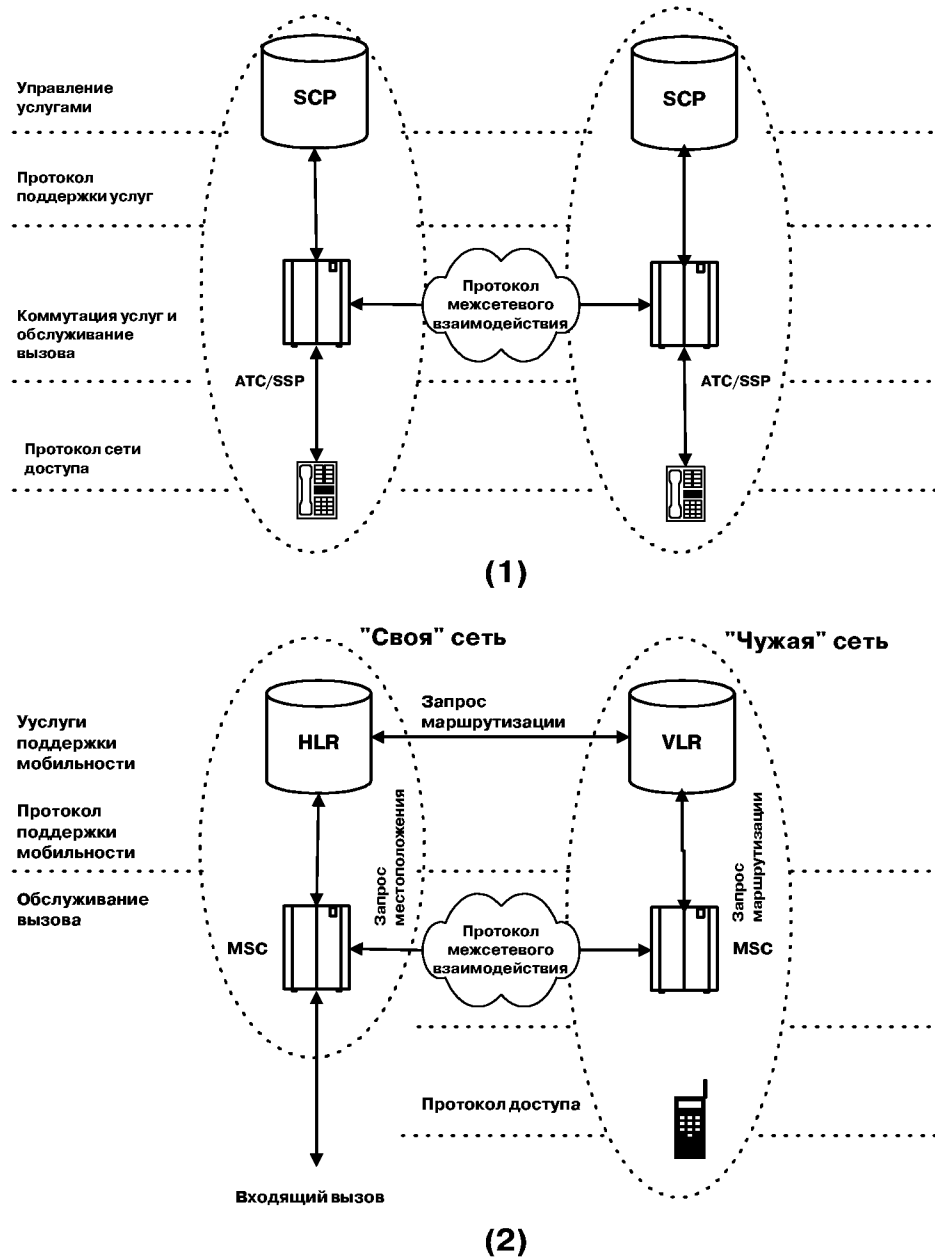


Рис. 5.3.1 Архитектура IN (1) и сети подвижной связи (2)

Во-первых, в отличие от проводных сетей, в коммутаторах подвижной связи триггерные точки и профиль услуг, как правило, не представляются в виде статических данных, а определяются при регистрации. Возможности «своей» и «чужой» сетей могут оказаться раз-

ными, что, скорее всего, скажется на перечне и на характеристиках услуг, которые смогут быть предложены пользователю, переместившемуся в «чужую» сеть. Во-вторых, в процессе обслуживания одного вызова может участвовать более одного SSP.

5.3.2 Способы перехода к беспроводной IN

Возможны два основных подхода к конвергенции мобильных сетей и IN. Первый – сформировать или «наложить» концепцию IN на архитектуру существующих сетей подвижной связи; второй – дополнить свойствами поддержки мобильности концепцию IN, ориентированную преимущественно на стационарные сети. Выбор того или другого подхода зачастую определяется заинтересованной стороной (то есть администрацией сети подвижной связи или сети IN).

Второй подход, которому следует ITU-T, предполагает, что реализовать полную поддержку мобильности в IN можно будет не ранее реализации набора CS-4, после завершения работ по спецификации систем связи третьего поколения. Первый подход более прагматичен и может быть реализован достаточно простыми средствами в ближайшем будущем. Однако его сторонники тоже разделились на две группы.

Первая группа придерживается мнения, что протоколы сигнализации, используемые в сетях подвижной связи (MAP IS-41 или MAP GSM), фактически уже являются протоколами IN. Такая точка зрения основана на убеждении, что процесс доставки вызова к мобильному абоненту есть услуга IN, и что сетевые объекты, которые выполняют эту функцию (HLR), по существу представляют собой специализированные пункты управления услугами (SCP). Сказанное подтверждает сравнение процедур запроса данных о местоположении мобильного терминала и запроса услуги IN – обе процедуры приводят к обмену инструкциями, нужными для маршрутизации и для установления соединения. В связи с этим предлагается модифицировать существующий протокол подвижной связи в соответствии с концепцией IN и адаптировать его к более унифицированным требованиям, после чего любое различие между запросами, специфическими для подвижной связи, и запросами услуг IN будет «размыто».

Вторая группа признает схожесть прикладных протоколов сетей подвижной связи и сетей IN, однако считает первые недостаточно общими для того, чтобы они могли поддерживать концептуальные идеи IN. Поэтому предлагается рассматривать обращение к услуге IN в сети подвижной связи как процесс, который происходит, в значительной степени, независимо от сигнализации, служащей для установления соединения, и свести к минимуму роль HLR в реализа-

ции услуг IN. Доставка вызова мобильному абоненту считается основной функцией, а не услугой IN. Операции, используемые для доставки вызова, не изменяются с введением операций IN, поскольку последние не зависят от протокола установления соединения. Различие между сигнализацией, специфической для подвижной связи, и сигнализацией для поддержки услуги усиливается, поскольку та и другая остаются логически разными.

Учитывая потребность в конвергенции концепции IN и свойств мобильности, организации, занимающиеся стандартизацией, активно работают в этой области. В частности:

- Ассоциация Промышленности Связи (TIA) в лице своего филиала – Американского Национального Института Стандартов (ANSI) – разработала стандарт для беспроводной Интеллектуальной Сети под названием WIN (Wireless Intelligent Network);
- Европейский Институт Стандартов в области связи (ETSI) разработал стандарт поддержки услуг IN в сетях GSM под названием CAMEL (Customized Application for Mobile Network Enhanced Logic);
- Международный союз электросвязи (ITU-T) продолжает работу над развитием концепции IN, в спецификации которой должны частично войти соответствующие разделы пакета рекомендаций для системы подвижной связи следующего (третьего) поколения FPLMTS (Future Public Land Mobile Telephone System), переименованной недавно в IMT-2000 (International Mobile Telecommunications System – 2000).

Рассмотрим различия подходов, выбранных каждой из этих организаций для решения некоторых проблем объединения концепций IN и сетей подвижной связи.

5.3.3 Стандарт WIN (ANSI TIA)

WIN представляет собой попытку TIA ввести концепцию IN в существующий стандарт ANSI-41 (ранее IS-41). Разработка стандартов TIA традиционно была ориентирована на конкретные услуги, что первоначально приводило к определению операций и параметров, специфических для каждой услуги. Сочетание такой ориентации со стремлением повысить эффективность сигнализации привело к тому, что запросы, относящиеся к подвижной связи, часто становились также и запросами, касающимися услуг.

Неоднократные изменения стандарта IS-41 приносили в протокол все более унифицированные операции. Это обстоятельство, а также традиционно длинный цикл стандартизации услуг, вызвали интерес TIA к использованию концепции IN. За основу стандарта WIN

был взят набор CS-2 ITU-T. Общая архитектура WIN представлена на рисунке 5.3.2.

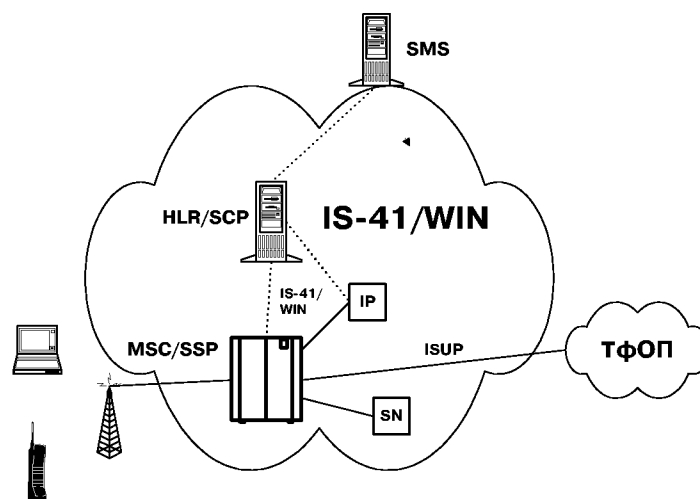


Рис. 5.3.2 Общая архитектура WIN

На рисунке 5.3.3 представлена архитектура распределенной функциональной плоскости WIN. В затененных овалах на рисунке показаны дополняющие CS-2 функциональные объекты, необходимые для поддержки WIN. Поясним кратко их основные функции.

- ACF (Authentication control function) – функциональный объект управления процедурой аутентификации.
- RACF (Radio access control function) – функциональный объект управления радиодоступом; поддерживает процедуру поиска мобильного терминала, управление радиоканалом и процедуру эстафетной передачи управления.
- LRF (Location registration function) – функциональный объект регистрации местонахождения; содержит логику и данные для административного управления мобильностью.
- RTF (Radio terminal function) – функциональный объект поддержки функций радиотерминала; является шлюзом между мобильным пользователем и сетевыми функциями управления связью пользователя, выполняя функции, аналогичные функциям CCAF в стационарной IN.
- RCF (Radio control function) – функциональный объект управления радиопортами; управляет генерацией несущих радиочастот, усилением сигнала, модуляцией/демодуляцией, назначением радиоканалов, контролем канала.

Модель базового процесса обслуживания вызова BCSM CS-2 полностью принята для WIN. Однако имеются и существенные различия в процессах обработки вызова в IN и WIN – в возможностях динамич-

ческого назначения триггерных точек и в типах критериев триггеров. Кроме того, BCSM уже не содержит всей информации, необходимой для запроса услуги. Недостающую информацию должны предоставлять другие объекты, в первую очередь LRF.

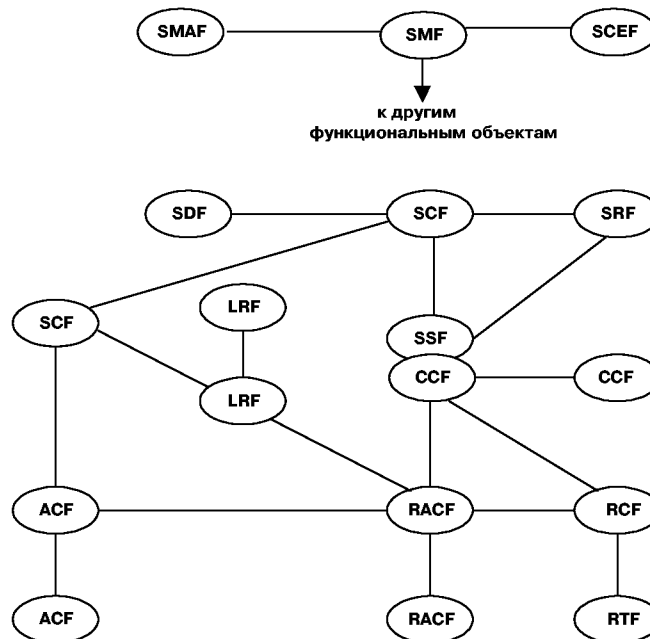


Рис. 5.3.3 Архитектура распределенной функциональной плоскости WIN

Чтобы предоставить производителям оборудования достаточно гибкие возможности создания его новых реализаций, стандарт WIN не определяет элементы физической плоскости (PE). В отличие от физической плоскости IN, стандарт WIN оперирует сетевыми элементами (NE – Network element). Кроме IP, SCP и SN типовая модель сети стандарта ANSI-41.1 содержит:

- AC:** центр аутентификации; содержит функциональный объект ACF и управляет информацией аутентификации подвижных станций. AC может быть совмещен с регистром HLR и способен обслуживать несколько регистров HLR.
- EIR:** регистр идентификации оборудования; хранит информацию идентификации оборудования пользователей.
- HLR:** регистр местонахождения «своих» абонентов; содержит функциональные объекты LRF, SCF и SDF. HLR может быть совмещен с MSC и обслуживать несколько MSC.
- MC:** центр сообщений; записывает и передает короткие сообщения.
- MS:** подвижная станция; содержит функциональный объект RTF.
- MSC:** центр коммутации подвижной связи; содержит функциональные объекты CCF, SSF и RACF (может также содержать LRE и SRF) и является шлюзом, пропускающим трафик между мобильной и стационарной сетями и/или другими мобильными сетями.
- VLR:** регистр местонахождения «чужих» абонентов; содержит функциональные объекты LRF и ACF и управляет данными о «чужих» абонентах.

Как физические, так и сетевые элементы могут содержать в себе каждый несколько функциональных объектов. Однако сетевые элементы, в отличие от физических, могут быть объединены в одной единице оборудования. В качестве протокола для интерфейсов между сетевыми элементами стандартом WIN определен протокол MAP стандарта ANSI 41.

Возможности открытых интерфейсов между функциональными объектами WIN достаточны для поддержки любой услуги из набора CS-2. Таким образом, использование BCSM CS-2 и новой модели состояний LRF представляет собой значительный шаг в направлении интеграции принципов IN в инфраструктуру беспроводных сетей стандарта IS-41.

5.3.4 Стандарт CAMEL (ETSI)

5.3.4.1 Концепция и архитектура CAMEL

При перемещении мобильного пользователя из сети GSM одного оператора в сеть GSM другого оператора ему (вплоть до фазы 2) мог предоставляться только тот набор услуг, который определен стандартом. Однако, чтобы привлечь большее число клиентов, операторы искали способы отличаться друг от друга, в первую очередь, спектром предоставляемых услуг. Применение разными производителями концепции IN CS-1 в сетях GSM приводило к несовместимым реализациям и затрудняло (а иногда и вовсе исключало) возможность взаимодействия изготовленного ими оборудования. Особенно это касалось протокола INAP и тех функциональных средств, которые должны быть заложены в SSP и SCP. Кроме того, предоставление услуг ограничивалось в каждой сети GSM лишь «своими» пользователями.

CAMEL – это попытка комитета SMG ETSI разработать стандарт для поддержки национального и международного роуминга услуг, не специфицированных стандартом GSM. CAMEL можно рассматривать как интеграцию IN и архитектуры GSM путем (1) адаптации существующего протокола сигнализации MAP стандарта GSM к расширенным требованиям и (2) введением сигнализации IN для поддержки не стандартизированных GSM услуг.

В новой архитектуре функции IN и функции, специфические для подвижной связи, логически разделены. По существу, CAMEL заимствовал протокол INAP CS-1 и приспособил его к особенностям процесса обслуживания вызова в сетях GSM, определив необходимые триггерные точки DP. В протоколе MAP GSM были сделаны изменения для поддержки передачи информации о триггерных точках к коммутаторам «своих» и «чужих» сетей. CAMEL сохраняет протокол MAP GSM независимым от протокола INAP CS-1, позволяя обоим развиваться

самостоятельно. Такой прагматический подход предоставляет относительно простые механизмы обслуживания вызовов, требующих услуг IN, в инфраструктуре сетей подвижной связи.

Как платформа, CAMEL позволяет операторам сетей GSM определять и вводить новые услуги, не требуя их стандартизации в рамках стандарта GSM. Тем самым операторы получают возможность отличаться друг от друга спектром предоставляемых услуг и, что самое важное, обеспечивается национальный и международный роуминг этих услуг.

Фаза 1 стандарта CAMEL является первым шагом интеграции IN в стандарт GSM. Эта фаза содержит в себе возможности CS-1 в отношении операций протокола и точек обнаружения и выходит за рамки CS-1 в части усовершенствований, относящихся к мобильным приложениям. В CAMEL предусмотрены новые механизмы, обеспечивающие одновременно процедуры запроса как стандартных баз данных GSM, так и баз данных CAMEL, и способы корректного взаимодействия со стандартными дополнительными услугами GSM, например, такими как «переадресация». Кроме того, определены процедуры подавления иноязычных речевых сообщений автоинформатора и процедуры запросов регистра HLR узлом управления услугами.

Для реализации первой фазы CAMEL определена самостоятельная прикладная подсистема OKC-7 (CAMEL Application Part) и соответствующий протокол CAP (CAMEL Application Protocol), базирующийся на стандарте ETSI INAP CS-1.

Рисунок 5.3.4 представляет функциональную архитектуру, необходимую для поддержки первой фазы CAMEL. Архитектура второй фазы CAMEL отличается наличием интерфейса CAP между функциональным объектом gsmSCF и внешним функциональным объектом специализированных ресурсов gsmSRF. Представленная на рисунке функциональная архитектура содержит как традиционные для сети GSM элементы, так и новые, определенные стандартом CAMEL.

HLR хранит информацию об абонентах, требующих поддержки CAMEL, т.е. информацию о текущей подписке, и обеспечивает интерфейс в сторону функционального объекта gsmSCF для осуществления процедур запроса. Логика услуг CAMEL, поддерживающая предоставление нестандартных для сети GSM дополнительных услуг, содержится в gsmSCF. В процессе предоставления услуги gsmSCF взаимодействует с gsmSSF, gsmSRF (для второй фазы CAMEL) и HLR. Интерфейсом между центром коммутации подвижной связи и gsmSCF служит gsmSSF, принципы работы которого аналогичны SSF CS-1, но предусматривают новые триггерные механизмы, обусловленные особенностями мобильной связи.

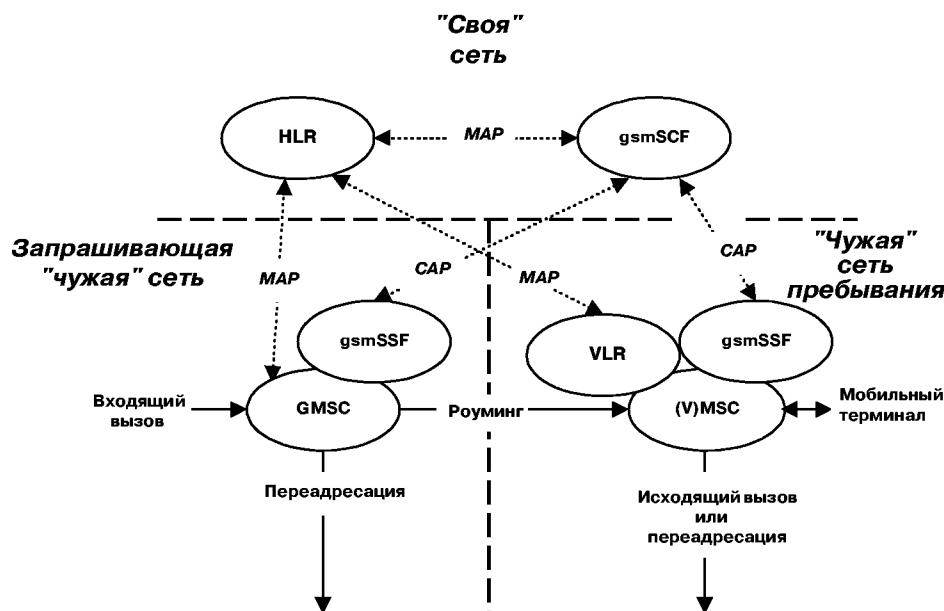


Рис. 5.3.4 Функциональная архитектура первой фазы стандарта CAMEL

При обработке входящих/исходящих вызовов, требующих поддержки CAMEL, в направлении от/к другой сети, шлюзовый центр коммутации подвижной связи GMSC (Gateway MSC) запрашивает от HLR информацию о наличии подписки абонента на услуги CAMEL. Если подписка существует, GMSC запрашивает от gsmSSF необходимые инструкции. На основе полученных инструкций GMSC отслеживает изменения состояний процесса обслуживания вызова и информирует о них gsmSSF, позволяя тем самым gsmSSF управлять обслуживанием вызова в GMSC. Интерфейс между GMSC и gsmSSF является внутренним. При обнаружении запроса услуги gsmSSF информирует об этом gsmSCF.

Регистр местонахождения «чужих» абонентов (VLR – Visitor location register) хранит информацию о подписке CAMEL при исходящем вызове и о подписке на дополнительные услуги как часть данных о «чужих» абонентах, находящихся в обслуживаемой этим VLR зоне. При исходящем внутрисетевом вызове MSC получает от VLR информацию о наличии подписки абонента на услуги CAMEL. Наличие подписки указывает (V)MSC (MSC «чужой» сети) на необходимость обмена инструкциями с gsmSSF. При обработке запроса дополнительной услуги (V)MSC получает от VLR информацию о том, что абонент на нее подписан, и это указывает gsmSSF на необходимость информировать gsmSCF о запросе услуги.

5.3.4.2 Интерфейсы CAMEL

Интерфейс HLR – VLR используется для передачи информации об абоненте в «чужую» сеть GSM и для предоставления номера при роуминге подвижного абонентского терминала. Посредством этого интерфейса также извлекаются данные о статусе и местонахождении подвижного абонента, определяется необходимость подавления иноязычных речевых сообщений для услуг CAMEL, и передаются указания подавить эти сообщения. Интерфейс GMSC – HLR используется при входящих вызовах для обмена информацией о маршрутизации, о статусе и местонахождении абонента, а также информацией о текущей подписке абонента на услуги и о необходимости подавления иноязычных речевых сообщений автоинформатора. Через этот же интерфейс в запрашивающую сеть GSM передается информация о подписке абонента на услуги CAMEL при входящем и исходящем вызовах.

Интерфейс gsmSSF – gsmSCF используется для обмена инструкциями в процессе обслуживания вызова и для передачи запросов соединения с gsmSRF. Важно отметить, что, в отличие от IN CS-1 и CS-2, интерфейс gsmSSF – gsmSCF определен также и для межсетевых (международного) взаимодействия. Интерфейс gsmSCF – HLR используется функциональным объектом gsmSCF для запроса информации от HLR. Предполагается, что HLR может отказать в предоставлении информации, запрошенной gsmSCF.

Интерфейс gsmSCF – gsmSRF применяется на второй фазе реализации стандарта CAMEL и используется gsmSCF для пересылки функциональному объекту gsmSRF инструкций о передаче пользователю акустических сигналов или речевых сообщений. Интерфейс MSC – gsmSCF используется MSC для извещения gsmSCF о запросе пользователем дополнительных услуг. Интерфейсы GMSC – gsmSSF и (V)MSC – gsmSSF являются внутренними и описываются лишь для того, чтобы облегчить понимание процесса обработки точек обнаружения DP.

5.3.5 Система IMT-2000 (FPLMTS)

5.3.5.1 Архитектура систем связи третьего поколения

ITU-T продолжает работу над стандартами для систем подвижной связи третьего поколения, архитектура которых будет использовать принципы построения Интеллектуальных сетей. Пакет проектов рекомендаций, имевший название FPLMTS, был переименован в IMT-2000 – International Mobile Telecommunications System 2000 (международная система подвижной связи 2000 года). Принципы IMT-2000 нацелены на достижение максимальной общности между

различными радио-интерфейсами с целью упростить использование многорежимных терминалов. Сотовая структура применима в самых разных условиях: от квартирных беспроводных телефонов до спутниковых систем с большой областью охвата. Система IMT-2000 предназначена для поддержки широкого спектра услуг, включая мультимедийные, принципы предоставления которых будут основаны на концепции IN.

Функциональная архитектура системы обеспечивает полную интеграцию контроля передвижения и функций IN. Первоначально предполагалось, что SCP, кроме управления услугами, будет также отвечать за контроль местоположения, управление профилями услуг и аутентификацию. Однако, под давлением сторонников CAMEL и WIN, был поддержан принцип раздельного рассмотрения аспектов мобильности и аспектов обслуживания. Архитектура системы IMT-2000 описывается набором из четырех высокоуровневых функциональных подсистем, каждая из которых содержит ряд функциональных объектов. Подсистемы и интерфейсы, через которые осуществляется связь между ними, представлены на рисунке 5.3.5.

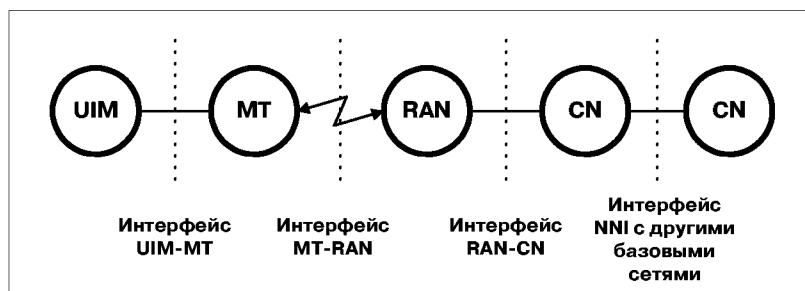


Рис. 5.3.5 Функциональные подсистемы и интерфейсы IMT-2000

Модуль идентификации пользователя (UIM) обеспечивает защиту информации пользователя, может быть реализован в виде отдельной карты, вставляемой в мобильный терминал, или быть встроен в него. Мобильный терминал (MT) взаимодействует с UIM и сетью радиодоступа, обеспечивая поддержку услуг и мобильности пользователя. Сеть радиодоступа (RAN) обеспечивает взаимодействие MT с базовой сетью, выполняя роль маршрутизатора и шлюза для обмена информацией. Базовая сеть (CN) взаимодействует с сетью радиодоступа и с другими базовыми сетями, обеспечивая поддержку услуг и мобильности пользователя.

На рисунке 5.3.6 изображена сетевая модель IMT-2000 и возможный вариант объединения функциональных объектов в физические элементы. Функциональные объекты модели разделены на три группы: (1) группа объектов управления радиодоступом, (2) группа объектов управления связью и услугами и (3) группа объектов управления пакетными данными.

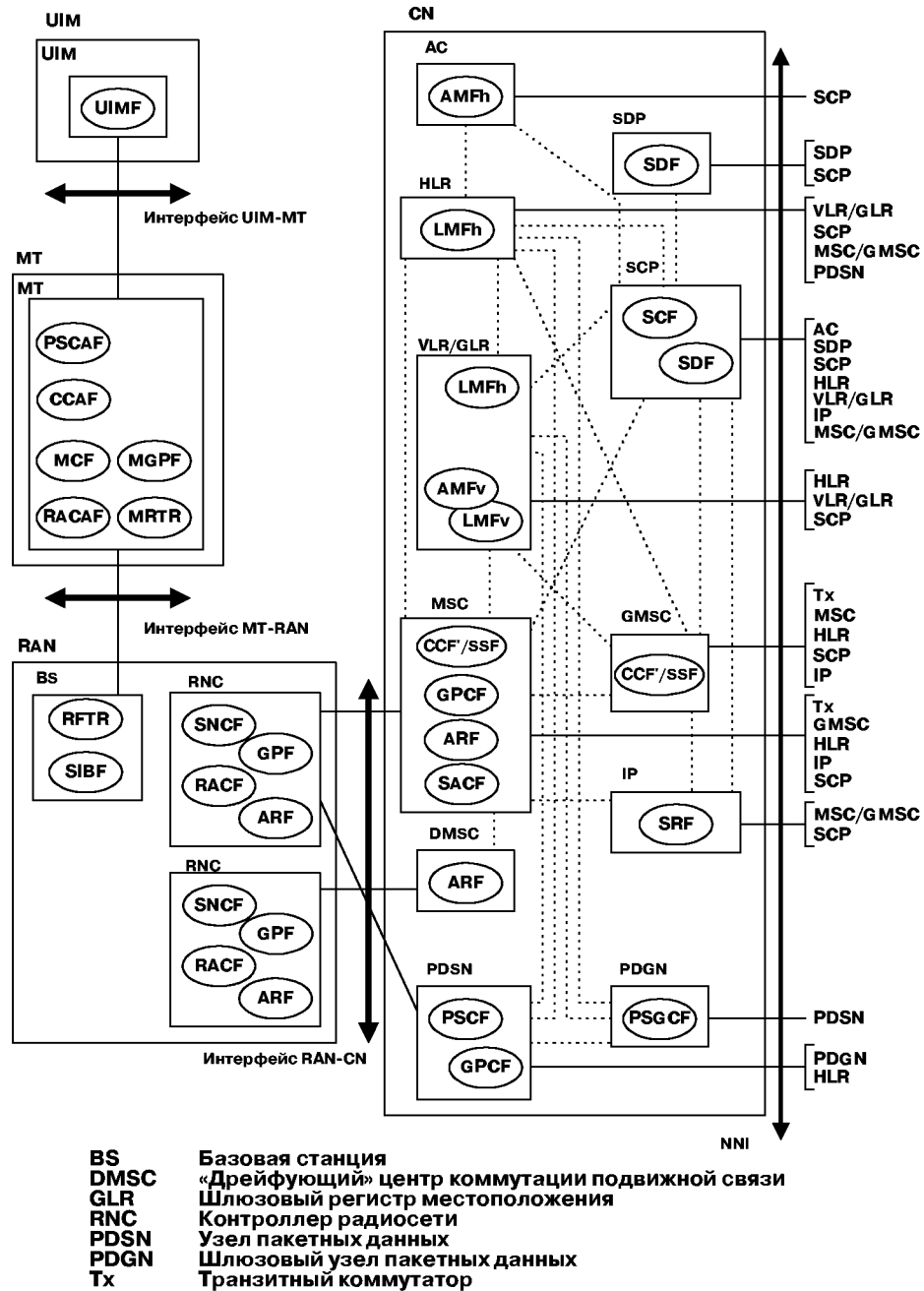


Рис. 5.3.6 Сетевая модель IMT-2000

Первая группа содержит функциональные объекты управления радиодоступом (RACF), вспомогательного управления (RACAF), приема и передачи радиочастот (RFTR), приема и передачи радиосигналов мобильного терминала (MRTR), ретрансляции канала доступа (ARF), распространения информации системного доступа (SIBF), управления спутниковой сетью (SNCF), а также объект, связанный с обработкой задач поиска мобильного терминала на стороне радиодоступа (GPF).

Функциональные объекты, входящие во вторую группу, в свою очередь, делятся на объекты, относящиеся к стороне сети, и объекты, относящиеся к стороне мобильного терминала. К стороне сети относятся функциональные объекты SCF, SRF, модифицированные для мобильной связи объекты CCF, SSF, SDF и SMF, объект управления установлением соединений (CnCF), объект административного управления определением местоположения (LMF), объект административного управления аутентификацией (AMF), объект управления доступом к услугам (SACF) и объект контроля географического местоположения на сетевой стороне (GPCF). К стороне мобильного терминала относятся функциональные объекты логики управления доступом к услугам (MCF) и административного управления идентификацией пользователя (UIMF), модифицированный для мобильной связи объект CCAF', объект вспомогательного управления соединением (CnCAF) и объект, связанный с управлением поиском мобильного терминала (MGPF).

