

Б.С. Гольдштейн, А.В. Пинчук, А.Л. Суховицкий

iP

ТЕЛЕФОНИЯ

Б.С. Гольдштейн, А.В. Пинчук, А.Л. Суховицкий

IP-ТЕЛЕФОНИЯ

Санкт-Петербург
«БХВ-Петербург»
2014

УДК 621.395.34

Г63

ББК 32.881

Гольдштейн Б.С., Пинчук А.В., Суховицкий А.Л.

IP-Телефония. — СПб.: БХВ-Петербург, 2014. —

336 с.: ил.

Г63 **ISBN 978-5-9775-3384-3**

Успехи IP-телефонии являются сегодня наиболее наглядным доказательством необходимости и неизбежности конвергенции сетей и услуг связи. Книга посвящена этой новой и перспективной технологии. Рассматриваются системно-сетевые аспекты IP-телефонии, методы и алгоритмы кодирования речевой информации, основные подходы и протоколы H.323, SIP, MGCP, MEGACO, вопросы качества обслуживания QoS, аспекты реализации оборудования IP-телефонии и его тестирования.

Для инженеров, программистов, менеджеров и специалистов, занятых разработкой и эксплуатацией систем и средств IP-телефонии. Для студентов и аспирантов соответствующих специальностей. Для всех, кого интересуют современные технологии телекоммуникаций.

Научно-техническое издание

ISBN 978-5-9775-3384-3

© Гольдштейн Б.С., Пинчук А.В., Суховицкий А.Л., 2001, 2014

Компьютерная вёрстка *М.А. Фрост*

Издательство «БХВ-Петербург», 190005, Санкт-Петербург, Измайловский пр., 29

Содержание

Предисловие	9
1 Конвергенция сетей связи	13
1.1 Пропорции в телекоммуникациях	13
1.2 Перспективы развития ТфОП и IP-сетей	15
1.3 Транспортные технологии пакетной коммутации	19
1.4 Уровни архитектуры IP-телефонии	21
1.5 Различные подходы к построению сетей IP-телефонии	23
1.5.1 Построение сети по рекомендации H.323	23
1.5.2 Сеть на базе протокола SIP	30
1.5.3 Сеть на базе MGCP и MEGACO	35
1.5.4 Сравнение подходов к построению сетей IP-телефонии ...	41
2 Сетевые аспекты IP-телефонии	45
2.1 Три основных сценария IP-телефонии	45
2.2 Проект TIPHON	54
2.3 Установление телефонного соединения в IP-сети	62
2.4 Эффективность IP-телефонии	64
3 Передача речи по IP-сеть	67
3.1 Особенности передачи речевой информации по IP-сетям	67
3.1.1 Задержки	67
3.1.2 Эхо	71
3.1.3 Устройства ограничения эффектов эха	72
3.2 Принципы кодирования речи	74
3.2.1 Кодирование формы сигнала	76
3.2.2 Кодеры источника информации (вокодеры) и гибридные алгоритмы	78
3.2.3 Цифровые процессоры обработки сигналов для речевых кодеков	81
3.2.4 Основные алгоритмы кодирования речи, используемые в IP телефонии	82
3.3 Кодеки, стандартизованные ITU-T	86
3.3.1 Кодек G.711	86
3.3.2 Кодек G.723.1	86
3.3.3 Кодек G.726	87

3.3.4	Кодек G.728	87
3.3.5	Кодек G.729	87
3.4	Алгоритмы кодирования ETSI	88
3.5	Передача сигналов DTMF	88
3.6	Передача факсимильной информации	90
3.7	О реализации «стандартных» алгоритмов	92
4	Протоколы сети интернет	95
4.1	Интернет ab ovo	95
4.2	Стандарты в сфере Интернет	99
4.3	Адресация	100
4.4	Уровни архитектуры Интернет	104
4.5	Протокол IP версии 4	106
4.6	Протокол IP версии 6	109
4.7	Протокол TCP	115
4.7.1	Потоки, стек протоколов, порты и мультиплексирование	116
4.7.2	Установление TCP-соединения и передача данных	117
4.7.3	Механизмы обеспечения достоверности	118
4.7.4	Механизм управления потоком данных	119
4.7.5	Состав и назначение полей заголовка	120
4.8	Протокол UDP	121
4.9	Требования к современным IP-сетям	122
4.10	Протоколы RTP и RTCP	125
4.11	Многоадресная рассылка	128
5	Архитектура H.323	131
5.1	Стандарты мультимедийной связи	131
5.2	Архитектура систем видеотелефонии в узкополосных ISDN ...	134
5.3	Мультимедийная связь в IP-сетях	137
5.4	Терминал H.323	139
5.5	Шлюз H.323	141
5.6	Привратник	142

5.7 Устройство управления конференциями	144
5.8 Реализация оборудования H.323	146
6 Сигнализация H.323	153
6.1 Семейство протоколов H.323	153
6.2 Протокол RAS	154
6.2.1 Обнаружение привратника	155
6.2.2 Регистрация окончного оборудования	156
6.2.3 Доступ к сетевым ресурсам	159
6.2.4 Определение местоположения оборудования в сети	160
6.2.5 Изменение полосы пропускания	161
6.2.6 Опрос текущего состояния оборудования	162
6.2.7 Освобождение полосы пропускания	163
6.2.8 Метка доступа	163
6.3 Сигнальный канал H.225.0	167
6.4 Управляющий канал H.245	172
6.4.1 Определение ведущего и ведомого	173
6.4.2 Обмен данными о функциональных возможностях	174
6.4.3 Открытие и закрытие логических каналов	177
6.4.4 Выбор режима обработки информации	179
6.5 Алгоритмы установления, поддержания и разрушения соединения.	183
6.5.1 Базовое соединение с участием привратника	183
6.5.2 Базовое соединение без участия привратника	187
6.5.3 Туннелирование управляющих сообщений	189
6.5.4 Процедура быстрого установления соединения	189
6.5.5 Установление соединения с участием шлюза	191
7 Протокол инициирования сеансов связи – SIP	193
7.1 Принципы протокола SIP	193
7.2 Интеграция протокола SIP с IP-сетями	195
7.3 Адресация	197
7.4 Архитектура сети SIP	198
7.4.1 Терминал	198
7.4.2 Прокси-сервер	199
7.4.3 Сервер переадресации	200
7.4.4 Сервер определения местоположения пользователей	200
7.4.5 Пример SIP-сети	201
7.5 Сообщения протокола SIP	202

7.5.1 Структура сообщений	202
7.5.2 Заголовки сообщений	204
7.5.3 Запросы	208
7.5.4 Ответы на запросы	211
7.6 Алгоритмы установления соединения	217
7.6.1 Установление соединения с участием сервера переадресации	217
7.6.2 Установление соединения с участием прокси-сервера ...	218
7.7 Реализация дополнительных услуг на базе протокола SIP	220
7.8 Сравнительный анализ H.323 и SIP	222
8 Протокол управления шлюзами MGCP	229
8.1 Принцип декомпозиции шлюза	229
8.2 Классификация шлюзов	232
8.3 Модель организации связи	233
8.4 Команды протокола MGCP	235
8.5 Структура команд	243
8.6 Структура ответов на команды	247
8.7 Описания сеансов связи	249
8.8 Установление, изменение и разрушение соединений	251
8.9 Реализация оборудования с поддержкой протокола MGCP	254
8.10 Возможности и перспективы протокола MGCP	256
9 Протокол MEGACO/H.248	257
9.1 История создания и особенности протокола MEGACO/H.248	257
9.2 Модель процесса обслуживания вызова	258
9.3 Сравнительный анализ протоколов MGCP и MEGACO	262
9.4 Структура команд и ответов	265
9.5 Пример установления и разрушения соединения	271
10 Качество обслуживания в сетях IP-телефонии	283
10.1 Что понимается под QoS?	283
10.2 Качество обслуживания в сетях пакетной коммутации	285

10.3	Трафик реального времени в IP сетях	286
10.4	Дифференцированное обслуживание разнотипного трафика – Diff Serv	287
10.5	Интегрированное обслуживание IntServ	289
10.6	Протокол резервирования ресурсов – RSVP	290
10.6.1	Общие принципы протокола	290
10.6.2	Процедура резервирования ресурсов	291
10.7	Технология MPLS	295
10.8	Обслуживание очередей	299
10.8.1	Алгоритмы организации очереди	299
10.8.1.1	Алгоритм Tail Drop	299
10.8.1.2	Алгоритм Random Early Detection (RED)	300
10.8.2	Алгоритмы обработки очередей	300
10.8.2.1	Стратегия FIFO	301
10.8.2.2	Очередь с приоритетами	301
10.8.2.3	Class Based Queuing (CBQ)	302
10.8.2.4	Взвешенные очереди	303
10.8.3	Алгоритмы сглаживания пульсации трафика	304
10.8.3.1	Алгоритм Leaky Bucket	304
10.8.3.2	Алгоритм «Token Bucket»	305
11	Принципы реализации	307
11.1	Оборудование IP-телефонии	307
11.2	Особенности оборудования IP телефонии для России	311
11.3	Шлюз IP телефонии Протей-ITG	312
11.4	Привратник Протей-GK и варианты организации связи	315
11.5	Экономические аспекты применения оборудования IP-телефонии	316
11.6	Виртуальная телефонная линия	318
11.7	Центр обработки вызовов	322
11.8	Модуль IPU как средство интеграции цифровых АТС с IP-сетями	323
11.9	Тестирование протоколов IP-телефонии	325
	Глоссарий	328
	Список литературы	332

Предисловие

В 1829 году губернатор Нью-Йорка Мартин ван Бюрен отправил президенту США Эндрю Джексону письмо следующего содержания:

Уважаемый господин Президент!

Системе каналов в нашей стране угрожает распространение новой формы транспорта, называемой «железные дороги». Правительство должно сохранить каналы по следующим причинам.

1) Если суда будут вытеснены железными дорогами, это приведёт к большой безработице.

2) Производители судов сильно пострадают, а поставщики буксирных канатов, кнутов и конной упряжи останутся без средств к существованию.

3) Суда абсолютно необходимы для обеспечения обороны страны.

Эти слова настолько похожи на рекомендации относительно IP-телефонии, услышанные авторами всего лишь два года назад на одном научно-техническом совете, что заставляют удивиться такому совпадению уровней и мотивов в разные времена и в разных странах. Однако технический прогресс определяется не этим, и сегодняшняя IP-телефония обслуживает около двадцати миллионов абонентов во всем мире, а операторские компании постепенно превращают IP-телефонию в индустрию, не зависящую от административно-командных решений. Примером для отечественных операторов может служить компания AT&T, уже применяющая передачу речи по IP-сетям и объявившая о долгосрочном плане перевода всего своего речевого трафика дальней связи на платформу IP. В составе совместного глобального проекта AT&T и British Telecom в течение четырех лет создают новую глобальную IP-сеть стоимостью 10 миллиардов долларов, которая будет предоставлять услуги интегрированной передачи речи и данных многонациональным бизнес-абонентам.

А начиналось все отнюдь не так безоблачно. Первая попытка реализовать IP-телефонию была предпринята в 1983 году в Кембридже,

Массачусетс. В состав оборудования рабочих станций, закрепленных за отдельными проектами Интернет, была включена так называемая «речевая воронка», выполнявшая функции цифровизации речи, пакетирования и передачи пакетов через Интернет между офисами Bolt Beranek and Newman (BBN) на Восточном и Западном побережьях США. С позиций приписываемого А. Эйнштейну высказывания – «открытия делаются тогда, когда все знают, что этого сделать нельзя, а потом появляется кто-то, кто этого не знает и совершает открытие» – те эксперименты 80-х годов относились к первой части данной формулировки. Немногочисленные студенты и энтузиасты IP-телефонии первого поколения были должны использовать на каждом конце одно и то же клиентское программное обеспечение, находиться в режиме подключения к системе в момент вызова, проводить значительную часть времени, терзая регулировки громкости и компрессии в попытках устранить эхо, чтобы получше слышать друг друга. Качество речи портили длинные паузы, вызванные переменной задержкой пакетов, обрезанная речь, получавшаяся в результате выбрасывания пакетов, эхо обратной связи из-за близкого расположения громкоговорителя компьютера и микрофона.

Открытие IP-телефонии как профессиональной технологии совершила израильская компания VocalTec, сумевшая к 1995 году собрать воедино достижения в областях цифровой обработки сигналов (DSP), кодеков, компьютеров и протоколов маршрутизации, чтобы сделать реальными разговоры через Интернет без оглядки на расстояние между абонентами и длительность разговора. О системных аспектах, основных сценариях и алгоритмах IP-телефонии говорится в главах 1 и 2.

Начиная с 1995 года, для IP-телефонии стали использоваться два метода звуковой компрессии – GSM, с близкой к 5:1 степенью компрессии исходного звукового сигнала, и TrueSpeech компании DSP Group, Inc., обеспечивающей коэффициент компрессии 18:1 с малозаметной потерей качества звука при декомпрессии. Это обсуждается в главе 3 данной книги. Там же рассматриваются аудио-стандарты G.7xx, включенные в рекомендованный Международным союзом электросвязи (ITU-T) «зонтичный» стандарт H.323, которому целиком посвящены главы 5 и 6. Другие концептуальные подходы и стандартные протоколы IP-телефонии SIP, MGCP и MEGACO рассмотрены в главах 7, 8 и 9, соответственно.

В дополнение к алгоритмам компрессии/декомпрессии выборки речи и стандартным протоколам, IP-телефония занимается техникой борьбы с задержками в Интернет. Пакеты могут следовать к месту назначения по разным путям и могут не все поступить к месту сборки

вовремя и в надлежащем порядке. Если бы это были обычные данные, то запоздавшие или поврежденные пакеты можно было бы просто отбросить, а протокол контроля ошибок в рабочей станции запросил бы повторную передачу этих пакетов. Но такая концепция не может быть принята для пакетов, содержащих компрессированную речь, без опасности значительного ухудшения качества разговоров, которые, разумеется, должны происходить в реальном времени. Только если отбрасывается небольшой процент пакетов, скажем, 15%, пользователи на каждом конце могут не заметить пробелов в разговоре. Когда потеря пакетов достигает 20%, качество разговора ощутимо ухудшается. Общему анализу протоколов Интернет для IP-телефонии посвящена глава 4, а проблемы качества обслуживания (QoS) для IP-телефонии рассматриваются в главе 10.

Изделия для современной IP-телефонии предоставляют множество функциональных возможностей и позволяют решить проблемы качества передачи речи, что и обеспечивает рост коммерчески привлекательных и высококачественных услуг IP-телефонии. Выигрыш от использования компьютера для телефонной связи – по отношению к обычному телефону – заключается в том, что пользователь получает преимущества услуг интегрированной передачи речи и данных. Наиболее общие функциональные возможности, встречающиеся в широком спектре изделий IP-телефонии, рассматриваются в заключительной, 11 главе книги. В этой главе излагаются некоторые принципы и идеи отечественной платформы Протей, реализующей самые современные услуги IP-телефонии применительно к условиям Взаимоувязанной сети связи России.