В третью группу входят функциональные объекты, связанные с управлением пакетными данными, более подробно ознакомиться с которыми можно в проекте рекомендации ITU-T Q.FIN.

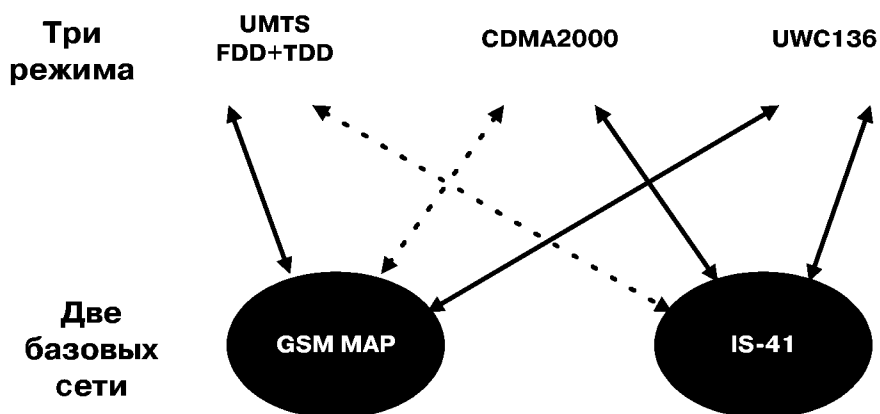


Рис. 5.3.7 Соотношение между сетевыми и радиоинтерфейсами

В спецификациях IMT-2000 ITU-T сформулировал общие требования к системам мобильной связи третьего поколения. Не заставили себя ждать и предложения с конкретными проектами от институтов по стандартизации. Предложенная ETSI (и наиболее известная в Европе) система UMTS – это лишь одно среди более чем десятка предложений в этой области. Так, японские специалисты предложили концепцию широкополосного доступа CDMA – WCDMA, а США представили сразу две концепции: CDMA2000, как эволюцию стандарта IS-95 (коммерчески известную под именем CDMAOne), и UWC-136, которая является развитием стандарта IS-136.

UMTS является результатом эволюции мира GSM, а CDMA2000 – эволюции мира IS-41. В ходе исследований, направленных на то, чтобы обеспечить совместимость GSM и IS-41, были определены три технологии для радиоинтерфейса, совместимого с двумя базовыми сетевыми стандартами. На рисунке 5.3.7 показано, как соотносятся с разными базовыми сетями различные режимы радиодоступа.

5.3.5.2 Система UMTS

Универсальная система мобильной связи (UMTS – Universal mobile telecommunication system) известна специалистам, большей частью, в связи обсуждением соглашения о радиоинтерфейсе для систем третьего поколения. Однако имеются и другие, не менее важные аспекты будущего стандарта. UMTS должна открыть путь для перехода к унифицированной сети и новым услугам. При этом основная задача, которая должна быть решена UMTS, – построение мобильных систем, способных предложить рынку мобильной связи, наряду с услугами передачи речи, и услуги мультимедиа.

В дополнение к ограниченному набору «классических» услуг связи (речь, экстренные вызовы, передача коротких сообщений, факсимильная связь и доступ в Internet), UMTS должна будет поддерживать, в рамках концепции *виртуально «домашней» обстановки* (VHE – Virtual home environment), услуги, максимально ориентированные на конечного пользователя. Согласно концепции VHE, одинаковые услуги будут предлагаться абоненту как в своей, так и в чужой сети, независимо от типа используемого им терминала.

На базе архитектуры VHE (рис. 5.3.8), UMTS реализует новое поколение технологии мобильной связи. Персональная связь будет предлагать услуги, не зависящие от местоположения абонента, от типа его терминала и от средств передачи (проводных или беспроводных). Перечень услуг персональной связи будет включать в себя как услуги, предоставляемые стационарными сетями, так и услуги, доступные сейчас только в инфраструктуре мобильной и беспроводной связи.



Рис. 5.3.8 Архитектура VNE

В то время как в радиодоступе применение новых технологий CDMA и ATM четко разделило системы GSM и UMTS, архитектура базовой сети для UMTS определена как результат эволюции сетевой подсистемы GSM, что допускает возможность плавного перехода существующих сетей к этой новой архитектуре.

Первая спецификация систем связи третьего поколения для базовой сети UMTS (3GPP), датированная 1999 годом, основывается на «смешанном» подходе, который предусматривает использование

двух технологий: коммутацию каналов и коммутацию пакетов. Одновременно с этим подходом операторы и поставщики оборудования находятся в процессе определения следующих спецификаций 3GPP, в которые войдет архитектура, полностью базирующаяся на транспортной IP-сети. Ожидается, что в течение последующих лет базовая сеть будет оптимизирована для передачи большого объема данных и, если мобильные сети будут основаны на технологии IP, одни и те же сетевые ресурсы будут использоваться для предоставления услуг и мобильной, и стационарной связи.

Глава 5.4

IN и Internet

5.4.1 Совместить несовместимое

С конца 80-х годов Интеллектуальная сеть играет важную роль в развитии быстро изменяющегося рынка средств и услуг связи. Применение концепции IN позволило значительно уменьшить сроки введения новых телефонных услуг, исключив необходимость адаптации к каждой новой услуге всех коммутационных узлов сети. Однако поскольку платформы IN следуют принятому в телефонных сетях принципу «закрытости» для внешнего доступа, они ограничены в части предоставления средств создания услуг сторонним поставщикам (провайдерам), что в значительной степени ограничивает количество и разнообразие услуг.

С конца 90-х годов, наряду с увеличением популярности «чисто» телефонных дополнительных сетевых услуг, стал расти спрос на простые в использовании и индивидуально настраиваемые интерактивные услуги мультимедиа. Всемирная сеть Internet, развитие которой шло параллельно развитию IN, прочно заняла эту нишу и предлагает пользователям широкий диапазон приложений, ориентированных изначально, в основном, на передачу данных. Новые услуги Internet внедряются на рынке быстрее услуг IN, а их популярность связана с присущей сети Internet открытостью. По этим причинам производители оборудования и поставщики услуг стали интенсивно исследовать возможности введения услуг, предоставляемых телефонными сетями «поверх» Internet. Основная возникающая при этом проблема – низкое качество связи, обусловленное характеристиками надежности Internet, большими (и, к тому же, переменными по величине) задержками при транспортировке речевых пакетов и отсутствием гарантии их доставки в правильной последовательности.

Несмотря на общеизвестные недостатки сетей, организованных по протоколу IP (Internet Protocol) или, проще говоря, IP-сетей, взрывной рост услуг Internet явился одним из наиболее значительных событий в современных телекоммуникациях. Общепризнано, что в ближайшем будущем объем трафика передачи данных превысит объем речевого трафика (для большинства операторов США и Западной Европы это – уже свершившийся факт). Такое изменение характеристик спроса может стать драматичным для инфраструктуры телекоммуникационных сетей и систем эксплуатационного управления. Похоже на то, что часть сетей связи, ориентированных на передачу ре-

чевого трафика, будет заменена сетями, ориентированными на передачу данных, скорее всего – IP-сетями. Однако этот процесс будет иметь эволюционный, а не революционный характер, поскольку операторы телефонных сетей постараются сначала окупить крупные инвестиции, вложенные сравнительно недавно в модернизацию сетей коммутации каналов.

В этих условиях оптимальное продвижение на телекоммуникационном рынке новых классов услуг связано с сетевой архитектурой, способной объединить общедоступность услуг телефонии и «дружественную» оболочку услуг Internet. Новые классы услуг и потребность в управлении ими через Internet диктуют целесообразность создания архитектуры, единой для IN и для Internet. Объединение высокой надежности и защищенности IN, которыми традиционно обладают телефонные сети общего пользования, с богатством возможностей и с открытостью информационных технологий, присущих сетям IP, ведет к новому видению «интеллектуальности» сети.

Интеллектуальность сети теперь рассматривается не как решение, специфическое для определенной технологии, а, напротив, как средство объединения различных технологий, механизмов и методологий для развертывания новых классов услуг. При этом интеллектуальность сети играет двоякую роль. Она должна, во-первых, предоставить разным поставщикам услуг возможности создания услуг и управления ими в разнородных сетях, во-вторых, обеспечить пользователям доступ к услугам из разных сетей вне зависимости от типа используемого терминала.

До недавнего времени телефонные сети и IP-сети рассматривались как два отдельных мира, взаимно дополняющих друг друга (телефонная сеть – для доступа, IP-сеть – для транспорта), и как потенциальные конкуренты (передача речи «поверх» протокола IP и традиционная телефония). Сети для обмена речевой информацией и сети для передачи данных основывались на разных философиях – коммутации каналов и коммутации пакетов, соответственно, – и потому предлагали различные механизмы доставки информации. Это определило и различие в подходах к предоставлению услуг (см. рис. 5.4.1). Так, –

- в области традиционной телефонии для введения новых услуг в инфраструктуре сетей общего пользования используется архитектура централизованного управления услугами на основе концепции IN;
- в IP-сетях «интеллект» (логика услуг и сопутствующие данные) распределен по множествам приложений, размещенных в конечных пунктах сети (пользовательских компьютерах и серверах).

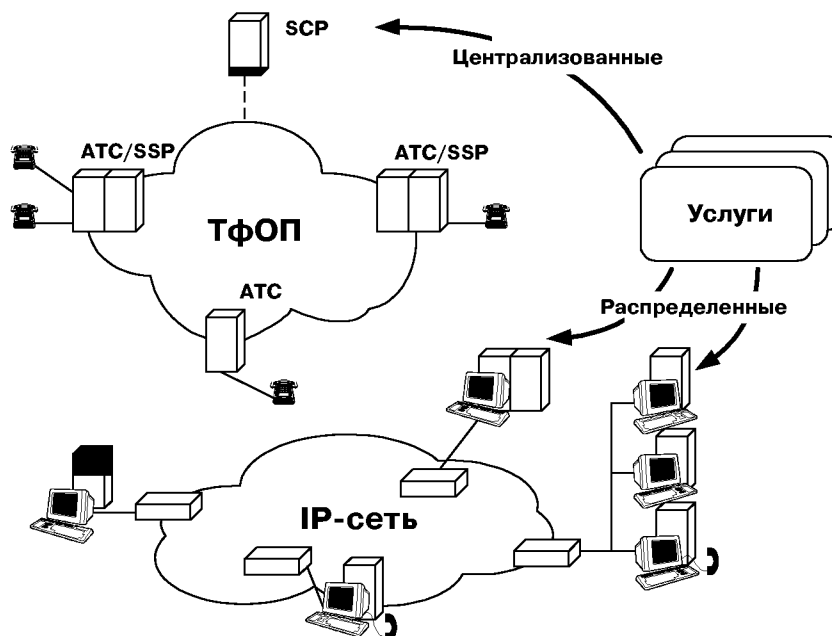


Рис. 5.4.1 Подходы к предоставлению услуг в IN и IP-сетях

Однако рынок услуг связи диктует потребность видеть эти два мира объединенными в одной глобальной сети, способной предложить лучшее от каждого из них. Для такой конвергенции требуются унифицированные принципы предоставления услуг и управления ими. Конвергенция услуг возможна лишь тогда, когда абонентские данные, хранящиеся в разных сетях, совместимы или, по крайней мере, преобразуемы из одного формата в другой. В процессе конвергенции IP-сетей и телефонных сетей ведущую роль играет Интеллектуальная сеть. В силу главной особенности IN-сетей, заключающейся в их физическом и логическом размещении *над* базовыми сетями, именно IN может успешно выступать в качестве « сетевого интеллекта » объединенной сети.

5.4.2 Интегрированные IN/Internet-услуги

Платформа IN, в зависимости от потребностей пользователей и рынка, может быть использована в широком спектре приложений, начиная с услуг управления абонентскими данными для операторов, Internet-провайдеров и поставщиков информации, услуг передачи речи через IP-сети, услуг преобразования телефонных адресов в IP-адреса (и обратно) и заканчивая услугами электронной коммерции, в которых сетевой оператор выступает в роли « доверительной стороны » для пользователей, Internet-провайдеров и поставщиков информации.

С конца 90-х годов операторы сетей связи начали предлагать решения, базирующиеся на интеграции телефонных услуг и услуг Internet. Первым примером такого рода решений может служить использование протокола IP для передачи междугородных факсов. Вскоре на рынке появились услуги типа «Click-to-Dial» (телефонный вызов подачей команды со стороны компьютера, включенного в IP-сеть), которым прочат большую популярность.

УСЛУГА «CLICK-TO-DIAL» (CTD) дает пользователю возможность во время сеанса связи с Internet произвести телефонный вызов путем простой активизации соответствующей пиктограммы на экране своего компьютера. Адрес вызываемой и/или вызывающей стороны, может быть как IP-адресом, так и номером телефонной линии.

УСЛУГА «VIRTUAL SECOND LINE» (VSL) позволяет абоненту ответить на входящий телефонный вызов, не прерывая сеанса связи с Internet. Для этого может быть использован специальный шлюз, преобразующий речевые телефонные сигналы в поток передачи речи по протоколу IP (VoIP – Voice over IP) к терминалу пользователя, подключенному к Internet.

УСЛУГА «INTERNET CUSTOMER PROFILE MANAGEMENT» (ICPM) позволяет управлять профилем услуги IN с персонального компьютера прямо из Web-страницы. В настоящее время абонент или пользователь услугой IN может управлять профилем своей услуги при помощи сигналов DTMF или с помощью оператора.

УСЛУГА «INTERNET CALL WAITING» (ICW) позволяет известить пользователя во время сеанса связи с Internet о поступившем телефонном вызове. После того как пользователь будет оповещен, у него имеется на выбор несколько опций: ответить на вызов, приостановив сеанс с Internet, переадресовать вызов в почтовый ящик речевых сообщений, передать вызывающему абоненту сигнал ожидания или вовсе игнорировать вызов. Если пользователь решил ответить на вызов, то устанавливается соединение в телефонной сети, а сеанс с Internet прекращается или приостанавливается с отключением модема. Схема предоставления услуги ICW представлена на рисунке 5.4.2.

УСЛУГА «IMPROVED VPN» позволяет использовать большинство привлекательных особенностей VPN, распространяя их на сети IP. Кроме традиционных атрибутов услуги VPN, «продвинутая» VPN обеспечивает:

- объединение терминалов разных типов (например, стационарных, мобильных, персональных компьютеров) в единую виртуальную частную сеть;
- использование перекрестных связей между терминалами разных типов;

- управление соединениями с целью выбора для них оптимального по стоимости маршрута как внутри VPN, так и вне ее.

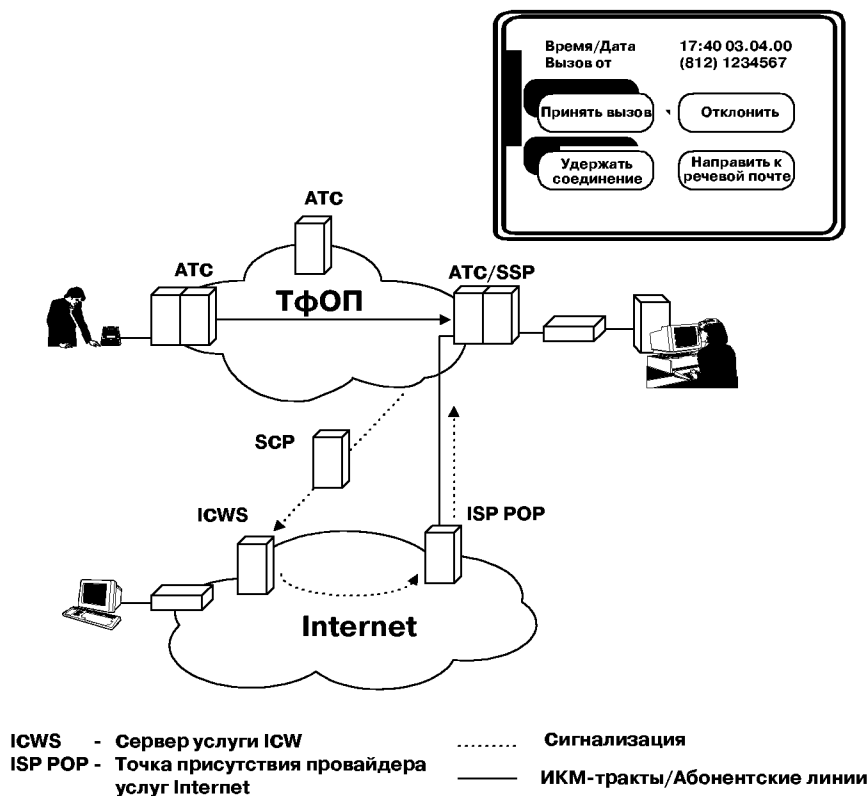


Рис. 5.4.2 Схема предоставления услуги ICW

В составе инженерной проблемной группы Internet – IETF (Internet engineering task force) организована рабочая группа под названием PINT (PSTN and Internet interworking), определившая протокол взаимодействия между телефонной сетью и Internet – PINT Profile. Этот протокол позволяет привлекать IP-сеть к предоставлению телефонных услуг (см. рис. 5.4.3). К таким услугам относятся организация связи между абонентами телефонной сети, а также передача и прием содержания факсов по телефону. При этом Internet используется для управляющих воздействий, а передачу речи (или данных) целиком берет на себя телефонная сеть. Работы группы PINT не затрагивают проблем передачи речи через границу между телефонными сетями и IP-сетями.

Таким образом, при объединении возможностей телефонных сетей и IP-сетей возможно предоставление двух новых классов услуг: передача речи поверх протокола IP (VoIP) и управление соединени-

ем в телефонной сети из IP-сети. Аспекты взаимодействия услуг/приложений IP и услуг IN являются предметом стандартизации в рамках набора CS-4. Прежде чем продолжить обсуждение архитектурных особенностей реализации принципов конвергенции услуг, целесообразно отойти немного в сторону и рассмотреть вкратце архитектурные принципы организации IP-сетей.

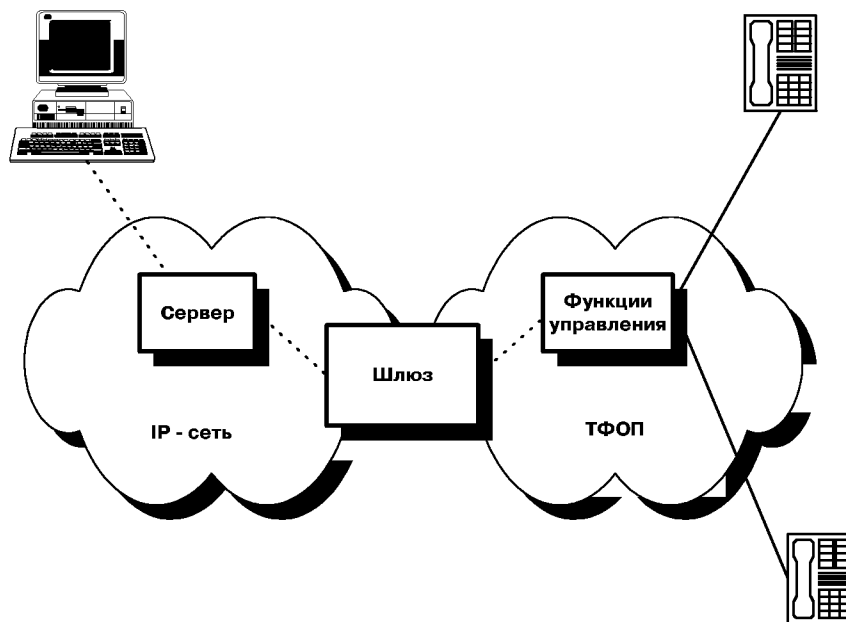


Рис. 5.4.3 Общая конфигурация для PINT-услуг

5.4.3 Сигнализация поддержки услуг мультимедиа в сетях IP

Для обеспечения функций сигнализации в IP-телефонии существуют два основных стандарта: семейство протоколов H.323, разработанное ITU-T, и протокол SIP (Session initiation protocol), разработанный IETF.

Протоколы семейства H.323 и SIP представляют различные подходы к решению одних и тех же задач. Если первые близки к традиционным системам сигнализации, то второй реализует более простой, «интернетовский» подход на основе протокола HTTP (Hyper text transfer protocol). В рекомендации H.323 описано большое количество различных вариантов прохождения соединений, что с большой вероятностью гарантирует совместимость разных систем. Протокол SIP с сообщениями, базирующимися на текстовом формате, проще в реализации и с точки зрения добавления новых функций. Систему

объектов H.323 можно рассматривать как прикладную сеть, наложенную на сеть передачи данных (IP-сеть), тогда как услуги SIP ориентированы на интеграцию с услугами Internet. Технология H.323 предоставляет больше возможностей управления конкретной услугой связи, как в части аутентификации, так и в части контроля использования сетевых ресурсов.

5.4.3.1 Технология H.323

Рекомендация ITU-T H.323 определяет основы передачи данных в сетях с коммутацией пакетов, протоколы и процедуры мультимедийной связи, в том числе, и в IP-сетях. H.323 является «зонтичной» рекомендацией, в ней описаны компоненты сети и даны комментарии к применению других рекомендаций, в частности H.225 и H.245, а также протоколов, разработанных IETF. Рекомендация H.323 определяет четыре компонента системы: *привратник* (GK – Gatekeeper), *шлюз* (GW – Gateway), *блок управления конференциями* (MCU – Multipoint Control Unit) и *H.323-терминал*.

Привратник (GK) служит для остальных компонентов управляющим устройством. В его функции входит преобразование адресов плана нумерации телефонных сетей по рекомендации ITU-T E.164 в IP-адреса, управление доступом к сети и обеспечение защиты, управление полосой пропускания, управление связью и маршрутизация.

Шлюз (GW) обеспечивает взаимодействие с другими сетями, в первую очередь, с телефонной сетью, преобразуя способы передачи сигналов, используемые в сетях коммутации каналов, в транспортный протокол RTP (Real time transport protocol) для передачи информации по IP-сети, а также выполняет часть функций управления полосой пропускания.

Блок управления конференциями (MCU) необходим для организации многосторонней связи трех и более конечных точек, соответствующих требованиям рекомендации H.323, например, персональных компьютеров.

H.323-терминал – это конечная точка сети, ориентированная на двунаправленное соединение в реальном времени с другим H.323-терминалом, шлюзом или блоком управления конференциями. Это соединение служит для передачи между двумя точками информации управления и пользовательской информации различных видов – речи, движущихся изображений и/или данных.

В сетях IP-телефонии между вызывающей и вызываемой сторонами существует информационная связь двух типов: двунаправленный поток пользовательских данных и обмен сигнальными сообщениями для управления соединением и характеристиками потока пользовательских данных.

Процедура организации связи между конечными точками проходит в три этапа: (1) регистрация конечной точки и управление доступом, (2) маршрутизация в сети и установление соединения между конечными точками, (3) согласование параметров передачи информации между конечными точками. После того как соединение установлено, и параметры согласованы, передача мультимедийной информации осуществляется по протоколу RTP.

Поток данных пользователя организован в виде двух отдельных RTP-сеансов, по одному для каждого направления передачи, причем для мультимедийного трафика разных типов используются разные логические каналы RTP. Рекомендуется использовать разные логические каналы и для потоков однотипной информации, если разные потоки одного типа предъявляют разные требования к качеству обслуживания.

В рекомендации H.225, входящей в семейство H.323, описаны механизмы сигнализации, необходимые для регистрации, контроля доступа и индикации состояния (RAS – Registration, admission control and status), для управления связью (на основе протокола Q.931 DSS1) и передачи трафика. В рекомендации H.245 определены типы сигнальных сообщений, которые передаются между конечными точками, и процедуры согласования параметров.

5.4.3.2 Протокол SIP (IETF)

SIP – это протокол прикладного уровня, который позволяет организовать и провести сеанс многосторонней мультимедийной связи, обеспечивая его создание, модификацию и завершение. Протокол работает по схеме клиент-сервер, в которой клиент запрашивает сервис определенного типа, а сервер обрабатывает его запрос и обеспечивает предоставление этого сервиса. Согласно протоколу SIP, пользовательская система может не только создавать, но и принимать запросы. Следовательно, она должна иметь и клиентскую, и серверную часть.

Обслуживание вызовов осуществляется сервером SIP, который может работать в режиме непосредственного установления связи или в режиме переадресации. В обоих режимах сервер принимает запрос сведений о местоположении нужного пользователя, но если в первом режиме он сам доводит вызов до адресата, то во втором – возвращает адрес конечного пункта запрашивающему клиенту.

Протоколом SIP определены два типа сигнальных сообщений – запрос и ответ, которые имеют текстовый формат и базируются на протоколе HTTP. В запросе указываются процедуры, вызываемые для выполнения требуемых операций, а в ответе – результаты их выполнения. SIP использует адресацию, основанную на унифицированном указателе ресурсов URL, в котором может быть записано имя домена или IP-адрес.

Предназначенный для инициирования сеансов, протокол SIP, кроме определения адреса пользователя и установления соединения с ним, служит также базой для применения других протоколов, реализующих функции защиты, аутентификации, описания характеристик канала мультимедийной связи и начисления платы.

5.4.3.3 Декомпозиция шлюза

5.4.3.3.1 Проект TIPHON (ETSI)

Работа над проектом TIPHON (Telecommunications and internet protocol harmonization over networks) была начата институтом ETSI в апреле 1997 года с целью решения проблем взаимодействия между IP-сетями и сетями коммутации каналов (CSN – Circuit switched network), крупнейшей из которых является мировая телефонная сеть. В рамках проекта TIPHON рассматриваются четыре возможных сценария передачи речи:

- от пользователя IP-сети к абоненту сети CSN с целью предоставить пользователям IP-сетей доступ к речевым услугам сетей с коммутацией каналов, в том числе, к услугам IN;
- от абонента сети CSN к пользователю IP-сети с идентификацией вызываемой стороны адресом в телефонной сети (по рекомендации E. 164) или IP-адресом;
- между абонентами сетей CSN с использованием IP на магистрали;
- между пользователями IP-сетей с использованием CSN на магистрали и с идентификацией вызываемой стороны адресом в телефонной сети (по E. 164) или IP-адресом.

Базируясь на стандартах H.323 ITU-T для IP-сетей, спецификации TIPHON ETSI дополняют их некоторыми обязательными процедурами, а также механизмами взаимодействия с сетями коммутации каналов. Функциональная модель TIPHON состоит из тех же компонентов, что и модель H.323, однако в ней предусмотрено разделение шлюза на три сетевых элемента: *шлюз сигнализации* (SG – Signalling gateway), *транспортный шлюз* (MG – Media gateway) и *контроллер транспортного шлюза* (MGC – Media gateway controller). Схема взаимодействия элементов архитектуры TIPHON с телефонной сетью приведена на рисунке 5.4.4.

Шлюз сигнализации (SG) служит промежуточным звеном обеспечения сигнализации между IP-сетями и сетями CSN. В задачи *транспортного шлюза* (MG) входит преобразование и/или перекодирование передаваемой информации, преобразование адреса, подавление эха, воспроизведение различных сообщений для абонентов, прием и передача сигналов DTMF и т.д. *Контроллер транспортного шлюза*

(MGC) выполняет процедуры сигнализации, которые определены рекомендацией H.323, и преобразует сигнальные сообщения сети с коммутацией каналов в сообщения H.323. Его основная задача – управлять работой транспортного шлюза, т.е. осуществлять управление соединениями, использованием ресурсов, преобразованием протоколов и т.п.

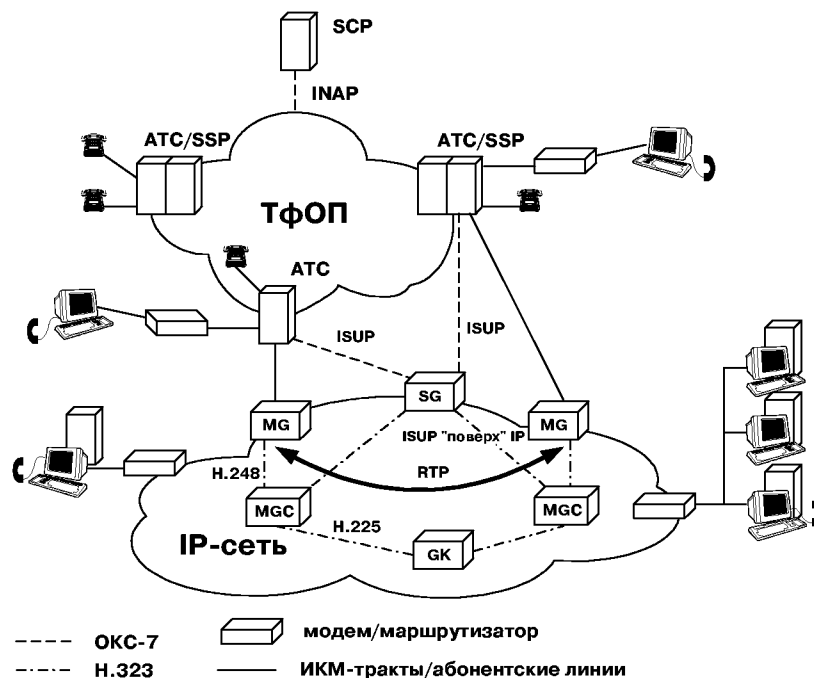


Рис. 5.4.4 Архитектура взаимодействия IN и IP-сетей через распределенные шлюзы

В архитектуре TIPHON привратник (GK), помимо функций, определенных для него в H.323, дополнительно отвечает за начисление платы, взаиморасчеты и составление отчетов об использовании ресурсов.

5.4.3.3.2 Декомпозиция шлюза IETF

Распределенная архитектура телефонного шлюза предлагается и рабочими группами IETF. Согласно подходу IETF, основу модели шлюза также составляют три уже знакомых нам элемента: шлюз сигнализации (SG), контроллер транспортного шлюза (MGC) и собственно транспортный шлюз (MG).

Функции каждого из этих элементов тоже практически не отличаются от тех, что определены рабочими группами TIPHON. При использовании системы сигнализации ОКС-7 в контроллер IP-сети будут

направляться сообщения подсистемы ISUP. При использовании сигнализации по выделенным сигнальным каналам (BCK), сигнальные сообщения сперва, вместе с информацией абонента, поступят в транспортный шлюз, а затем – в контроллер транспортного шлюза. Для инкапсуляции телефонных протоколов сигнализации (ISUP, BCK, PRI) и передачи переносимой ими информации в контроллер транспортного шлюза используется протокол MDTP (Multi-Network datagram transmission protocol).

Контроллер MGC анализирует сигнальные сообщения и передает управляющую информацию в шлюз MG, используя специальный протокол, позволяющий управлять ресурсами (системой интерактивного речевого отклика, мостами конференцсвязи и т. д.). Этот же протокол обеспечивает управление приемом и формированием сигналов DTMF и акустических сигналов, подавлением эха, использованием различных кодеков (G.711, G.723.1, G.729, GSM и т. д.), сбором статистики, тестированием конечных точек (например, испытаниями по шлейфу), резервированием, отключением и блокировкой конечных точек и шифрованием.