Впрочем, относительно современности обольщаться не следует ни авторам, ни читателям. Мы имеем дело с новой, бурно развивающейся областью, идеи и изделия появляются с ошеломляющей частотой, и самым трудным для авторов при подготовке книги было поставить точку.

В том, что это, в конце концов, удалось, заслуга тех, кто поддерживал авторов и помогал им советами, информацией и просто созданием стимулирующей атмосферы, в первую очередь – В.А. Соколова, Ю.В. Аксенова, А.Е. Кузнецова, В.Д. Кадыкова, а также студентов Санкт-Петербургского университета телекоммуникаций им. проф. М.А.Бонч-Бруевича – А.Б.Гольдштейна, В.В. Саморезова, С.Б. Шурыгиной. Результат всех этих усилий перед читателем.

Замечания и предложения по материалам книги просьба направлять по адресу nio1@loniis.spb.ru, а информацию о других книгах и разработках можно найти на Web-сайте www.loniis.ru.

Глава 1

Конвергенция сетей связи

1.1 Пропорции в телекоммуникациях

Гуляя в тенистой роще, греческий философ Анаксимен беседовал со своим учеником. «Скажи мне, – спросил юноша, – почему тебя часто одолевают сомнения? Ты прожил долгую жизнь, умудрен опытом и учился у великих эллинов. Как же так вышло, что и для тебя осталось столь много неясных вопросов?». В ответ философ очертил посохом перед собой два круга: маленький и большой. «Твои знания – это маленький круг, а мои – большой. Но все, что осталось вне этих кругов, – неизвестность. Маленький круг с неизвестностью соприкасается мало. Чем шире круг твоих знаний, тем протяженнее его граница с неизвестностью. И впредь, чем больше ты будешь узнавать нового, тем больше у тебя будет возникать неясных вопросов».

Классическая телефония с ее традиционными телефонными услугами POTS (Plain Old Telephone Service), достаточно хорошо изученная за свою более чем столетнюю историю, соответствует малому кругу из этой поучительной притчи. Большой круг представляет нарождающуюся индустрию инфокоммуникаций, являющуюся результатом взаимопроникновения (конвергенции) информационных и телекоммуникационных технологий и услуг и действительно порождающую больше неясных вопросов, чем готовых ответов. Не планируя в этой главе (да и во всей книге) рассмотреть множество разнообразных аспектов инфокоммуникаций за исключением одного – IP-телефонии, – коснемся лишь их общей базы – телекоммуникаций.

Со времени своего возникновения телекоммуникации базируются на передаче электромагнитных сигналов через транспортную среду, каковой могут быть:

- металлический кабель,
- оптоволокно,
- радиоканал.

Передаваемая в виде электромагнитных сигналов информация может представлять собой:

- речь,
- данные,
- видеоизображение

или любую их комбинацию, называемую мультимедийной информацией.

Эти три источника и три составные части телекоммуникаций в полной мере отражают их современное состояние, причем современность здесь понимается в широком смысле. Передача по сетям связи информации трех перечисленных выше видов благополучно осуществляется не одно десятилетие, пока не сработал принцип, давно известный в сфере искусств, – все дело в *пропорциях*.

Еще в 1996 г. в США трафик передачи данных впервые превысил речевой (рис. 1.1) и продолжает демонстрировать завидные темпы роста (до 30% в год по сравнению с 3% в год для телефонии). То же произошло в Европе в 1999 году. Все это послужило толчком к началу новой эры в телекоммуникациях – эры интегрированных решений и конвергенции всех видов связи. Протокол IP получил мировое признание и, в известной степени, стал «де-факто» стандартом для передачи мультимедийной информации.

Если добавить сюда феномен сети Интернет, где, по самым скромным подсчетам, рост числа пользователей составляет 5% в месяц, то станет совершенно ясно, что все эти события самым непосредственным образом влекут за собой коренное изменение подходов к построению информационных сетей. Речь и данные меняются местами. Традиционные сети передачи данных базировались на магистралях с коммутацией каналов, предназначенных для телефонного трафика. При новом подходе – все наоборот: телефония будет надстраиваться над инфраструктурой сети передачи данных.

Смещение центра тяжести в область передачи данных поставило вопрос о поиске удобного способа встраивания речи в мультимедийный цифровой поток. Причина популярности IP как раз и заключается в его восприимчивости к требованиям со стороны не только услуг передачи данных, но и приложений реального времени. Примером может служить успешно реализованная технология передачи речевой информации по сетям с маршрутизацией пакетов IP – Voice over IP (VoIP) или IP-телефония.

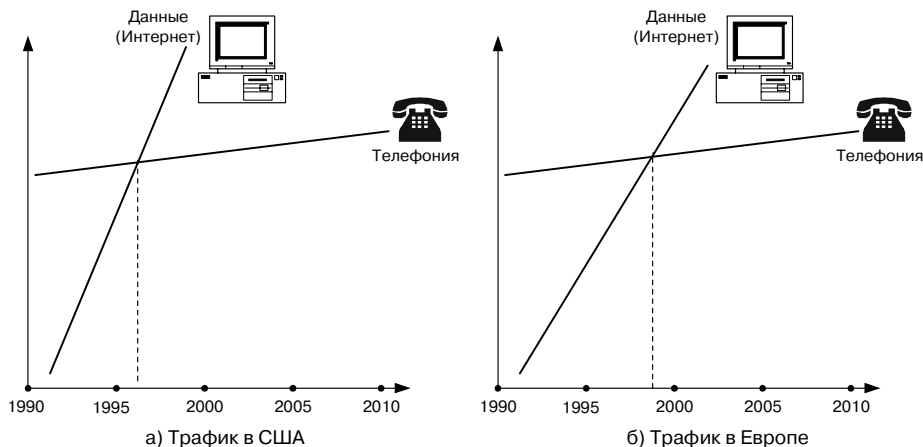


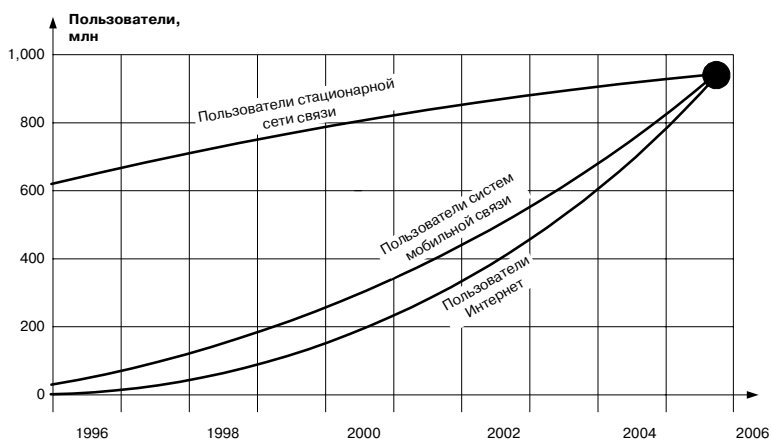
Рис. 1.1 Рост трафика Интернет (данные) и телефонного трафика

Но понятие Voice over IP подразумевает не только и не столько использование сети Интернет в качестве среды передачи речи, сколько сам протокол IP и технологии, обеспечивающие надежную и высококачественную передачу речевой информации в сетях пакетной коммутации. Отсутствие гарантированного качества обслуживания при передаче речи компенсируется появлением таких технологий, как многопротокольная коммутация по меткам – Multiprotocol Label Switching (MPLS), протокол резервирования ресурсов – Resource Reservation Protocol (RSVP), дифференциальное обслуживание разнотипного трафика – Differentiated Services (DiffServ) и других. Все большую популярность приобретает передача пакетов IP, упакованных в контейнеры систем синхронной цифровой иерархии – Synchronous Digital Hierarchy (SDH), а также технология спектрального мультиплексирования – Wave Division Multiplexing (WDM). Во всех случаях необходимым условием является подчинение каждого узла системы единой политике управления трафиком. Этому же призваны помочь протоколы RTP, RTSP, Differentiated Services и другие механизмы, рассматриваемые в следующих главах книги. Здесь же достаточно отметить, что стандартизация речевых технологий на основе стека TCP/IP и их поддержка лидерами рынка пакетной телефонии обеспечат совместимость оборудования разных производителей и позволят создавать системы, в которых возможны вызовы с аналогового телефонного аппарата, подключенного к порту маршрутизатора, на персональный компьютер, или с персонального компьютера на номер ТфОП, в рамках трех сценариев IP-телефонии, рассматриваемых в следующей главе.

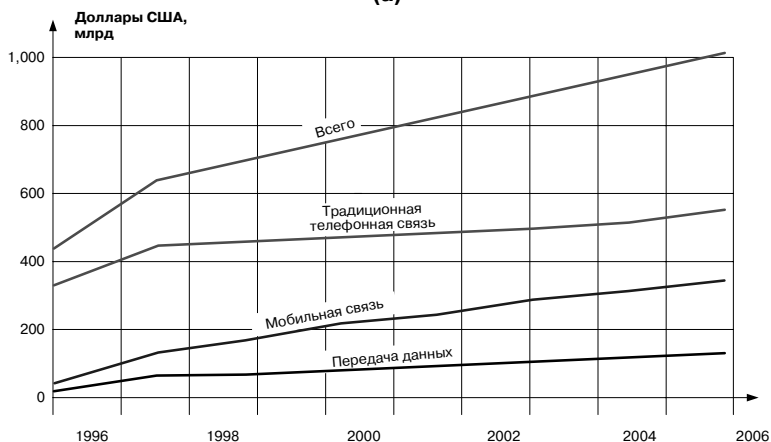
1.2 Перспективы развития ТфОП и IP-сетей

Продолжая анализ роста трафика данных и речи, представленного в виде графиков на рис.1 в предыдущем параграфе, авторы по-

зволили себе привести прогноз роста количества абонентов (графики на рис. 1.2а). Суть прогноза отнюдь не в том, что количество пользователей сетей стационарной связи, мобильной связи и Интернет к 2004-2006 годам достигнет миллиарда, а в том, что емкости этих сетей сближаются. В контексте данной главы последнее обстоятельство, согласно закону диалектики о переходе количества в качество, приводит к принципиально новым мыслям по поводу конвергенции этих сетей. Немаловажным стимулом таких мыслей является прогноз общемировых доходов от телекоммуникационных услуг, сделанный Dataquest (рис. 1.2б), графическое представление которого почти совпадает с верхней кривой на рис. 1.2а. Пороговая величина в этом прогнозе составляет триллион долларов США совокупного дохода по сегментам рынка (речь, данные, мобильная связь), а переход за этот порог ожидается еще раньше – в 2002-2003 гг.



(а)



(б)

Рис. 1.2 Рост численности абонентов, их перераспределение (а) и общемировые показатели доходов от телекоммуникационных услуг по сегментам рынка (б)

Одним из аспектов, способствующих упомянутой выше конвергенции, является ключевой принцип отделения организации услуг от транспортировки информации, составляющий основу идеи Интеллектуальных сетей. Суть концепции Интеллектуальной сети (IN) заключается в построении универсальной среды, обеспечивающей наибольшую эффективность создания и предоставления новых телефонных услуг. Постепенно эта концепция стала средством глобального нагнетания вычислительной мощности в телефонную сеть общего пользования (ТфОП), о чем немало сказано в только что вышедшей монографии [8].

Здесь же представляется полезным продолжить количественные оценки и попробовать представить себе краткосрочный и долгосрочный прогнозы развития телекоммуникационных услуг.

Краткосрочный прогноз авторы связывают с упомянутыми выше аспектами конвергенции сетей и услуг связи. Долгосрочный прогноз предполагает, что преобладание приложений типа клиент-сервер на основе IP-сетей (например, поиск информации, почта и др.) сохранится. Но в отдаленной перспективе внутренняя природа сети, базирующейся на протоколе IP, может стать тормозом для выполнения требований интерактивной мультимедиа: высокое быстродействие в реальном времени и «сквозная» широкополосная интерактивность. Для такого рода приложений в будущем потребуются более мощная платформа.

Рис. 1.3 иллюстрирует эволюцию телекоммуникационных приложений на основе IP.

Приняв во внимание то обстоятельство, что IP-телефония является одним из важнейших приложений на базе протокола IP, на основании рис. 1.3 читатель может принять решение о том, насколько целесообразно прочесть данную книгу. Основной вывод авторов из этого рисунка заключается в том, что Internet Protocol безусловно будет доминирующим протоколом в сетях следующего поколения, которым предстоит поддерживать передачу речи, данных, факсимиле, видеоинформации и мультимедиа.

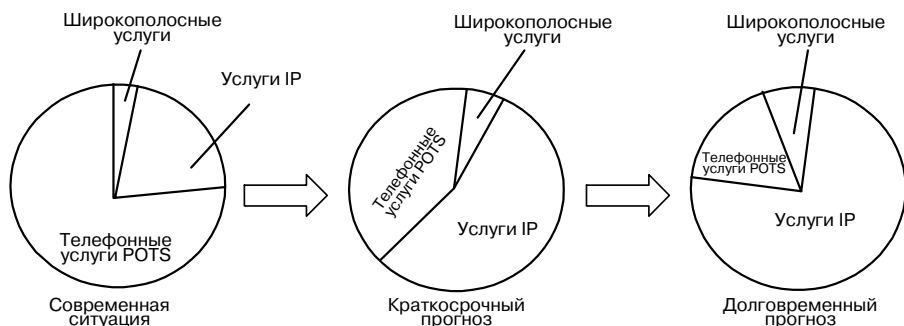


Рис. 1.3 Тенденции развития телекоммуникационных услуг

Первоочередная цель конвергенции сетей на базе протокола IP – это снижение общих расходов, складывающихся не только из капитальных затрат на приобретение и установку телекоммуникационного оборудования, но и из затрат на его содержание. Теоретически одна объединенная сеть уменьшила бы потребность в квалифицированном персонале – одни и те же люди стали бы заниматься и телефонией, и системами передачи данных. Наличие всего одного канала доступа к распределенной сети тоже основательно снизило бы ежемесячные расходы. Направляя речевой трафик через корпоративную магистральную сеть передачи данных, можно существенно уменьшить затраты на традиционные телефонные услуги. И, наконец, сокращение единиц используемого оборудования значительно уменьшит стоимость его технического обслуживания. Как отметил представитель одного международного оператора связи, переход на технологию IP-телефонии позволит ему сэкономить порядка 70% средств на капитальные затраты, 60-80% средств, выделяемых на организацию каналов доступа, и 50% средств на текущее обслуживание и ремонт сети [13].

Однако экономия на стоимости инфраструктуры – это не то, ради чего замышлялся переход к объединенным сетям. Революция произойдет тогда, когда появятся новые приложения, например, когда центры обслуживания клиентов смогут в реальном времени «сопровождать» каждого покупателя с момента его появления на домашней странице компании в сети Интернет до оформления заказа на покупку нужного продукта, «проводя» его через такие этапы, как демонстрация каталога предлагаемых изделий и выяснение неясных вопросов в ходе телефонного общения с представителем компании. Другой пример применения новых технологий – использование сотрудниками телефонного сервиса своей корпоративной УАТС независимо от того, где они находятся, например, при работе дома. К этим применениям IP-телефонии авторы вернутся в главе 11.

При всех оптимистических прогнозах, изложенных выше, не следует забывать, что традиционная телефонная связь опирается на мощную базу, создававшуюся на протяжении многих десятилетий, и такая система не может не обладать определенной инерцией. Исходя из этого, вряд ли стоит ожидать, что не сегодня-завтра произойдет мгновенный революционный скачок в области связи, и Интернет-телефония вытеснит все остальные технологии. Скорее наоборот: на протяжении ближайших 5-10 лет традиционная телефония будет по-прежнему занимать доминирующие позиции. Переход на новые, более прогрессивные методы будет происходить постепенно эволюционным путем, в разных странах с разной скоростью. А это значит, что в течение длительного времени ТФОП и IP-сети будут вынуждены существовать параллельно, обеспечивая взаимную прозрачность и объединяя свои усилия в обслуживании разнородного абонентского трафика.

Согласно известной формуле о невозможности находиться в каком-то обществе и быть вне его законов, при вхождении IP-телефонии в давно сформировавшееся глобальное телефонное общество необходимо соблюдение основных законов существующей ТфОП: эксплуатационная надежность с тремя девятками после запятой, жесткие нормы качества передачи речи в реальном времени и т.п.

Не менее законов, правил и норм важны традиции, сформировавшиеся за более чем столетний период существования ТфОП. И. Губерманом дана точная формулировка важности традиций:

Владыка наш – традиция. А в ней –
свои благословенья и препоны;
неписанные правила сильней,
чем самые свирепые законы.

Поэтому не менее важно сохранить все привычные для пользователя действия – набор номера, способ доступа к телефонным услугам и т. д. Таким образом, абонент не должен ощущать разницы между IP-телефонией и обычной телефонной связью ни по качеству речи, ни по алгоритму доступа.

По тем же причинам весьма желательно обеспечить между ТфОП и IP-сетями полную прозрачность передачи пользовательской информации и сигнализации. Дело в том, что в отличие, например, от большинства корпоративных сетей связи, сети общего пользования не имеют национальных и ведомственных границ. IP-телефония должна обладать возможностью поддерживать совместную работу и обеспечивать информационную прозрачность с множеством стандартов связи, принятых в разных странах мира. Речь идет не только об электрической стыковке – необходимо найти взаимоприемлемое решение таких задач, как взаимодействие протоколов верхних уровней и приложений, начисление платы и др.

1.3 Транспортные технологии пакетной коммутации

Большинство производителей, располагающих широким ассортиментом продукции для пакетной телефонии, занимают «технологически нейтральное» положение и предоставляют покупателю возможность самому выбирать ту технологию, которая лучше всего соответствует его интеграционной стратегии.

Основные технологии пакетной передачи речи – Frame Relay, ATM и маршрутизация пакетов IP – различаются эффективностью использования каналов связи, степенью охвата разных участков сети, надежностью, управляемостью, защитой информации и доступа, а также стоимостью. Ограниченный объем книги не позволяет дать глубокий сравнительный анализ этих технологий с точки зрения передачи речи, поэтому здесь приводятся в наиболее компактной графической форме только результаты такого анализа (рис. 1.4).



а) Речь по ATM



б) Речь по Frame Relay



в) Речь по IP

Рис. 1.4 Сравнение технологий пакетной передачи речи:
а)VoATM, б)VoFR, в)VoIP

Транспортная технология ATM уже несколько лет успешно используется в магистральных сетях общего пользования и в корпоративных сетях, а сейчас ее начинают активно использовать и для высокоскоростного доступа по каналам xDSL (для небольших офисов) и SDH/Sonet (для крупных предприятий). Главные преимущества этой технологии – ее зрелость, надежность и наличие развитых средств экс-

платационного управления сетью. В ней имеются непревзойденные по своей эффективности механизмы управления качеством обслуживания и контроля использования сетевых ресурсов. Однако ограниченная распространенность и высокая стоимость оборудования не позволяют считать АТМ лучшим выбором для организации сквозных телефонных соединений от одного конечного узла до другого.

Технологии Frame Relay суждено было сыграть в пакетной телефонии ту же роль, что и квазиэлектронным АТС в телефонии с коммутацией каналов: они показали пример эффективной программно управляемой техники, но имели ограниченные возможности дальнейшего развития. Пользователями недорогих услуг Frame Relay, обеспечивающих вполне предсказуемую производительность, стали многие корпоративные сети, и большинство из них вполне довольны своим выбором. В краткосрочной перспективе технология передачи речи по Frame Relay будет вполне эффективна для организации мультисервисного доступа и каналов дальней связи. Но сети Frame Relay распространены незначительно: как правило, на практике используются некоммутируемые соединения в режиме точка-точка.

Технология передачи речевой информации по сетям с маршрутизацией пакетов IP привлекает, в первую очередь, своей универсальностью – речь может быть преобразована в поток IP-пакетов в любой точке сетевой инфраструктуры: на магистрали сети оператора, на границе территориально распределенной сети, в корпоративной сети и даже непосредственно в терминале конечного пользователя. В конце концов, она станет наиболее широко распространенной технологией пакетной телефонии, поскольку способна охватить все сегменты рынка, будучи при этом хорошо адаптируемой к новым условиям применения. Несмотря на универсальность протокола IP, внедрение систем IP-телефонии сдерживается тем, что многие операторы считают их недостаточно надежными, плохо управляемыми и не очень эффективными. Но грамотно спроектированная сетевая инфраструктура с эффективными механизмами обеспечения качества обслуживания, рассматриваемыми в главе 10, делает эти недостатки малозначительными. В расчете на порт стоимость систем IP-телефонии находится на уровне (или немного ниже) стоимости систем Frame Relay, и заведомо ниже стоимости оборудования АТМ. При этом уже сейчас видно, что цены на продукты IP-телефонии снижаются быстрее, чем на другие изделия, и что происходит значительное обострение конкуренции на этом рынке.

1.4 Уровни архитектуры IP-телефонии

Архитектура технологии Voice over IP может быть упрощенно представлена в виде двух плоскостей. Нижняя плоскость – это базовая сеть с маршрутизацией пакетов IP, верхняя плоскость – это открытая архитектура управления обслуживанием вызовов (запросов связи).

Нижняя плоскость, говоря упрощенно, представляет собой комбинацию известных протоколов Интернет: это – RTP (Real Time Transport Protocol), который функционирует поверх протокола UDP (User Datagram Protocol), расположенного, в свою очередь, в стеке протоколов TCP/IP над протоколом IP. Таким образом, иерархия RTP/UDP/IP представляет собой своего рода транспортный механизм для речевого трафика. Этот механизм будет более подробно рассмотрен в главе 4, посвященной протоколам Интернет для передачи речи в реальном времени. Здесь же отметим, что в сетях с маршрутизацией пакетов IP для передачи данных всегда предусматриваются механизмы повторной передачи пакетов в случае их потери. При передаче информации в реальном времени использование таких механизмов только ухудшит ситуацию, поэтому для передачи информации, чувствительной к задержкам, но менее чувствительной к потерям, такой как речь и видеoinформация, используется механизм негарантированной доставки информации RTP/UDP/IP. Рекомендации ITU-T допускают задержки в одном направлении не превышающие 150 мс. Если приемная станция запросит повторную передачу пакета IP, то задержки при этом будут слишком велики. Эти проблемы более подробно рассматриваются в главе 10, посвященной качеству обслуживания.

Теперь перейдем к верхней плоскости управления обслуживанием запросов связи. Вообще говоря, управление обслуживанием вызова предусматривает принятие решений о том, куда вызов должен быть направлен, и каким образом должно быть установлено соединение между абонентами. Инструмент такого управления – телефонные системы сигнализации, начиная с систем, поддерживаемых декадно-шаговыми АТС и предусматривающих объединение функций маршрутизации и функций создания коммутируемого разговорного канала в одних и тех же декадно-шаговых искателях. Далее принципы сигнализации эволюционировали к системам сигнализации по выделенным сигнальным каналам, к многочастотной сигнализации, к протоколам общеканальной сигнализации №7 [6, 7] и к передаче функций маршрутизации в соответствующие узлы обработки услуг Интеллектуальной сети [8].

В сетях с коммутацией пакетов ситуация более сложна. Сеть с маршрутизацией пакетов IP принципиально поддерживает одновременно целый ряд разнообразных протоколов маршрутизации. Такими протоколами на сегодня являются: RIP – Routing Information Protocol, IGRP – Interior Gateway Routing Protocol, EIGRP – Enhanced Interior Gateway Routing Protocol, IS-IS – Intermediate System-to-Intermediate System, OSPF – Open Shortest Path First, BGP – Border Gateway Protocol и др. Точно так же и для IP-телефонии разработан целый ряд протоколов. Рассматриваемые в этой книге стандарты содержат положения, относящиеся к передаче речи по IP-сетям (глава 3) и к сигнализации для IP-телефонии (главы 6, 7, 8 и 9).

Наиболее распространенным является протокол, специфицированный в рекомендации H.323 ITU-T, в частности, потому, что он стал применяться раньше других протоколов, которых, к тому же, до внедрения H.323 вообще не существовало. Этот протокол подробно рассматривается в главах 5 и 6.

Другой протокол плоскости управления обслуживанием вызова – SIP – ориентирован на то, чтобы сделать оконечные устройства и шлюзы более интеллектуальными и поддерживать дополнительные услуги для пользователей. Этот протокол подробно рассматривается в главе 7.

Еще один протокол – SGCP – разрабатывался, начиная с 1998 года, для того, чтобы уменьшить стоимость шлюзов за счет реализации функций интеллектуальной обработки вызова в централизованном оборудовании. Протокол IPDC очень похож на SGCP, но имеет много больше, чем SGCP, механизмов эксплуатационного управления (OAM&P). В конце 1998 года рабочая группа MEGACO комитета IETF разработала протокол MGCP, базирующийся, в основном, на протоколе SGCP, но с некоторыми добавлениями в части OAM&P. Протокол MGCP подробно рассматривается в главе 8.

Рабочая группа MEGACO не остановилась на достигнутом, продолжала совершенствовать протокол управления шлюзами и разработала более функциональный, чем MGCP, протокол MEGACO. Его адаптированный к H.323 вариант (под названием Gateway Control Protocol) ITU-T предлагает в рекомендации H.248. Протоколу MEGACO/H.248 посвящена глава 9.

1.5 Различные подходы к построению сетей IP-телефонии

Чтобы стало понятно, чем конкретно отличаются друг от друга перечисленные в предыдущем параграфе протоколы, кратко рассмотрим архитектуру сетей, построенных на базе этих протоколов, и процедуры установления и завершения соединения с их использованием.

1.5.1 Построение сети по рекомендации H.323

Первый в истории подход к построению сетей IP-телефонии на стандартизированной основе предложен Международным союзом электросвязи (ITU) в рекомендации H.323 [42]. Сети на базе протоколов H.323 ориентированы на интеграцию с телефонными сетями и могут рассматриваться как сети ISDN, наложенные на сети передачи данных. В частности, процедура установления соединения в таких сетях IP-телефонии базируется на рекомендации Q.931 [44] и аналогична процедуре, используемой в сетях ISDN.

Рекомендация Н.323 предусматривает довольно сложный набор протоколов, который предназначен не просто для передачи речевой информации по IP-сетям с коммутацией пакетов. Его цель – обеспечить работу мультимедийных приложений в сетях с негарантированным качеством обслуживания. Речевой трафик – это только одно из приложений Н.323, наряду с видеоинформацией и данными. А так как ничего в технике (как и в жизни) не достается даром, обеспечение совместимости с Н.323 различных мультимедийных приложений требует весьма значительных усилий. Например, для реализации функции переключения связи (call transfer) требуется отдельная спецификация Н.450.2.

Вариант построения сетей IP-телефонии, предложенный Международным союзом электросвязи в рекомендации Н.323, хорошо подходит тем операторам местных телефонных сетей, которые заинтересованы в использовании сети с коммутацией пакетов (IP-сети) для предоставления услуг междугородной и международной связи. Протокол RAS, входящий в семейство протоколов Н.323, обеспечивает контроль использования сетевых ресурсов, поддерживает аутентификацию пользователей и может обеспечивать начисление платы за услуги.

На рис 1.5. представлена архитектура сети на базе рекомендации Н.323. Основными устройствами сети являются: терминал (Terminal), шлюз (Gateway), привратник (Gatekeeper) и устройство управления конференциями (Multipoint Control Unit – MCU).

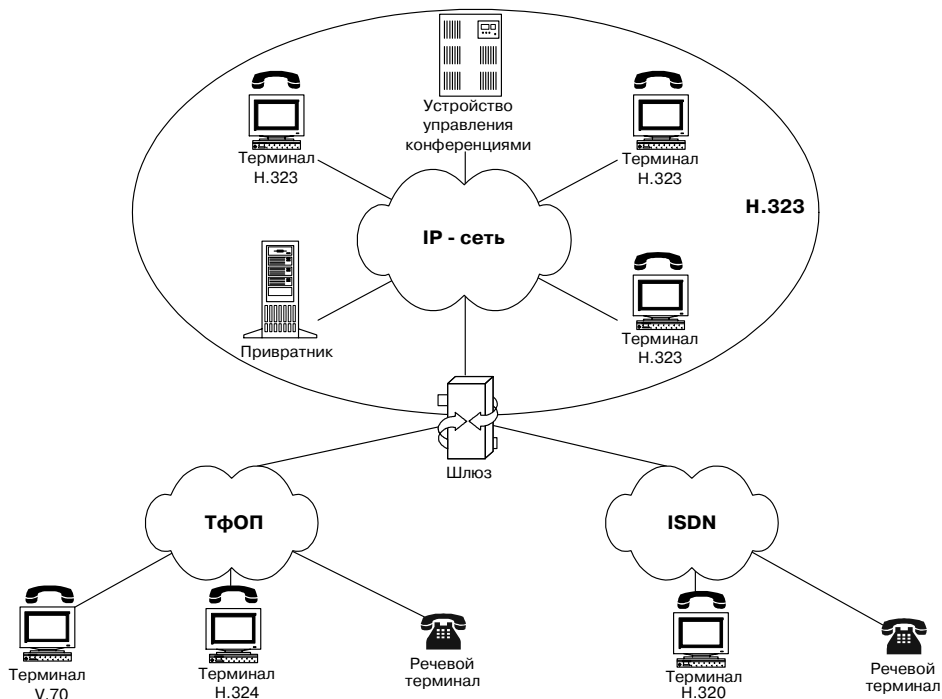


Рис. 1.5 Архитектура сети Н.323

Терминал Н.323 – оконечное устройство пользователя сети IP-телефонии, которое обеспечивает двухстороннюю речевую (мультимедийную) связь с другим терминалом Н.323, шлюзом или устройством управления конференциями.