5.4.3.3.3 Протокол MGCP

MGCP (Media gateway control protocol) представляет собой довольно простой протокол типа «клиент-сервер», предложенный рабочей группой IETF MEGACO (Media gateway control) для управления транспортным шлюзом. MGCP дополняет протоколы H.323 или SIP, поддерживающие сигнализацию, функциями управления соединениями. Впервые протокол управления транспортным шлюзом под названием SGCP (Simple gateway control protocol) был разработан компанией Telcordia (бывшей BellCore). Впоследствии компанией Level 3 был предложен протокол IPDC (IP device control protocol). Оба они впоследствии были объединены в протокол MGCP.

Согласно модели MEGACO логика управления соединениями в MG контролируется централизованно специальным элементом – агентом запрашиваемой связи (CA – Call agent) – аналогом MGC в архитектуре TIPHON, находящимся вне транспортного шлюза. Сам же шлюз MG представляется в виде объекта, состоящего из (1) конечных точек – точек входа/выхода информационных потоков и (2) соединений – двух или более связанных между собой конечных точек. Модель определяет физические конечные точки (например, окончания соединительных линий) и виртуальные конечные точки (скажем, аудиоисточники). Протокол MGCP использует принцип «ведущий/ведомый», согласно которому агент CA передает транспортному шлюзу команды для управления конечными точками и соединениями.

В ITU-T обсуждается протокол под рабочим названием H.248, предназначенный для управления транспортным шлюзом. Протокол

MGCP является основным (но не единственным) кандидатом на то, чтобы составить основу для H.248.

5.4.4 Функциональная архитектура поддержки IP-сетей в IN

В то время как в проекте ETSI TIPPHON рассматриваются проблемы взаимодействия с сетями коммутации каналов с точки зрения IP-сетей, ITU-T сфокусировал свои исследования на проблемах взаимодействия IP-сетей и IN. Первый вариант функциональной архитектуры интегрированной IN/IP-сети был предложен в 1998 году. Начиная с этого варианта, ITU-T проводит исследование двух основных групп аспектов, связанных с взаимодействием IN и IP сетей, – передача операций протокола INAP поверх протокола IP и атрибуты услуг IN/IP для предположительного включения в наборы CS-3 и CS-4.

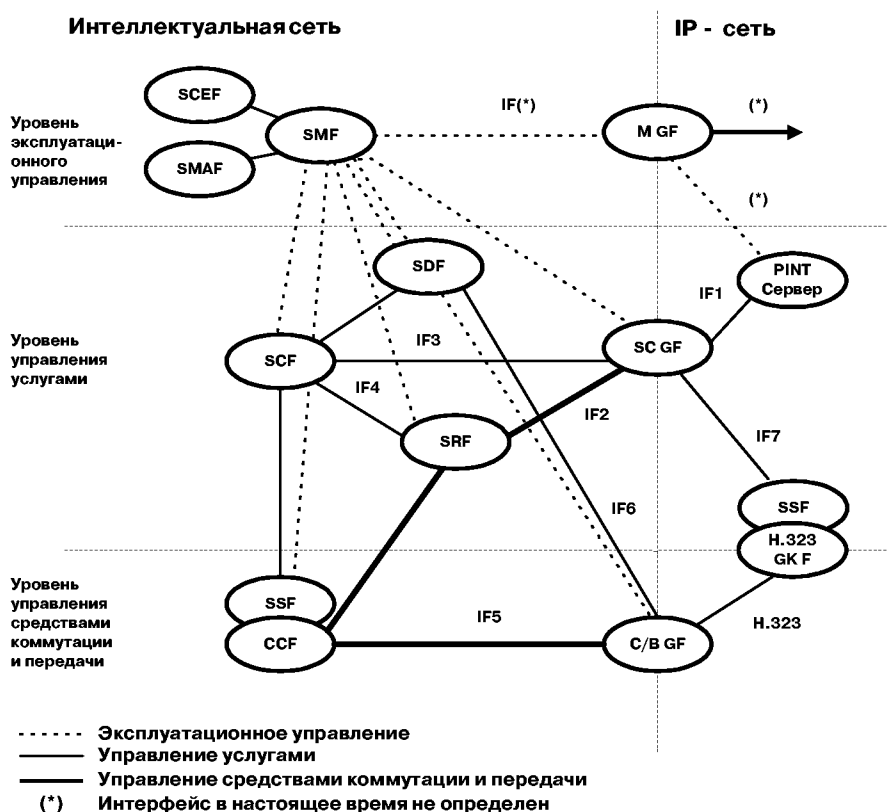


Рис. 5.4.5 Функциональная архитектура Интеллектуальной сети для поддержки услуг, предоставляемых IP-сетями

На рисунке 5.4.5 представлена расширенная функциональная архитектура поддержки услуг, предоставляемых совместно телефонными и IP-сетями по правилам концепции IN. Как видно, модель представляет собой расширение функциональной модели CS-2 IN и содержит новые функциональные объекты, обеспечивающие взаимодействие с IP-сетями. Кроме того, на рисунке показаны функциональные объекты самой IP-сети, вовлеченные в процесс коммутации и управления услугами. Архитектура должна обеспечивать как передачу речи на базе технологии VoIP, так и установление соединений в телефонной сети по запросу из IP-сети.

5.4.4.1 Новые объекты распределенной функциональной плоскости

Функциональный объект *PINT-сервер* предназначен для приема и обработки запросов организации связи в телефонной сети, поступающих от PINT-клиентов, которые являются пользователями IP-сети, и для передачи сведений о результате выполнения таких запросов. PINT-сервер направляет запросы к SCF, передавая их через функциональный объект – *шлюз управления услугами* (SC GF – Service control gateway function).

Функциональный объект *привратник* (GK F – Gatekeeper function) может рассматриваться как логический коммутатор, реализующий функции управления связью (CCF) в IP-сети. Сигнализация управления связью (H.225) и сигнализация управления соединением (H.245) в случае передачи речи способом VoIP направляется к GK F, который решает задачи маршрутизации, при необходимости обращаясь к помощи SCF, например, в случае вызова, требующего услуги FPH. Для этого GK, кроме функций GK F, должен содержать функции SSF.

Функциональный объект *шлюз к SCF* (SC GF – Service control gateway function) обеспечивает взаимодействие между уровнями управления услугами в интеллектуальных и в IP-сетях. SC GF позволяет «скрыть» от элементов IP-сети функциональные объекты SCF и SRF, выступая в роли промежуточного согласующего устройства. SC GF принимает запросы выполнения услуг от PINT-сервера в домене IP-сети и доставляет их в SCF. Запрос содержит информацию, которая требуется SCF для управления услугой, идентификации пользователей и аутентификации данных, а также для предотвращения некорректного использования IP-сетью ресурсов IN. При запросе услуги IN (бесплатной связи, виртуальной частной сети и т.п.) в случае, когда один из участников связи расположен в IP-сети, т.е. является H.323-терминалом, SC GF выполняет функции сопряжения, обеспечивая преобразование протоколов нижнего уровня и адресной информации, а также взаимодействие с несколькими GK F.

Функциональный объект *шлюз управления связью и параметрами доставки информации* (C/B GF – Call/bearer gateway function) вы-

полняет функции доступа к IP-сети через телефонную сеть, например, установление связи с Internet с помощью модемного соединения (режим dial-up), и функции взаимодействия VoIP и телефонной сети. С/В GF преобразует формат, используемый для доставки информации в сети одного типа, в формат, используемый в сети другого типа.

Функции *шлюза эксплуатационного управления* (MGF – Management gateway function) пока не определены.

5.4.4.2 Интерфейсы функциональной модели

Интерфейс IF3 используется для передачи от GK F к SCF запросов активизации логики услуг и предоставляет SCF возможность передавать к GK F команды управления шлюзом с целью сбора сведений, необходимых для исполнения услуги (информации об идентификации, аутентификации и начислении платы). В направлении к SC GF через интерфейс IF3 от SCF передаются запросы предоставления услуг в IP-сети. Например, со стороны SCF через IF3 к пользователю, во время сеанса его связи с Internet, передается уведомление о входящем телефонном вызове. Для переноса через IF3 информационных потоков используется протокол INAP.

Интерфейс IF4 предусматривает расширение возможностей связи между SCF и SRF. Он используется для передачи со стороны SCF к SRF запросов выборки указанных данных из SC GF. Кроме того, через IF4 со стороны SCF к SRF передается команда преобразовать выбранные данные в формат, требующийся для их передачи по телефонной сети. Интерфейс IF5 между CCF и C/BG F нужен для предоставления услуг, базирующихся на VoIP. Работы над его спецификацией ведутся в ITU-T, ETSI и IETF; ожидается, что он не будет обладать особенностями, специфическими для IN. Через интерфейс IF6 к SDF доставляется информация о состоянии C/BGF с целью выбора шлюза средствами IN. Интерфейс IF7 необходим для передачи от H.323-GK F/SSF к SC GF запросов активизации управления услугой со стороны H.323-терминала.

5.4.5 Услуга «CLICK TO DIAL»

Рисунок 5.4.6 иллюстрирует алгоритм предоставления (и связанный с этим информационный обмен) услуги «Вызов щелчком мыши» «CLICK-TO-DIAL» (Вызов щелчком мыши) для установления связи между двумя абонентами телефонной сети по запросу от персонального компьютера пользователя, находящегося в IP-сети. Во время сеанса связи с Internet пользователь запрашивает услугу CTD с указанием телефонных номеров вызывающего (#A) и вызываемого (#B) абонентов. PINT-сервер передает в шлюз управления услугами

(SC GF) запрос активизировать услугу CTD. SC GF передает запрос активизации услуги CTD к SCF. SCF отправляет в SSF/CCF запрос инициировать вызов к абоненту А (операция INAP *InitiateCallAttempt*), а также запрос отчета SSF/CCF об ответе абонента А (операция INAP *RequestReportBCSMEvent*).

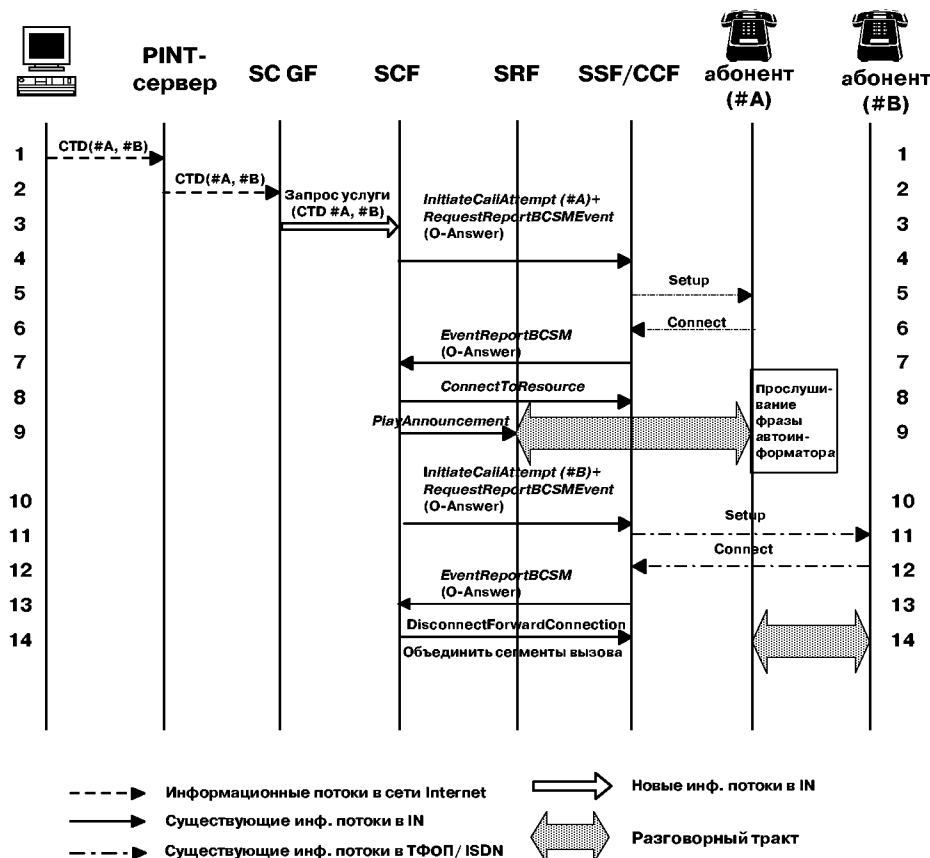


Рис. 5.4.6 Информационный обмен при предоставлении услуги CLICK TO DIAL

Устанавливается соединение между SSF/CCF и абонентом А (сообщения DSS1 Setup и Connect). SSF/CCF сообщает SCF, что абонент А ответил (операция INAP *EventReportBCSM*). SCF отправляет в SSF/CCF запрос установить соединение между абонентом А и SRF (операция INAP *ConnectToResource*). SCF дает команду SRF передать речевое сообщение абоненту А (операция INAP *PlayAnnouncement*).

SCF передает в SSF/CCF запрос инициировать вызов абонента В (операция INAP *InitiateCallAttempt*), а также запрос отчета SSF/CCF об ответе абонента В (операция INAP *RequestReportBCSMEvent*). Устанавливается соединение между SSF/CCF и абонентом В (сообщение DSS1 Setup и Connect).

ния DSS1 Setup и Connect). SSF/CCF сообщает SCF, что абонент В ответил (операция *INAP EventReportBCSM*).

SCF передает в SSF/CCF запрос нарушить соединение между абонентом А и SRF (операция *INAP DisconnectForwardConnection*) и объединить сегменты связи (абонент А – абонент В).

5.4.6 Перспективы интегрированных IN/IP-платформ

Исследование основных направлений конвергенции сетей IN и Internet и доступных для этого технологий проводится в рамках начатого в 1999 году проекта EURESCOM P909 «Enabling technologies for IN evolution and IN-Internet integration». Проект должен дать решение следующих задач:

- идентификация новых классов услуг;
- определение архитектуры для поддержки новых классов услуг (наиболее приемлемая кандидатура – адаптированная к существующим продуктам архитектура TINA – будет подробно рассматриваться в главе 5.6);
- тестирование оборудования на предмет готовности реализации требований новой архитектуры как со стороны IN, так и со стороны Internet;
- интеграция существующего оборудования или прототипов для создания в соответствии с определенной архитектурой платформ поддержки новых классов услуг; анализ необходимости использовать для доступа к SCP шлюзы на базе CORBA (см. главу 5.6) и возможностей интеграции технологий управления, применяемых в шлюзах разного назначения – VoIP GW, PINT GW, GK GW и MCU GW;
- реализация услуг на базе интегрированных IN/Internet-платформ и сравнение с реализацией их на базе существующего оборудования.

Основной предпосылкой данного проекта стало то, что, в отличие от компьютерных систем, платформы IN были и остаются закрытыми. До сих пор нет иного способа взаимодействия с ними, кроме как посредством протокола INAP системы ОКС-7. Протоколы эксплуатационного управления и создания услуг остаются не стандартизированными (и различными в платформах разных поставщиков). В совокупности с другими факторами эти причины обусловили монополию небольшого количества крупнейших компаний – поставщиков телекоммуникационного оборудования и, как следствие, весьма высокие цены платформ IN.

Новый стандарт открытого интерфейса для доступа к телекоммуникационным системам, каковыми являются платформы IN, дол-

жен дать сотням новых малых и средних компаний потенциальную возможность разрабатывать и оперативно поставлять на рынок приложения, доступные в инфраструктуре сетей общего пользования. Открытые сетевые интерфейсы способствуют созданию интегрированных компьютерно-телекоммуникационных приложений, появление которых должно улучшить использование ресурсов платформ IN за счет подключения к ним многомиллионной армии обладателей персональных компьютеров и увеличить потребность в новых услугах.

От оператора сети связи требуется, чтобы он мог предоставить доступ к ресурсам своей сети третьей стороне, открыть для нее возможность управления ими (например, управления соединениями в коммутационной системе или управления интеллектуальной периферией). Однако при этом на первый план выдвигаются проблемы, связанные с аутентификацией пользователей и с защитой ресурсов. Существующий протокол INAP должен быть защищен от внешнего несанкционированного доступа, например, путем использования шлюзов IN/Internet, обладающих специальными средствами (firewall). Должна сохраняться целостность всей сети, то есть обеспечение предоставления сетью любой услуги не должно зависеть от воздействий (как преднамеренных, так и непреднамеренных), обусловленных доступом в сеть третьей стороны. Должна быть гарантирована защита абонентов сети в части конфиденциальности данных о деталях подписки и оплате услуг.

Как уже упоминалось выше, открытые интерфейсы в сторону Internet опираются на протокол IP, изначально созданный для не ориентированной на соединение передачи данных. Качество обслуживания, поддерживаемое протоколами высшего уровня (TCP – Transmission control protocol или UDP – User datagram protocol), приемлемо, в основном, для данных, которые нечувствительны к задержкам пакетов. Рост потребности в услугах мультимедиа, требующих передачи информации в реальном времени, привел к новой концепции сети *Internet интегрального обслуживания* (по аналогии с ISDN), в которой должны обеспечиваться более жесткие нормы качества обслуживания, такие как гарантированная длительность задержки и сквозная синхронизация. Протокол RSVP (Resource reservation protocol) предусматривает средства для передачи пользовательским приложением запроса обеспечить определенный класс обслуживания (один из трех) к маршрутизатору, который резервирует для потока пользовательских данных адекватные сетевые ресурсы.

Второе направление эволюции протокола IP связано с тем, что он был создан для поддержки соединений сегментов вида «точка-точка» с обеспечением возможности связи между несколькими участниками только внутри сегмента. Для расширения этой функции разработан протокол поддержки многоточечной связи (Internet protocol

multicast) и делаются попытки интеграции его и протокола RSVP с системами эксплуатационного управления соединениями в ATM-сетях с тем, чтобы приложения протокола IP могли полностью использовать достоинства этих сетей при установлении широкополосных многоточечных соединений с гарантированным качеством обслуживания.

Проблемы, изложенные в данной главе, составляют одну из областей, наиболее привлекательных как для разработчиков Internet-приложений, так и для разработчиков IN, а это, в свою очередь, дает все основания предполагать, что сформулированные выше задачи будут успешно решены в самом ближайшем будущем.

Глава 5.5

Интеграция IN и B-ISDN

5.5.1 Особенности архитектуры

В дополнение к услугам IN, предоставляемым узкополосными сетями, широкополосные Интеллектуальные сети (B-IN – Broadband IN) должны быть способны предоставлять услуги мультимедиа, комбинируя возможности широкополосных АТМ-сетей с функциональными возможностями сетей, использующих концепцию IN. Адекватная поддержка услуг мультимедиа не может быть получена простым переносом архитектуры IN на широкополосные сети интегрального обслуживания (B-ISDN) – требуется дополнить архитектуру новыми элементами и отражающими их работу моделями.

Услуги мультимедиа, кроме высоких скоростей передачи, нужных для широкополосной связи, требуют, чтобы имелась возможность свободно задавать параметры качества обслуживания и поддерживать сложные конфигурации связи, например, многоточечные и многосторонние. Кроме того, необходимо поддерживать сложные алгоритмы взаимодействия между сетью и пользователем при выборе им услуги того или иного поставщика и параметров этой услуги.

С точки зрения концепции IN архитектура должна, как и прежде, поддерживать оперативность развертывания и модификации услуг без изменения инфраструктуры сети, гибкость системы начисления платы и доступность базовых атрибутов, таких, например, как переадресация. Некоторые из перечисленных требований (высокие скорости передачи, возможность задавать требуемые параметры качества обслуживания) выполняются при комбинировании B-ISDN и IN автоматически, другие же (поддержка сложных конфигураций связи и новых алгоритмов взаимодействия с пользователем) требуют введения в архитектуру IN дополнительных функциональных средств.

При предоставлении большинства услуг мультимедиа требуется устанавливать несколько соединений, каждое из которых связано, так или иначе, с обращением к услуге. Так, при предоставлении услуги совместной работы на нескольких компьютерах (CSCW – Computer-supported co-operative work) может быть соединено между собой более двух участников связи, причем для такой «конференции» потребуется иметь одновременно несколько соединений, поддерживающих обмен информацией разных видов (текстовой, графической, видео и т.д.). Архитектура сети должна быть такой,

чтобы освободить конечного пользователя от необходимости управления конфигурацией связи как при начальном обращении к услуге, так и во время пользования ею. Кроме того, плата должна начисляться сетью не за использование отдельных соединений, а за использование всех сетевых ресурсов, обеспечивающих предоставление услуги.

Совокупность таких ресурсов (участников связи, соединений, создаваемых для переноса информации пользователей, и соединений, не связанных с доставкой этой информации, а необходимых, например, для сигнализации) отображается в архитектуре широкополосной IN абстрактным понятием *сессия*. На рисунке 5.5.1 показано концептуальное различие между архитектурами узкополосной и широкополосной Интеллектуальных сетей. Применительно к узкополосной IN состояния CCF описывают модель базового процесса обслуживания вызова (BCSM – Basic call state model), которая отражает «видение» этого процесса логикой услуги (SL – Service logic). Протокол прикладного уровня – INAP обеспечивает обмен информацией между CCF/SSF и SCF в терминах изменения состояния BCSM. Основное допущение в архитектуре узкополосной IN состоит в том, что реализация услуги всегда соотносится с одним вызовом (с одним запросом связи).

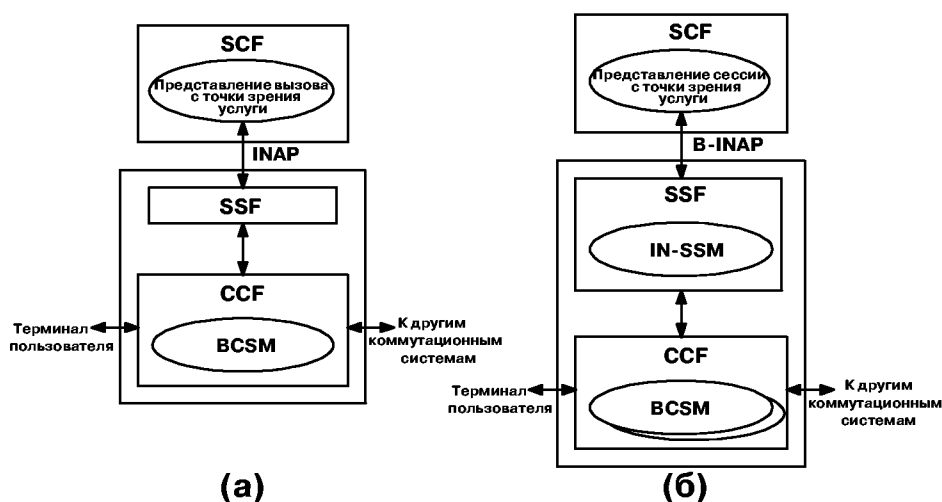


Рис. 5.5.1 Управление услугами в узкополосных (а) и в широкополосных (б) сетях IN

Ниже рассматривается подход к предоставлению широкополосных мультимедиа-услуг IN в рамках международного проекта INSIGNIA (1995-1998 гг.), входящего в паневропейскую программу ACTS, которая ставит своей целью разработку и практическое внедрение новейших телекоммуникационных систем и услуг. Проект INSIGNIA (IN and B-ISDN signalling integration on ATM platform), реализуемый

14 участниками из восьми европейских стран, основан на прагматическом эволюционном подходе к обеспечению поддержки услуг мультимедиа и связан с попыткой объединить последние разработки в области сигнализации B-ISDN и технологию IN. Для реализации проекта был образован одноименный консорциум, в который вошли научно-исследовательские институты, операторы сетей связи и производители оборудования. Полученные результаты представлены ITU-T и рассматриваются в качестве основы для архитектуры CS-4, которая должна будет поддерживать услуги B-ISDN.

Названия функциональных объектов и физических элементов широкополосной IN получают соответствующую приставку (B-, Broadband). В архитектуре, предложенной проектом для широкополосных IN, функции SSF обеспечивают абстрактное представление информации о нескольких запросах связи, относящихся к одной услуге. Архитектура по-прежнему использует модель BCSM, но одновременно вводится более общий уровень абстракции – понятие *сессия*. Для того, чтобы сохранить преемственность терминологии узкополосной IN, в отношении сессии используется термин *модель коммутации услуг IN-SSM* (IN switching state model). Прикладной протокол обмена информацией между B-CCF/B-SSF и B-SCF теперь формулируется в терминах изменения состояний этой более абстрактной модели. Существующий протокол INAP может использоваться в качестве внутреннего протокола B-SSP между функциями B-CCF и B-SSF, а также для связи с существующими SCP (или SSP) узкополосных сетей.

Конфигурация физических элементов B-IN, представлена на рисунке 5.5.2. Терминал пользователя TE (terminal equipment) – это персональный компьютер или рабочая станция, оснащенная интерфейсной платой ATM (STB – Set top box). B-SSP – это коммутатор ATM широкополосной сети общего пользования, содержащий функции B-CCF, B-SSF и B-CUSF (Broadband call unrelated service functions).

Предназначенный для управления широкополосными услугами B-SCP содержит функции B-SCF и B-SDF. Содержащая функции B-SRF интеллектуальная периферия B-IP находится в собственности оператора или поставщика услуги и поддерживает взаимодействие пользователя с элементами Интеллектуальной сети, необходимое на этапе доступа к услуге. Доступ, например, к услуге «видео по требованию», предполагает сложный диалог с сетью, в котором пользователь, используя возможности своего компьютера, просматривает меню, файлы с видеоклипами и выполняет другие действия.

Основным расширением B-SSF, связанным с введением модели сессии, является возможность раздельного управления со стороны одного объекта несколькими соединениями, как непосредственно относящимися к услуге, так и находящимися вне ее контекста. Ме-

менеджер коммутации услуг IN SM (IN switching manager) хранит информацию о состоянии услуги и координирует несколько соединений, относящихся к одной услуге. IN-SSM позволяет моделировать каждое соединение в виде группы объектов. IN SM интерпретирует события, происходящие в BCSM, и транслирует их в события IN-SSM (изменения состояний SSM) для передачи сведений о них к B-SCF. В противоположном направлении приходящие от B-SCF инструкции транслируются в инструкции BCSM. Менеджер коммутации услуг может потребовать от B-CCF инициировать один или несколько вызовов на основании полученных от B-SCF инструкций или дать указание B-CUSF установить сигнальное соединение для взаимодействия пользователя с услугой.

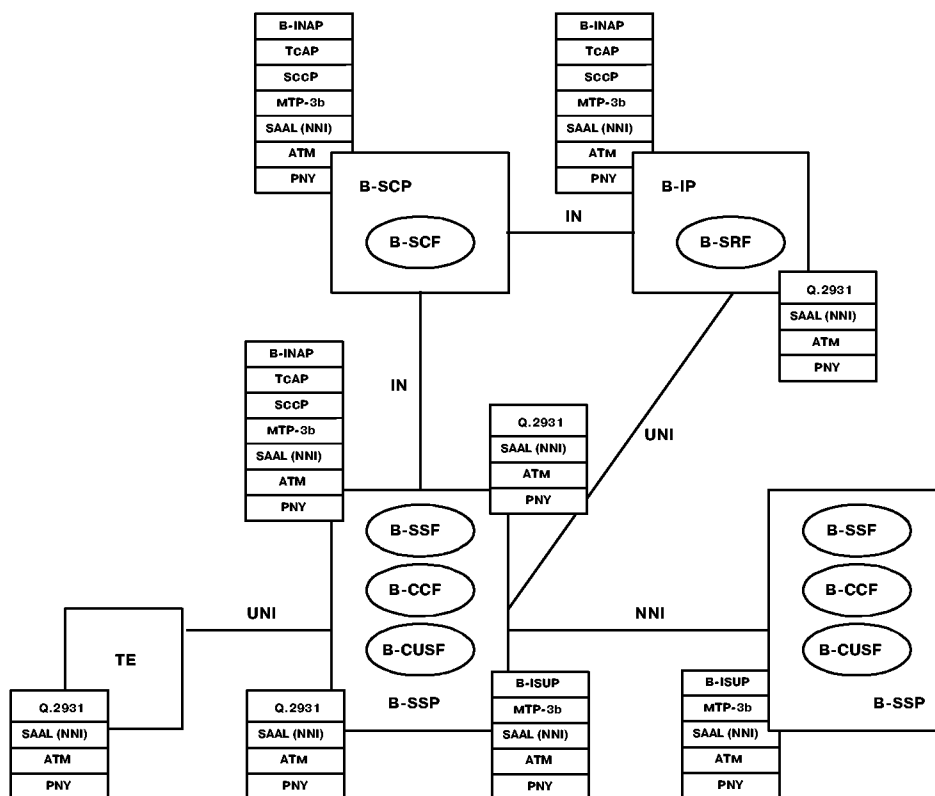


Рис. 5.5.2 Конфигурация физических элементов B-IN

Главное различие между SCF и B-SCF – в особенностях логики управления широкополосными услугами (например, в наличии средств управления сессией), адекватно поддерживаемых протоколом B-INAP (Broadband INAP) в интерфейсе между B-SSF и B-SCF. Предложенный для узкополосных услуг способ взаимодействия между SLP и SSF оставлен неизменным.

Новые группы ресурсов введены в B-SRF. Причина их введения состоит в том, что взаимодействие с пользователем, основанное на графическом интерфейсе, требует большей полосы пропускания для загрузки программных компонентов в терминал пользователя и частотного их обновления. Для хранения программных компонентов широкополосных услуг в B-SRF используются базы данных значительного объема.

Интерфейс между TE (или STB) и B-SSP, как и интерфейс между B-SSP и B-IP, представляет собой стандартный интерфейс пользователь-сеть UNI (User-network interface), интерфейс между двумя B-SSP – межузловой интерфейс NNI (Network-node interface); оба они определены рекомендациями ITU-T по сигнализации в широкополосных сетях. Интерфейсы между B-SCP и B-SSP и между B-SCP и B-IP являются интерфейсами IN и находятся в процессе разработки в ITU-T.

5.5.2 Модель состояний IN-SSM

Модель IN-SSM дает объектно-ориентированное описание машины конечных состояний процесса обслуживания вызова (управления соединением) в терминах состояний связи/соединения IN. Вследствие довольно простой топологии соединений, связанных с услугами первого набора возможностей, в CS-1 модель IN-SSM ориентирована только на процесс управления соединением. В CS-2 модель IN-SSM дополнена аспектами управления участниками многосторонней связи.

Проблема создания адекватной модели IN-SSM применительно к широкополосным сетям состоит в том, что набор возможностей, предоставляемых сетями B-ISDN, не является фиксированным, а постепенно развивается. Следуя этому, идет и создание наборов стандартов сигнализации (SCS – Signalling capabilities set) для поддержки каждого нового набора возможностей B-ISDN. Это означает постоянное обновление модели BCSM. Нельзя забывать и о том, что любая широкополосная услуга IN может соотноситься с несколькими вызовами (запросами связи) и предоставлять несколько соединений.

На рисунке 5.5.3 представлены реализации сессии в B-SSF и B-SCF для случая трех соединений и трех участников связи. Между участниками A и B установлены два соединения (1 и 2) для доставки информации пользователей, а между участниками B и C установлено одно соединение (3), не связанное с доставкой такой информации.

При спецификации услуг B-ISDN применяется принцип, согласно которому на сложность сценария никаких ограничений не накладывается. Реализация таких услуг требует, чтобы процессы управления

связью пользователей были отделены от процессов управления соединениями. Однако трудно ожидать, что переход от сетей ATM, поддерживающих только полупостоянные соединения, к сетям коммутируемых виртуальных каналов, поддерживающим полный спектр услуг B-ISDN, произойдет внезапно.

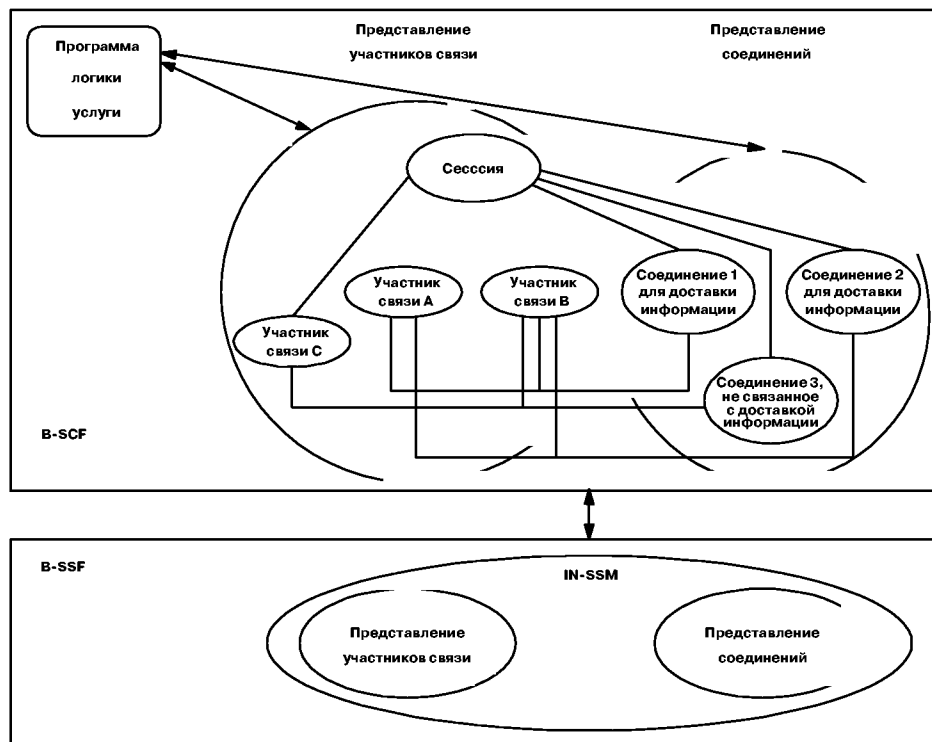


Рис. 5.5.3 Модель сессии в B-SSF и B-SCF

ITU-T определены три фазы введения возможностей сигнализации SCS (Signalling capability set) в B-ISDN. Первая из этих фаз, SCS-1, рекомендации для которой утверждены в 1995 году, ориентирована на поддержку простейших широкополосных услуг связи между двумя пользователями. Вторая фаза разделена на два этапа – SCS-1.1 и SCS-1.2. Одобренный в 1997 году пакет рекомендаций этапа SCS-1.1 поддерживает связи/соединения между одной точкой и несколькими с возможностью изменения характеристик соединения во время связи. В 2000 году утверждены рекомендации для SCS-1.2, которые предусматривают дополнительно поддержку связей, требующих нескольких соединений. В настоящее время ведется работа над спецификацией протоколов с полным разделением процессов управления связью пользователей и управления соединениями.

Овалы на рисунке 5.5.3 соответствуют реализациям объектов (в терминах объектно-ориентированного программирования). Каждый объект содержит внутри себя набор локальных данных и присущие ему функции или операции. Однородные объекты в абстрактном описании образуют класс (например, класс «участник» или класс «соединение»). Графическое представление модели IN-SSM с использованием объектно-ориентированной технологии (OMT – Object modeling technique) – так называемая диаграмма классов – приведено на рисунке 5.5.4. Диаграмма классов содержит классы объектов IN-SSM, их атрибуты (не путать с атрибутами услуг!) и взаимные связи, называемые в OMT ассоциациями.

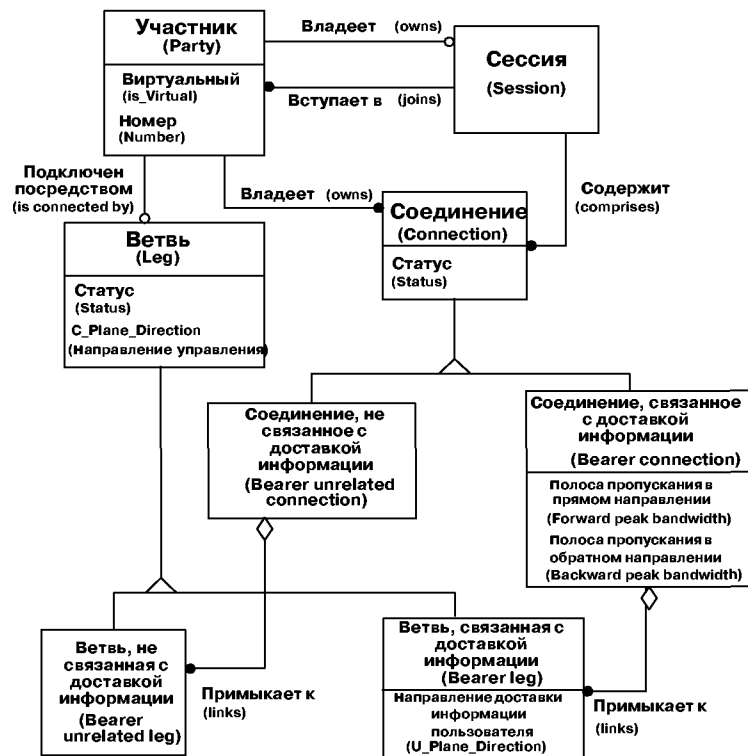


Рис. 5.5.4 Диаграмма связи классов для IN-SSM

«Сессия» (Session) является представлением конфигурации связи с точки зрения функциональных объектов IN. В сессию может быть введен один «участник» (Party) или несколько участников. В роли участника может выступать конечный пользователь или сетевой элемент, например В-SCP. В последнем случае участник характеризуется тем, что атрибут «виртуальный» (Is_Virtual) имеет значение True (Верно). Это позволяет моделировать действия, инициируемые SCP, например, установление или нарушение соединения. Один из участников

сессии всегда является ее «*владельцем*» (owner). Во время пользования услугой к сессии могут динамически добавляться новые участники и удаляться прежние.

Между участниками сессии устанавливаются *соединения*. Объект класса «соединение» может быть двух видов: «соединение для доставки информации пользователей» (bearer connection) и «соединение, не связанное с доставкой информации» (bearer unrelated connection). Если используются возможности сигнализации B-ISDN SCS-1, то существует однозначное соответствие между связью пользователей и соединением для доставки информации. Объект «соединение, не связанное с доставкой информации» служит для моделирования ресурсов, которые поддерживают диалог между пользователем и логикой услуги в B-SCP.

Соединение компонуется из *ветвей*. Термин «ветвь» (leg) обозначает объект, представляющий тракт связи, который ведет к участнику сессии, связанному с другими участниками посредством соединения. Объекты «ветвь для доставки информации» (bearer leg) и «ветвь, не связанная с доставкой информации» (bearer unrelated leg), используются в соединениях соответствующих видов. Оба эти объекта характеризуются атрибутами «статус» и «направление в плоскости управления», а первый из них – также атрибутом «направление в плоскости пользователя».

Комбинация связей между ветвями и соединением определяет его топологию. Если соединение содержит только две ветви, то это соединение «точка – точка». Соединение для доставки информации пользователя, имеющее конфигурацию «точка – группа точек», содержит более двух ветвей соответствующего вида.

Различие между объектами «соединение для доставки информации» и «соединение, не связанное с доставкой», состоит еще и в том, что первый из них может содержать несколько ветвей (в моделях многоточечных конфигураций связи), тогда как второй – только две ветви (каждая для одного направления в конфигурации «точка-точка»). Атрибут «номер» (number) в классе объектов «участник», используется для идентификации оконечных пользователей. Атрибут «статус» соединений для доставки информации может иметь значения:

«Устанавливается»	BEING SETUP
«Активно»	SETUP
«Нарушается»	BEING RELEASED

Атрибут «статус» в классе объектов «ветвь» может иметь значения:

«Ветвь создается»	PENDING
«Маршрут выбран»	DESTINED
«Ветвь создана»	JOINED

«Отказ вызывающего пользователя от связи во время установления соединения»	ABANDONED
«Отказ вызываемой стороны принять вызов»	REFUSED

Атрибут «направление в плоскости управления» нужен, чтобы указать направление сигнальной связи. Он имеет значение «входящее» (INCOMING) для «ветви» вызывающего пользователя, и «исходящее» (OUTGOING) – для «ветви» вызываемого пользователя.

Атрибут, характеризующий направление передачи информации пользователя для объекта «ветвь», может иметь значения «источник», «потребитель» и «двунаправленное». Атрибут «максимальная полоса пропускания в прямом направлении» задает пиковое значение полосы для направления от владельца сессии к другим участникам, а атрибут «максимальная полоса пропускания в обратном направлении» задает аналогичное значение для направления к владельцу сессии.

Инновационный аспект модели состоит в том, что логика услуги в B-SCF не имеет прямого представления о состояниях BCSM в B-CCF (т.е. о триггерных точках DP). Связи между B-SCF и B-SSF организованы на более высоком уровне абстракции в терминах сессии, ее участников, соединений и ветвей.

С точки зрения B-SCF модель сессии влияет на принципы построения программ SLP, которые теперь основаны на представлении топологии всей сессии, а не отдельных соединений. Исполнение логики услуги в B-SCF сводится к созданию и модификации параметров SSM. Действия, выполняемые менеджером коммутации услуг SM над IN-SSM, осуществляются по командам либо со стороны B-SCF, либо со стороны B-CCF или B-SSF, переносимым в информационных элементах протокола B-INAP.

5.5.3 Услуга «Видео-по-требованию»

Поиск в сети по запросу пользователя мультимедийных приложений (IMR – Interactive multimedia retrieval) и их предоставление является одной из комплексных услуг, использующих гибкость управления соединениями B-IN. В качестве примера применения рассмотренных выше подходов приведем краткое описание реализации разновидности IMR – «видео-по-требованию» (VoD – Video on demand).

Услуга VoD предоставляет пользователям возможность выбирать и просматривать видеоинформацию, которая хранится в цифровом виде в сетевом информационном центре одного из поставщиков.

Информация эта передается пользователям по индивидуальным запросам. Пользователь выбирает поставщика услуги, задает время начала демонстрации и управляет просмотром со своего терминала.

Предоставление услуги основано на сквозном использовании АТМ-технологии. Для выбора услуги и управления просмотром служит имеющаяся у пользователя специальная приставка к телевизору или персональный компьютер. Информация хранится на видеосervere в помещении поставщика. Схема предоставления услуги VoD приведена на рисунке 5.5.5.

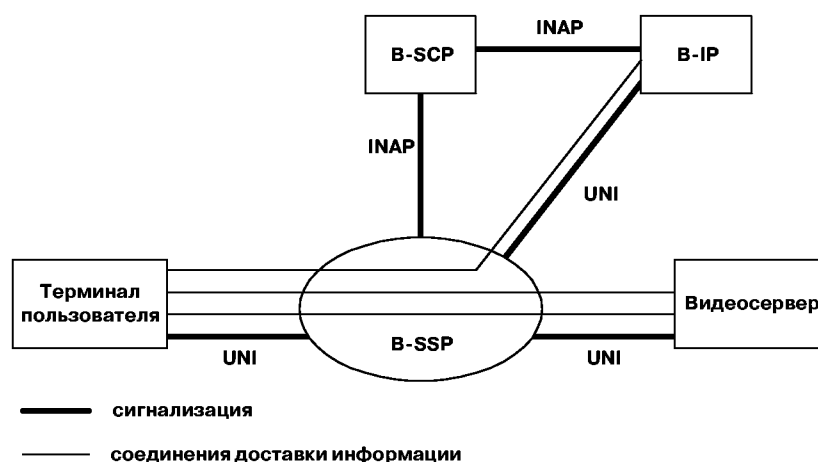


Рис. 5.5.5 Сетевая конфигурация предоставления услуги VoD

С точки зрения сигнализации B-ISDN при реализации услуги использованы возможности согласования и модификации полосы пропускания в интерфейсах UNI и NNI. С точки зрения IN задействованы атрибуты пересчета номерной информации, возможность выбора поставщика услуги, инициирование вызова со стороны B-SCP, организация нескольких соединений в контексте одной сессии и некоторые другие возможности.

Процесс предоставления услуги VoD содержит несколько фаз (см. рис. 5.5.6 и 5.5.7). Сначала пользователь выбирает услугу путем набора соответствующего логического номера и подключается к B-IP для аутентификации и выбора одного из поставщиков. По завершении первой фазы IN отключает пользователя от B-IP и соединяет его с видеосервером выбранного поставщика. Пользователь выбирает нужный видеоматериал, после чего начинается «проигрывание» видеоизображения, во время которого имеется возможность управлять цифровым потоком посредством команд остановки, быстрой прокрутки, повтора и т.д. По окончании просмотра пользователь снова подключается к B-IP, чтобы либо выбрать другого поставщика, либо отказаться от пользования услугой.

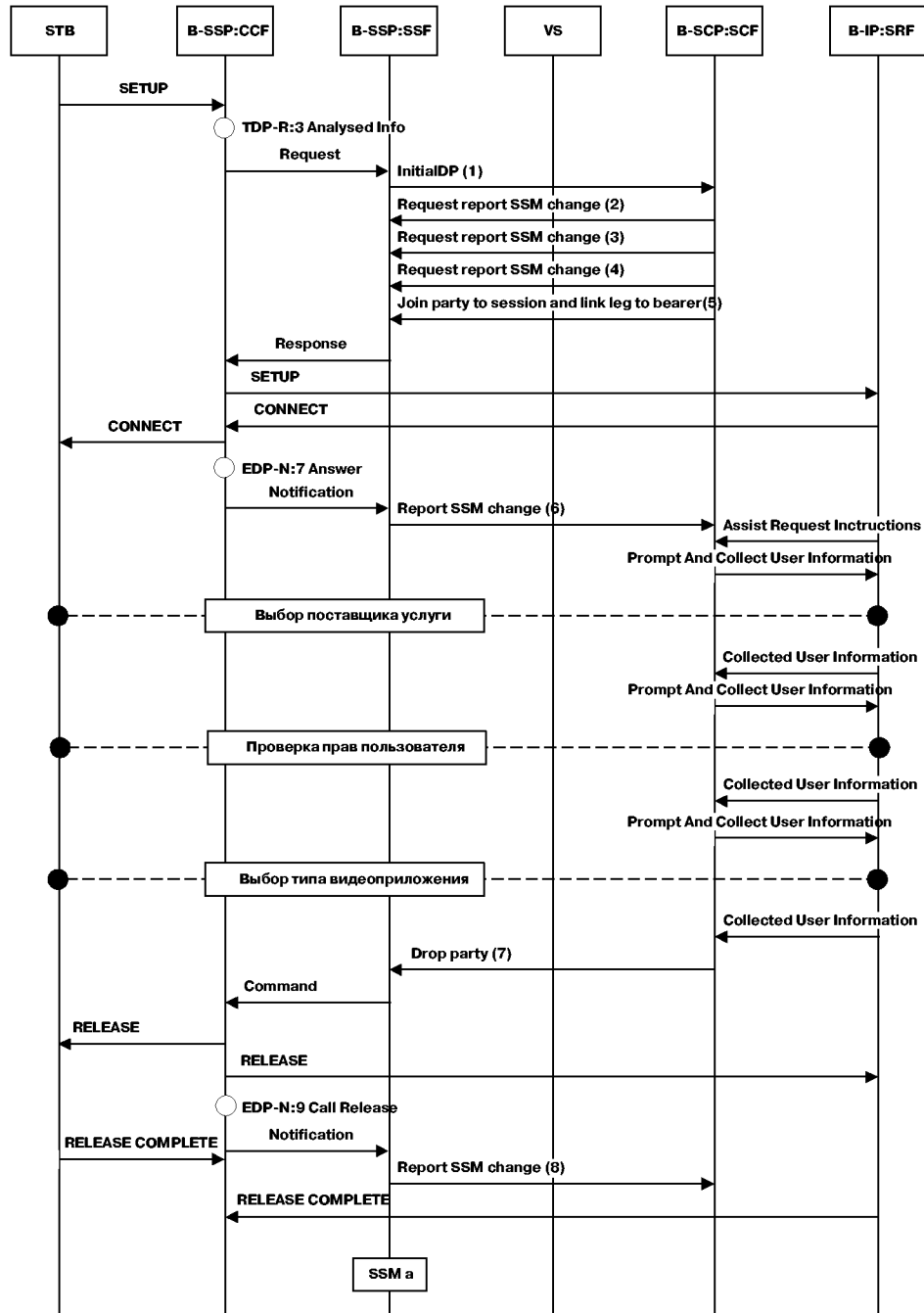


Рис. 5.5.6 Установление и нарушение связи пользователя с В-IP

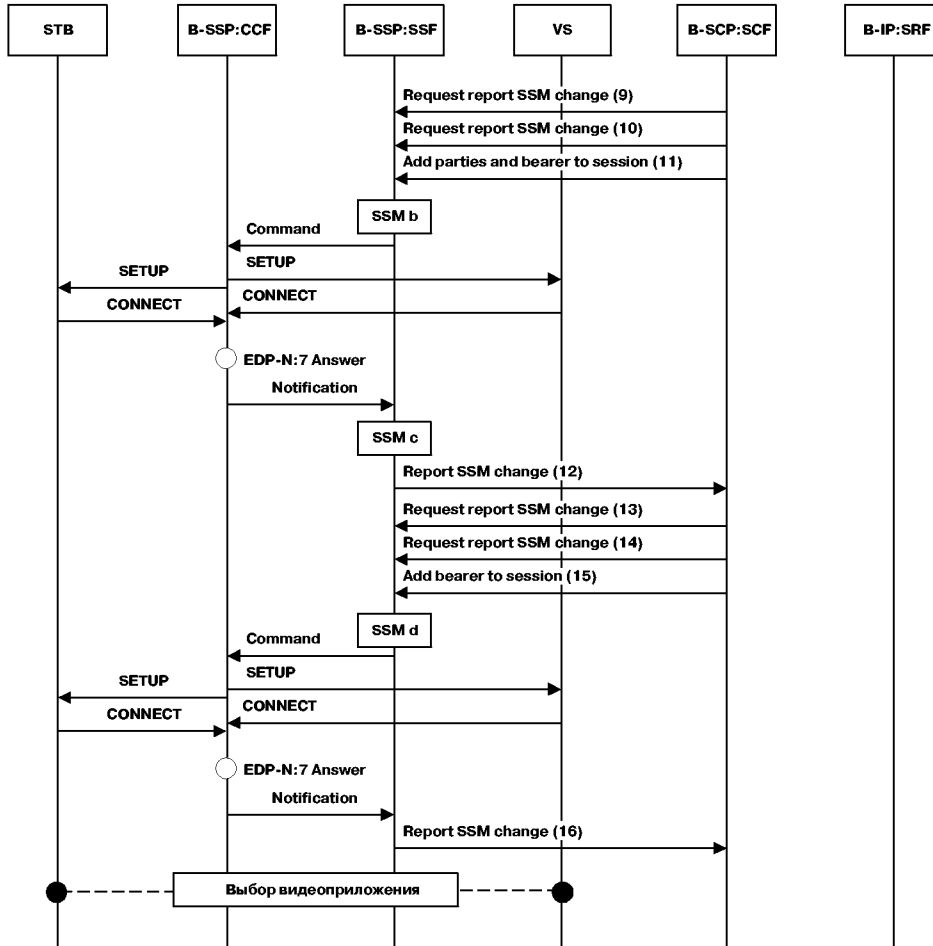


Рис. 5.5.7 Установление соединения пользователя с видеосервером

Через интерфейс UNI от терминала пользователя к B-SSP передается сообщение SETUP сигнализации DSS2 (Digital subscriber signalling system 2), в котором содержится код услуги VoD. Процесс обслуживания вызова приостанавливается в точке TDP-R 3: Analysed Info. B-SSP реагирует на это передачей к B-SCP операции *InitialDP* с запросом организовать сессию услуги VoD. Приняв запрос, B-SCP предписывает B-SSP следить за изменениями состояний объектов SSM, передавая к нему несколько операций *RequestReportSSM-Change* (запрос отчета об изменении состояния SSM), каждая из которых предназначена для отслеживания события одного определенного вида. Затем B-SCP передает B-SSP операцию *JoinPartyToSessionAndLinkLegToBearer* (подключить участника к сессии и примкнуть ветвь к среде доставки информации) с данными, нужными для установления соединения к B-IP.

В-SSP устанавливает соединение с В-IP, передавая сообщение SETUP сигнализации DSS2. Известие о приеме от В-IP сообщения CONNECT (EDP-N 7: Answer notification) передается к В-SCP в операции *ReportSSMChange* (отчет об изменении состояния SSM). Чтобы запросить инструкции о ведении диалога с пользователем, со стороны В-IP в направлении В-SCP передается операция *AssistRequest-Instructions*. Результаты отдельных этапов диалога между пользователем и В-IP передаются к В-SCP (посредством операций *Collected-UserInformation*), который каждый раз передает к В-IP инструкции о дальнейших действиях (в операциях *PromptAndCollectUserInformation*). Назначение операций в интерфейсе между В-SCP и В-IP аналогично рассмотренному в главе 3.1 для INAP CS-1.

После того как В-SCP определит окончание диалога, с его стороны к В-SSP передается операция *DropParty* (отключить участника), предписывающая нарушить соединение пользователя с В-IP; это событие транслируется в сообщение RELEASE (разъединить) сигнализации DSS2. Информация о завершении разъединения (EDP 9: Call Released) передается к В-CSF в операции *ReportSSMChange*.

Затем В-SCP дает В-SSP указание, чтобы тот отслеживал изменения состояний объектов, привлекаемых к обслуживанию дальнейших, иницируемых со стороны В-SCP, вызовов. После приема операции *AddPartiesAndBearerToSession* (добавить к сессии участников и средства доставки информации) между терминалом пользователя и видеосервером создаются нужные соединения, а отчет о соответствующих событиях направляется к В-SCP в операциях *ReportSSMChange*. На основании выбора, сделанного пользователем, видеосервер модифицирует должным образом ширину полосы пропускания, предоставляемой для доставки информации. Аналогичным путем производится отключение пользователя по окончании сеанса.

Рассмотрим более подробно ту часть алгоритма, которая связана с изменениями состояний SSM, когда В-SCP иницирует два вызова для подключения терминала пользователя к видеосерверу. К этому моменту сессия уже организована, а нумерация рассматриваемых далее операций соответствует рисункам 5.5.6 и 5.5.7. Состояние SSM после того, как соединение между В-IP и терминалом пользователя нарушено, приведено на рисунке 5.5.8.

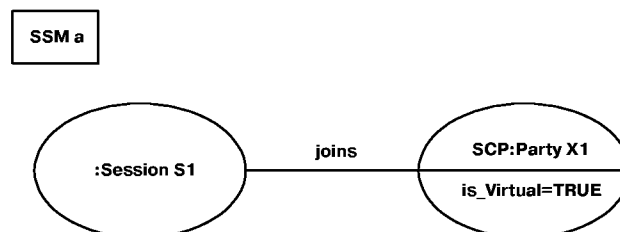


Рис. 5.5.8 Состояния объектов сессии (SSM a)

<i>(9) RequestReportSSMChange</i>	
Идентификатор сессии	S1
Тип объекта	Соединение
Список идентификаторов объектов:	
Идентификатор объекта	Z2
Список состояний объектов:	
Состояние 1:	
Атрибут	Статус соединения
Значение атрибута	Установлено
Режим наблюдения	Уведомить и продолжить
<i>(10) RequestReportSSMChange</i>	
Идентификатор сессии	S1
Тип объекта	Ветвь
Список идентификаторов объектов:	
Идентификатор объекта	Y3, Y4
Список состояний объектов:	
Состояние 1:	
Атрибут	Статус ветви
Значение атрибута	Отказ принять вызов
Режим наблюдения	Уведомить и продолжить
Состояние 2:	
Атрибут	Статус ветви
Значение атрибута	Ветвь создается
Режим наблюдения	Уведомить и продолжить
<i>(11) AddPartiesAndBearerToSession</i>	
Идентификатор сессии	S1
Список пунктов доставки информации:	
Пункт доставки 1:	
Идентификатор участника	X4 (терминал пользователя)
Идентификатор ветви	Y3
Направление в плоскости пользователя	Двунаправленная доставка
Направление в плоскости управления	Исходящее
Номер участника	Номер ТФОП/ISDN для X4
Информация о совместимости в верхних уровнях	Зависит от терминала пользователя
Информация о совместимости в нижних уровнях	Зависит от терминала пользователя
Пункт доставки 2:	
Идентификатор участника	X5 (видеосервер)
Идентификатор ветви	Y4
Направление в плоскости пользователя	Двунаправленная доставка
Направление в плоскости управления	Исходящее
Номер участника	Номер ТФОП/ISDN для X5
Информация о совместимости в верхних уровнях	Зависит от терминала пользователя
Информация о совместимости в нижних уровнях	Зависит от терминала пользователя
Характеристики среды доставки информации:	
Пиковая скорость в прямом направлении	167 ячеек/с
Пиковая скорость в обратном направлении	167 ячеек/с
Класс среды доставки	BCOV-X
Тип уровня адаптации AAL	AAL5
Идентификатор владельца сессии	X5

После обмена операциями 9-11 реализация объекта SSM переходит в состояние, показанное на рисунке 5.5.9.

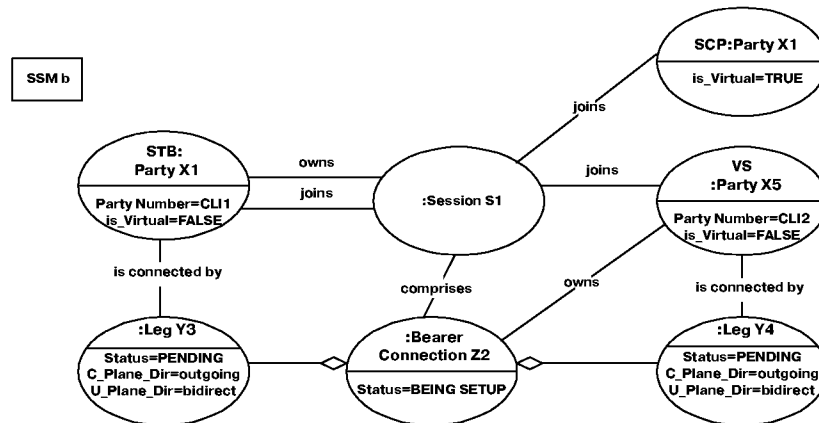


Рис. 5.5.9 Состояния объектов сессии (SSM c)

Рассмотренные в данной главе аспекты конвергенции двух концепций: Интеллектуальная сеть и шировополосная сеть В-ISDN позволяют оптимистично оценивать перспективы IN (а заодно и небесполезность данной книги) в ближайшем будущем.

Глава 5.6

Интеллектуальные сети и TINA

5.6.1 Что такое TINA?

В центре концепции IN лежит принцип отделения функций коммутации от функций управления услугами. Применение этого принципа позволило операторам добиться большей независимости от поставщиков коммутационного оборудования при создании новых услуг в инфраструктуре телефонных сетей, сохранив при этом инвестиции, вложенные туда в течение многих лет. С переходом спецификаций IN в стабильное состояние и с приобретением за последние 3-6 лет положительного опыта практического внедрения услуг IN мировое телекоммуникационное сообщество обратило свое внимание на возможности расширения и развития апробированных в концепции принципов для применения в инфраструктуре широкополосных сетей и сетей IP. Оказалось, что требующуюся для такой инфраструктуры гибкость можно получить, применив дополнительно новый принцип – разделение приложений и используемых ими ресурсов.

Консорциум из 40 компаний – лидеров телекоммуникационного и компьютерного рынка Alcatel, AT&T, Bellcore, British Telecom, Cable and Wireless, CSELT, Deutsche Telecom, Digital Equipment Corp., EricssonNnc, ETRI, Eurescom, France Telecom, Fujitsu, GPT, Hewlett-Packard, Hitachi, IBM, ISIS Distributed Systems, KDD, Korea Telecom, MCI, NEC, Nokia, Northern Telecom, NTT, Oki, PTT Telecom Netherlands, Portugal Telecom, Samsung, Siemens, Stentor, Swiss Telecom PTT, Telecom Italia, TeleDanmark, Telefonica, Telenor, Telia, Telstra, Unisys – образовался в конце 1992 года в связи с проектом новой Архитектуры сетевого информационного обеспечения систем связи (TINA – Telecommunications information networking architecture). К консорциуму в дальнейшем присоединились также компании Broadcom, C-DOT, IONA Technologies, KPN, Softwire, Sprint, SUN Microsystems, Telecom Malaysia.

Актуальность проекта была обусловлена потребностью разработчиков телекоммуникационного оборудования в архитектуре, позволяющей, во-первых, унифицировать средства и способы управления разнородными сетями связи, во-вторых, совместно использовать уже существующие и вновь разрабатываемые средства связи, в-третьих, объединить достоинства традиционных систем коммутации с достоинствами широкополосных сетей и сети Internet.

Такая объединительная задача ставилась с самого начала проекта, когда TINA рассматривалась как главный инструмент конвергенции двух основных телекоммуникационных направлений: IN и TMN. В литературе даже встречалась формула

$$TINA = IN + TMN,$$

первое слагаемое которой наиболее интересно в контексте данной книги. Но попытаемся сначала ответить на вопрос, вынесенный в название параграфа.

5.6.1.1 Основные принципы TINA

TINA – это интегрированная архитектура, применимая к любым типам услуг и сетей, но нацеленная, в основном, на поддержку предоставления широкополосных услуг, услуг мобильности и информационных услуг. Архитектура базируется на четырех принципах:

- 1) Объектно-ориентированный подход, позволяющий представить систему в виде набора объектов и тем самым скрыть ее сложность.
- 2) Распределение программных компонентов по разным частям сети в соответствии как с требованиями пользователя, так и с реальными сетевыми возможностями (характеристики трафика, загрузка сети, надёжность). Этот принцип обеспечивается с помощью среды распределённой обработки DPE.
- 3) Независимость программных компонентов друг от друга, что гарантирует возможность замены любого компонента (например, в связи с изменением алгоритмов его функционирования) без модификации смежных компонентов.
- 4) Разделение полномочий, применяемое в двух плоскостях. Во-первых, TINA предусматривает разделение приложений и среды DPE, в которой эти приложения функционируют. Телекоммуникационные приложения представляются в виде набора взаимодействующих друг с другом объектов, реализующих функциональные возможности системы, а DPE поддерживает взаимодействие этих распределённых приложений. Во-вторых, TINA разделяет всё программное обеспечение на приложения, обеспечивающие услуги, и приложения, обеспечивающие общий контроль.

Принципы TINA нацелены на отделение вполне стабильных функций оперативного и эксплуатационного управления от требующих гибкости и динамичности функций разработки услуг и быстро меняющихся сетевых технологий. Иерархию архитектуры TINA определяют следующие четыре составные части:

- 1) Архитектура услуг (service architecture) определяет концепции и принципы разработки, спецификации и реализации телекоммуникационных услуг и управления ими.
- 2) Сетевая архитектура (network architecture) описывает правила взаимодействия приложений с транспортной сетью.
- 3) Архитектура вычислений (computing architecture) определяет принципы построения распределённого программного обеспечения с позиций простоты взаимодействия, надёжности и возможности перемещения программных компонентов из одной части сети в другую.
- 4) Архитектура эксплуатационного управления (management architecture) определяет принципы построения программных систем для эксплуатационного управления услугами, ресурсами и приложениями.