Шлюз IP-телефонии реализует передачу речевого трафика по сетям с маршрутизацией пакетов IP по протоколу Н.323. Основное назначение шлюза – преобразование речевой информации, поступающей со стороны ТФОП, в вид, пригодный для передачи по сетям с маршрутизацией пакетов IP. Кроме того, шлюз преобразует сигнальные сообщения систем сигнализации DSS1 и ОКС7 в сигнальные сообщения Н.323 и производит обратное преобразование в соответствии с рекомендацией ITU Н.246.

В *привратнике* сосредоточен весь интеллект сети IP-телефонии. Сеть, построенная в соответствии с рекомендацией Н.323, имеет зонную архитектуру (рис. 1.6). Привратник выполняет функции управления одной зоной сети IP-телефонии, в которую входят: терминалы, шлюзы, устройства управления конференциями, зарегистрированные у данного привратника. Отдельные фрагменты зоны сети Н.323 могут быть территориально разнесены и соединяться друг с другом через маршрутизаторы.

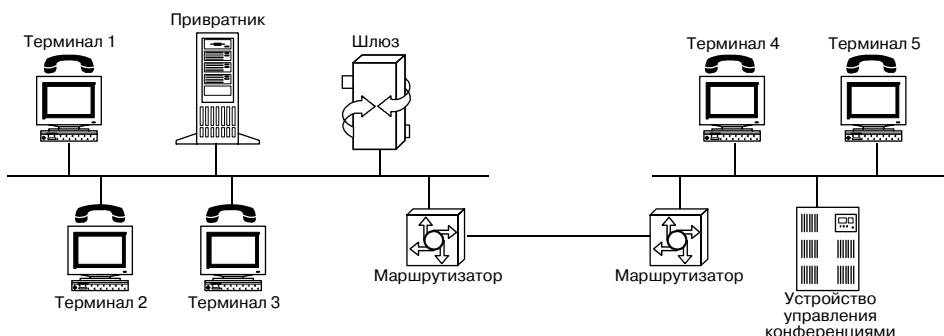


Рис. 1.6 Зона сети Н.323

Наиболее важными функциями привратника являются:

- регистрация оконечных и других устройств;
- контроль доступа пользователей системы к услугам IP-телефонии при помощи сигнализации RAS;
- преобразование *alias*-адреса вызываемого пользователя (объявленного имени абонента, телефонного номера, адреса электронной почты и др.) в транспортный адрес сетей с маршрутизацией пакетов IP (IP адрес + номер порта TCP);
- контроль, управление и резервирование пропускной способности сети;

- ретрансляция сигнальных сообщений H.323 между терминалами.

В одной сети IP-телефонии, отвечающей требованиям рекомендации ITU H.323, может находиться несколько привратников, взаимодействующих друг с другом по протоколу RAS.

Кроме основных функций, определенных рекомендацией H.323, привратник может отвечать за аутентификацию пользователей и начисление платы (биллинг) за телефонные соединения.

Устройство управления конференциями обеспечивает возможность организации связи между тремя или более участниками. Рекомендация H.323 предусматривает три вида конференции (рис. 1.7): централизованная (т.е. управляемая MCU, с которым каждый участник конференции соединяется в режиме точка-точка), децентрализованная (когда каждый участник конференции соединяется с остальными ее участниками в режиме точка-группа точек) и смешанная.

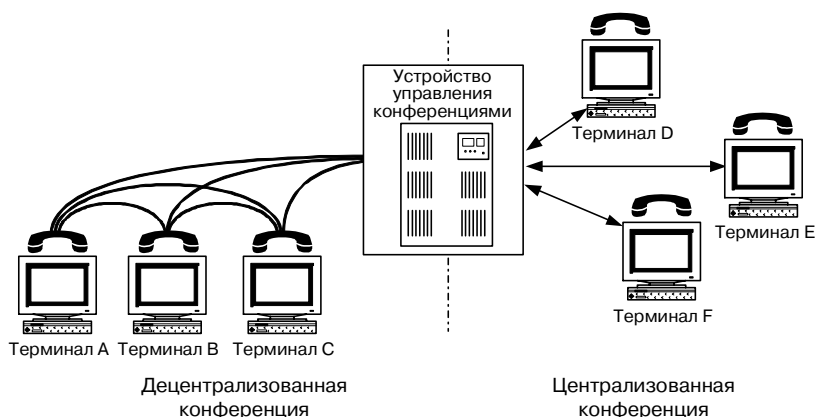


Рис. 1.7 Виды конференции в сетях H.323

Преимуществом централизованной конференции является сравнительно простое терминальное оборудование, недостатком – большая стоимость устройства управления конференциями.

Для децентрализованной конференции требуется более сложное терминальное оборудование и желательно, чтобы в сети IP поддерживалась передача пакетов IP в режиме многоадресной рассылки (IP multicasting). Если этот режим в сети не поддерживается, терминал должен передавать речевую информацию каждому из остальных участников конференции в режиме точка-точка.

Устройство управления конференциями состоит из одного обязательного элемента – контроллера конференций (Multipoint Controller – MC), и, кроме того, может включать в себя один или более про-

цессоров для обработки пользовательской информации (Multipoint Processor – MP). Контроллер может быть физически совмещен с привратником, шлюзом или устройством управления конференциями, а последнее, в свою очередь, может быть совмещено со шлюзом или привратником.

Контроллер конференций используется для организации конференции любого вида. Он организует обмен между участниками конференции данными о режимах, поддерживаемых их терминалами, и указывает, в каком режиме участники конференции могут передавать информацию, причем в ходе конференции этот режим может изменяться, например, при подключении к ней нового участника.

Так как контроллеров в сети может быть несколько, для каждой вновь создаваемой конференции должна быть проведена специальная процедура выявления того контроллера, который будет управлять данной конференцией.

При организации централизованной конференции, кроме контроллера MC, должен использоваться процессор MP, обрабатывающий пользовательскую информацию. Процессор MP отвечает за переключение или смешивание речевых потоков, видеoinформации и данных. Для децентрализованной конференции процессор не нужен.

Существует еще один элемент сети H.323 – *прокси-сервер H.323*, т.е. сервер-посредник. Этот сервер функционирует на прикладном уровне и может проверять пакеты с информацией, которой обмениваются два приложения. Прокси-сервер может определять, с каким приложением (H.323 или другим) ассоциирован вызов, и осуществлять нужное соединение. Прокси-сервер выполняет следующие ключевые функции:

- подключение через средства коммутируемого доступа или локальные сети терминалов, не поддерживающих протокол резервирования ресурсов (RSVP). Два таких прокси-сервера могут образовывать в IP-сети туннельное соединение с заданным качеством обслуживания;
- маршрутизацию трафика H.323 отдельно от обычного трафика данных;
- обеспечение совместимости с преобразователем сетевых адресов, поскольку допускается размещение оборудования H.323 в сетях с пространством адресов частных сетей;
- защиту доступа – доступность только для трафика H.323.

Более подробно архитектура сети H.323 будет рассмотрена в главе 5, а сейчас целесообразно сказать несколько слов о протоколах сигнализации, входящих в семейство H.323.

Протокол RAS (Registration, Admission, Status) обеспечивает взаимодействие оконечных и других устройств с привратником. Основными функциями протокола являются: регистрация устройства в системе, контроль его доступа к сетевым ресурсам, изменение полосы пропускания в процессе связи, опрос и индикация текущего состояния устройства. В качестве транспортного протокола используется протокол с негарантированной доставкой информации UDP.

Протокол H.225.0 (Q.931) поддерживает процедуры установления, поддержания и разрушения соединения. В качестве транспортного протокола используется протокол с установлением соединения и гарантированной доставкой информации TCP.

По протоколу H.245 происходит обмен между участниками соединения информацией, которая необходима для создания логических каналов. По этим каналам передается речевая информация, упакованная в пакеты RTP/UDP/IP, которые рассматриваются в главе 4.

Выполнение процедур, предусмотренных протоколом RAS, является начальной фазой установления соединения с использованием сигнализации H.323. Далее следуют фаза сигнализации H.225.0 (Q.931) и обмен управляющими сообщениями H.245. Разрушение соединения происходит в обратной последовательности: в первую очередь закрывается управляющий канал H.245 и сигнальный канал H.225.0, после чего привратник по каналу RAS оповещается об освобождении ранее занимавшейся полосы пропускания.

Сложность протокола H.323 демонстрирует рис. 1.8, на котором представлен упрощенный сценарий установления соединения между двумя пользователями. В данном сценарии предполагается, что конечные пользователи уже знают IP-адреса друг друга. В обычном случае этапов бывает больше, поскольку в установлении соединения участвуют привратники и шлюзы; это будет рассмотрено в главе 6.

Рассмотрим шаг за шагом этот упрощенный сценарий.

1. Оконечное устройство пользователя А посылает запрос соединения – сообщение SETUP – к оконечному устройству пользователя В на TCP-порт 1720.
2. Оконечное устройство вызываемого пользователя В отвечает на сообщение SETUP сообщением ALERTING, означающим, что устройство свободно, а вызываемому пользователю подается сигнал о входящем вызове.
3. После того, как пользователь В принимает вызов, к вызывающей стороне А передается сообщение CONNECT с номером TCP-порта управляющего канала H.245.
4. Оконечные устройства обмениваются по каналу H.245 информацией о типах используемых речевых кодеков (G.729, G.723.1 и т.д.),

а также о других функциональных возможностях оборудования, и оповещают друг друга о номерах портов RTP, на которые следует передавать информацию.

5. Открываются логические каналы для передачи речевой информации.
6. Речевая информация передаётся в обе стороны в сообщениях протокола RTP; кроме того, ведётся контроль передачи информации при помощи протокола RTCP.

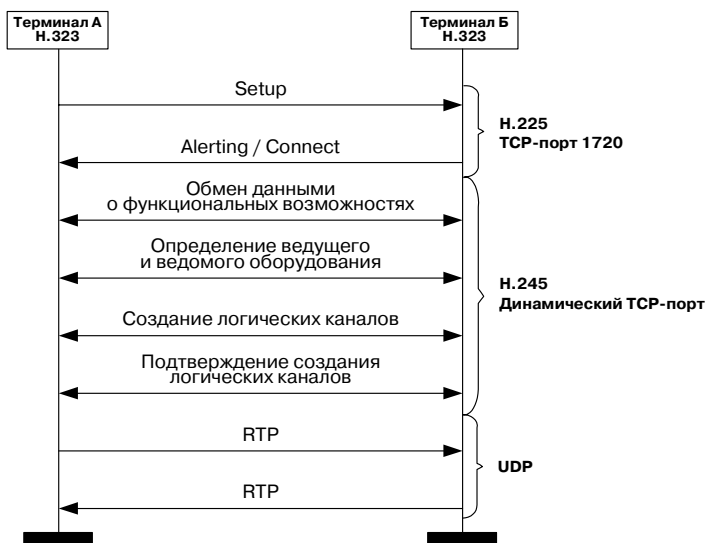


Рис. 1.8 Упрощённый сценарий установления соединения в сети H.323

Приведенная процедура обслуживания вызова базируется на протоколе H.323 версии 1. Версия 2 протокола H.323 позволяет передавать информацию, необходимую для создания логических каналов, непосредственно в сообщении **SETUP** протокола H.225.0 без использования протокола H.245. Такая процедура называется «быстрый старт» (**Fast Start**) и позволяет сократить количество циклов обмена информацией при установлении соединения. Кроме организации базового соединения, в сетях H.323 предусмотрено предоставление дополнительных услуг в соответствии с рекомендациями ITU H.450.x. Более детальный обзор сигнализации H.323 приводится в главе 6.

Следует отметить еще одну важную проблему – качество обслуживания в сетях H.323. Оконечное устройство, запрашивающее у привратника разрешение на доступ, может, используя поле *transportQoS* в сообщении **ARQ** протокола **RAS**, сообщить о своей способности резервировать сетевые ресурсы. Рекомендация H.323 определяет

протокол резервирования ресурсов (RSVP) как средство обеспечения гарантированного качества обслуживания, что предъявляет к терминалам требование поддержки протокола RSVP. К сожалению, протокол RSVP используется отнюдь не повсеместно, что оставляет сети H.323 без основного механизма обеспечения гарантированного качества обслуживания. Это – общая проблема сетей IP-телефонии, характерная не только для сетей H.323.

Мониторинг качества обслуживания обеспечивается протоколом RTCP, однако обмен информацией RTCP происходит только между оконечными устройствами, участвующими в соединении. Более подробно эта проблематика рассматривается в главе 10, целиком посвященной качеству обслуживания вызовов IP-телефонии.

1.5.2 Сеть на базе протокола SIP

Второй подход к построению сетей IP-телефонии, предложенный рабочей группой MMUSIC комитета IETF в документе RFC 2543 [54], основан на использовании протокола SIP – Session Initiation Protocol. SIP представляет собой текст-ориентированный протокол, который является частью глобальной архитектуры мультимедиа, разработанной комитетом Internet Engineering Task Force (IETF). Эта архитектура также включает в себя протокол резервирования ресурсов (Resource Reservation Protocol, RSVP, RFC 2205), транспортный протокол реального времени (Real-Time Transport Protocol, RTP, RFC 1889), протокол передачи потоков в реальном времени (Real-Time Streaming Protocol, RTSP, RFC 2326), протокол описания параметров связи (Session Description Protocol, SDP, RFC 2327), протокол уведомления о связи (Session Announcement Protocol, SAP). Однако функции протокола SIP не зависят от любого из этих протоколов.

Сразу следует отметить, что хотя на сегодня наиболее широкое распространение получил протокол H.323, всё большее количество производителей старается предусмотреть в своих новых продуктах поддержку протокола SIP. Пока это – единичные явления и серьезной конкуренции протоколу H.323 они составить не могут. Однако, учитывая темпы роста популярности протокола SIP, весьма вероятно, что в ближайшем будущем решения на его базе займут значительную нишу рынка IP-телефонии.

Подход SIP к построению сетей IP-телефонии намного проще в реализации, чем H.323, но меньше подходит для организации взаимодействия с телефонными сетями. В основном это связано с тем, что протокол сигнализации SIP, базирующийся на протоколе HTTP, плохо согласуется с системами сигнализации, используемыми в ТФОП. Поэтому протокол SIP более подходит поставщикам услуг Интернет для предоставления услуги IP-телефонии, причем эта услуга будет являться всего лишь частью пакета услуг.

Тем не менее, протокол SIP поддерживает услуги интеллектуальной сети (IN), такие как преобразование (мэппинг) имён, переадресация и маршрутизация [8], что существенно для использования SIP в качестве протокола сигнализации в сети общего пользования, где приоритетной задачей оператора является предоставление широкого спектра телефонных услуг. Другой важной особенностью протокола SIP является поддержка мобильности пользователя, т.е. его способности получать доступ к заказанным услугам в любом месте и с любого терминала, а также способности сети идентифицировать и аутентифицировать пользователя при его перемещении из одного места в другое. Это свойство SIP не уникально, и, например, протокол H.323 тоже в значительной степени поддерживает такую возможность. Сейчас настал момент, когда эта возможность станет главной привлекательной чертой сетей IP-телефонии нового поколения. Данный режим работы потребует дистанционной регистрации пользователей на сервере идентификации и аутентификации.

Перейдем непосредственно к архитектуре сетей, базирующихся на протоколе SIP (рис. 1.9).

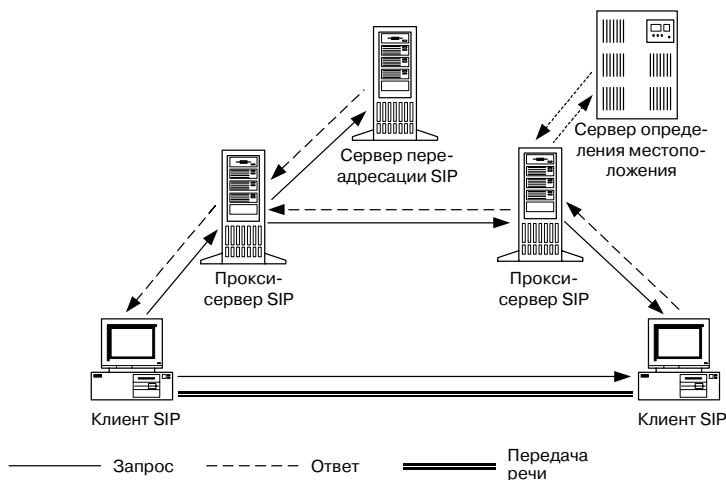


Рис. 1.9 Пример сети на базе протокола SIP

Сеть SIP содержит основные элементы трех видов: агенты пользователя, прокси-серверы и серверы переадресации.

Агенты пользователя (User Agent или SIP client) являются приложениями терминального оборудования и включают в себя две составляющие: агент пользователя – клиент (User Agent Client – UAC) и агент пользователя – сервер (User Agent Server – UAS), иначе известные как *клиент* и *сервер* соответственно. Клиент UAC инициирует SIP-запросы, т.е. выступает в качестве вызывающей стороны. Сервер UAS принимает запросы и возвращает ответы, т.е. выступает в качестве вызываемой стороны.

Кроме того, существует два типа сетевых серверов SIP: прокси-серверы (серверы-посредники) и серверы переадресации. Серверы SIP могут работать как в режиме с сохранением состояний текущих соединений (statefull), так и в режиме без сохранения состояний текущих соединений (stateless). Сервер SIP, функционирующий в режиме stateless, может обслужить сколь угодно большое количество пользователей, в отличие от привратника H.323, который может одновременно работать с ограниченным количеством пользователей.

Прокси-сервер (Proxy-server) действует «от имени других клиентов» и содержит функции клиента (UAC) и сервера (UAS). Этот сервер интерпретирует и может перезаписывать заголовки запросов перед отправкой их к другим серверам (рис. 1.10). Ответные сообщения следуют по тому же пути обратно к прокси-серверу, а не к клиенту.

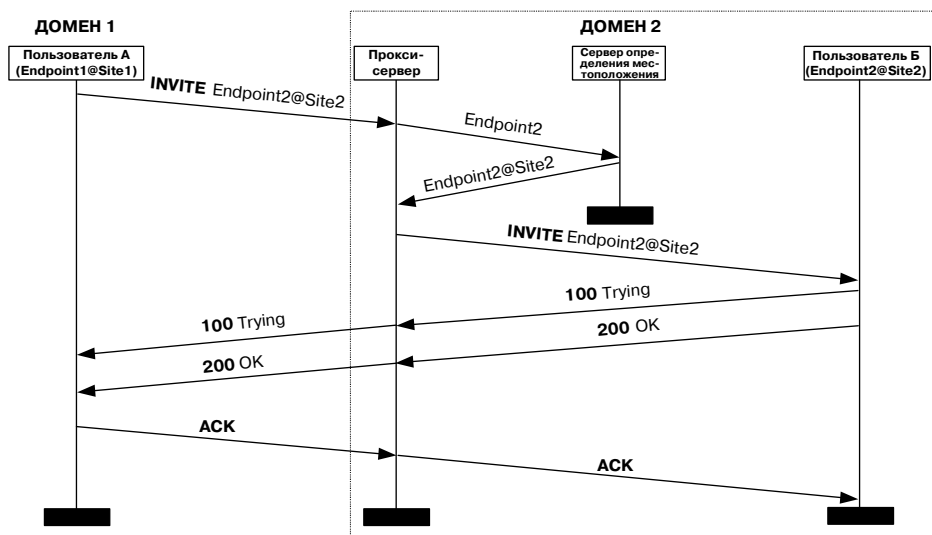


Рис. 1.10 Сеть SIP с прокси-сервером

Ниже представлен алгоритм установления соединения с помощью протокола SIP при участии прокси-сервера:

1. Прокси-сервер принимает запрос соединения INVITE от оборудования вызывающего пользователя.
2. Прокси-сервер устанавливает местонахождение клиента с помощью сервера определения местоположения (location server).
3. Прокси-сервер передает запрос INVITE вызываемому пользователю.
4. Оборудование вызываемого пользователя уведомляет последнего о входящем вызове и возвращает прокси-серверу сообщение о том, что запрос INVITE обрабатывается (код 100). Прокси-сервер, в свою очередь, направляет эту информацию оборудованию вызывающего пользователя.

5. Когда вызываемый абонент принимает вызов, его оборудование извещает об этом прокси-сервер (код 200), который переправляет информацию о том, что вызов принят, к оборудованию вызывающего пользователя.
6. Вызывающая сторона подтверждает установление соединения передачей запроса ACK, которое прокси-сервер переправляет вызываемой стороне. Установление соединения закончено, абоненты могут обмениваться речевой информацией.

Сервер переадресации (Redirect server) определяет текущее местоположение вызываемого абонента и сообщает его вызывающему пользователю (рис. 1.11). Для определения текущего местоположения вызываемого абонента сервер переадресации обращается к *серверу определения местоположения*, принципы работы которого в документе RFC 2543 не специфицированы.

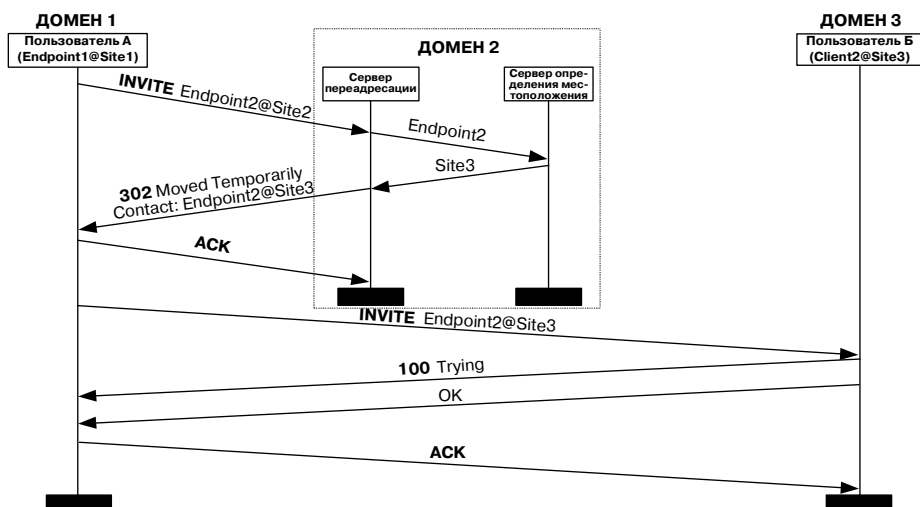


Рис. 1.11 Сеть SIP с сервером переадресации

Алгоритм установления соединения с использованием протокола SIP при участии сервера переадресации выглядит следующим образом:

1. Сервер переадресации принимает от вызывающей стороны запрос соединения INVITE и связывается с сервером определения местонахождения, который выдает текущий адрес вызываемого клиента.
2. Сервер переадресации передает этот адрес вызывающей стороне. В отличие от прокси-сервера, запрос INVITE к оборудованию вызываемого пользователя сервер переадресации не передает.

3. Оборудование вызывающего пользователя подтверждает завершение транзакции с сервером переадресации запросом ACK.
4. Далее оборудование вызывающего пользователя передает запрос INVITE на адрес, полученный от сервера переадресации.
5. Оборудование вызываемого пользователя уведомляет последнего о входящем вызове и возвращает вызывающему оборудованию сообщение о том, что запрос INVITE обрабатывается (код 100).
6. Когда вызываемый абонент принимает вызов, об этом извещается оборудование вызывающего пользователя (код 200). Установление соединения закончено, абоненты могут обмениваться речевой информацией.

Существует также и бессерверный вариант соединения, когда один терминал может передать запрос другому терминалу непосредственно.

Дадим краткую характеристику самого протокола SIP. Следует заметить, что сообщения SIP могут переноситься как протоколом TCP, так и протоколом UDP.

Протокол SIP предусматривает 6 запросов и ответов на них. Сигнализация SIP дает возможность пользовательским агентам и сетевым серверам определять местоположение, выдавать запросы и управлять соединениями.

INVITE – запрос привлекает пользователя или услугу к участию в сеансе связи и содержит описание параметров этой связи. С помощью этого запроса пользователь может определить функциональные возможности терминала своего партнера по связи и начать сеанс связи, используя ограниченное число сообщений и подтверждений их приема.

ACK – запрос подтверждает прием от вызываемой стороны ответа на команду INVITE и завершает транзакцию.

OPTIONS – запрос позволяет получить информацию о функциональных возможностях пользовательских агентов и сетевых серверов. Однако этот запрос не используется для организации сеансов связи.

BYE – запрос используется вызывающей и вызываемой сторонами для разрушения соединения. Перед тем как разрушить соединение, пользовательские агенты отправляют этот запрос к серверу, сообщая о намерении прекратить сеанс связи.

CANCEL – запрос позволяет пользовательским агентам и сетевым серверам отменить любой ранее переданный запрос, если ответ на нее еще не был получен.

REGISTER – запрос применяется клиентами для регистрации информации о местоположении с использованием серверов SIP.

Более подробная информация о протоколе SIP приведена в главе 7.

1.5.3 Сеть на базе MGCP и MEGACO

Третий подход к построению сетей IP–телефонии, основанный на использовании протокола MGCP [56], также предложен комитетом IETF, рабочей группой MEGACO.

При разработке этого протокола рабочая группа MEGACO опиралась на сетевую архитектуру, содержащую основные функциональные блоки трех видов (рис. 1.12):

- шлюз – Media Gateway (MG), который выполняет функции преобразования речевой информации, поступающей со стороны ТФОП с постоянной скоростью передачи, в вид, пригодный для передачи по сетям с маршрутизацией пакетов IP (кодирование и упаковку речевой информации в пакеты RTP/UDP/IP, а также обратное преобразование);
- контроллер шлюзов – Call Agent, который выполняет функции управления шлюзами;
- шлюз сигнализации – Signaling Gateway (SG), который обеспечивает доставку сигнальной информации, поступающей со стороны ТФОП, к контроллеру шлюзов и перенос сигнальной информации в обратном направлении.

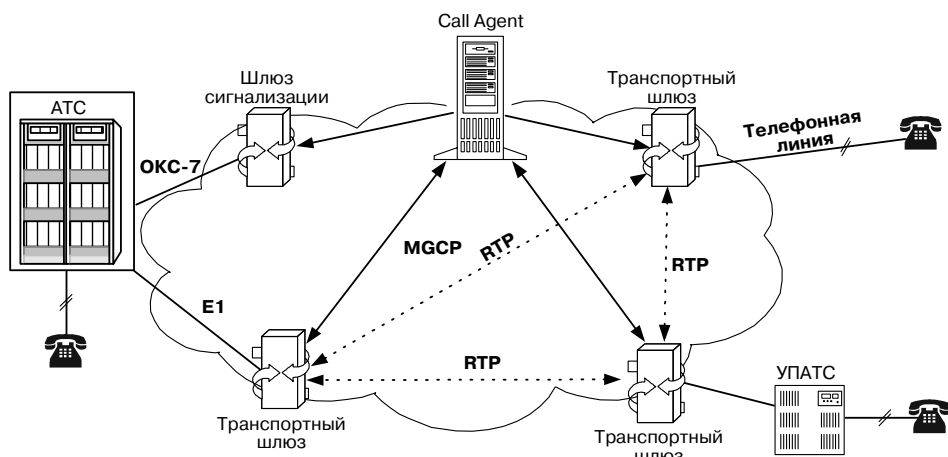


Рис. 1.12 Архитектура сети на базе протокола MGCP

Таким образом, весь интеллект функционально распределенного шлюза сосредоточен в контроллере, функции которого могут быть распределены между несколькими компьютерными платформами.

Шлюз сигнализации выполняет функции STP – транзитного пункта сети сигнализации ОКС7. Сами шлюзы выполняют только функции преобразования речевой информации. Один контроллер управляет одновременно несколькими шлюзами. В сети могут присутствовать несколько контроллеров. Предполагается, что они синхронизованы между собой и согласованно управляют шлюзами, участвующими в соединении. Вместе с тем, MEGACO не определяет протокола для синхронизации работы контроллеров. В ряде работ, посвященных исследованию возможностей протокола MGCP, для этой цели предлагается использовать протоколы H.323, SIP или ISUP/IP.

Сообщения протокола MGCP переносятся протоколом без гарантированной доставки сообщений UDP. Рабочая группа SIGTRAN комитета IETF в настоящее время разрабатывает механизм взаимодействия контроллера шлюзов и шлюза сигнализации.

Шлюз сигнализации должен принимать поступающие из ТФОП пакеты трех нижних уровней системы сигнализации ОКС7 (уровней подсистемы переноса сообщений МТР) и передавать сигнальные сообщения верхнего, пользовательского, уровня к контроллеру шлюзов. Шлюз сигнализации также должен уметь передавать по IP-сети приходящие из ТФОП сигнальные сообщения Q.931.

Основное внимание рабочей группы SIGTRAN уделяется вопросам разработки наиболее эффективного механизма передачи сигнальной информации по IP-сетям. Следует отметить, что существует несколько причин, по которым пришлось отказаться от использования для этой цели протокола TCP. Рабочая группа SIGTRAN предлагает использовать для передачи сигнальной информации протокол Stream Control Transport Protocol (SCTP), имеющий ряд преимуществ перед протоколом TCP, основным из которых является значительное снижение времени доставки сигнальной информации и, следовательно, времени установления соединения – одного из важнейших параметров качества обслуживания.

Если в ТФОП используется сигнализация по выделенным сигнальным каналам (ВСК), то сигналы сначала поступают вместе с пользовательской информацией в транспортный шлюз, а затем передаются в контроллер шлюзов без посредничества шлюза сигнализации.

Отметим, что протокол MGCP является внутренним протоколом для обмена информацией между функциональными блоками распределенного шлюза, который извне представляется одним шлюзом. Протокол MGCP является master/slave протоколом. Это означает, что контроллер шлюзов является ведущим, а сам шлюз – ведомым устройством, которое должно выполнять все команды, поступающие от контроллера Call Agent.

Вышеописанное решение обеспечивает масштабируемость сети и простоту управления сетью через контроллер шлюзов. Шлюзы не

должны быть интеллектуальными устройствами, требуют меньшей производительности процессоров и, следовательно, становятся менее дорогими. Кроме того, очень быстро вводятся новые протоколы сигнализации или дополнительные услуги, так как эти изменения затрагивают только контроллер шлюзов, а не сами шлюзы.