Построенная на основе этой архитектуры типичная TINA-система может быть разделена несколько уровней. В данной книге выделяются следующие четыре основных уровня:

- 1) Уровень аппаратных ресурсов, к которому относят процессоры, память, коммуникационные и другие периферийные устройства.
- 2) Уровень среды выполнения (NCCE – Native computing and communications environment) включает в себя программные средства, обеспечивающие работу среды DPE и TINA-приложений, – операционную систему, средства обмена данными и другие программные утилиты.
- 3) Уровень DPE обеспечивает технологически независимое представление аппаратных ресурсов, облегчая разработку и свободу перемещения TINA-компонентов. Уровень DPE поддерживает распределённость TINA-приложений, позволяя компонентам приложений динамически размещаться в разных узлах TINA-системы и взаимодействовать друг с другом через интерфейсы DPE.
- 4) Уровень TINA-приложений располагается выше уровня DPE и состоит из двух основных групп компонентов: компонентов услуг и компонентов сетевых ресурсов.

Наличие уровня NCCE определяет возможность одновременной работы в одном и том же узле сети как TINA-приложений, так и иных, не-TINA-приложений, что позволяет обеспечить совместимость и преемственность с уже существующими системами. Необходимый для работы DPE стек протоколов обеспечивается NCCE, в противном случае забота об этом перекладывается на DPE.

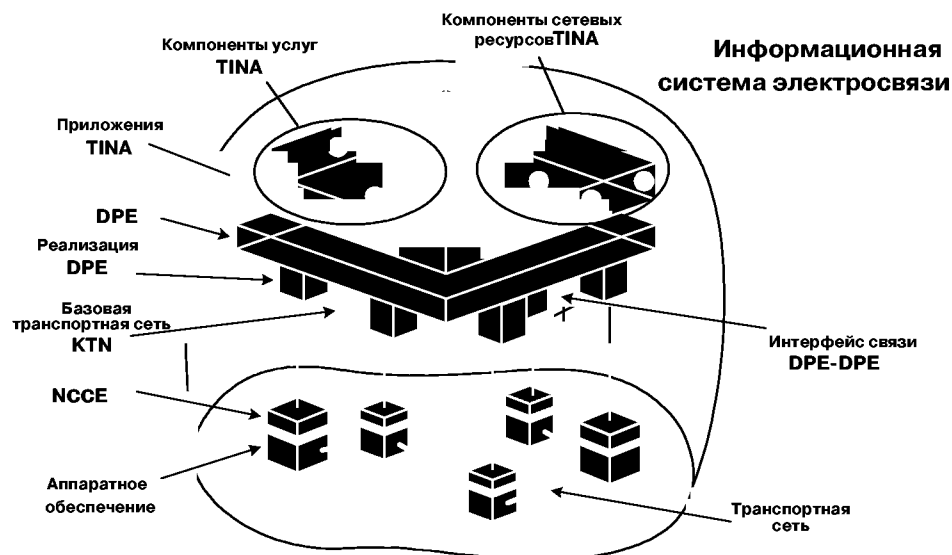


Рис. 5.6.1 Архитектура TINA

5.6.1.2 Архитектура вычислений

Архитектура вычислений базируется на специфицированной в рекомендациях X.901 – X.903 ITU-T Модели открытой распределённой обработки RM-ODP (Reference model for open distributed processing) и определяет концепцию моделирования при проектировании телекоммуникационного программного обеспечения и среды распределённой обработки DPE. Архитектура вычислений TINA уточняет стандарты RM-ODP в отношении разработки телекоммуникационных систем. Предусматривается декомпозиция сложной телекоммуникационной системы на несколько отображений, каждое из которых охватывает некоторое подмножество характеристик системы. Такими отображениями являются: бизнес-отображение; информационное отображение; вычислительное отображение и отображение проектирования. Архитектура TINA определяет модели для этих четырёх отображений и средства абстрактного описания, используемые для разработки телекоммуникационных систем.

БИЗНЕС-МОДЕЛЬ (BUSINESS MODEL) описывает систему с точки зрения участников телекоммуникационного рынка, которые намерены ее использовать. Эта модель оперирует понятиями «Участники» (люди и организации, связанные с TINA-системой) и «Роли участников» (типы активности, которую могут проявлять участники). В частности, введены роли «Потребитель», «Розничный поставщик услуг», «Сторонний поставщик услуг», «Брокер», «Поставщик сетевых ресурсов».

Согласно бизнес-модели вся телекоммуникационная система TINA разделяется на административные области (домены), каждая из которых относится к какому-либо участнику. Роли взаимодействуют между собой через интерфейсы, которые обеспечиваются опорными точками. На рисунке 5.6.2 представлены отношения между ролями и опорными точками.

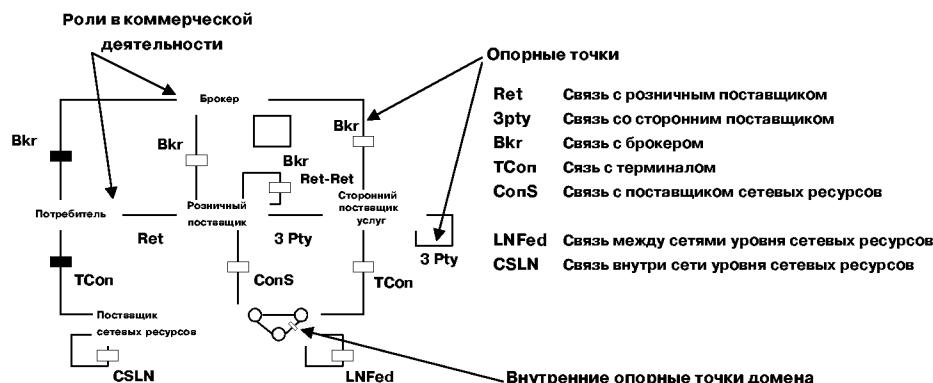


Рис. 5.6.2 Роли и опорные точки бизнес-модели TINA

ИНФОРМАЦИОННАЯ МОДЕЛЬ (INFORMATION MODEL) определяет средства для составления информационных спецификаций, т.е. для описания данных, обмен которыми происходит в системе. Модель включает в себя следующие понятия: информационные объекты, с помощью которых представляется информация, передаваемая в системе; классификацию информационных объектов по типам; отношения между объектами; ограничения и правила, которые определяют поведение объектов, включая правила их создания и удаления.

Информационная спецификация описывает также проблемную область, для которой предназначается система. Разработчик информационной спецификации определяет набор данных, которыми будет оперировать система, безотносительно к тому, как это будет реализовано.

Для записи информационных спецификаций используется язык GDMO (Guidelines for the definition of managed objects) и GRM (General relationship model). В пользу выбора стандарта GDMO/GRM говорит его широкое распространение в сфере телекоммуникаций, в первую очередь – в области TMN. Однако TINA использует только те аспекты GDMO и GRM, которые подходят для информационной модели TINA (квази-GDMO/GRM или q-GDMO/GRM). Для графического представления информационных спецификаций были адаптированы средства OMT (Object modelling technique) и созданы инструменты для перевода OMT-диаграмм в q-GDMO/GRM.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (COMPUTATIONAL MODEL) определяет средства создания вычислительных спецификаций, базирующихся на объектно-ориентированном подходе и описывающих распределённые телекоммуникационные приложения в терминах вычислительных объектов (CO – Computational object). Чтобы обеспечить функционирование приложения, объекты взаимодействуют друг с другом, причем это взаимодействие происходит через интерфейсы двух типов:

- операционный интерфейс (operational interface) – набор операций, которые одни объекты (серверы) предоставляют для использования другим объектам (клиентам); операция может иметь аргументы и возвращать результат;
- потоковый интерфейс (stream interface) – не имеет операций и поддерживает потоковую передачу структурированных данных.

Графическое представление объектов дано на рисунке 5.6.3. В качестве системы нотации для вычислительных спецификаций выбран язык TINA ODL (Object definition language). TINA ODL является усовершенствованной версией OMG IDL (Object management group Interface definition language). Основная идея вычислительной модели заключается в том, чтобы скрыть от разработчика приложения сложность механизма взаимодействия распределённых программных компонентов.

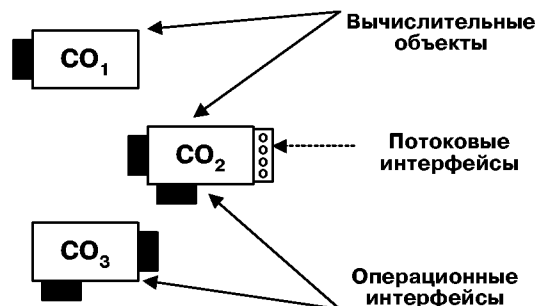


Рис. 5.6.3 Графическое представление объектов TINA

МОДЕЛЬ ПРОЕКТИРОВАНИЯ (ENGINEERING MODEL) предоставляет средства для определения структуры распределённого TINA-приложения в терминах компонентов, распределённых по узлам сети, а также моделей и механизмов для организации их взаимодействия. Как уже отмечалось выше, для поддержания взаимодействия компонентов используется DPE, а модели проектирования описывают способ размещения вычислительных объектов для работы в инфраструктуре DPE. Модель проектирования определяет архитектуру DPE, включающую в себя:

- ядро DPE – обеспечивает жизненный цикл объектов (создание и удаление), а также средства коммуникации между объектами (например, вызов интерфейсных операций); подразумевается, что ядро DPE присутствует во всех узлах, где используется DPE;
- базовую транспортную сеть (КТН – Kernel transport network) – служит для обеспечения взаимодействия объектов удалённых узлов; КТН является виртуальной сетью, которая логически отделена от транспортной сети и обеспечивает технологически независимое представление коммуникационных услуг, предоставляемых NCCE;
- сервисы DPE – обеспечивают операционный интерфейс для поддержки функционирования и взаимодействия объектов, динамическое размещение интерфейсов удалённых объектов и генерацию уведомлений для объектов.

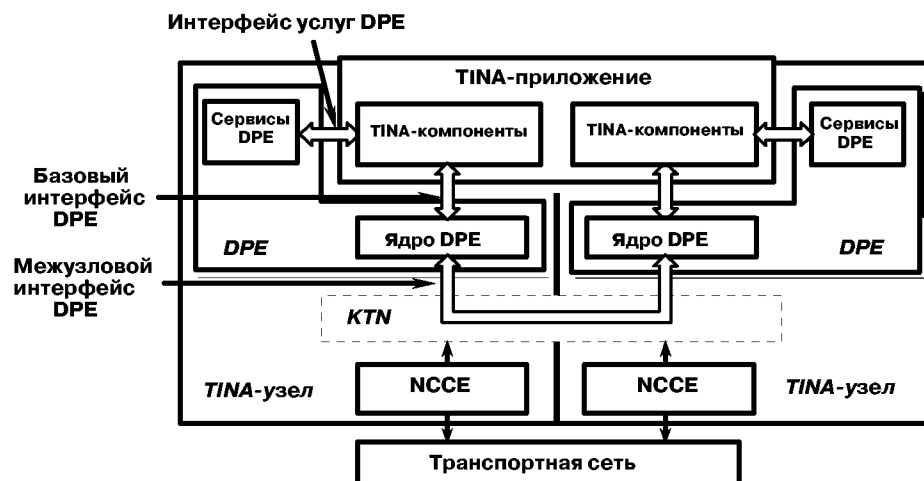


Рис. 5.6.4 Взаимодействие элементов архитектуры DPE

Разграничение между ядром DPE и сервисами DPE проведено для того, чтобы отделить набор средств и возможностей, которые должны существовать во всех узлах DPE, от остальных средств и возможностей. Рисунок 5.6.4 иллюстрирует архитектуру DPE и взаимодействие компонентов DPE при предоставлении услуг компонентам TINA.

5.6.1.3 Сетевая архитектура

Сетевая архитектура определяет общие принципы описания транспортной сети способом, не зависящим от каких-либо конкретных технологий, и обеспечивает механизмы установления, изменения и разрушения сетевых соединений.

Моделируемая посредством сетевой архитектуры TINA транспортная сеть предназначена, в основном, для переноса мультимедийной информации. Поток информации может быть разнородным в смысле форматов данных, требований к полосе пропускания и к другим характеристикам качества обслуживания. Сетевая модель TINA рассматривает интерфейсы приложений в виде потоковых интерфейсов, ассоциированных с TINA-приложением или мультимедиа-устройством.

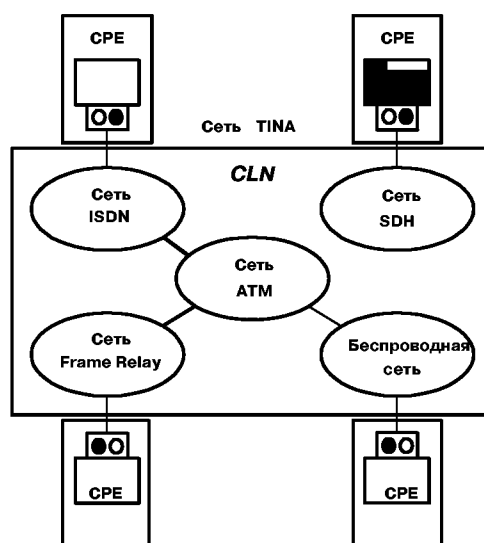


Рис. 5.6.5 Физическое представление TINA-сети

Физическое представление TINA-сети приведено на рисунке 5.6.5 и разделяется на два компонента: совокупность сетей уровня сетевых ресурсов и оборудование пользователя. Совокупность сетей уровня сетевых ресурсов (CLN – connectivity layer network) – это транспортная система, включающая в себя оборудование, которое позволяет сетевым ресурсам, построенным на базе различных технологий (ATM, FrameRelay, узкополосная ISDN, SDH и т.п.) транспортировать информацию разного типа. Оборудование пользователя (CPE – customer premises equipment) может представлять из себя либо простой терминал, либо компьютерную систему, в которой установлены TINA-приложения.

5.6.1.4 Архитектура услуг

Архитектура услуг определяет понятия и принципы построения, развертывания и использования телекоммуникационных услуг и эксплуатационного управления ими с применением для этих целей вычислительных объектов (CO) в качестве компонентов много-

разового использования. Процесс выполнения операций объединяется в понятие сессии, трактуемое как промежуток времени, в течение которого разные компоненты производят совместно определённые действия, направленные на получение пользователем какой-либо услуги. Концепция сессии определяет множество действий, специфических для данной услуги, и ряд общих операций, поддерживающих их использование. Определяют сессии четырех видов (см. рис. 5.6.6):

- **СЕССИЯ ДОСТУПА** открывается для каждого пользователя и управляет его участием в одной или нескольких сессиях услуг одновременно. Внутри контекста сессии доступа вводится понятие агента. Предусматриваются два типа агентов:
 - агент пользователя (UA – user agent) – вычислительный объект на стороне поставщика, который принимает запрос пользователя присоединить его к сессии услуги или создать такую сессию; агент пользователя также принимает запросы от сессии услуги, созданной другим пользователем, который желает, чтобы первый пользователь присоединился к ней;
 - агент поставщика (PA – provider agent) – вычислительный объект на стороне пользователя, ответственный за определение точного места расположения терминала пользователя, например, при подключении к сети переносного компьютера.

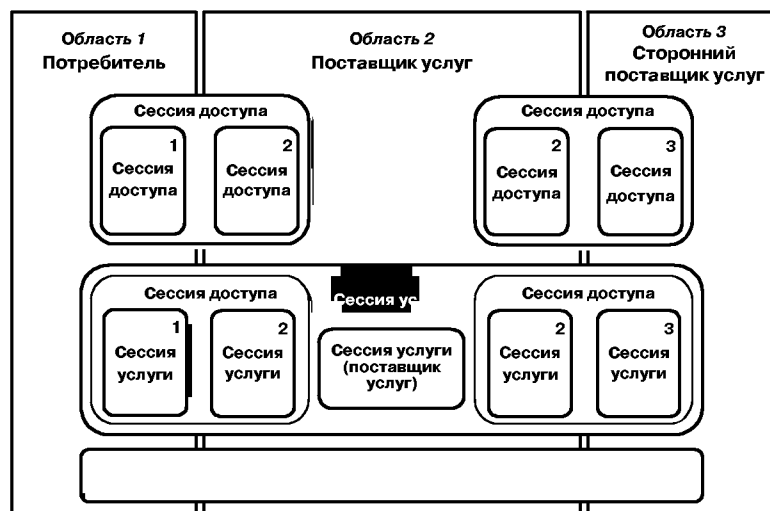


Рис. 5.6.6 Концепция сессии TINA

Чтобы получить доступ к услуге, пользователь должен ассоциировать своего пользовательского агента с агентом поставщика. Если

один пользователь одновременно ассоциирован с несколькими терминалами, то при входящем запросе агент пользователя должен определить, с каким именно агентом поставщика он должен связаться. Если пользователь уже имеет доступ к системе, то запрос поступает на один из доступных терминалов по указанию пользователя. В противном случае пользовательский агент определяет, через какой терминал пришёл запрос, и обращается к агенту поставщика, который способен обслужить пользователя. Таким образом, реальное местонахождение пользователей и терминалов может определяться через соответствующих агентов, что позволяет пользователям (и терминалам) перемещаться по территории, обслуживаемой сетью. Обеспечение концепции возложено на DPE.

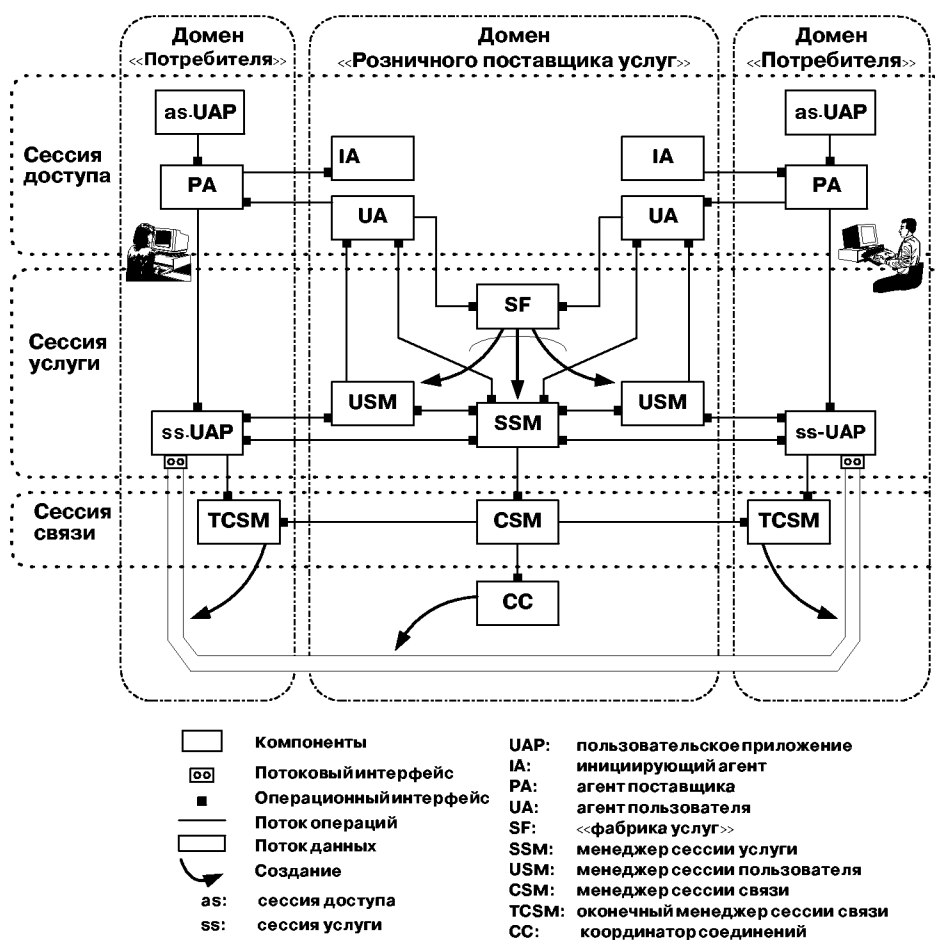


Рис. 5.6.7 Основные компоненты архитектуры услуг

- **СЕССИЯ УСЛУГИ** – отдельная активизация услуги, во время которой пользователям предоставляются сетевые ресурсы согласно логике этой услуги. Вычислительную часть сессии исполняет менеджер сессии услуги (SSM – Service session manager), имеющий операционные интерфейсы двух типов: общий интерфейс управления сессией (обеспечивает подключение и удаление пользователей, а также приостановку участия в сессии) и специфический интерфейс, обеспечивающий логику услуги. Сессию услуги, в свою очередь, разделяют на две части – сессию пользователя и сессию поставщика. Часть, ассоциируемая с поставщиком, представляет поведение (логику) услуги, общее для любого типа пользователей, вовлеченных в процесс ее использования. Сессия пользователя начинается с момента его вхождения в сессию услуги и заканчивается с его выходом из этой сессии. Во время сессии сохраняются индивидуальные настройки пользователя и специфические данные.
- **СЕССИЯ СВЯЗИ** – это ориентированная на услуги абстракция соединения в транспортной сети. Во время сессии связи происходит установление соединений, определение конечных точек, путей связи и показателей ее качества. Вычислительную основу сессии обеспечивает менеджер сессии связи (CSM – Communication session manager).

Разделение сессий услуги и доступа позволяет каждому пользователю выбрать свою технологию и метод доступа, менять свое расположение во время сеанса, а также временно приостанавливать и возобновлять участие в сессии. Разделение сессий услуги и связи делает услугу независимой от существующих соединений, благодаря чему отпадает необходимость сохранять соответствие между пользователями услуги и транспортными соединениями. Основные компоненты архитектуры услуг приведены на рисунке 5.6.7.

5.6.1.5 Архитектура эксплуатационного управления

Архитектура эксплуатационного управления определяет концепции и принципы создания систем эксплуатационного управления объектами в TINA-системах. Различают два типа управления (см. рис. 5.6.8):

- управление вычислениями – включает в себя управление NCCE (компьютером) и средой DPE; этот вид управления не принимает во внимание функциональные особенности приложений, работающих в DPE, а управляет размещением, конфигурацией, запуском и поддержкой выполнения соответствующих программ;
- телекоммуникационное управление – охватывает эксплуатационное управление транспортной сетью и приложениями, которые используют и контролируют транспортную сеть, а также управление услугами.

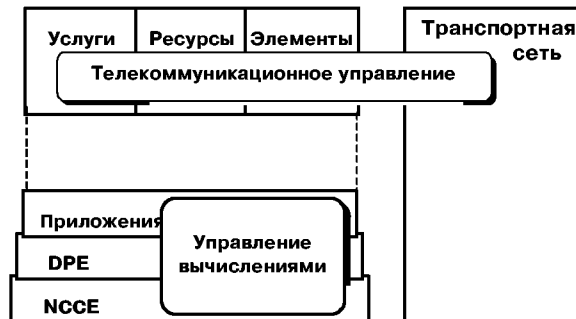


Рис. 5.6.8 Архитектура эксплуатационного управления TINA

5.6.2 Общие принципы CORBA

Для среды DPE TINA-систем в качестве стандарта де-факто рассматривается Общая архитектура с передачей запросов к объекту через посредника (CORBA – Common object request broker architecture), разработанная группой OMG. CORBA нацелена на обеспечение работы и взаимодействия разнородных (написанных на разных языках) приложений в распределенной среде. Центральным компонентом CORBA является посредник (ORB – Object request broker), который отвечает, главным образом, за обеспечение связи между клиентами и объектами. По сути дела, ORB является распределенной программной шиной (совокупностью связанных через транспортную среду ORB-компонентов), обеспечивающей взаимодействие между удаленными объектами, причем детали этого взаимодействия (поиск объекта в сети, его активизация, транспортировка сообщений и данных) скрыты от пользователя и существуют как уже реализованная часть шины ORB.

Кроме ORB, в состав архитектуры CORBA входит ряд компонентов, обеспечивающих доступ к шине со стороны приложения:

- ядро шины ORB,
- язык описания интерфейса (IDL – Interface definition language),
- хранилище интерфейсов (IR – Interface repository),
- отображения для языков программирования (Language mappings),
- концепции входных и выходных блоков IDL (Stubs и skeletons),
- динамический вызов и диспетчеризация (Dynamic invocation and dispatch),
- адаптеры объектов (Object adapters),
- протоколы взаимодействия ORB-компонентов.

Через шину ORB доставляются запросы к объектам и принимаются ответы клиентам на эти запросы. Объект, которому необходимо доставить запрос от клиента, называется целевым объектом. Ключевой особенностью ORB является то, что процесс связи клиент/сер-

вер в значительной степени скрыт от клиента. Обычно бывают скрыты следующие элементы взаимодействия.

- Местоположение объекта: клиент не знает, где располагается целевой объект – в отдельном процессе на другой машине сети, в другом процессе той же самой машины или внутри текущего процесса.
- Реализация объекта: клиент не знает, каким образом реализован объект, какой язык программирования или язык сценариев использован для его реализации, какая операционная система или какое оборудование работает в объекте.
- Состояние объекта: при передаче запроса к целевому объекту клиент не должен знать, активизирован ли этот объект (то есть, запущен ли соответствующий процесс) и готов ли он принять запрос. В случае необходимости, перед тем как передать к объекту запрос услуги, ORB – компонент невидимо для клиента запускает процесс на объекте.
- Механизм связи с объектом: клиенту неизвестен механизм (например, протокол, разделяемая память, вызов локального метода и т. д.), с помощью которого к объекту будет доставлен запрос и от объекта будет получен ответ.

Эти особенности ORB дают возможность разработчикам приложений сконцентрировать внимание на самих приложениях и не беспокоиться о проблемах распределенного системного программирования нижнего уровня. Архитектура CORBA представлена на рисунке 5.6.9.

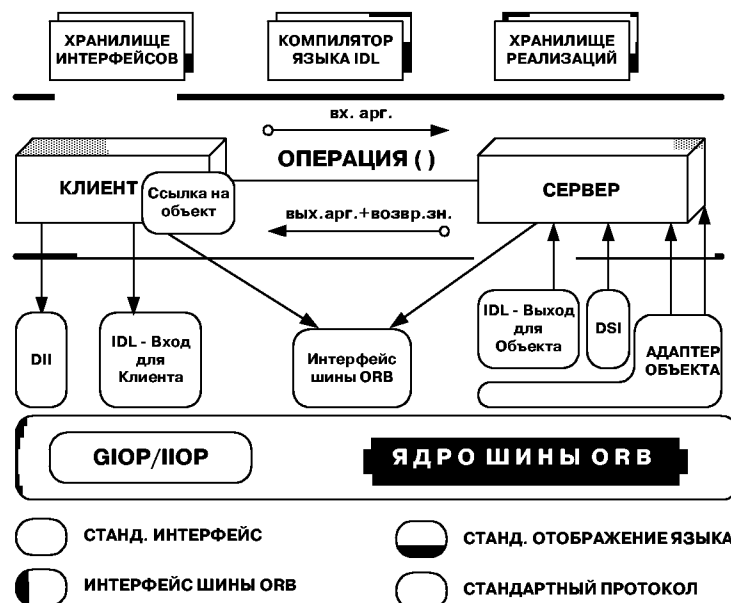


Рис. 5.6.9 Архитектура CORBA

Для передачи запроса клиент определяет целевой объект посредством ссылки, которая создается для каждого объекта CORBA в момент создания самого объекта. Используемая любым клиентом ссылка всегда соответствует именно тому объекту, для которого она была создана. Ссылка существует столь долго, сколько существует сам объект. Другими словами, ссылка всегда соответствует одному единственному объекту. Ссылки всегда постоянны и «закрыты» от клиента, так что он не может модифицировать их по своему усмотрению. «Внутренности» ссылки известны только ORB. Ссылки на объекты могут иметь стандартизованные форматы, например, в соответствии со стандартами OMG для Протокола взаимодействия между ORB-компонентами по правилам Internet (IIOP – Internet inter ORB protocol) или же для Общего протокола взаимодействия между ORB-компонентами (GIOP – General inter ORB protocol). Могут быть использованы и другие подходящие форматы.

Перед тем как передать запрос к объекту, клиент должен знать типы операций, поддерживаемых этим объектом. Операции и типы данных, которые поддерживаются объектом, определяются с помощью интерфейса объекта. Интерфейс определяет те запросы, которые могут быть переданы к объекту и обслужены им. Интерфейсы для объекта определяются на языке IDL. Интерфейсы CORBA похожи на классы языка C++ и на интерфейсы языка Java. Так, интерфейс статических вызовов (SII – Static invocation interface) представляет собой не более, чем вызов процедуры на языке C к скомпилированному модулю. Интерфейс динамических вызовов (DII – Dinamic invocation interface) – это вызов процедуры C, но с дополнительным кодом. DII используется, когда на момент компиляции доступна не вся необходимая для вызова объекта информация.

Язык OMG IDL не является полноценным языком программирования, а представляет собой (как и GDMO/ASN.1) декларативный язык. Например, в нем отсутствуют средства построения управляющих конструкций (циклы, переходы). Он не предназначен для непосредственной реализации распределенных приложений; для этого определяются отображения языка – правила преобразования структур и типов OMG IDL в эквивалентные конструкции одного из стандартных языков программирования.

Каждое приложение, реализованное на базе CORBA, в процессе выполнения задачи требует доступа к системе типов языка OMG IDL. Это необходимо, поскольку приложению должны быть известны типы значений, передаваемых в качестве аргументов запроса. Кроме того, приложению должны быть известны типы интерфейсов, поддерживаемых используемыми объектами. Этим целям служит «хранилище интерфейсов» системы CORBA, которое позволяет получить программно управляемый доступ к системе типов OMG IDL во время выполнения задачи. Само хранилище представляет собой объект

CORBA, вызов операций которого осуществляется по тем же правилам, что и операций других объектов CORBA. Язык IDL по своим возможностям практически не уступает языку GDMO/ASN.1 для описания информационных объектов.

5.6.3 IN и TINA

В предыдущих главах книги приведено немало доводов и примеров, подтверждающих эффективность концепции Интеллектуальной сети. Реализация платформы IN в инфраструктуре узкополосных сетей, потребовавшая от операторов сетей значительных капиталовложений, не предполагает замены такого дорогостоящего фундамента без очень серьёзных мотиваций. Однако повод для размышлений на эту тему уже просматривается. Это – все возрастающая потребность в мультимедийных услугах на базе широкополосных сетей B-ISDN, заставляющая операторов анализировать возможности усовершенствования и эволюции своих платформ IN для удовлетворения новых требований заказчиков, ради чего, собственно говоря, и была разработана TINA.

Хотя обе системы, и IN, и TINA, представляют собой архитектуры управления услугами информационных сетей произвольного типа, между функционально-ориентированной архитектурой IN и объектно-ориентированной архитектурой TINA существуют принципиальные различия, усложняющие их взаимодействие и взаимопроникновение. IN предусматривает декомпозицию услуг в группы многократно используемых блоков SIB и определяет функциональные сетевые элементы для их распределенной реализации. Архитектура IN основана на идее централизации функций управления услугами в специальных сетевых узлах и управления удаленными коммутаторами через сеть ОКС-7. Согласно этому для IN определен ряд функциональных объектов, в которых размещены программы логики услуг (SCF), относящиеся к услугам данные (SDF), функции обнаружения запросов услуг в коммутаторе (SSF), а также средства эксплуатационного управления услугами (SMF). Связь между этими элементами осуществляется посредством информационных потоков, которые передаются по сети в операциях прикладного протокола INAP системы ОКС-7.