Третий подход, предлагаемый организацией IETF (рабочая группа MEGACO), хорошо подходит для развертывания глобальных сетей IP-телефонии, приходящих на смену традиционным телефонным сетям. Более подробная информация о протоколе MGCP приведена в главе 8.

Рассмотрим алгоритмы установления и разрушения соединения с использованием протокола MGCP. Первый пример охватывает взаимодействие протокола MGCP с протоколом OKC7 (рис. 1.13).

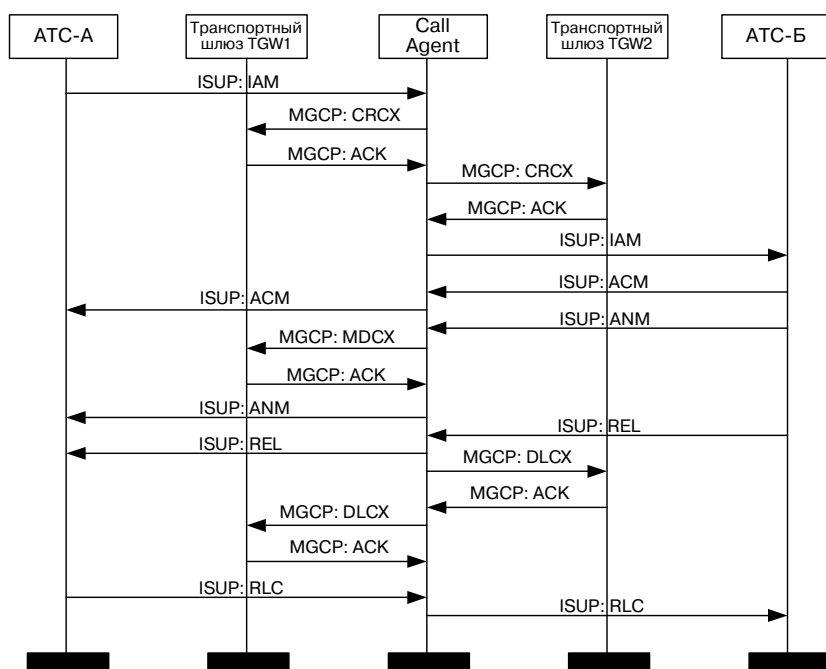


Рис. 1.13 Установление и разрушение соединения с использованием протокола MGCP (Пример 1)

1. От телефонной станции ATC-A к шлюзу сигнализации SG1 по общему каналу сигнализации поступает запрос соединения в виде сообщения IAM протокола ISUP [6]. На рис. 1.13 шлюз сигнализации SG1 и SG2 совмещены с транспортными шлюзами TGW1 и TGW2 соответственно. Шлюз SG1 передает сообщение IAM к контроллеру шлюзов, который обрабатывает запрос и определяет, что вызов должен быть направлен к ATC-B посредством шлюза TGW2.

2. Контроллер резервирует порт шлюза TGW1 (разговорный канал). С этой целью он передает к шлюзу команду CreateConnection. Отметим, что порт шлюза TGW1 может только принимать информацию (режим «recvonly»), так как он еще не осведомлен о том, по какому адресу и каким образом ему следует передавать информацию.
3. В ответе на эту команду шлюз TGW1 возвращает описание параметров сеанса связи.
4. Приняв ответ шлюза TGW1, контроллер передает команду CRCX второму шлюзу TGW2 с целью зарезервировать порт в этом шлюзе.
5. Шлюз TGW 2 выбирает порт, который будет участвовать в соединении, и подтверждает прием команды CRCX. При помощи двух команд CRCX создается односторонний разговорный канал для передачи вызываемому абоненту акустических сигналов или речевых подсказок и извещений. В то же время, порт шлюза TGW2 уже может не только принимать, но и передавать информацию, так как он получил описание параметров связи от встречного шлюза.
6. Далее контроллер шлюзов передает сообщение IAM к АТС-Б.
7. На сообщение IAM станция АТС-Б отвечает подтверждением ACM, которое немедленно пересылается к станции АТС-А.
8. После того как вызываемый абонент примет вызов, АТС-Б передает к контроллеру шлюзов сообщение ANM.
9. Далее контроллер заменяет в шлюзе TGW1 режим «recvonly» на полнодуплексный режим при помощи команды MDCX.
10. Шлюз TGW1 выполняет и подтверждает изменение режима.
11. Контроллер передает сообщение ANM к АТС-А, после чего начинается разговорная фаза соединения.
12. Завершение разговорной фазы происходит следующим образом. В нашем случае вызвавший абонент Б дает отбой первым. АТС-Б передает через шлюз сигнализации сообщение REL к контроллеру шлюзов.
13. Приняв сообщение REL, контроллер шлюзов завершает соединение с вызванным абонентом.
14. Шлюз подтверждает завершение соединения и передает к контроллеру собранные за время соединения статистические данные.
15. Контроллер шлюзов передает сообщение RLC к АТС-Б с целью подтвердить разъединение.

16. Параллельно контроллер завершает соединение с вызвавшей стороной
17. Шлюз TGW1 подтверждает завершение соединения и передает к контроллеру собранные за время соединения статистические данные.
18. АТС-А подтверждает завершение соединения передачей сообщения RLC, после чего соединение считается разрушенным.

Второй пример иллюстрирует взаимодействие протокола MGCP с протоколами ОКС7 и H.323 (рис. 1.14).

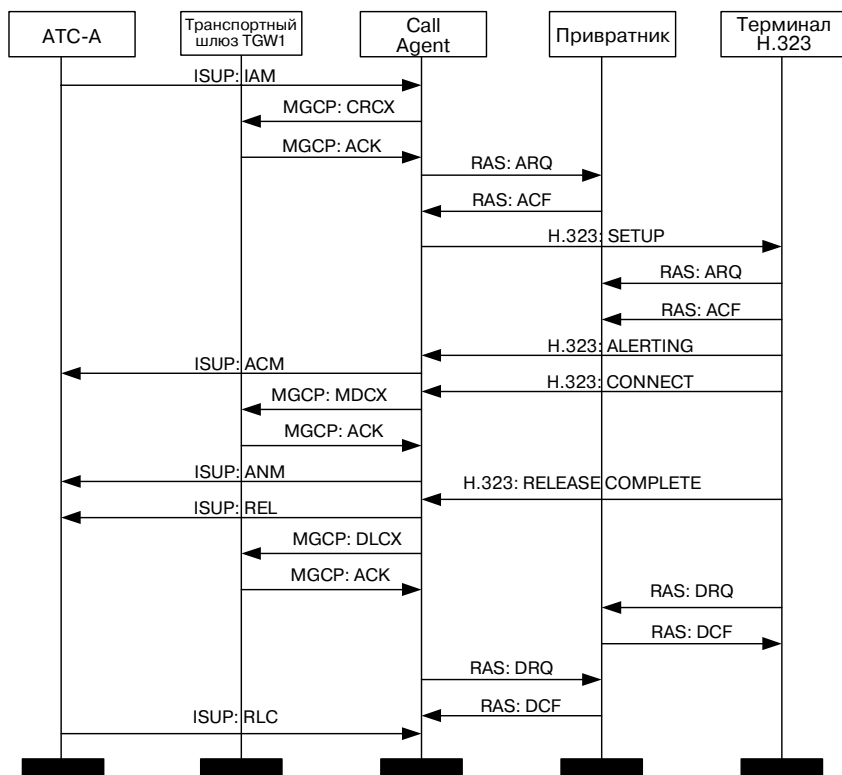


Рис. 1.14 Установление и разрушение соединения с использованием протокола MGCP (Пример 2)

1. С телефонной станции АТС-А к шлюзу сигнализации SG1 по общему каналу сигнализации поступает запрос соединения (сообщение IAM). На рис. 1.14 шлюз сигнализации SG1 также совмещен с транспортным шлюзом TGW1. Шлюз SG1 передает сообщение IAM контроллеру шлюзов, который обрабатывает запрос и определяет, что вызов должен быть направлен к окончательному устройству вызываемого пользователя – терминалу H.323.

2. Контроллер шлюзов резервирует порт шлюза TGW1 (разговорный канал). С этой целью он передает к шлюзу команду CreateConnection. И в этом примере порт шлюза TGW1 может только принимать информацию (режим «recvonly»).
3. В ответе на принятую команду шлюз TGW1 возвращает описание параметров связи.
4. Приняв ответ от шлюза TGW1, контроллер передает к привратнику сети H.323 сообщение ARQ с alias адресом вызываемого абонента.
5. В ответ на сообщение ARQ привратник передает сообщение ACF с указанием транспортного адреса своего сигнального канала.
6. Контроллер передает запрос соединения SETUP на транспортный адрес сигнального канала привратника, при этом используется процедура Fast Start. Привратник пересылает сообщение SETUP к вызываемому терминалу.
7. Вызываемый терминал передает запрос допуска к ресурсам сети ARQ.
8. В ответ на запрос ARQ привратник передает подтверждение запроса ACF.
9. Вызываемый терминал передает сообщение ALERTING, которое привратник маршрутизирует к контроллеру шлюзов. При этом вызываемому пользователю подается визуальный или акустический сигнал о входящем вызове, а вызывающему пользователю подается индикация того, что вызываемый пользователь не занят и получает сигнал о вызове.
10. Контроллер преобразует сообщение ALERTING в сообщение ACM, которое немедленно пересылается к АТС-А.
11. После того как вызываемый пользователь примет входящий вызов, контроллер получит сообщение CONNECT.
12. Контроллер шлюзов меняет в шлюзе TGW1 режим «recvonly» на полнодуплексный режим.
13. Шлюз TGW1 выполняет и подтверждает изменение режима соединения.
14. Контроллер передает сообщение ANM к АТС-А, после чего начинается разговорная фаза соединения, в ходе которой оборудование вызвавшего пользователя передает речевую информацию, упакованную в пакеты RTP/UDP/IP, на транспортный адрес RTP-канала терминала вызванного абонента, а тот передает пакетированную речевую информацию на транспортный адрес RTP-канала терминала вызвавшего абонента. При помощи канала RTCP ведется контроль передачи информации по RTP каналу.

15. После окончания разговорной фазы начинается фаза разрушения соединения. Оборудование пользователя, инициирующего разрушение соединения, должно прекратить передачу речевой информации, закрыть логические каналы и передать сообщение RELEASE COMPLETE, после чего сигнальный канал закрывается.
16. Контроллер шлюзов передает сообщение RELEASE к ATC-A с целью завершения соединения.
17. Кроме того, контроллер передает к шлюзу команду DLCX.
18. Шлюз подтверждает завершение соединения и передает к контроллеру собранные за время соединения статистические данные.
19. После вышеописанных действий контроллер и окончное оборудование извещают привратник об освобождении занимавшейся полосы пропускания. С этой целью каждый из участников соединения посылает привратнику по каналу RAS запрос выхода из соединения DRQ, на который привратник должен передать подтверждение DCF.
20. От ATC-A приходит подтверждение разъединения RLC, после чего соединение считается разрушенным.

Следует заметить, что алгоритм взаимодействия протоколов SIP и MGCP не сильно отличается от вышеописанного алгоритма.

Рабочая группа MEGACO комитета IETF продолжает работу по усовершенствованию протокола управления шлюзами, в рамках которой разработан более функциональный, чем MGCP, протокол MEGACO.

Международный союз электросвязи в проекте версии 4 рекомендации H.323 ввел принцип декомпозиции шлюзов. Управление функциональными блоками распределенного шлюза будет осуществляться контроллером шлюза – Media Gateway Controller – при помощи адаптированного к H.323 протокола MEGACO, который в рекомендации H.248 назван Gateway Control Protocol.

Сообщения протокола MEGACO отличаются от сообщений протокола MGCP, но процедуры установления и разрушения соединений с использованием обоих протоколов идентичны, поэтому описание процедуры установления соединения на базе протокола MEGACO здесь не приводится. Эти процедуры, вместе с детальным анализом протокола MEGACO, рассматриваются в главе 9.

1.5.4 Сравнение подходов к построению сети IP-телефонии

В настоящее время для построения хорошо функционирующих и совместимых с ТФОП сетей IP-телефонии подходят протоколы H.323 и MGCP. Как уже отмечалось, протокол SIP несколько хуже взаимодействует с системами сигнализации, используемыми в ТФОП (сравнительный анализ протоколов H.323 и SIP приведен в главе 7).

Подход, основанный на использовании протокола MGCP, обладает весьма важным преимуществом перед подходом, предложенным ITU в рекомендации Н.323: поддержка контроллером шлюзов сигнализации ОКС7 и других видов сигнализации, а также прозрачная трансляция сигнальной информации по сети IP-телефонии. В сети, построенной на базе рекомендации Н.323, сигнализация ОКС7, как и любая другая сигнализация, конвертируется шлюзом в сигнальные сообщения Н.225.0 (Q.931).

Основным недостатком третьего из приведенных в данном параграфе подходов является незаконченность стандартов. Функциональные составляющие распределенных шлюзов, разработанные разными фирмами-производителями телекоммуникационного оборудования, практически несовместимы. Функции контроллера шлюзов точно не определены. Не стандартизированы механизмы переноса сигнальной информации от шлюза сигнализации к контроллеру и в обратном направлении. К недостаткам можно отнести также отсутствие стандартизированного протокола взаимодействия между контроллерами. Кроме того, протокол MGCP является протоколом управления шлюзами, но не предназначен для управления соединениями с участием терминального оборудования пользователей (IP-телефонов). Это означает, что в сети, построенной на базе протокола MGCP, для управления терминальным оборудованием должен присутствовать привратник или сервер SIP.

Стоит также отметить, что в существующих приложениях IP-телефонии, таких как предоставление услуг международной и междугородной связи, использовать протокол MGCP (так же, как и протокол SIP) нецелесообразно в связи с тем, что подавляющее количество сетей IP-телефонии сегодня построено на базе протокола Н.323. Оператору придется строить отдельную сеть IP-телефонии на базе протокола MGCP (или SIP), что связано со значительными капиталовложениями. В то же время, оператор связи, имеющий оборудование стандарта Н.323, может присоединиться к существующим сетям IP-телефонии.

В последнем из упомянутых подходов (в проекте версии 4 рекомендации Н.323) ITU-T ввел принцип декомпозиции шлюзов, использованный в третьем подходе. Управление функциональными блоками распределенного шлюза будет осуществляться контроллером шлюза – MGC (Media Gateway Controller) при помощи протокола MEGACO/Н.248. В проекте версии 4 рекомендации Н.323 предусмотрена также возможность прозрачной передачи сигнализации ОКС7 и других видов сигнализации по сетям IP-телефонии и обработка сигнализации всех видов привратником без преобразования в сигнальные сообщения Н.225.0.

Приведенных в этой главе сведений отнюдь не достаточно для окончательных выводов относительно перспектив использования того или другого протокола IP-телефонии, хотя первое впечатление уже может сложиться. В следующих главах авторы постараются представить более глубокие сведения по данной тематике, однако обязуются не навязывать читателю какую-либо одну точку зрения, а дать ему все необходимое для того, чтобы он мог сам сделать надлежащие выводы.