TINA базируется на технологии среды распределённой обработки (DPE), которая должна поддерживаться каждым сетевым узлом. Услуги в TINA моделируются путём взаимодействия между вычислительными объектами (CO), включающими в себя логику, интерфейсы и операции. Архитектурой TINA определена группа порождающих CO, которые могут использоваться при создании услуг посредством объединения разных CO из этой группы и настройки их параметров.

Услуги создаются на основе постоянно развивающегося набора компонентов так, что каждый новый продукт (услуга) может быть создан на основе предыдущего, а потому нет нужды начинать всякий раз с заранее определенного набора компонентов, как это делается в случае с блоками SIB при создании услуг IN. Среда DPE поддерживает произвольное распределение объектов CO по узлам DPE, а взаимодействие между узлами DPE обеспечивает опорная транспортная сеть (КТН), так что не требуется иметь специализированные сетевые узлы с предопределенными функциями. В отличие от IN, использование в качестве КТН какой-либо из существующих сетей архитектурой TINA явно не обозначается.

Взаимодействие между CO системы TINA (поддерживаемое DPE и КТН) соответствует информационным потокам между FE платформы IN (через INAP сети ОКС-7). Однако в отличие от IN, где запрос выполнения услуги, обнаруженный размещенными в коммутаторе функциями SSF, проходит по сети «снизу вверх» к SCF, TINA использует ориентированный на административное управление подход к обеспечению связи, при котором функции управления соединениями выполняются «сверху вниз».

Необходимо подчеркнуть, что большинство функциональных возможностей IN относится к гибким системам проверки полномочий пользователей, маршрутизации в зависимости от разных условий и начисления платы. В TINA подобные функциональные возможности относятся, по большей части, к сессии доступа, а не к сессии услуги. Все это приводит к тому, что взаимно однозначного отображения услуг IN на услуги TINA и объектов IN на объекты TINA не существует.

Некоторые сравнительные оценки этих двух архитектур приведены в таблице 5.6.1. Конвергенция представленных (и ряда не представленных) в таблице характеристик концепций IN и TINA идет в двух направлениях, рассматриваемых в следующих параграфах данной главы: миграции и взаимодействия. Эти термины взяты из названия проекта P508 EURESCOM «Эволюция, пути миграции к TINA и аспекты взаимодействия».

Таблица 5.6.1

Критерии	IN	TINA
Принципы организации интерфейсов	Сообщения	Объекты, методы
Поддержка расширяемости	Хорошая, но ограничена возможностями INAP	Отличная, благодаря платформе DPE
Разделение услуг и сети	Физическое	Логическое
Эффективность создания услуг	Услуги создаются очень быстро, если не требуется модификация INAP	Начальные вложения могут быть большими, чем в IN, но потом используются многократно

Миграция или эволюция платформы на базе IN в направлении платформы TINA означает пошаговую «TINАризацию» элементов платформы Интеллектуальной сети, т.е. пошаговую замену функциональных объектов IN соответствующими компонентами услуг TINA – CO верхнего уровня DPE. По очевидным соображениям практического характера такая замена начнётся с тех функциональных объектов IN, которые установлены в небольшом количестве, например SMF и/или SCF/SDF. Образующаяся при этом платформа будет состоять из двух взаимодействующих частей (подсистем): исконной части IN и новой «TINАризованной» части. Для отображения операций INAP на операции, обеспечиваемые объектами TINA, и наоборот должны быть определены соответствующие адаптерные блоки (AU).

Взаимодействие между системами Интеллектуальной сети и TINA становится необходимым и в тех случаях, когда оператор развернул платформу на базе TINA и ему нужно обеспечить ее согласованную работу с платформами IN других операторов для предоставления межсетевых услуг. Возникает необходимость иметь блок сопряжения (IWU), обеспечивающий взаимодействие на прикладном уровне, к примеру, между SCF/SDF в зоне оператора IN и набором компонентов на верхнем уровне DPE в зоне оператора системы TINA.

Таким образом, общим для аспектов миграции и взаимодействия IN/TINA является определение блоков, подходящих для того, чтобы установить соответствие между операциями INAP системы ОКС-7 и интерфейсными операциями CO TINA в среде DPE. Этот процесс взаимного проникновения IN и TINA активно внедряется в реальные продукты.

Так, Alcatel, развивавший первоначально закрытую технологию классической IN, вводит в своей новой версии SCE интерфейсы на основе CORBA для приложений, разработанных сторонними поставщиками. Ведутся также работы в рамках исследовательского проекта ALCIN, ориентированного на оценку концепции TINA на уровне прототипов. Alcatel Corporate Research Center и KPN Research совместно разработали несколько услуг в управляемой по принципам TINA инфраструктуре, которая демонстрировалась на Telecom '99. Разработка базируется на платформе Alcatel Mu3S, которая представляет собой прототип, выполненный в соответствии с архитектурой TINA.

Siemens и Italtel ведут работы по проекту SISTINA, призванному создать конкурентоспособную архитектуру, базирующуюся на концепции TINA и позволяющую оператору сети общего пользования получать доходы от продажи услуг IN в условиях будущей интегрированной сети с продвижением «интеллекта» к краям сети.

Отмеченные выше тенденции развития концепций и технологий предоставления услуг позволяют утверждать, что архитектура TINA будет определять облик телекоммуникационных сетей XXI века. Более того, даже вне связи с конкретной архитектурой TINA в практике разработки телекоммуникационного ПО широчайшее применение найдет технология CORBA.

5.6.4 Этапы миграции IN к TINA

Миграция IN к TINA означает постепенную замену отдельных компонентов Интеллектуальной сети соответствующим оборудованием TINA, названную в предыдущем параграфе пошаговой «TINАризацией» элементов платформы IN, т.е. замену функциональных объектов IN адекватными наборами взаимодействующих вычислительных объектов TINA на базе DPE. Сказанное означает, что в вычислительных объектах TINA реализуются услуги квази-IN и одновременно определяются соответствующие функции адаптации – так называемые адаптерные блоки AU. Эти блоки должны поддерживать взаимодействие между существующей подсистемой IN и новой «TINАризованной» подсистемой. В данном контексте возможны следующие основные шаги эволюции, намеченные в проекте P508 EURESCOM и представленные на рисунке 5.6.10.

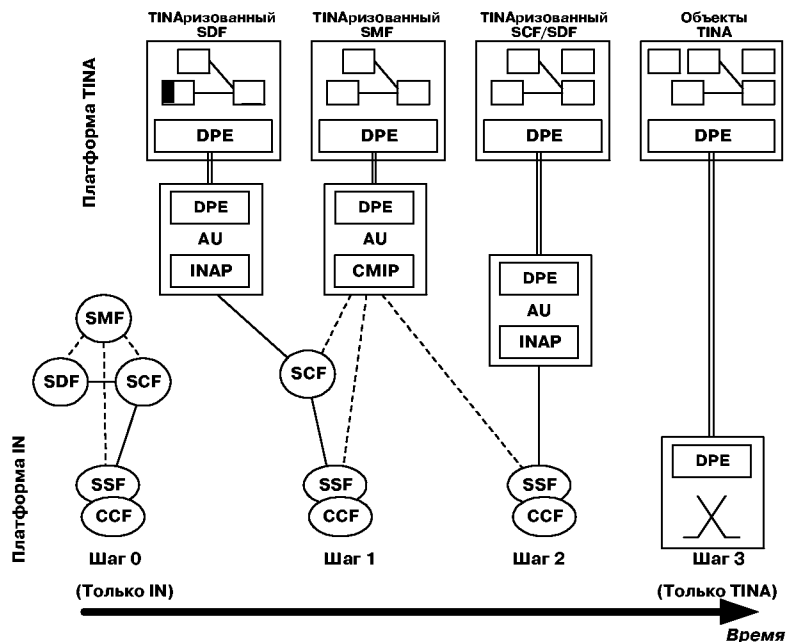


Рис. 5.6.10 Пошаговая миграция IN к TINA

Шаг 1: «TINАризация» базы данных об услугах IN. На этом этапе функции SDF и/или SMF начинают выполняться компонентами TINA. Такой подход позволяет воспользоваться методами моделирования данных, применяемыми в TINA, и прозрачным распределением данных, которое обеспечивает DPE. При этом требуется иметь блок AU, обеспечивающий доступ SCF Интеллектуальной сети к рабочим данным об услугах и данным для управления услугами платформы TINA. Сценарий отдельной «TINАризации» на шаге 1 функций SDF и эксплуатационного управления SMF, скорее всего, нереален, поскольку разделяет логику услуги и базу данных об услугах, что вступает в противоречие с объектно-ориентированной концепцией TINA, предполагающей их интеграцию.

Шаг 2: Согласованная «TINАризация» функций SMF, SCF и SDF с целью максимального использования преимуществ архитектуры услуг TINA и объектно-ориентированного подхода. Это означает, что «TINАризуются», т.е. реализуются посредством объектов CO системы TINA в среде DPE, функции оперативного и эксплуатационного управления услугами IN. Из архитектуры IN остаются только функции SSF, поскольку на них падает основная доля сделанных ранее инвестиций в ТФОП и IN. Данный сценарий позволяет операторам сети продолжать использовать существующее коммутационное оборудование и интерфейсы INAP. При этом требуются специальные блоки AU, чтобы обеспечить взаимодействие объектов CO системы TINA, в которых реализованы функции SCF/SDF и SMF, с функциональными объектами SSF сети IN.

Шаг 3: Внедрение TINA в коммутационные станции. Данный сценарий предусматривает замену вычислительными объектами TINA также и функций SSF, что становится реальным при внедрении новых широкополосных систем коммутации. На этом шаге модифицируются или вовсе удаляются функции, связанные с моделью базового процесса BCSM, а объекты DPE вводятся непосредственно в коммутационный узел или в соответствующий блок AU, адаптирующий станционные интерфейсы управления к TINA-архитектуре управления услугами и соединениями. При таком подходе интерфейс на основе INAP больше не требуется, и коммутационные узлы могут вызывать операции, предусмотренные в CO TINA, непосредственно через DPE.

Ограниченный объем книги не позволяет рассмотреть пошаговую «TINАризацию» более подробно. Отметим лишь, что блок AU между SSF и подсистемой TINA, называемый IN(-SCF)-AU, должен отображать операции INAP системы ОКС-7, вызываемые FE SSF, на вызовы по методу CO TINA через DPE, и наоборот. Таким образом, IN(-SCF)-AU дает возможность устанавливать управляющие связи между SSF в области IN и менеджером сессии услуги (SSM) в области TINA, т.е.

SSM в TINA управляет SSF в IN через AU. Определение IN(-SCF)-AU в значительной степени зависит от способа моделирования услуг IN и пользователей услуг в среде TINA.

5.6.5 Взаимодействие IN и TINA

Уже отмечалось, что между функционально-ориентированной архитектурой IN и объектно-ориентированной архитектурой TINA существуют принципиальные различия, и что различна организация связи между системными компонентами в IN и в TINA: в то время как IN использует систему сигнализации OKC-7, в TINA используются услуги среды распределенной обработки (DPE). Тем не менее, общая функциональная направленность IN и TINA (предоставление услуг связи) ставит вопрос об их взаимодействии, т.е. о возможности доступа пользователей TINA к уже существующим услугам IN. Обратный доступ от IN к логике услуг TINA также может быть востребован для предоставления пользователям IN тех услуг и возможностей TINA, которые не поддерживаются в IN.

Чтобы обеспечить взаимодействие систем с разными технологиями, между ними должен быть помещен блок сопряжения IWU. Для связи между IN и IWU, с одной стороны, как и между TINA и IWU, с другой, должны использоваться стандартизованные интерфейсы. На стороне IN наиболее приемлемым протоколом обмена информацией логики услуг выглядит протокол INAP для интерфейса SCF-SCF, определённый в CS-2. На стороне TINA должна быть выбрана подходящая опорная точка TINA.

В дополнение к взаимодействию на уровне логики услуг потребуется также и взаимодействие на уровне средств доставки информации. Сказанное касается всех услуг, при предоставлении которых должно быть установлено соединение между пользователем в IN и пользователем в TINA. Поскольку относится это, главным образом, к взаимодействию между узкополосной ISDN (N-ISDN) и TINA, на стороне IN для связи с IWU может использоваться протокол ISUP. На стороне TINA должна быть выбрана подходящая опорная точка TINA.

Взаимодействие на уровне средств доставки информации при установлении соединения, инициированного со стороны IN (т.е. в N-ISDN), становится необходимым, когда в процессе предоставления услуги IN запрос установить соединение передается ко входящей стороне, находящейся в сети TINA. Согласование протоколов управления услугами IN (INAP) и TINA в этом случае не требуется, но зато требуется обеспечить взаимодействие протоколов N-ISDN (ISUP) и TINA. При взаимодействии на уровне средств доставки сеть TINA должна восприниматься со стороны IN (N-ISDN) как часть сети N-ISDN, т.е. подчиняться правилам N-ISDN. На другой стороне соединения сеть N-ISDN рассматривается с точки зрения TINA как совместимая с TINA сеть, принадлежащая другому домену. Запрос со-

единения со стороны N-ISDN воспринимается как запрос этим вторым доменом соединения в собственном домене сети. Чтобы преодолеть барьер, разделяющий разные технологии, между N-ISDN и TINA также должен быть помещён блок IWU.

Рассмотрим пример реализации услуги Freephone в случае TINA-ризованного SCP. Ни один из участников предполагаемой связи не находится в области TINA, но квази-IN возможности управления услугами полностью моделируются вычислительными объектами TINA, то есть пользовательским приложением UAP_{IN} и менеджером сессии услуги SSM_{IN} , которые используют необходимые именно для Freephone объекты (SSO – Service specific object), например базы абонентских данных и логических номеров. Со стороны SSF блок AU выглядит как SCF, а со стороны TINA – как терминал пользователя и по-этому содержит, кроме UAP_{IN} , агента поставщика PA и конечную точку сессии общего рода (GSEP – Generic session endpoint). Следовательно, AU описанного типа может быть соотнесён с опорной точкой RetRP бизнес-модели TINA (см. рис. 5.6.11).

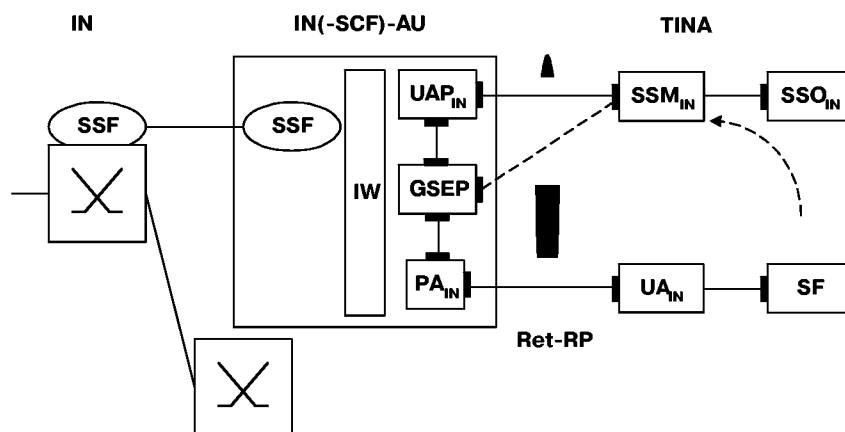


Рис. 5.6.11 Применение TINA для реализации услуги квази-IN

После набора пользователем логического номера запрос услуги обнаруживается функциями SSF сети IN, и соответствующая индикация передается в операции *INAP InitialDP* к блоку адаптации IN(-SCF)-AU. Приняв запрос, AU передает его к UAP_{IN} , который при помощи PA инициирует организацию сессии услуги. Чтобы организовать сессию доступа, PA взаимодействует с UA в административной области поставщика услуги. После того как «фабрика услуг» SF создаст SSM_{IN} , UAP_{IN} взаимодействует с SSM_{IN} с целью обмена информацией управления. В контексте этих диалогов производится обмен информацией, относящейся к преобразованию логического номера в физический. Получив физический номер, AU передает операцию *Connect* к SSF. Преобразование операций INAP в вызовы объектов TINA и обратно обеспечивается UAP_{IN} .

Следует отметить, что в отличие от миграции, которая может начаться очень скоро, взаимодействие понадобится только тогда, когда будут реализованы первые сети TINA. Сегодня трудно предсказать, через какой интерфейс будет осуществляться это взаимодействие, ибо это будет зависеть также от дальнейшего развития IN. Таким образом, выбор конкретных сценариев, имеющих хороший шанс воплотиться на практике, вряд ли можно считать сделанным. Именно поэтому в данном параграфе названы два сценария взаимодействия: один на уровне INAP, а другой на уровне ISUP.

5.6.6 Интеграция ОКС-7 и CORBA

На рисунке 5.6.12 представлена общая схема взаимодействия систем IN и TINA. Структура и функционирование блоков адаптации AU и сопряжения IWU в обеих системах будет сильно зависеть от конкретных реализаций IN и TINA в соответствующих сетях. Однако общее для всех блоков адаптации и сопряжения состоит в необходимости преобразования операций INAP ОКС-7 в вызовы объектов, размещённых в DPE. В TINA среда DPE базируется на архитектуре CORBA. Следовательно, нужен «шлюз», способный конвертировать INAP ОКС-7 в CORBA и обратно.

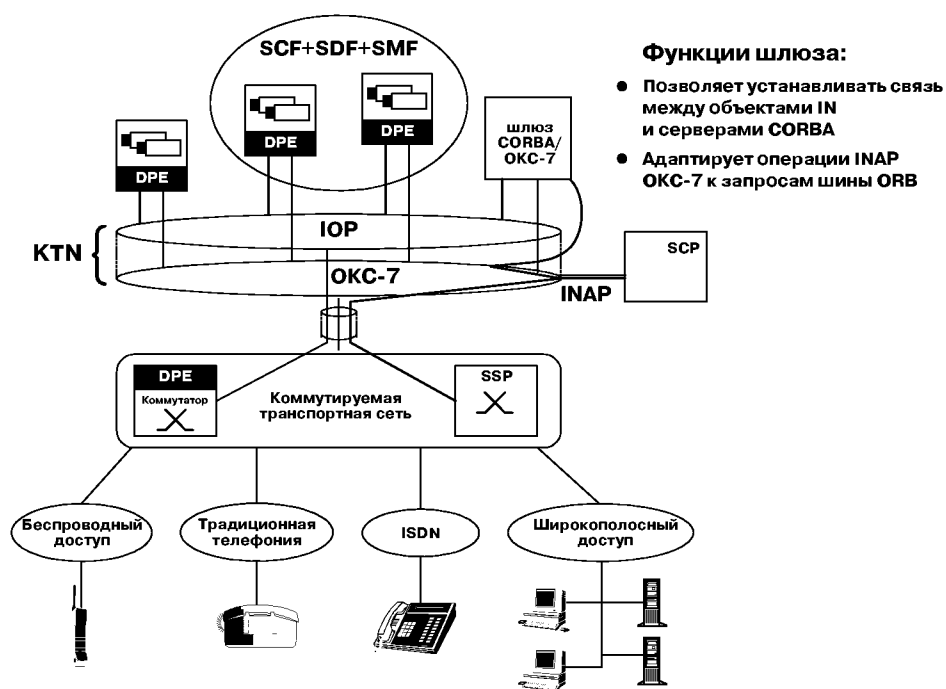


Рис. 5.6.12 Схема взаимодействия IN и TINA

Существенными при этом являются несколько важных требований, которые должны быть выполнены до того, как продукты CORBA можно будет использовать в сетях связи. Таковы, например, фундаментальное требование обеспечить режим реального времени, а также требования высокой надежности и отказоустойчивости. Именно на решение этих проблем нацелен документ IN/CORBA White Paper, разрабатываемый в настоящее время в рамках Рабочей группы по управлению объектами (OMG) в области электросвязи, а также упоминавшийся выше проект EURESCOM P508, в котором даны два прикладных сценария интеграции OKC-7 и CORBA.

Принцип связывания операций в TINA сильно зависит от принципов процедурных вызовов, определенных OMG. На момент написания книги существовала всего одна (также определенная OMG) спецификация общего протокола связи между узлами DPE (GIOP), а именно – протокол IIOP (Internet inter ORB protocol) взаимодействия между ORB-компонентами в архитектуре CORBA, версия 2.0.

Как известно, по ряду причин сеть Internet не может обеспечить гарантированное качество обслуживания ни в части потерь пакетов данных, ни в части их задержки. Поэтому до момента введения новых протоколов (некоторые из них находятся в стадии апробации, например IPv6 и RSVP) во все узлы сети Internet протокол IIOP целесообразно использовать лишь внутри одного или нескольких доменов, в которых оператор IP-сети может гарантировать качество обслуживания, требуемое для поддержки услуг в сетях общего пользования. С другой стороны, при введении объектов CORBA в инфраструктуру телефонных сетей возникает естественное желание использовать возможности сетей OKC-7 в качестве транспортной среды для распределенной шины ORB.

Сети, построенные с применением системы сигнализации OKC-7, широко распространены и обеспечивают необходимое качество обслуживания, в связи с чем производители телекоммуникационного оборудования ведут работы по использованию OKC-7 как ESIOP для переноса сообщений протокола GIOP. Для сопряжения TCAP OKC-7 и CORBA требуется преобразование протокола ROSE, который используется подсистемой TCAP, в язык определения интерфейсов IDL группы OMG и наоборот. Структура шлюза, осуществляющего такое преобразование протоколов, представлена на рисунке 5.6.13.

Одно из решений может заключаться в том, что первоначальные запросы от любого объекта CORBA направляются к объекту-посреднику с интерфейсом IDL в брокере объектных запросов (ORB) клиента. Этот объект имеет открытые прикладные программные интерфейсы API и отвечает за преобразование запроса в форму, понятную серверу ORB. Такая стратегия не очень эффективна, поскольку

запросы должны сперва передаваться к объекту-посреднику, и только потом – к другой стороне соединения. Вместе с тем, её легко реализовать, причем не требуется участие поставщиков ORB. Поэтому данную стратегию имеет смысл применять во всех случаях, когда производительность не является главным критерием. Соответствующие функциональные возможности реализуются в приложении нижнего (распределённого) уровня, в котором на одной стороне (на стороне сети сигнализации) обеспечивается интерфейс OKC-7, а на другой (на стороне CORBA) – интерфейс IDLAPI с приложениями.

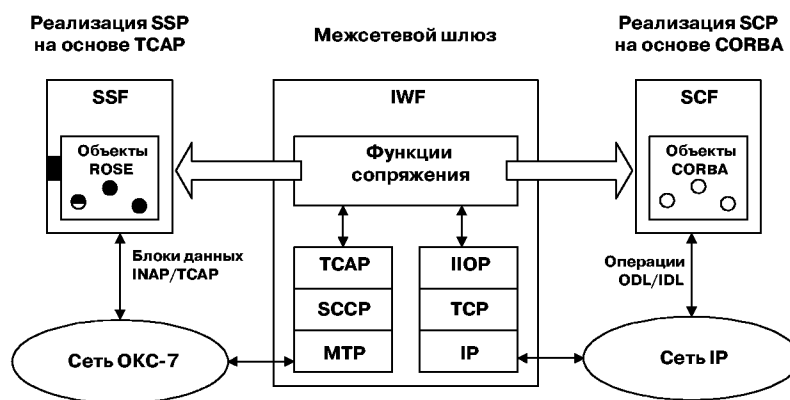


Рис. 5.6.13 Структура межсетевого шлюза OKC-7/CORBA

В связи с отсутствием достойной альтернативы OKC-7 может также использоваться и для соединения узлов CORBA DPE внутри TINA-сети. Имеется возможность реализовать на прикладном уровне OKC-7 общий протокол GIOP или, если необходимо, разработать новый протокол, ориентированный на конкретную среду (ESIOP – Environment specific inter ORB protocol).

В качестве ESIOP можно использовать стек MTP/SCCP (GIOP «поверх» SCCP) или стек MTP/SCCP/TCAP (GIOP «поверх» TCAP). В первом случае реализация GIOP более проста, но привязана к транспортной услуге SCCP, вследствие чего потребует изменений при введении в стек OKC-7 нового транспортного протокола. Во втором случае реализация GIOP несколько сложнее, так как требует преобразования используемых TCAP форматов данных ASN.1 в формат ODL CORBA, однако она обеспечивает простоту добавления новых прикладных протоколов, что может понадобиться в будущем.

Достоинством данного решения является то, что эффективно используются капиталовложения в сеть OKC-7, которая служит очень надёжным и высокопроизводительным средством связи между узлами. К недостаткам следует отнести то, что такая стратегия, разумеется, требует участия поставщиков ORB. Кроме того,

чтобы удовлетворить эксплуатационные требования, предъявляемые телекоммуникационными услугами, узлы DPE в составе коммутационных станций и узлы, содержащие интеллект услуг, должны обеспечивать высокую надёжность и производительность при взаимодействии.

5.6.7 Перспективы применения TINA

С точки зрения TINA телекоммуникационная архитектура выглядит системой, в которой приложения, связанные с услугами доставки информации, с управлением информационными услугами и с эксплуатационным управлением, можно будет легко создавать, гибко модифицировать и развивать экономически эффективным способом. При этом приложения не будут зависеть от физической транспортной сети, технологии, протоколов и географического расположения узлов. TINA ориентирована на решение проблем, возникающих в связи с глобализацией, либерализацией и административным регулированием быстро изменяющегося телекоммуникационного рынка с большим числом участников.

Архитектура удовлетворяет потребности всех потенциальных участников рынка с точки зрения открытости телекоммуникационного пространства и информационного обеспечения, обеспечивает большую свободу в выборе партнера, предусматривает защиту, возможность многократного использования и надежность программного обеспечения. TINA основана на совершенно новом подходе, учитывающем самые последние достижения в области разработки объектно-ориентированного программного обеспечения.

Реализация сформулированных в TINA требований будет означать кардинальное изменение ситуации, сложившейся на рынке телекоммуникаций, где существует множество разнородных систем создания услуг и эксплуатационного управления ими, сетевых протоколов и коммутационных платформ. Каждая из этих систем изначально предназначалась для решения специфических задач, при создании разных систем использовались разные технологии и стандарты разных международных и национальных организаций.

В настоящее время TINA входит в фазу консолидации спецификаций и результатов. По завершении этой работы индустрия телекоммуникаций получит для апробации и оценки новую архитектуру с полным набором спецификаций. К сожалению, сегодня еще очень невелик опыт практического использования новой архитектуры, особенно в части ее производительности, стабильности и масштабируемости применительно к системам реального времени,ковыми являются программно-аппаратные комплексы систем связи. Однако ситуация, по-видимому, улучшится уже в ближайшем буду-

щем, когда станут известны результаты применения архитектуры в нескольких прототипах разных международных исследовательских проектов.

TINA – очень интересное и многообещающее решение, рассчитанное на среднесрочную и долгосрочную перспективу, но не лишённое, однако, определенных недостатков. Основным недостаток состоит том, что TINA предлагает совершенно новую архитектуру, уделяя при этом недостаточно внимания проблеме ее совместимости с существующими системами. Кроме того, предложенная TINA архитектура управления ресурсами транспортной сети показала себя несколько «медлительнее» существующей. Таким образом, ключевой фактор, определяющий успех этой архитектурной концепции, состоит в возможности постепенной интеграции существующих систем, таких как IN, N-ISDN, B-ISDN, TMN и Internet, в единую TINA-систему при максимально долгом использовании существующих систем и стандартов протоколов.

Глава 5.7

Сравнительный анализ способов предоставления услуг

5.7.1 Сеть связи как большая система

Теперь, когда уже рассмотрены концепция, архитектура и техника Интеллектуальной сети и основные тенденции ее конвергенции с другими перспективными телекоммуникационными технологиями, перейдем к несколько более общему подходу к проблематике предоставления телекоммуникационных услуг. Как известно, современная сеть связи представляет собой сложную систему с распределенным «интеллектом» и функционирует в соответствии с правилами, определяющими взаимодействие между ее элементами, каждый из которых, в свою очередь, тоже является сложной системой.

Большая система – это такая система, которая обладает, по крайней мере, одним из следующих свойств:

- требует сложного программного обеспечения,
- имеет разветвленную структуру,
- покрывает значительную территорию,
- обслуживает большое количество пользователей.

Сеть связи обладает всеми вышеперечисленными свойствами, поэтому ее можно рассматривать как большую систему. Развитие сети, как и любой большой системы, происходит циклически («волнами»). Первая фаза цикла связана с инвестированием и созданием технической платформы для предоставления услуг. На второй фазе осуществляется эксплуатация сети и возврат вложенных инвестиций. Через некоторое время система морально устаревает и возникает необходимость в следующей фазе – инвестировании средств для создания более современной технической платформы – модернизированной или совершенно новой.

Моральное старение сети связи является неизбежным следствием технического прогресса и роста потребностей пользователей. Для описания процесса морального старения введем *фактор современности* – величину, которая в существующей сети непрерывно снижается (см. рис. 5.7.1). После того, как этот фактор достигнет определенного уровня, сеть связи становится для оператора скорее обузой, чем источником доходов. В идеальном случае система связи должна быть модернизирована или заменена прежде,

чем наступит этот момент, но на практике, с учетом сложности сети, сделать это бывает непросто и дорого, а потому некоторые сетевые элементы какое-то время эксплуатируются, уже будучи морально устаревшими.

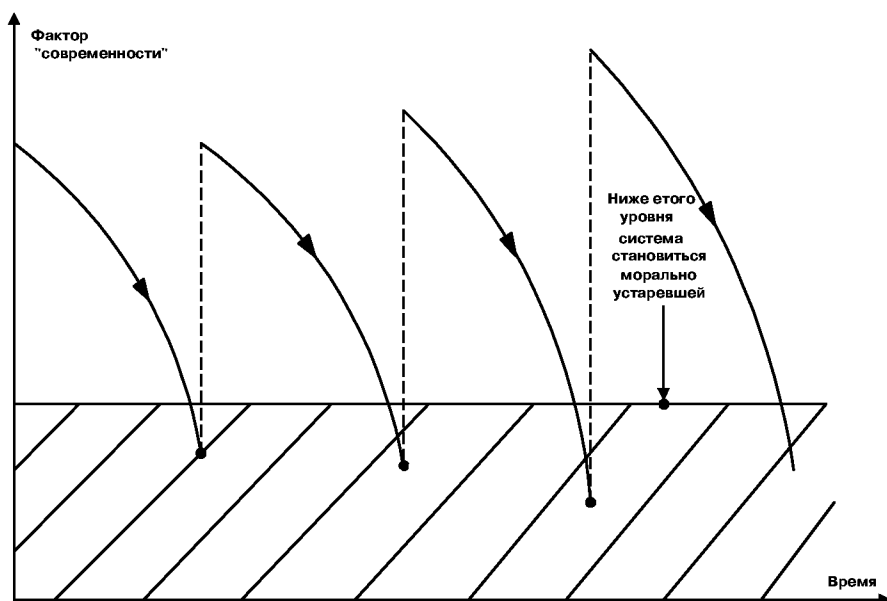


Рис. 5.7.1 Фактор современности

Рассмотренная выше цикличность развития сети связи явно прослеживается начиная с 70-х годов: создание цифровых систем коммутации с программным управлением, внедрение узкополосных цифровых сетей интегрального обслуживания, развитие концепции Интеллектуальной сети и широкополосные технологии.

С точки зрения пользователей, технические достижения, используемые при построении сетей, проявились, прежде всего, в повышении их надежности, а также в доступности новых дополнительных услуг. В 70-х годах новые услуги были доступны только пользователям учрежденческих АТС, в 80-х – абонентам отдельных АТС сетей общего пользования, а затем, в 90-х, – любому абоненту в пределах всей сети связи.

Можно выделить следующие основные факторы, влияющие на модернизацию системы связи:

- **Стоимость модернизации.** Из-за постоянного усложнения систем связи каждая следующая модернизация их функций обходится дороже предыдущей. Например, изменить план нумерации в определенном регионе становится тем дороже, чем больше в нем абонентов.

- *Особенности существующей системы.* Модернизация, в отличие от построения новой системы, предполагает необходимость решения проблем, связанных с совместимостью оборудования разных поколений.
- *Техническая эволюция в целом.* Растущий темп технического прогресса ведет к уменьшению интервала времени между появлениями на рынке новых поколений оборудования, функционирующих в составе системы связи.

Цикл жизни системы связи можно разделить на три составляющие: (1) время, затрачиваемое на модернизацию оборудования – T_a , (2) интервал времени, в котором система полностью удовлетворяет требованиям оператора – T_b , (3) интервал времени, когда система эксплуатируется, но уже не удовлетворяет этим требованиям – T_c .

Если T_g – интервал времени между появлением двух следующих друг за другом поколений системы связи, то справедливы следующие предположения (см. рис. 5.7.2):

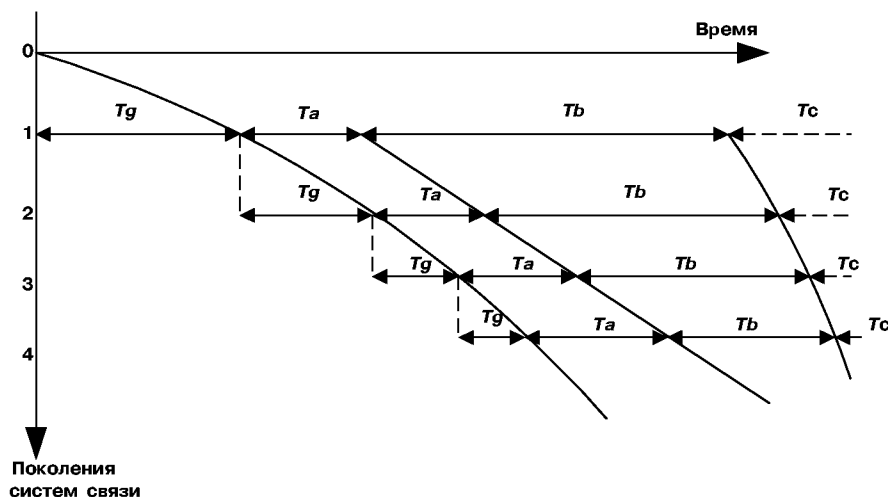


Рис. 5.7.2 Эволюция поколений системы связи

1. Чем быстрее идет технологический прогресс, тем быстрее устареваеет система (T_g и T_b уменьшаются).
2. Система становится более сложной, то есть содержит большее количество функций. Кроме того, увеличивается число одновременно находящихся в эксплуатации систем разных поколений. Оба эти фактора приводят к увеличению времени модернизации системы (T_a увеличивается).

Очевидно, что при сохранении описанных тенденций необходимо принять меры к тому, чтобы (1) T_a не достигло T_b , т.е. чтобы время, затрачиваемое на модернизацию, не сравнялось со временем,

в течение которого еще не возникают новые требования к оборудованию со стороны пользователя, и (2) чтобы T_d не стало меньше T_a , т.е. чтобы период между появлением на рынке новых поколений оборудования, не стал меньше времени, затрачиваемого на модернизацию.

Большая система – сеть связи – может, в свою очередь, рассматриваться как совокупность нескольких больших систем, каждая из которых предназначена для решения круга задач определенного уровня: уровня транспортных сетей, уровня коммутации, уровня управления услугами и уровня сигнализации. Последние три уровня тесно взаимосвязаны с точки зрения предоставления услуг и могут рассматриваться как самостоятельная распределенная большая система, основной задачей которой является предоставление услуг связи конечному пользователю.

5.7.2 Использование технологии IN для модернизации сети связи

Большое количество, сложность и частота введения новых функций в сети связи потребовали нового подхода, способного кардинально изменить все аспекты создания и предоставления услуг, а также эксплуатационного управления ими. Определилась необходимость перехода от использовавшегося в течение долгих лет консервативного подхода, предусматривающего предоставление небольшого перечня услуг, к созданию платформы, позволяющей вводить широкий спектр нетрадиционных услуг и дающей возможность «подстраивать» их под индивидуальные требования клиента. С точки зрения аналитического представления эволюции сети это адекватно решению проблемы уменьшения T_a – времени, которое затрачивается на модернизацию сети.

На рис. 5.7.3. показано, как различается характер зависимости фактора современности от времени при реализации новых функциональных возможностей телефонной сети путем модернизации существующих коммутационных систем и путем создания сети IN. В первом случае каждый раз требуется модификация или замена программного обеспечения во всех компонентах системы. Во втором случае процесс обновления требует существенно меньше изменений программного обеспечения и занимает гораздо меньше времени благодаря ограниченному количеству интерфейсов с платформой и возможности создания новых услуг посредством комбинирования логики выполнения программных блоков SIB, заложенных в эту платформу.

Чем меньше времени тратится на создание услуги (как это имеет место при наличии платформы IN), тем меньшим может быть интервал времени между моментами внедрения новых услуг, тем чаще воз-

можно обновление системы и тем меньше риск ее морального устаревания. Использование для создания услуг технологии IN снижает риск того, что услуга станет несвоевременной и перестанет удовлетворять пользователя. Уменьшение времени, требуемого на внедрение новой услуги, снижает и стоимость обновления системы. Рассмотрим, как изменяются перечисленные выше факторы, влияющие на модернизацию системы.

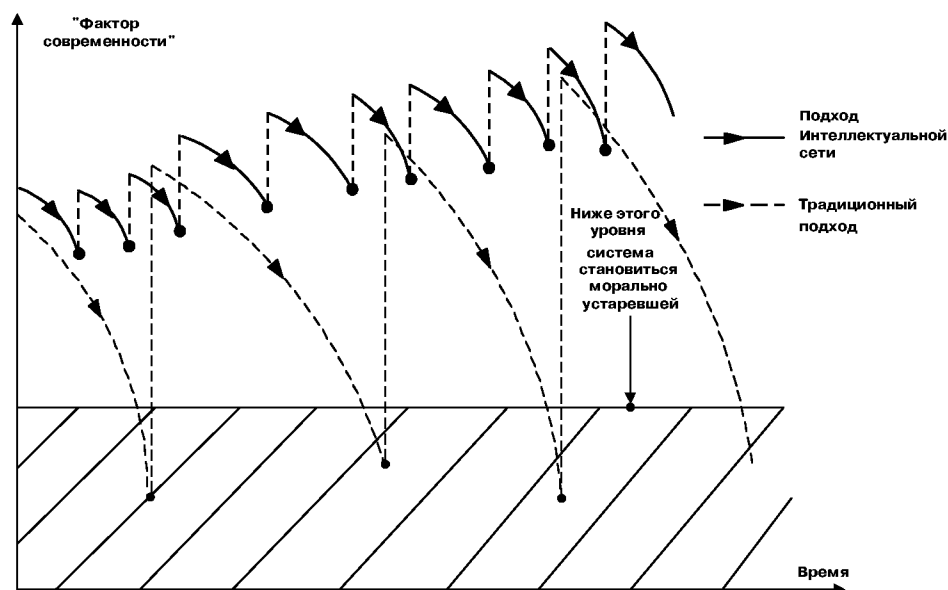


Рис. 5.7.3 Изменение "фактора современности" для двух вариантов создания услуг

Стоимость модернизации системы может быть поделена на две составляющие (а) стоимость модернизации платформы Интеллектуальной сети и (б) стоимость введения новых услуг на базе платформы Интеллектуальной сети.

Для (а) справедливо заключение, сделанное ранее и состоящее в том, что чем больше и сложнее система, тем дороже стоит ее расширение. Расширение платформы IN включает в себя введение новых протоколов сигнализации (от INAP CS-1 к INAP CS-2), увеличение числа SCP с целью резервирования, увеличение производительности оборудования и т.п. Что касается (б), то стоимость введения новой услуги на базе существующей платформы в случае применения технологии IN существенно снижается по сравнению с традиционным способом.

Особенности существующей системы сказываются в том, что модернизация каждого поколения платформы должна обеспечивать совместимость «снизу вверх», совместимость же старых и новых услуг обеспечивается проще благодаря возможности создания новых

услуг из блоков SIB, являющихся частью существующей платформы. Таким образом, инвестиции, необходимые для модернизации сети, при введении новых услуг на базе платформы IN могут быть ниже, чем при традиционном способе модернизации. Сравнение характера роста инвестиций для двух способов введения услуг приведены на рисунке 5.7.4.

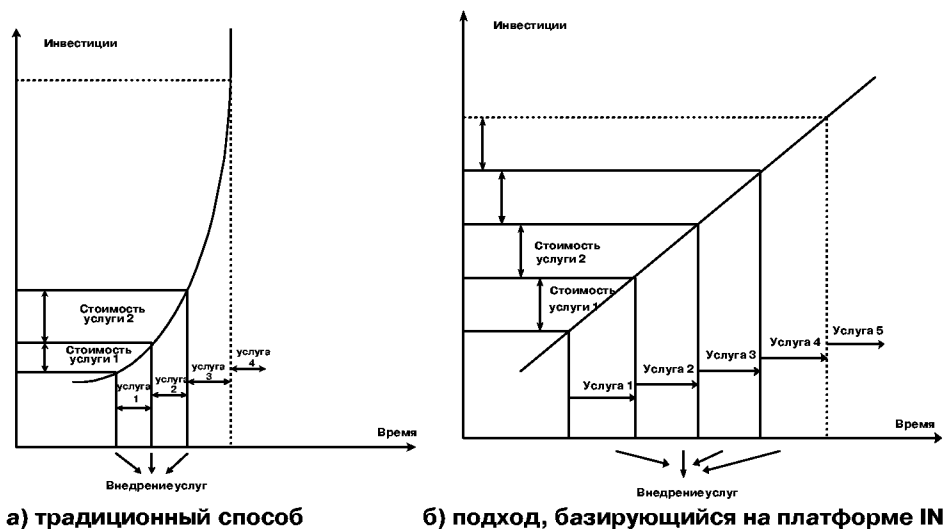


Рис. 5.7.4 Сравнение инвестиций, необходимых для внедрения услуг

Общие аспекты эволюции технологии. На рисунке 5.7.5 изображен цикл жизни услуги при традиционном способе ее предоставления до внедрения Интеллектуальной сети и при наличии интеллектуальной платформы.

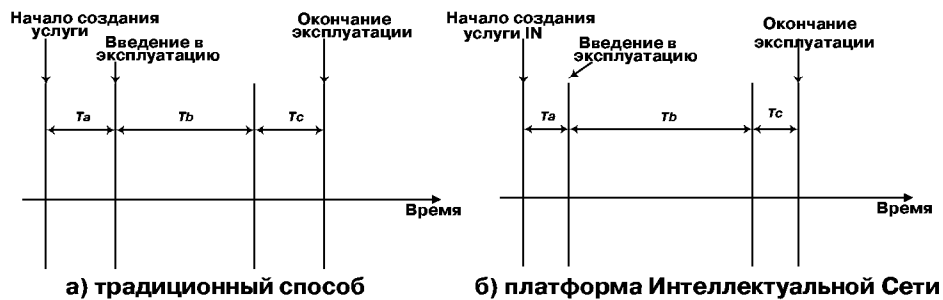


Рис. 5.7.5 Сравнение жизненного цикла услуг при двух способах внедрения

В Интеллектуальной сети величина T_a мало зависит от размера сети (см. рис. 5.7.6), то есть от того, сколько в ней коммутационных станций. Это обусловлено тем, что логика новой услуги Интеллектуальной сети должна быть введена только в SCP, который обслуживает всю сеть

(или значительную часть крупной сети), и не затрагивает программное обеспечение остальных ATC с функциями SSP.

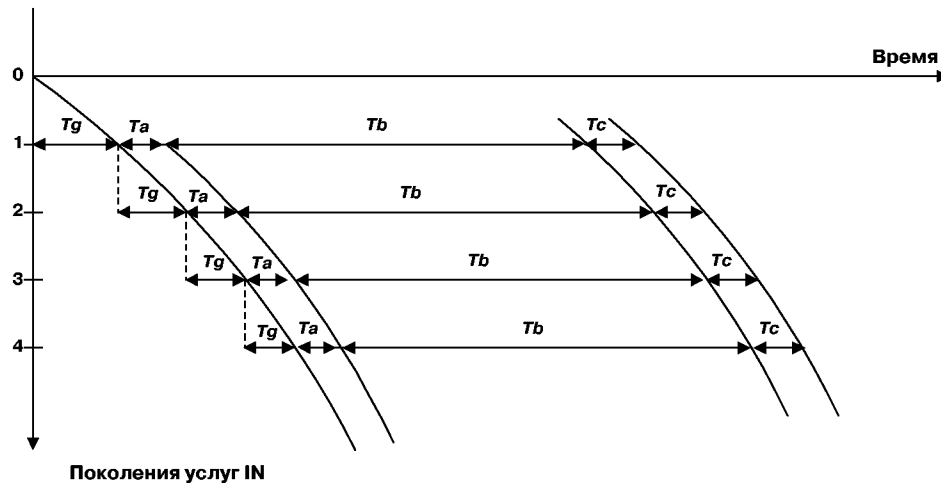


Рис. 5.7.6 Эволюция поколений платформ IN

Кроме того, по сравнению с традиционными способами предоставления услуг, в Интеллектуальной сети снижается T_c – время, в течение которого находится в эксплуатации услуга, больше не удовлетворяющая потребности оператора сети или пользователя. Это обусловлено тем, что процесс вывода услуги IN из эксплуатации тоже гораздо проще, чем процесс вывода из эксплуатации услуги, реализованной традиционным способом.

Отметим, что значение T_b – интервал времени, в течение которого услуга удовлетворяет оператора, – в случае IN может быть гораздо меньше. Однако отношение T_b к T_a и к T_c всегда больше, чем при традиционном способе реализации услуг.

За счет снижения интервала T_a , при одном и том же фиксированном его значении, допустимый размер системы, который для платформы IN может быть оценен, прежде всего, как число одновременно доступных услуг, может быть значительно увеличен прежде, чем значение T_a сравняется с T_b . Кроме того, существенно снижается риск того, что интервал T_g , уменьшаясь, достигнет T_a .

Таким образом, можно сделать следующие выводы, справедливые не только для данной главы, но и для книги в целом.

1. Реализация услуги в рамках концепции IN увеличивает общее время эффективной эксплуатации услуги за счет уменьшения времени, затрачиваемого на ее создание и развертывание.
2. Общее число одновременно находящихся в эксплуатации услуг для одного поколения коммутационного оборудования при наличии платформы IN увеличивается.

3. Инвестиции, требующиеся для введения каждой новой услуги, в случае использования подхода **IN** меньше, чем при традиционном способе реализации услуг.

Эти простые выводы (как и весь предыдущий материал книги в целом) соответствуют сформулированному А. Эйнштейном принципу: «Все должно быть изложено так просто, как только возможно, но не проще». Именно этот принцип был принят авторами в качестве базового при написании книги, а насколько это им удалось – судить читателю.

Приложения

Приложение 1

Действующие рекомендации ITU-T по Интеллектуальной сети

Приложение 1 содержит перечень рекомендаций ITU-T по Интеллектуальной сети.

Таблица П1.1 содержит перечень действующих рекомендаций, определяющих основу для определения наборов возможностей (CS) архитектурной концепции интеллектуальной сети.

Таблица П1.2 содержит перечень действующих рекомендаций, определяющих CS-1 архитектурной концепции интеллектуальной сети.

Таблица П1.3 содержит перечень действующих рекомендаций, определяющих CS-2 архитектурной концепции интеллектуальной сети.

Таблица П1.1

Рекомендация	Название
Q.1200	Структура рекомендаций по Интеллектуальной сети серии Q.
Q.1201	Принципы архитектуры Интеллектуальной сети.
Q.1202	Архитектура плоскости услуг Интеллектуальной сети.
Q.1203	Архитектура глобальной функциональной плоскости Интеллектуальной сети.
Q.1204	Архитектура распределенной функциональной плоскости Интеллектуальной сети.
Q.1205	Архитектура физической плоскости Интеллектуальной сети.
Q.1208	Основные аспекты прикладного протокола Интеллектуальной сети.
Q.1209	Словарь терминов, используемых при определении Интеллектуальной сети.

Таблица П1.2

Рекомендация	Название
Q.1210	Структура рекомендаций по Интеллектуальной сети серии Q.121x.
Q.1211	Введение в CS-1 Интеллектуальной сети.
Q.1213	Архитектура глобальной функциональной плоскости для CS-1 Интеллектуальной сети.
Q.1214	Архитектура распределенной функциональной плоскости для CS-1 Интеллектуальной сети.
Q.1215	Архитектура физической плоскости для CS-1 Интеллектуальной сети.
Q.1218	Интерфейсы Интеллектуальной сети для CS-1.
Q.1219	Руководство пользователя по CS-1 Интеллектуальной сети.

Таблица П1.2

Рекомендация	Название
Q.1220	Структура рекомендаций по Интеллектуальной сети серии Q.122х.
Q.1221	Введение в CS-2 Интеллектуальной сети.
Q.1223	Архитектура глобальной функциональной плоскости для CS-2 Интеллектуальной сети.
Q.1224	Архитектура распределенной функциональной плоскости для CS-2 Интеллектуальной сети.
Q.1225	Архитектура физической плоскости для CS-2 Интеллектуальной сети.
Q.1228	Интерфейсы Интеллектуальной сети для CS-2.
Q.1229	Руководство пользователя по CS-2 Интеллектуальной сети.

Приложение 2

Расширенный код ASN.1 протокола ETSI INAP CS-1

initialDP OPERATION

ARGUMENT

```
SEQUENCE {
  serviceKey [0] IMPLICIT INTEGER (0..2147483647),
  calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  callingPartysCategory [5] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
  cGEncountered [7] IMPLICIT ENUMERATED {
    manualCGencountered (1),
    scpOverload (2)} OPTIONAL,
  iPSSPCapabilities [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  iPAvailable [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  locationNumber [10] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  originalCalledPartyID [12] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  extensions [15] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
      type INTEGER,
      criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore
      value [1] ANY DEFINED BY type } OPTIONAL,
  highLayerCompatibility [23] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
  serviceInteractionIndicators [24] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  additionalCallingPartyNumber [25] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  forwardCallIndicators [26] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
  bearerCapability [27] CHOICE {
    bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
  eventTypeBCSM [28] IMPLICIT ENUMERATED {
    origAttemptAuthorized (1),
    collectedInfo (2),
    analyzedInformation (3),
    routeSelectFailure (4),
    oCalledPartyBusy (5),
    oNoAnswer (6),
    oAnswer (7),
    oMidCall (8),
    oDisconnect (9),
    oAbandon (10),
    termAttemptAuthorized (12),
    tCalledPartyBusy (13),
    tNoAnswer (14),
    tAnswer (15),
    tMidCall (16),
    tDisconnect (17),
    tAbandon (18)} OPTIONAL,
```

redirectingPartyID [29] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [30] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL}

ERRORS {

- missingCustomerRecord – localValue 6,
- missingParameter – localValue 7,
- systemFailure – localValue 11,
- taskRefused – localValue 12,
- unexpectedComponentSequence – localValue 14,
- unexpectedDataValue – localValue 15,
- unexpectedParameter – localValue 16}

::= localValue 0

assistRequestInstructions OPERATION

ARGUMENT

SEQUENCE {

correlationID [0] IMPLICIT OCTET STRING (SIZE (??..??)),
iPAvailable [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iPSSPCapabilities [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [3] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore
value [1] ANY DEFINED BY type } OPTIONAL}

ERRORS {

- missingCustomerRecord – localValue 6,
- missingParameter – localValue 7,
- taskRefused – localValue 12,
- unexpectedComponentSequence – localValue 14,
- unexpectedDataValue – localValue 15,
- unexpectedParameter – localValue 16}

::= localValue 16

establishTemporaryConnection OPERATION

ARGUMENT

SEQUENCE {

assistingSSPIPRoutingAddress [0] IMPLICIT OCTET STRING (SIZE (??..??)),
correlationID [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
scfID [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore
value [1] ANY DEFINED BY type } OPTIONAL,
serviceInteractionIndicators [30] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}

ERRORS {

- eTCFailed – localValue 3,
- missingParameter – localValue 7,

```

    – systemFailure – localValue 11,
    – taskRefused – localValue 12,
    – unexpectedComponentSequence – localValue 14,
    – unexpectedDataValue – localValue 15,
    – unexpectedParameter – localValue 16}
 ::= localValue 17

```

disconnectForwardConnection OPERATION

```

ERRORS {
    – systemFailure – localValue 11,
    – taskRefused – localValue 12,
    – unexpectedComponentSequence – localValue 14}
 ::= localValue 18

```

connectToResource OPERATION

```

ARGUMENT
    SEQUENCE {
        resourceAddress CHOICE {
            ipRoutingAddress [0] IMPLICIT OCTET STRING (SIZE (??..??)),
            none [3] IMPLICIT NULL},
        extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF
            SEQUENCE {
                type INTEGER,
                criticality ENUMERATED {
                    ignore (0),
                    abort (1)} DEFAULT ignore
                value [1] ANY DEFINED BY type } OPTIONAL,
        serviceInteractionIndicators [30] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}
ERRORS {
    – missingParameter – localValue 7,
    – systemFailure – localValue 11,
    – taskRefused – localValue 12,
    – unexpectedComponentSequence – localValue 14,
    – unexpectedDataValue – localValue 15,
    – unexpectedParameter – localValue 16}
 ::= localValue 19

```

connect OPERATION

```

ARGUMENT
    SEQUENCE {
        destinationRoutingAddress [0] IMPLICIT SEQUENCE SIZE (1) OF
            OCTET STRING (SIZE (??..??)),
        alertingPattern [1] IMPLICIT OCTET STRING (SIZE (3)) OPTIONAL,
        correlationID [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        cutAndPaste [3] IMPLICIT INTEGER (0..22) OPTIONAL,
        originalCalledPartyID [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        routeList [7] IMPLICIT SEQUENCE SIZE (1..3) OF
            OCTET STRING (SIZE (??..??)) OPTIONAL,
        scfID [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        extensions [10] IMPLICIT SEQUENCE SIZE (1..??) OF
            SEQUENCE {

```

```

    type    INTEGER,
    criticality ENUMERATED {
        ignore    (0),
        abort     (1)} DEFAULT ignore
    value    [1] ANY DEFINED BY type    } OPTIONAL,
    serviceInteractionIndicators [26] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    callingPartyNumber [27] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    callingPartysCategory [28] IMPLICIT OCTET STRING (SIZE (1) ) OPTIONAL,
    redirectingPartyID [29] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    redirectionInformation [30] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL}
ERRORS {
    – missingParameter – localValue 7,
    – systemFailure – localValue 11,
    – taskRefused – localValue 12,
    – unexpectedComponentSequence – localValue 14,
    – unexpectedDataValue – localValue 15,
    – unexpectedParameter – localValue 16}
::= localValue 20

```

releaseCall OPERATION

```

ARGUMENT
    OCTET STRING (SIZE (2..??))
::= localValue 22

```

requestReportBCSMEvent OPERATION

```

ARGUMENT
    SEQUENCE {
        bcsmEvents [0] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {
            eventTypeBCSM [0] IMPLICIT ENUMERATED {
                origAttemptAuthorized (1),
                collectedInfo (2),
                analyzedInformation (3),
                routeSelectFailure (4),
                oCalledPartyBusy (5),
                oNoAnswer (6),
                oAnswer (7),
                oMidCall (8),
                oDisconnect (9),
                oAbandon (10),
                termAttemptAuthorized (12),
                tCalledPartyBusy (13),
                tNoAnswer (14),
                tAnswer (15),
                tMidCall (16),
                tDisconnect (17),
                tAbandon (18)},
            monitorMode [1] IMPLICIT ENUMERATED {
                interrupted (0),
                notifyAndContinue (1),
                transparent (2)},

```



```

legID    [2] CHOICE {
    sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
    receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1)) } OPTIONAL,
dPSpecificCriteria [30] CHOICE {
    numberOfDigits [0] IMPLICIT INTEGER (1..255),
    applicationTimer [1] IMPLICIT INTEGER (0..2047) } OPTIONAL,
extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore
    value [1] ANY DEFINED BY type } OPTIONAL}
ERRORS {
    – missingParameter – localValue 7,
    – systemFailure – localValue 11,
    – taskRefused – localValue 12,
    – unexpectedComponentSequence – localValue 14,
    – unexpectedDataValue – localValue 15,
    – unexpectedParameter – localValue 16}
 ::= localValue 23

```

eventReportBCSM OPERATION

ARGUMENT

```

SEQUENCE {
    eventTypeBCSM [0] IMPLICIT ENUMERATED {
        origAttemptAuthorized (1),
        collectedInfo (2),
        analyzedInformation (3),
        routeSelectFailure (4),
        oCalledPartyBusy (5),
        oNoAnswer (6),
        oAnswer (7),
        oMidCall (8),
        oDisconnect (9),
        oAbandon (10),
        termAttemptAuthorized (12),
        tCalledPartyBusy (13),
        tNoAnswer (14),
        tAnswer (15),
        tMidCall (16),
        tDisconnect (17),
        tAbandon (18)},
    eventSpecificInformationBCSM [2] CHOICE {
        collectedInfoSpecificInfo [0] IMPLICIT SEQUENCE {
            calledPartyNumber [0] IMPLICIT OCTET STRING (SIZE (??..??)),
            analyzedInfoSpecificInfo [1] IMPLICIT SEQUENCE {
                calledPartyNumber [0] IMPLICIT OCTET STRING (SIZE (??..??)),
                routeSelectFailureSpecificInfo [2] IMPLICIT SEQUENCE {
                    failureCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
                    oCalledPartyBusySpecificInfo [3] IMPLICIT SEQUENCE {

```

```

    busyCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
    oNoAnswerSpecifidnfo [4] IMPLICIT SEQUENCE {},
    oAnswerSpecifidnfo [5] IMPLICIT SEQUENCE {},
    oMidCallSpecifidnfo [6] IMPLICIT SEQUENCE {},
    oDisconnectSpecifidnfo [7] IMPLICIT SEQUENCE {
        releaseCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
    tCalledPartyBusySpecifidnfo [8] IMPLICIT SEQUENCE {
        busyCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
    tNoAnswerSpecifidnfo [9] IMPLICIT SEQUENCE {},
    tAnswerSpecifidnfo [10] IMPLICIT SEQUENCE {},
    tMidCallSpecifidnfo [11] IMPLICIT SEQUENCE {},
    tDisconnectSpecifidnfo [12] IMPLICIT SEQUENCE {
        releaseCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL}} OPTIONAL,
    legID [3] CHOICE {
        sendingSidelD [0] IMPLICIT OCTET STRING (SIZE (1)),
        receivingSidelD [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
    miscCallInfo [4] IMPLICIT SEQUENCE {
        messageType [0] IMPLICIT ENUMERATED {
            request (0),
            notification (1)}}, DEFAULT {
        messageType request },
    extensions [5] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
            ignore (0),
            abort (1)}}, DEFAULT ignore
        value [1] ANY DEFINED BY type } OPTIONAL}
::= localValue 24

```

requestNotificationChargingEvent OPERATION

ARGUMENT

SEQUENCE SIZE (1..??) OF

```

SEQUENCE {
    eventTypeCharging [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    monitorMode [1] IMPLICIT ENUMERATED {
        interrupted (0),
        notifyAndContinue (1),
        transparent (2)},
    legID [2] CHOICE {
        sendingSidelD [0] IMPLICIT OCTET STRING (SIZE (1)),
        receivingSidelD [1] IMPLICIT OCTET STRING (SIZE (1)) } OPTIONAL}

```

ERRORS {

- missingParameter – localValue 7,
- systemFailure – localValue 11,
- taskRefused – localValue 12,
- unexpectedComponentSequence – localValue 14,
- unexpectedDataValue – localValue 15,
- unexpectedParameter – localValue 16}

::= localValue 25

eventNotificationCharging OPERATION**ARGUMENT**

```

SEQUENCE {
    eventTypeCharging [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    eventSpecifidnformationCharging [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    legID [2] CHOICE {
        sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
        receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))) OPTIONAL,
    extensions [3] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {
            type INTEGER,
            criticality ENUMERATED {
                ignore (0),
                abort (1)} DEFAULT ignore
            value [1] ANY DEFINED BY type } OPTIONAL,
    monitorMode [30] IMPLICIT ENUMERATED {
        interrupted (0),
        notifyAndContinue (1),
        transparent (2)} DEFAULT notifyAndContinue }
::= localValue 2 6

```

collectInformation OPERATION**ARGUMENT**

```

SEQUENCE {
    extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {
            type INTEGER,
            criticality ENUMERATED {
                ignore (0),
                abort (1)} DEFAULT ignore
            value [1] ANY DEFINED BY type } OPTIONAL}
ERRORS {
    – missingParameter – localValue 7,
    – systemFailure – localValue 11,
    – taskRefused – localValue 12,
    – unexpectedComponentSequence – localValue 14,
    – unexpectedDataValue – localValue 15,
    – unexpectedParameter – localValue 16}
::= localValue 27

```

continue OPERATION

```

::= localValue 31

```

initiateCallAttempt OPERATION**ARGUMENT**

```

SEQUENCE {
    destinationRoutingAddress [0] IMPLICIT SEQUENCE SIZE (1) OF
        OCTET STRING (SIZE (??..??)),
    alertingPattern [1] IMPLICIT OCTET STRING (SIZE (3)) OPTIONAL,
    extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {

```

```

    type    INTEGER,
    criticality ENUMERATED {
        ignore    (0),
        abort     (1)} DEFAULT ignore
    value    [1] ANY DEFINED BY type    } OPTIONAL,
    serviceInteractionIndicators [29] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    callingPartyNumber [30] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL)

```

ERRORS {

- missingParameter – localValue 7,
- systemFailure – localValue 11,
- taskRefused – localValue 12,
- unexpectedComponentSequence – localValue 14,
- unexpectedDataValue – localValue 15,
- unexpectedParameter – localValue 16}

::= localValue 32

resetTimer OPERATION

ARGUMENT

```

SEQUENCE {
    timerID [0] IMPLICIT ENUMERATED {
        tssf    (0)} DEFAULT tssf
    timervalue [1] IMPLICIT INTEGER (0..2147483647) ,
    extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {
            type    INTEGER,
            criticality ENUMERATED {
                ignore    (0),
                abort     (1)} DEFAULT ignore
            value    [1] ANY DEFINED BY type    } OPTIONAL}

```

ERRORS {

- missingParameter – localValue 7,
- taskRefused – localValue 12,
- unexpectedComponentSequence – localValue 14,
- unexpectedDataValue – localValue 15,
- unexpectedParameter – localValue 16}

::= localValue 33

furnishchargingInformation OPERATION

ARGUMENT

```

OCTET STRING (SIZE (??..??))