Глава 2

Сетевые аспекты IP-телефонии

2.1 Три основных сценария IP-телефонии

Материал предыдущей главы дал в первом приближении ответ на вопрос: *что такое IP-телефония?* Прежде чем обсудить более подробно различные подходы к архитектуре, протоколам и вариантам построения систем и оборудования, полезно обратить внимание на другой вопрос: *для чего нужна IP-телефония?* В качестве ответа на этот вопрос рассмотрим три наиболее часто используемых сценария IP-телефонии:

- «компьютер-компьютер»;
- «компьютер-телефон»;
- «телефон-телефон».

Сценарий «компьютер-компьютер» реализуется на базе стандартных компьютеров, оснащенных средствами мультимедиа и подключенных к сети Интернет.

Компоненты модели IP-телефонии по сценарию «компьютер-компьютер» показаны на рис. 2.1. В этом сценарии аналоговые речевые сигналы от микрофона абонента А преобразуются в цифровую форму с помощью аналого-цифрового преобразователя (АЦП), обычно при 8000 отсчетов/с, 8 битов/отсчет, в итоге – 64 Кбит/с. Отсчеты речевых данных в цифровой форме затем сжимаются кодирующим устройством для сокращения нужной для их передачи полосы в отношении 4:1, 8:1 или 10:1. Алгоритмы сжатия речи подробно рассматриваются в следующей главе. Выходные данные после сжатия формируются в пакеты, к которым добавляются заголовки протоколов, после чего пакеты

передаются через IP-сеть в систему IP-телефонии, обслуживающую абонента Б. Когда пакеты принимаются системой абонента Б, заголовки протокола удаляются, а сжатые речевые данные поступают в устройство, развертывающее их в первоначальную форму, после чего речевые данные снова преобразуются в аналоговую форму с помощью цифроаналогового преобразователя (ЦАП) и попадают в телефон абонента Б. Для обычного соединения между двумя абонентами системы IP-телефонии на каждом конце одновременно реализуют как функции передачи, так и функции приема. Под IP-сетью, изображенной на рис. 2.1, подразумевается либо глобальная сеть Интернет, либо корпоративная сеть предприятия Intranet. Описанию протоколов, используемых в IP-сетях, в том числе протоколов передачи речевой информации по IP-сети, посвящена глава 4.

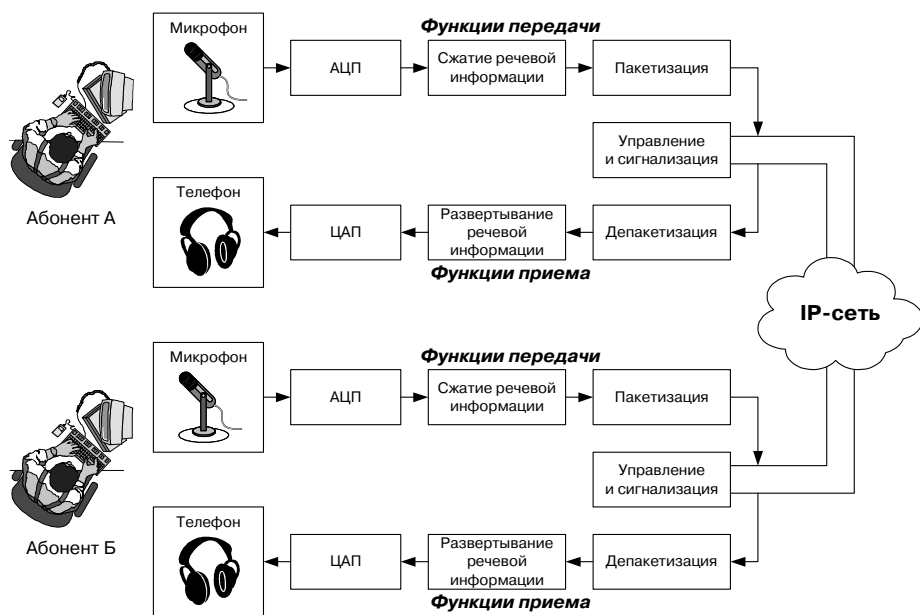


Рис. 2.1 Сценарий IP-телефонии "компьютер-компьютер"

Для поддержки сценария «компьютер – компьютер» поставщику услуг Интернет желательно иметь отдельный сервер (привратник), преобразующий имена пользователей в динамические адреса IP. Сам сценарий ориентирован на пользователя, которому сеть нужна, в основном, для передачи данных, а программное обеспечение IP-телефонии требуется лишь иногда для разговоров с коллегами. Эффективное использование телефонной связи по сценарию «компьютер-компьютер» обычно связано с повышением продуктивности работы крупных компаний, например, при организации виртуальной презентации в корпоративной сети с возможностью не только видеть доку-

менты на Web-сервере, но и обсуждать их содержание с помощью IP-телефона. При этом между двумя IP-сетями могут использоваться элементы ТфОП, а идентификация вызываемой стороны может осуществляться как на основе E.164, так и на основе IP-адресации. Наиболее распространенным программным обеспечением для этих целей является пакет Microsoft NetMeeting, доступный для бесплатной загрузки с узла Microsoft.

Рассмотрим представленный на рис. 2.1 сценарий установления соединения «компьютер-компьютер» более подробно.

Для проведения телефонных разговоров друг с другом абоненты А и Б должны иметь доступ к Интернет или к другой сети с протоколом IP. Предположим, что такая IP-сеть существует, и оба абонента подключены к ней. Рассмотрим возможный алгоритм организации связи между этими абонентами.

1. Абонент А запускает свое приложение IP-телефонии, поддерживающее протокол H.323.

2. Абонент Б уже заранее запустил свое приложение IP-телефонии, поддерживающее протокол H.323.

3. Абонент А знает доменное имя абонента Б элемент системы имен доменов – Domain Name System (DNS), вводит это имя в раздел «кому позвонить» в своем приложении IP-телефонии и нажимает кнопку Return.

4. Приложение IP-телефонии обращается к DNS-серверу (который в данном примере реализован непосредственно в персональном компьютере абонента А) для того, чтобы преобразовать доменное имя абонента Б в IP-адрес.

5. Сервер DNS возвращает IP-адрес абонента Б.

6. Приложение IP-телефонии абонента А получает IP-адрес абонента Б и отправляет ему сигнальное сообщение H.225 Setup.

7. При получении сообщения H.225 Setup приложение абонента Б сигнализирует ему о входящем вызове.

8. Абонент Б принимает вызов и приложение IP-телефонии отправляет ответное сообщение H.225 Connect.

9. Приложение IP-телефонии у абонента А начинает взаимодействие с приложением у абонента Б в соответствии с рекомендацией H.245.

10. После окончания взаимодействия по протоколу H.245 и открытия логических каналов абоненты А и Б могут разговаривать друг с другом через IP-сеть.

Несмотря на нарочитую простоту изложения, рассмотренный пример довольно сложен, что обусловлено сложностью технологии IP-те-

лефонии. В этом примере не показаны все шаги и опущены весьма существенные детали, которые необходимы поставщику услуг для развертывания сети IP-телефонии. Обо всех этих более сложных моментах будет сказано в главах 5 – 11 данной книги, а здесь сделаем еще одно упрощение.

Сам характер сценария «компьютер-компьютер» на рис. 2.1 обуславливает сосредоточение всех необходимых функций IP-телефонии в персональном компьютере или другом аналогичном устройстве конечного пользователя. При описании других сценариев в этой главе вместо громоздкого изображения компонентов оконечного устройства будет приводиться только упрощенное изображение терминала IP-телефонии. Таким аналогом рис. 2.1 является упрощенное представление того же сценария на рис. 2.2. К детальному рассмотрению процедур аналогово-цифрового и цифро-аналогового преобразования, сжатия, пакетизации и др. мы вернемся в следующей главе.



Рис. 2.2 Упрощенный сценарий IP-телефонии
"компьютер-компьютер" (аналог рис.2.1)

Замена изображений имеет и более глубокий смысл. Название сценария «компьютер – компьютер» отнюдь не означает, что в распоряжении пользователя обязательно должен быть стандартный РС с микрофоном и колонками, как это представлено на рис. 2.1. Главным требованием для такой схемы является то, что оба пользователя должны иметь подключенные к сети персональные компьютеры. И эти РС должны быть всегда включены, подсоединены к сети и иметь в запущенном виде программное обеспечение IP-телефонии для приема входящих вызовов. При всем этом должна быть полная совместимость между программно-аппаратными средствами IP-телефонии, полученными от разных поставщиков, т.е. пользователи, желающие разговаривать друг с другом, должны иметь идентичное программное обеспечение, например, реализующее протокол H.323.

Принимая во внимание эти обстоятельства, под названием «компьютер» во всех сценариях мы будем понимать терминал пользователя, включенный в IP-сеть, а под названием «телефон» – терминал пользователя, включенный в сеть коммутации каналов любого типа: ТФОП, ISDN или GSM.

И еще одно, более существенное замечание. До сих пор в обсуждении сценария «компьютер – компьютер» на рис. 2.1 и 2.2 полага-

лось, что оба пользователя включены в одну и ту же IP-сеть (Интернет, Интранет или другую сеть с протоколом IP). В рамках проекта TIPHON, которому посвящен следующий параграф этой главы, рассматривается другая, более сложная модификация сценария «компьютер – компьютер». Эта модификация, представленная на рис. 2.3, предусматривает организацию связи между абонентами IP-сети с учетом того, что вызов транзитом проходит через сеть коммутации каналов (СКК). Заметим, что на этом и на следующих рисунках в качестве СКК выступает телефонная сеть общего пользования (ТфОП), хотя излагаемые в данной главе материалы справедливы для ISDN, GSM и др.

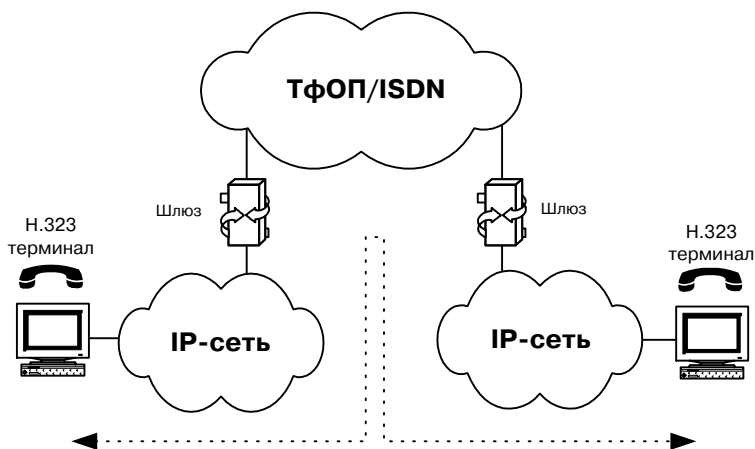


Рис. 2.3 Упрощенный сценарий IP-телефонии «компьютер-компьютер». Соединение пользователей IP-сетей через транзитную СКК

Следующий сценарий – «телефон-компьютер» – находит применение в разного рода справочно-информационных службах Интернет, в службах сбыта товаров или в службах технической поддержки. Пользователь, подключившийся к серверу WWW какой-либо компании, имеет возможность обратиться к оператору справочной службы. Этот сценарий в ближайшие несколько лет будет, по всей вероятности, более активно востребован деловым сектором. Компании будут использовать данную технологию для наращивания своих Web-страниц (и своего присутствия во всемирной паутине). Пользователи компьютеров смогут просматривать в «реальном времени» каталоги, почти мгновенно заказывать товары и получать множество других услуг. Это вполне соответствует стилю жизни современных потребителей, связанному с потребностью в дополнительных удобствах и экономии времени. Уже сегодня осознаются все выгоды и удобства централизованного приобретения предметов широкого потребления (например, компакт-дисков, книг, программного обеспечения и т. д.) и уже привычно совершаются операции электронной коммерции.

В рамках проекта TIPHON рассматриваются две модификации этого сценария IP-телефонии:

- от компьютера (пользователя IP-сети) к телефону (абоненту ТфОП), в частности, в связи с предоставлением пользователям IP-сетей доступа к телефонным услугам, в том числе, к справочно-информационным услугам и к услугам Интеллектуальной сети;
- от абонента ТфОП к пользователю IP-сети с идентификацией вызываемой стороны на основе нумерации по E.164 или IP-адресации.

Проект TIPHON заслуживает более пристального внимания, и уже было обещано посвятить ему целиком следующий параграф этой главы.

В первой из упомянутых модификаций сценария «компьютер – телефон» соединение устанавливается между пользователем IP-сети и пользователем сети коммутации каналов (рис. 2.4). Предполагается, что установление соединения инициирует пользователь IP-сети.

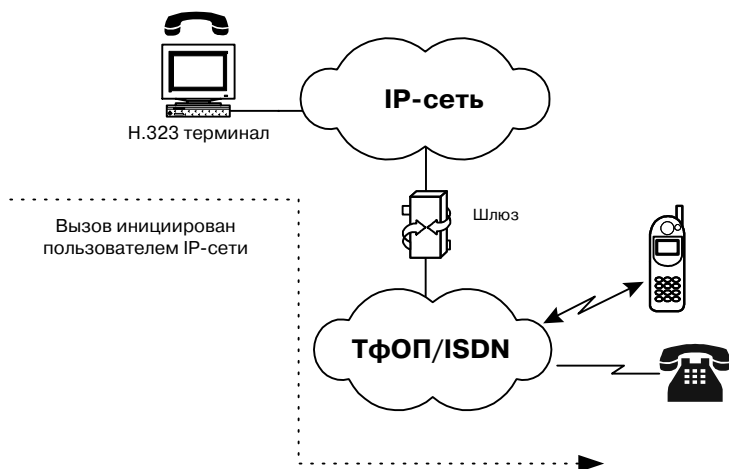


Рис. 2.4 Вызов абонента ТфОП пользователем IP-сети по сценарию "компьютер - телефон"

Шлюз (GW) для взаимодействия сетей ТфОП и IP может быть реализован в отдельном устройстве или интегрирован в существующее оборудование ТфОП или IP-сети. Показанная на рисунке сеть СКК может быть корпоративной сетью или сетью общего пользования.

В соответствии со второй модификацией сценария «компьютер – телефон» соединение устанавливается между пользователем IP-сети и абонентом ТфОП, но инициирует его создание абонент ТфОП (рис. 2.5).