```

ERRORS {

- missingParameter – localValue 7,
- taskRefused – localValue 12,
- unexpectedComponentSequence – localValue 14,
- unexpectedDataValue – localValue 15,
- unexpectedParameter – localValue 16}

::= localValue 34

applyCharging OPERATION

ARGUMENT

```

SEQUENCE {

```

```

aChBillingChargingCharacteristics [0] IMPLICIT OCTET STRING (SIZE (??..??)),
sendCalculationToSCPIndication [1] IMPLICIT BOOLEAN DEFAULT FALSE,
partyToCharge [2] CHOICE {
    sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
    receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
extensions [3] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore
        value [1] ANY DEFINED BY type } OPTIONAL}
ERRORS {
    - missingParameter - localValue 7,
    - unexpectedComponentSequence - localValue 14,
    - unexpectedParameter - localValue 16,
    - unexpectedDataValue - localValue 15,
    - parameterOutOfRange - localValue 8,
    - systemFailure - localValue 11,
    - taskRefused - localValue 12}
 ::= localValue 35

```

applyChargingReport OPERATION**ARGUMENT**

```
OCTET STRING (SIZE (??..??))
```

ERRORS {

```

    - missingParameter - localValue 7,
    - unexpectedComponentSequence - localValue 14,
    - unexpectedParameter - localValue 16,
    - unexpectedDataValue - localValue 15,
    - parameterOutOfRange - localValue 8,
    - systemFailure - localValue 11,
    - taskRefused - localValue 12}

```

```
::= localValue 3 6
```

callGap OPERATION**ARGUMENT****SEQUENCE {**

```

gapCriteria [0] CHOICE {
    calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    gapOnService [2] IMPLICIT SEQUENCE {
        serviceKey [0] IMPLICIT INTEGER (0..2147483647),
    calledAddressAndService [29] IMPLICIT SEQUENCE {
        calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        serviceKey [1] IMPLICIT INTEGER (0..2147483647),
    callingAddressAndService [30] IMPLICIT SEQUENCE {
        callingAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        serviceKey [1] IMPLICIT INTEGER (0..2147483647),
        locationNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
    gapIndicators [1] IMPLICIT SEQUENCE {
        duration [0] IMPLICIT INTEGER (-2..86400),

```



```

        numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
        duration [2] IMPLICIT INTEGER (0.. 32767) OPTIONAL,
        interval [3] IMPLICIT INTEGER (0.. 32767) OPTIONAL,
        tone [1] IMPLICIT SEQUENCE {
            toneID [0] IMPLICIT INTEGER (0..2147483647),
            duration [1] IMPLICIT INTEGER (0.. 2147483647) OPTIONAL,
            displayInformation [2] IMPLICIT IA5String (SIZE (??..??)),
            releaseCause [1] IMPLICIT OCTET STRING (SIZE (2..??))} OPTIONAL,
        extensions [4] IMPLICIT SEQUENCE SIZE (1..?) OF
        SEQUENCE {
            type INTEGER,
            criticality ENUMERATED {
                ignore (0) ,
                abort (1)} DEFAULT ignore
            value [1] ANY DEFINED BY type } OPTIONAL}
 ::= locaValue 41

```

activateServiceFiltering OPERATION

ARGUMENT

```

SEQUENCE {
    filteredCallTreatment [0] IMPLICIT SEQUENCE {
        sFBillingChargingCharacteristics [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        informationToSend [1] CHOICE {
            inbandinfo [0] IMPLICIT SEQUENCE {
                messageID [0] CHOICE {
                    elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
                    text [1] IMPLICIT SEQUENCE {
                        messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
                        attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
                        elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..?) OF
                        INTEGER (0..2147483647),
                        variableMessage [30] IMPLICIT SEQUENCE {
                            elementaryMessageID [0] IMPLICIT INTEGER (0.. 2147483647) ,
                            variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
                            CHOICE {
                                integer [0] IMPLICIT INTEGER (0..2147483647),
                                number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
                                time [2] IMPLICIT OCTET STRING (SIZE (2)),
                                date [3] IMPLICIT OCTET STRING (SIZE (3)),
                                price [4] IMPLICIT OCTET STRING (SIZE (4))}},
                        numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
                        duration [2] IMPLICIT INTEGER (0.. 32767) OPTIONAL,
                        interval [3] IMPLICIT INTEGER (0.. 32767) OPTIONAL,
                        tone [1] IMPLICIT SEQUENCE {
                            toneID [0] IMPLICIT INTEGER (0.. 2147483647) ,
                            duration [1] IMPLICIT INTEGER (0.. 2147483647) OPTIONAL,
                            displayInformation [2] IMPLICIT IA5String (SIZE (??..??)) OPTIONAL,
                            maximumNumberOfCounters [2] IMPLICIT INTEGER (1..100) OPTIONAL,
                            releaseCause [3] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
                            filteringCharacteristics [1] CHOICE {

```

```

interval [0] IMPLICIT INTEGER (-1.. 32000),
numberOfCalls [1] IMPLICIT INTEGER (0..2147483647)),
filteringTimeOut [2] CHOICE {
    duration [0] IMPLICIT INTEGER (-2..86400),
    stopTime [1] IMPLICIT OCTET STRING (SIZE (6))),
filteringCriteria [3] CHOICE {
    serviceKey [2] IMPLICIT INTEGER (0..2147483647) ,
    addressAndService [30] IMPLICIT SEQUENCE {
        calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        serviceKey [1] IMPLICIT INTEGER (0..2147483647) ,
        callingAddressValue [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        locationNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}},
startTime [4] IMPLICIT OCTET STRING (SIZE (6)) OPTIONAL,
extensions [5] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore
        value [1] ANY DEFINED BY type } OPTIONAL}
ERRORS {
    - missingParameter - localValue 7,
    - parameterOutOfRange - localValue 8,
    - systemFailure - localValue 11,
    - taskRefused - localValue 12,
    - unexpectedComponentSequence - localValue 14,
    - unexpectedParameter - localValue 16}
::= localValue 42

```

serviceFilteringResponse OPERATION

ARGUMENT

```

SEQUENCE {
    countersValue [0] IMPLICIT SEQUENCE SIZE (0..100) OF
        SEQUENCE {
            counterID [0] IMPLICIT INTEGER (0..99),
            counterValue [1] IMPLICIT INTEGER (0..2147483647)),
    filteringCriteria [1] CHOICE {
        serviceKey [2] IMPLICIT INTEGER (0..2147483647) ,
        addressAndService [30] IMPLICIT SEQUENCE {
            calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
            serviceKey [1] IMPLICIT INTEGER (0..2147483647) ,
            callingAddressValue [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
            locationNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}},
    extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {
            type INTEGER,
            criticality ENUMERATED {
                ignore (0),
                abort (1)} DEFAULT ignore
            value [1] ANY DEFINED BY type } OPTIONAL}
::= localValue 43

```


callInformationReport OPERATION**ARGUMENT**

```

SEQUENCE {
  requestedInformationList [0] IMPLICIT SEQUENCE SIZE (1..5) OF
    SEQUENCE {
      requestedInformationType [0] IMPLICIT ENUMERATED {
        callAttemptElapsedTime (0),
        callStopTime (1),
        callConnectedElapsedTime (2),
        calledAddress (3),
        releaseCause (30)},
      requestedInformationValue [1] CHOICE {
        callAttemptElapsedTimeValue [0] IMPLICIT INTEGER (0..255),
        callStopTimeValue [1] IMPLICIT OCTET STRING (SIZE (6)),
        callConnectedElapsedTimeValue [2] IMPLICIT INTEGER (0..2147483647),
        calledAddressValue [3] IMPLICIT OCTET STRING (SIZE (??..??)),
        releaseCauseValue [30] IMPLICIT OCTET STRING (SIZE (2..??))}},
  extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
      type INTEGER,
      criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore
      value [1] ANY DEFINED BY type } OPTIONAL}
 ::= localValue 44

```

callInformationRequest OPERATION**ARGUMENT**

```

SEQUENCE {
  requestedInformationTypeList [0] IMPLICIT SEQUENCE SIZE (1..5) OF
    ENUMERATED {
      callAttemptElapsedTime (0),
      callStopTime (1),
      callConnectedElapsedTime (2),
      calledAddress (3),
      releaseCause (30)},
  extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
      type INTEGER,
      criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore
      value [1] ANY DEFINED BY type } OPTIONAL}
ERRORS {
  – missingParameter – localValue 7,
  – parameterOutOfRange – localValue 8,
  – requestedInfoError – localValue 10,
  – systemFailure – localValue 11,
  – taskRefused – localValue 12,
  – unexpectedComponentSequence – localValue 14,
  – unexpectedParameter – localValue 16}
 ::= localValue 45

```

sendchargingInformation OPERATION**ARGUMENT**

```

SEQUENCE {
  sCBillingChargingCharacteristics [0] IMPLICIT OCTET STRING (SIZE (??..??)),
  legID [1] CHOICE {
    sendingSidelD [0] IMPLICIT OCTET STRING (SIZE (1)),
    receivingSidelD [1] IMPLICIT OCTET STRING (SIZE (1))},
  extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
      type INTEGER,
      criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore
      value [1] ANY DEFINED BY type } OPTIONAL}

```

ERRORS {

- missingParameter – localValue 7,
- unexpectedComponentSequence – localValue 14,
- unexpectedParameter – localValue 16,
- parameterOutOfRange – localValue 8,
- systemFailure – localValue 11,
- taskRefused – localValue 12,
- unknownLegID – localValue 17}

::= localValue 46

playAnnouncement OPERATION**ARGUMENT**

```

SEQUENCE {
  informationToSend [0] CHOICE {
    inbandinfo [0] IMPLICIT SEQUENCE {
      messageID [0] CHOICE {
        elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
        text [1] IMPLICIT SEQUENCE {
          messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
          attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
        elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF
          INTEGER (0..2147483647),
        variableMessage [30] IMPLICIT SEQUENCE {
          elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
          variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
            CHOICE {
              integer [0] IMPLICIT INTEGER (0..2147483647),
              number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
              time [2] IMPLICIT OCTET STRING (SIZE (2)),
              date [3] IMPLICIT OCTET STRING (SIZE (3)),
              price [4] IMPLICIT OCTET STRING (SIZE (4))}},
        numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
        duration [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
        interval [3] IMPLICIT INTEGER (0..32767) OPTIONAL},
    tone [1] IMPLICIT SEQUENCE {
      toneID [0] IMPLICIT INTEGER (0..2147483647),

```

```

        duration [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
        displayInformation [2] IMPLICIT IA5String (SIZE (??..??)),
        disconnectFromIPEorbidden [1] IMPLICIT BOOLEAN DEFAULT TRUE,
        requestAnnouncementComplete [2] IMPLICIT BOOLEAN DEFAULT TRUE,
        extensions [3] IMPLICIT SEQUENCE SIZE (1..?) OF
        SEQUENCE {
            type INTEGER,
            criticality ENUMERATED {
                ignore (0),
                abort (1)} DEFAULT ignore
            value [1] ANY DEFINED BY type } OPTIONAL}
    ERRORS {
        - cancelled – localValue 0,
        - missingParameter – localValue 7,
        - systemFailure – localValue 11,
        - unavailableResource – localValue 13,
        - unexpectedComponentSequence – localValue 14,
        - unexpectedDataValue – localValue 15,
        - unexpectedParameter – localValue 16}
    LINKED {
        - specializedResourceReport – localValue 49}
    ::= localValue 47

```

promptAndCollectUserInfo OPERATION

ARGUMENT

```

    SEQUENCE {
        collectedInfo [0] CHOICE {
            collectedDigits [0] IMPLICIT SEQUENCE {
                minimumNbOfDigits [0] IMPLICIT INTEGER (1..127) DEFAULT 1,
                maximumNbOfDigits [1] IMPLICIT INTEGER (1..127),
                endOfReplyDigit [2] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
                cancelDigit [3] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
                startDigit [4] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
                firstDigitTimeOut [5] IMPLICIT INTEGER (1..127) OPTIONAL,
                interDigitTimeOut [6] IMPLICIT INTEGER (1..127) OPTIONAL,
                errortreatment [7] IMPLICIT ENUMERATED {
                    stdErrorAndInfo (0),
                    help (1),
                    repeatPrompt (2)} DEFAULT StdErrorAndInfo ,
                interruptableAnnInd [8] IMPLICIT BOOLEAN DEFAULT TRUE,
                voiceInformation [9] IMPLICIT BOOLEAN DEFAULT FALSE,
                voiceBack [10] IMPLICIT BOOLEAN DEFAULT FALSE},
            disconnectFromIPForbidden [1] IMPLICIT BOOLEAN DEFAULT TRUE,
            informationToSend [2] CHOICE {
                inbandinfo [0] IMPLICIT SEQUENCE {
                    messageID [0] CHOICE {
                        elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647) ,
                        text [1] IMPLICIT SEQUENCE {
                            messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
                            attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
                        elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..?) OF

```

```

    INTEGER (0..2147483647),
    variableMessage [30] IMPLICIT SEQUENCE {
        elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647) ,
        variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
        CHOICE {
            integer [0] IMPLICIT INTEGER (0..2147483647) ,
            number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
            time [2] IMPLICIT OCTET STRING (SIZE (2)),
            date [3] IMPLICIT OCTET STRING (SIZE (3)),
            price [4] IMPLICIT OCTET STRING (SIZE (4))}},
        numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
        duration [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
        interval [3] IMPLICIT INTEGER (0..32767) OPTIONAL,
        tone [1] IMPLICIT SEQUENCE {
            toneID [0] IMPLICIT INTEGER (0..2147483647),
            duration [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
            displayInformation [2] IMPLICIT IA5String (SIZE (??..??)) OPTIONAL,
            extensions [3] IMPLICIT SEQUENCE SIZE (1..??) OF
            SEQUENCE {
                type INTEGER,
                criticality ENUMERATED {
                    ignore (0),
                    abort (1)} DEFAULT ignore
                value [1] ANY DEFINED BY type } OPTIONAL}
        }
    }
RESULT
    CHOICE {
        digitsResponse [0] IMPLICIT OCTET STRING (SIZE (??..??))
    }
ERRORS {
    – cancelled – localValue 0,
    – improperCallerResponse – localValue 4,
    – missingParameter – localValue 7,
    – systemFailure – localValue 11,
    – taskRefused – localValue 12,
    – unavailableResource – localValue 13,
    – unexpectedComponentSequence – localValue 14,
    – unexpectedDataValue – localValue 15,
    – unexpectedParameter – localValue 16}
::= localValue 48

```

specializedResourceReport OPERATION

ARGUMENT

NULL

::= localValue 49

cancel OPERATION

ARGUMENT

CHOICE {

 invokeID [0] IMPLICIT INTEGER (-128..127),

 allRequests [1] IMPLICIT NULL}

ERRORS {

 – cancelFailed – localValue 1}

::= localValue 53

activityTest OPERATION

RESULT

NULL

::= localValue 55

cancelled ERROR

::= localValue 0

cancelFailed ERROR

PARAMETER

SEQUENCE {

problem [0] IMPLICIT ENUMERATED {

unknownOperation (0),

tooLate (1),

operationNotCancellable (2)},

operation [1] IMPLICIT INTEGER (-128..127)}

::= localValue 1

eTCFailed ERROR

::= localValue 3

improperCallerResponse ERROR

::= localValue 4

missingCustomerRecord ERROR

::= localValue 6

missingParameter ERROR

::= localValue 7

parameterOutOfRange ERROR

::= localValue 8

requestedInfoError ERROR

PARAMETER

ENUMERATED {

unknownRequestedInfo (1),

requestedInfoNotAvailable (2)}

::= localValue 10

systemFailure ERROR

PARAMETER

ENUMERATED {

unavailableResources (0),

componentEailure (1),

basicCallProcessingException (2),

resourceStatusFailure (3),

endUserEailure (4)}

::= localValue 11

taskRefused ERROR

PARAMETER

```
ENUMERATED {  
    generic    (0) ,  
    unobtainable (1) ,  
    congestion  (2)}
```

::= localValue 12

unavailableResource ERROR

::= localValue 13

unexpectedComponentSequence ERROR

::= localValue 14

unexpectedDataValue ERROR

::= localValue 15

unexpectedParameter ERROR

::= localValue 16

unknownLegID ERROR

::= localValue 17

Список сокращений

АОН	автоматическое определение номера	
АМТС	автоматическая междугородная телефонная станция	
АТС	автоматическая телефонная станция	
ВСК	выделенный сигнальный канал	
ЗСЛ	заказно-соединительная линия	
ИКМ	импульсно-кодовая модуляция	
ОКС-7	система общеканальной сигнализации №7	
СЛ	соединительная линия	
СЛМ	соединительная линия междугородная	
СПС	сеть подвижной связи	
ТФОП	телефонная сеть общего пользования	
УСС	узел специальных служб	
AC	(Application context)	прикладной контекст
ACC	(Account card calling)	вызов с начислением платы на счет, соответствующий используемой карте
ACTS	(Advanced communications technologies and services)	паневропейская программа "Продвинутые технологии и услуги связи"
AD	(Adjunct)	вспомогательный узел управления услугами
AE	(Application entity)	прикладной объект
ANSI	(American National Standards Institute)	Американский Национальный Институт Стандартов
AP	(Application Process)	прикладной процесс
ASE	(Application Service Element)	прикладной сервисный элемент
ASN.1	(Abstract syntax notation one)	язык абстрактного описания
ATM	(Asynchronous transfer mode)	асинхронный режим переноса
B-ISDN	(Broadband integrated services digital network)	широкополосная сеть интегрального обслуживания
BCP	(Basic call process)	базовый процесс обслуживания вызова
BCSM	(Basic call state model)	модель состояний базового процесса обслуживания вызова
BER	(Basic encoding rules)	базовые правила кодирования
BRI	(ISDN basic rate interface)	(ISDN) интерфейс на базовой скорости
CCAF	(Call control agent function)	функциональный объект поддержки доступа
CCC	(Credit card calling)	вызов с оплатой по кредитной карте
CCF	(Call control function)	функциональный объект управления связью пользователя
CID	(Call instance data)	данные, зависящие от вызова
CIDFP	(CID field pointer)	указатель поля данных CID

CORBA	(Common object request broker architecture)	общая архитектура с передачей запросов к объекту через посредника
CS	(Capability set)	набор возможностей
CUSF	(Call unrelated service functions)	функции, предназначенные для приема информации от пользователя при взаимодействиях, не связанных с вызовом
DP	(Detection point)	точка обнаружения
DPC	(Destination point code)	код пункта назначения (в сети ОКС-7)
DPE	(Distributed processing environment)	среда распределенной обработки
DTMF	(Dual-tone multifrequency)	многочастотный набор номера
EDP	(Event detection point)	точка обнаружения события
EF	(Elementary function)	элементарная функция
ETSI	(European Telecommunications Standards Institute)	Европейский институт стандартов в области связи
FE	(Functional entity)	функциональный объект
FEA	(Functional entity action)	действие функционального объекта
FPH	(Freephone)	бесплатный вызов
GFP	(Global functional plane)	глобальная функциональная плоскость
GSL	(Global service logic)	глобальная логика услуги
GSM	(Global systems for mobile communications)	глобальная система мобильной связи
GT	(Global title)	глобальный заголовок
HLR	(Home location register)	домашний (опорный) регистр местоположения
IAM	(Initial address message)	начальное адресное сообщение (ISUP, ОКС-7)
IDL	(Interface definition language)	язык описания интерфейса
IETF	(Internet engineering task force)	Инженерная проблемная группа Internet
IF	(Information flow)	информационный поток
IN	(Intelligent network)	интеллектуальная сеть
INAP	(Intelligent network application protocol)	прикладной протокол интеллектуальной сети
INCM	(Intelligent network conceptual model)	концептуальная модель интеллектуальной сети
IP	(Intelligent peripheral)	интеллектуальная периферия
IP	(Internet protocol)	протокол сети Internet
ISUP	(ISDN user part)	подсистема пользователя ISDN (в сети ОКС-7)
ITU-T	(International Telecommunications Union - Telecommunication Standardization Sector)	Международный Союз Электросвязи - Сектор стандартизации в области связи
MACF	(Multiple association control function)	функция управления множественными ассоциациями
MAS	(Mass calling)	массовые вызовы

MTP	(Message transfer part)	подсистема передачи сообщений (в сети ОКС-7)
NAP	(Network access point)	узел сетевого доступа
NE	(Network element)	сетевой элемент
NNI	(Network node interface)	межузловой интерфейс
O-BCSM	(Originating basic call state model)	исходящая часть модели базового процесса обслуживания вызова
OCS	(Originating call screening)	просмотр списка исходящих вызовов
ODP	(Open distributed processing)	открытая распределенная обработка
OMG	(Object management group)	группа управления объектами
OMT	(Object modeling technique)	техника моделирования объектов
OPC	(Originating point code)	код исходящего пункта сигнализации (в сети ОКС-7)
ORB	(Object request broker)	посредник запросов объекта
PE	(Physical entity)	физический элемент
PCN	(Personal communications network)	сеть персональной связи
PCS	(Personal communications services)	услуги персональной связи
PDU	(Protocol data unit)	блоки данных протокола
PIC	(Point in call)	состояние процесса обслуживания вызова
PIN	(Personal identification number)	персональный идентификационный номер
POI	(Point of initiation)	инициирующая точка
POR	(Point of return)	точка возврата
PRI	(ISDN primary rate interface)	(ISDN) интерфейс на первичной скорости
PRM	(Premium rate)	услуга с дополнительной оплатой
ROSE	(Remote operations service element)	сервисный элемент удаленных операций
SACF	(Single association control function)	функция управления одиночной ассоциацией
SAO	(Single association object)	объект одиночной ассоциации
SCCP	(Signalling connection control part)	подсистема управления соединениями сигнализации (в сети ОКС-7)
SCE	(Service creation environment)	среда создания услуг
SCEF	(Service creation environment function)	функциональный объект среды создания услуг
SCEP	(Service creation environment point)	узел среды создания услуг
SCF	(Service control function)	функциональный объект управления услугами
SCP	(Service control point)	узел управления услугами
SDF	(Service data function)	функциональный объект предоставления данных для услуг
SDL	(Specification and description language)	язык описаний и спецификаций
SDP	(Service data point)	узел хранения данных для услуг

SF	(Service feature)	атрибут услуги
SIB	(Service independent building block)	независимый от услуг конструктивный блок
SLEE	(Service logic execution environment)	среда выполнения логики услуг
SLP	(Service logic program)	программа логики услуги
SMAF	(Service management access function)	функциональный объект доступа к системе эксплуатационного управления услугами
SMAP	(Service management access point)	узел доступа к системе эксплуатационного управления услугами
SMF	(Service management function)	функциональный объект эксплуатационного управления услугами
SMP	(Service management point)	узел системы эксплуатационного управления услугами
SN	(Service node)	узел услуг
SRF	(Specialized resource function)	функциональный объект специализированных ресурсов
SSCP	(Service switching and control point)	узел коммутации и управления услугами
SSD	(Service support data)	данные, относящиеся к услуге
SSF	(Service switching function)	функциональный объект коммутации услуг
SSN	(Subsystem number)	номер подсистемы (в сети ОКС-7)
SSP	(Service switching point)	узел коммутации услуг
STP	(Signalling transfer point)	транзитный пункт сигнализации (в сети ОКС-7)
T-BCSM	(Terminating basic call state model)	входящая часть модели состояний базового процесса обслуживания вызова
TCAP	(Transaction capabilities application part)	прикладная подсистема возможностей транзакций (в сети ОКС-7)
TDP	(Trigger detection point)	триггерная точка обнаружения
TINA	(Telecommunications information networking architecture)	архитектура сетевого информационного обеспечения связи
TINA-C	(TINA-Consortium)	консорциум TINA
TMN	(Telecommunications management network)	сеть управления сетями связи
UMTS	(Universal mobile telecommunications system)	универсальная система подвижной связи
UNI	(User network interface)	интерфейс пользователь-сеть
UPT	(Universal personal telecommunications)	универсальная персональная связь
VLR	(Visitor location register)	визитный (гостевой) регистр местоположения
VoD	(Video on demand)	услуга видео по требованию
VoIP	(Voice over IP)	услуга передачи речи по протоколу IP
VOT	(Tele voting)	телеголосование
VPN	(Virtual private network)	виртуальная частная сеть
WIN	(Wireless intelligent network)	беспроводная интеллектуальная сеть

Литература

1. Варакин Л.Е. Интеллектуальная сеть: эволюция сетей и услуг связи / Электросвязь, 1992, №1.
2. Варакин Л.Е., Кучерявый А.Е., Соколов Н.А., Филюшин Ю.И. Интеллектуальная сеть: концепция и архитектура/Электросвязь, 1992, № 1.
3. Гольдштейн Б.С. Сигнализация в сетях связи. – М.: Радио и связь, 1999. Т. 1, 2.
4. Гольдштейн Б., Ехриель И., Рерле Р. Интеллектуальная сеть: подходы и альтернативы / Вестник связи, 1999, № 5.
5. Гольдштейн Б., Ехриель И., Рерле Р., Мороз А., Шамбазов А. Предоставление услуг IN на сети МГТС / Сети и системы связи, 1998, № 11.
6. Голышко А., Виленский М.А., Мазникер Ю.М. Интеллектуальные сети и услуги / Сети и системы связи. — 1998, № 6.
7. Ефимушкин В.А., Филюшин Ю.И. Модели взаимодействия узлов платформы интеллектуальной сети связи при обслуживании вызовов / Труды Международной Академии Связи, 1998, № 3.
8. Интеллектуальные сети: разработка стандартов и внедрение услуг: Труды Междунар. конф., 17-19 февр. 1999 г., Москва, Россия = Intelligent Networks: Services and Standards: Internat. Conf., IFIP WG.6.7 Workshop SmartNet, Febr. 17-19, 1999, Moscow, Russia.— М.: Диалог-МГУ, 1999.— 342 с.— Доклады на русском и английском языках.
9. Лазарев В.Г. Интеллектуальные цифровые сети: Справочник / Под ред. Н.А.Кузнецова. — М.: Финансы и статистика, 1996.
10. Новые информационные технологии. Интеллектуальные сети и услуги: Тез. докл. семинара-совещания, 30 - 31 марта 1999 г., г. Челябинск.— М.: 1999.
11. Прогноз рынка услуг интеллектуальных сетей / В мире телекоммуникаций, 1998, № 8.
12. Рерле Р., Баева О., Видерман А., Виницкий А. Опыт программной реализации услуги “Безусловная переадресация вызова” в рамках концепции интеллектуальной сети АТСЦ-90 / Электросвязь, 1997, № 4.
13. Росляков А.В. Общеканальная система сигнализации №7. — М.: Эко-Тренд, 1999.
14. Самуйлов К.Е., Филюшин Ю.И. Роль интеллектуальной сети в эволюции систем связи / Открытые системы, 1996, №2.
15. Соколов Н.А. Эволюция местных телефонных сетей. Пермь: ТОО «Типография «Книга»», 1994.

16. Шнепс-Шнеппе М. Интеллектуальные сети как основа конвергенции систем связи / Вестник связи, 1999, № 5.
17. Ahamed S.V., Lawrence V.B. Intelligent Broadband Multimedia Networks: Genevic Aspects and Architectures: Wireless, ISDN, Current and Future Intelligent Networks: Kluwer Academic Publishers, 1997.
18. Bellcore. SR-NPL-001555. Advanced Intelligent Network Release 1 Baseline Architecture. USA. June, 1990.
19. Black U. The Intelligent Network. Customizing Telecommunication Networks and Services: Prentice Hall PTR, 1998.— XVI.
20. ETSI. ETS 300 374-1. Intelligent Network(IN); Intelligent Network Capability Set 1 (CS1); Core Intelligent Network Application Protocol (INAP); Part 1:Protocol Specification. France. September, 1994.
21. Faynberg I. et al. The Intelligent Network Standards: Their Applications to Services: McGraw-Hill, 1997.
22. Gerd Siegmund. Intelligente Netze: Technik, Dienste, Vermarktung / Heidelberg: Huthig, 1999// ISBN 3-7785-3908-6.
23. Herzog U., Magedanz T. Intellegent Network and TINA – Migration and Interworking Issues//ISS'97. XVI World Telecom Congress Proceedings.-1997.
24. Intelligent Broadband Networks / Edited by I.Venieris, H.Husmann.— Chichester: John Wiley & Sons, 1998.— XXV.
25. Intelligent Networks and Intelligence in Networks: IFIP TC6 WG 6.7 International Conference on Intelligent Networks, 2 - 5 Sept. 1997, Paris, France / Ed. by D.Gaiti.— London: Chapman & Hall, 1997.— X.
26. Intelligent Networks and New Technologies: Proceedings of the IFIP TC6 Conference on Intelligent Networks and New Technologies, Technical University of Denmark, Denmark, August 1995.— London: Chapman & Hall, 1996.— XXII.
27. Intelligent Networks: Proceedings of the IFIP Workshop on Intelligent Networks 1994.— London: Chapman & Hall, 1995.
28. Liebowitz J., Prerau D. Worldwide Intelligent Systems. Approaches to Telecommunications and Network Management.— Amsterdam: IOS Press, 1995.
29. Magedanz T., Popescu-Zeletin R. Intelligent Networks: International Thomson Computer Press, 1996.
30. Russel T. Signalling System #7. – McGraw-Hill, 1995.
31. Thorner J. Intelligent Networks.— Boston; London: Artech House, 1994.
32. TINA' 96 Conference: Proceedings. Convergence of Telecommunications and Distributed Computing Technologies. Heidelberg, Sept. 3-5, 1996.— Germany: 1996.



Б.С. Гольдштейн – автор более 130 работ по телекоммуникациям. Его перу принадлежат, в частности, «Сигнализация в сетях связи» и «Протоколы сетей доступа» (обе книги вышли в свет в издательстве «Радио и связь»). Будучи заместителем директора ЛОНИИС, начальником первого научно-исследовательского отделения этого института, а также профессором кафедры сетей связи Государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича занимается исследованиями телекоммуникационного рынка, проведением анализа и разработкой программно-аппаратных средств перспективных телекоммуникационных технологий и услуг. В 1982 г. защитил кандидатскую диссертацию «Исследование и разработка телефонной операционной системы электронного узла коммутации», а в 1994 г. – докторскую диссертацию «Численные методы анализа и проектирования программного обеспечения систем коммутации», академик МАС и МАИ, член IEEE.



И.М. Ехриель работает в ЛОНИИС, где в настоящее время отвечает за координацию работ по реализации протоколов ОКС-7 в программно-аппаратных средствах для различных областей применения, включая Интеллектуальные сети. И. Ехриель закончил ЛЭИС им. проф. М.А. Бонч-Бруевича и защитил кандидатскую диссертацию, посвященную качественным характеристикам систем сигнализации ISDN. Автор более 20 печатных работ.



Р.Д. Перле работает в ЛОНИИС, где в настоящее время отвечает за координацию работ по реализации единой программно-аппаратной платформы для создания новых телекоммуникационных услуг в инфраструктуре ТФОП/ISDN, мобильных и IP-сетей на базе концепции Интеллектуальной сети. Окончила ЛЭИС им. проф. М.А. Бонч-Бруевича и защитила кандидатскую диссертацию, посвященную проблемам организации алгоритмов межпроцессорного обмена в цифровых системах коммутации. Автор более 50 печатных работ.

Гольдштейн Борис Соломонович
Ехриель Илья Михайлович
Рерле Римма Дмитриевна
ИНТЕЛЛЕКТУАЛЬНЫЕ СЕТИ

Компьютерная вёрстка *М.А. Фрост*

ИБ № 2970 Лицензия № 010164 от 29.01.97

Подписано в печать 15.06.2000.

Формат 70×100¹/₆.

Бумага офсетная.

Гарнитура прагматика. Печать офсетная.

Объем 32 печ. л.

Тираж 5000 экз. Зак № 24266

Издательство «Радио и связь», 101000 Москва, Почтамт, а/я 693

Отпечатано с готовых диапозитивов
в Академической типографии «Наука» РАН
199034, Санкт-Петербург, ВО, 9-я линия, д. 12