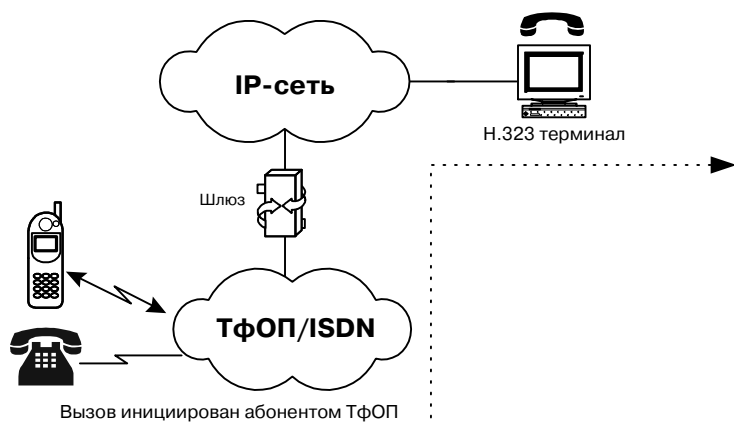


Рис. 2.5 Пользователя IP-сети вызывает абонент ТФОП по сценарию "компьютер - телефон"

Рассмотрим несколько подробнее пример представленной на рис. 2.5 упрощенной архитектуры системы IP-телефонии по сценарию «телефон-компьютер». При попытке вызвать справочно-информационную службу, используя услуги пакетной телефонии и обычный телефон, на начальной фазе абонент А вызывает близлежащий шлюз IP-телефонии. От шлюза к абоненту А поступает запрос ввести номер, к которому должен быть направлен вызов (например, номер службы), и личный идентификационный номер (PIN) для аутентификации и последующего начисления платы, если это служба, вызов которой оплачивается вызывающим абонентом. Основываясь на вызываемом номере, шлюз определяет наиболее доступный путь к данной службе. Кроме того, шлюз активизирует свои функции кодирования и пакетизации речи, устанавливает контакт со службой, ведет мониторинг процесса обслуживания вызова и принимает информацию о состояниях этого процесса (например, занятость, посылка вызова, разъединение и т.п.) от исходящей стороны через протокол управления и сигнализации. Разъединение с любой стороны передается противоположной стороне по протоколу сигнализации и вызывает завершение установленных соединений и освобождение ресурсов шлюза для обслуживания следующего вызова.

Для организации соединений от службы к абонентам (рис. 2.4) используется аналогичная процедура. Популярными программными продуктами для этого варианта сценария IP-телефонии «компьютер-телефон» являются IDT Net2Phone и DotDialer, организующие вызовы к обычным абонентским телефонным аппаратам в любой точке мира.

Эффективность объединения услуг передачи речи и данных является основным стимулом использования IP-телефонии по сценариям «компьютер-компьютер» и «компьютер-телефон», не нанося при

этом никакого ущерба интересам операторов традиционных телефонных сетей.

Сценарий «телефон-телефон» в значительной степени отличается от остальных сценариев IP-телефонии своей социальной значимостью, поскольку целью его применения является предоставление обычным абонентам ТфОП альтернативной возможности междугородной и международной телефонной связи. В этом режиме современная технология IP-телефонии предоставляет виртуальную телефонную линию через IP-доступ.

Как правило, обслуживание вызовов по такому сценарию IP-телефонии выглядит следующим образом. Поставщик услуг IP-телефонии подключает свой шлюз к коммутационному узлу или станции ТфОП, а по сети Интернет или по выделенному каналу соединяется с аналогичным шлюзом, находящимся в другом городе или другой стране.

Типичная услуга IP-телефонии по сценарию «телефон-телефон» использует стандартный телефон в качестве интерфейса пользователя, а вместо междугородного компонента ТфОП использует либо частную IP-сеть/Intranet, либо сеть Интернет. Благодаря маршрутизации телефонного трафика по IP-сети стало возможным обходить сети общего пользования и, соответственно, не платить за междугородную/международную связь операторам этих сетей.

Следует отметить, что сама идея использовать альтернативные транспортные механизмы для обхода сети ТфОП не является новой. Достаточно вспомнить статистические мультиплексоры, передачу речи по сети Frame Relay или оборудование передачи речи по сети ATM.

Как показано на рис. 2.6, поставщики услуг IP-телефонии предоставляют услуги «телефон-телефон» путём установки шлюзов IP-телефонии на входе и выходе IP-сетей. Абоненты подключаются к шлюзу поставщика через ТфОП, набирая специальный номер доступа. Абонент получает доступ к шлюзу, используя персональный идентификационный номер (PIN) или услугу идентификации номера вызывающего абонента (Calling Line Identification). После этого шлюз просит ввести телефонный номер вызываемого абонента, анализирует этот номер и определяет, какой шлюз имеет лучший доступ к нужному телефону. Как только между входным и выходным шлюзами устанавливается контакт, дальнейшее установление соединения к вызываемому абоненту выполняется выходным шлюзом через его местную телефонную сеть.

Полная стоимость такой связи будет складываться для пользователя из расценок ТфОП на связь с входным шлюзом, расценок Интернет-провайдера на транспортировку и расценок удалённой ТфОП на связь выходного шлюза с вызванным абонентом.

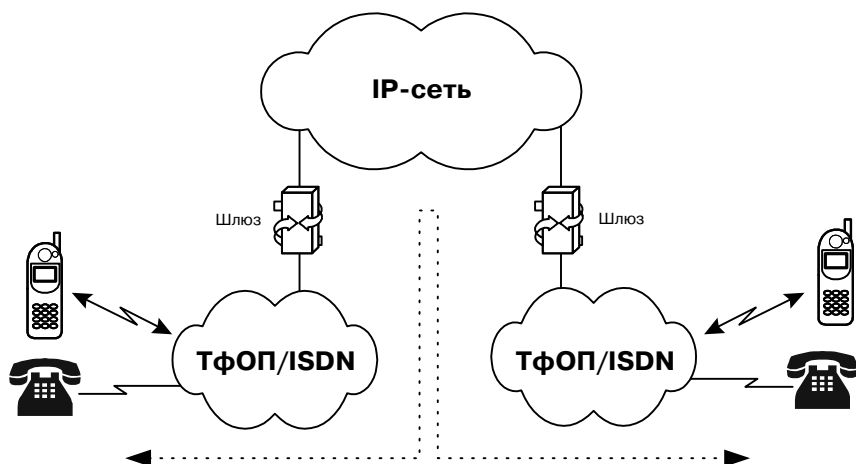


Рис. 2.6 Соединение абонентов ТфОП через транзитную IP-сеть по сценарию "телефон-телефон"

Одним из алгоритмов организации связи по сценарию «телефон-телефон» является выпуск поставщиком услуги своих телефонных карт. Имея такую карту, пользователь, желающий позвонить в другой город, набирает номер данного поставщика услуги, затем в режиме донабора вводит свой идентификационный номер и PIN-код, указанный на карте. После процедуры аутентификации он набирает телефонный номер адресата.

Возможны и другие алгоритмы реализации этого сценария: вместо телефонной карты может использоваться информация об альтернативном счете. Счет для оплаты может быть выслан абоненту и после разговора, аналогично тому, как это делается при междугородном соединении в ТфОП.

Рассмотренные выше сценарии сведены в таблице 2.1.

Таблица 2.1 Варианты межсетевого взаимодействия

Сценарий	Входящая сеть	Транзитная сеть	Исходящая сеть	Примечание
«компьютер – компьютер»	IP	IP	IP	Рис. 2.1 и 2.2
	IP	ТфОП	IP	Рис. 2.3
«компьютер – телефон»	IP	ТфОП	ТфОП	Рис. 2.5
	ТфОП	IP	IP	Рис. 2.4
	ТфОП	ТфОП	IP	Рис. 2.4
	IP	IP	ТфОП	Рис. 2.5
«телефон – телефон»	ТфОП	IP	ТфОП	Рис. 2.6
	ТфОП	ТфОП	ТфОП	Не рассм.

Из представленных в таблице девяти вариантов трех сценариев последний вариант остается за рамками данной книги по вполне очевидной причине – его принадлежности к классической (а не к IP-) телефонии, описанной в многих десятках других книг.

Следующий параграф посвящен анализу проекта TIPHON Европейского института стандартизации в области телекоммуникаций – Europe Telecommunications Standardization Institute (ETSI). Именно этот институт вплотную занимается сетевыми вопросами IP-телефонии, в то время как другие стандартизирующие телекоммуникационные организации основное внимание уделяют вопросам разработки протоколов сигнализации или механизмов переноса речевой информации по сетям с маршрутизацией пакетов IP. Так, например, область деятельности основоположников IP-телефонии ITU-T и IETF ограничивается только сетями с маршрутизацией пакетов IP. Вопросы взаимодействия телефонных и IP сетей рассматривались ITU-T, в основном, в части преобразования систем сигнализации [Н.246] и практически не затрагивались комитетом IETF. Более подробно деятельность ITU-T в области IP-телефонии освещена в главах 5 и 6, посвященных архитектуре и протоколам H.323, а результаты деятельности комитета IETF в этой же области рассмотрены в главах 7, 8 и 9.

В проекте TIPHON предполагается разработка новых стандартов и профилей существующих стандартов для каждого из приведенных в таблице 2.1 сценариев. Новые стандарты будут разрабатываться только для тех областей связи, для которых действующие стандарты отсутствуют. Там, где существуют действующие стандарты ETSI, ITU или других стандартизирующих организаций, будет проводиться разработка и преобразование профилей этих стандартов.

2.2 Проект TIPHON

Работа над проектом TIPHON (Telecommunication and Internet Protocol Harmonization over Networks) была начата институтом ETSI в апреле 1997 г. Основная задача проекта – решение проблем взаимодействия между сетями с маршрутизацией пакетов IP и сетями с коммутацией каналов в части поддержки прозрачной передачи речевой и факсимильной информации. Под сетями с коммутацией каналов подразумеваются ТфОП, ISDN и GSM.

В проекте принимают участие свыше 40 крупнейших телекоммуникационных компаний. Имеется восемь рабочих групп, последняя из которых – по защите информации – была организована во время 15-го совещания рабочих групп 4-8 октября 1999 г. в Лейпциге. Результатом деятельности рабочих групп TIPHON являются технические спецификации и отчеты.

Сама идея проекта TIPHON родилась под влиянием динамично развивающегося рынка телекоммуникационных услуг, предоставляемых как операторами сетей связи, базирующихся на технологии коммутации каналов, так и операторами сетей, построенных на основе технологии маршрутизации пакетов IP. Задачей проекта является претворение в жизнь идеологии конвергенции и создание единой сетевой инфраструктуры, привлекательной для операторов различных видов связи.

Была отмечена растущая потребность в организации связи в реальном времени, в том числе, телефонной связи, в сетях, реализующих технологию маршрутизации пакетов IP. Для удовлетворения этой потребности институт ETSI предлагает в проекте TIPHON концепцию «сети сетей», такой, что сети, входящие в ее состав, могут базироваться на технологиях коммутации каналов и маршрутизации пакетов IP (рис. 2.7).

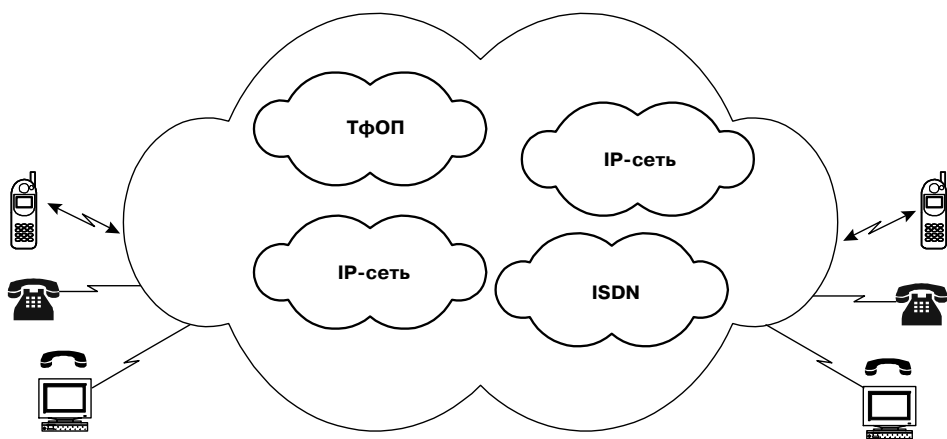


Рис. 2.7 Обобщенная структура сети TIPHON

В рамках проекта TIPHON сети, использующие различные технологии коммутации, имеют статус доменов «глобальной сети». В основу взаимодействия этих доменов положено обеспечение гарантированного качества обслуживания (QoS) и защиты межсетевых соединений. Кроме того, обеспечивается возможность управлять соединениями, используя стандартные протоколы сигнализации. Таким образом, сеть TIPHON можно определить как сеть высшего уровня, поддерживающую предоставление услуг телефонной связи и базирующуюся на совокупности сетей более низкого уровня.

В основу проекта TIPHON положены следующие правила:

- терминалами TIPHON могут быть персональные компьютеры и обычные телефоны;
- интерфейс «человек-машина» (MMI) строится по аналогии с телефонным интерфейсом;

- пользователи могут менять точки доступа к услугам глобальной сети; при этом должен сохраняться набор предоставляемых услуг и качество обслуживания (QoS).

Главной целью проекта TIPHON является разработка механизмов взаимодействия и связанных с ними параметров для обеспечения мультимедийной связи с гарантированным качеством обслуживания между пользователями сетей с коммутацией каналов и сетей с маршрутизацией пакетов. При этом акцент делается на взаимодействие сетей, а не на отдельные сети, для чего и создаются соответствующие спецификации и стандарты, ориентированные на промышленные предприятия, операторские компании, администрации связи, органы сертификации и стандартизации и др.

Проект TIPHON предусматривает решение ряда технических задач, связанных с обеспечением приемлемого качества услуг телефонной связи. В число этих задач входит разработка эталонных конфигураций и функциональных моделей, требований к взаимодействию различных функциональных объектов, процедур управления соединением и протоколов; преобразование адресов в формате E.164 в IP-адреса; рассмотрение технических аспектов защиты; изучение вопросов мобильности и обеспечения качества обслуживания. Ранее указывалось, что в работе над проектом TIPHON участвуют несколько групп, каждая из которых отвечает за решение определенной задачи. Ниже представлены основные направления деятельности рабочих групп TIPHON.

Разработка требований к единой межсетевой политике, определяющих выявление неисправностей, выбор уровня качества обслуживания, поддержку необходимой сигнализации и передачи акустических сигналов, трассировку соединения, идентификацию вызывающего абонента.

Разработка эталонных конфигураций и функциональных моделей, включая функциональную модель шлюза между IP-сетями и сетями с коммутацией каналов, а также спецификацию интерфейсов шлюза. Модели должны отражать все аспекты функциональности шлюзов, в том числе взаимодействие с привратниками и с Интеллектуальными сетями.

Разработка процедур обработки вызовов и протоколов, алгоритмов установления и разрушения соединения, процедур обнаружения привратника, регистрации оконечного оборудования, аутентификации пользователя. Здесь же рассматриваются вопросы использования DTMF-сигнализации и специфицируются функции транспортного уровня.

Преобразование адреса в формате E.164 в IP-адрес. Пользователям IP-сетей, как правило, адреса выделяются динамически, поэтому идентифицировать пользователей по их IP-адресам невозможно. Необходимо разработать новый механизм адресации, обеспечивающий технологическую прозрачность при преобразовании номера E.164 в IP-адрес.

Технические аспекты начисления платы и выставления счетов. Должны быть предусмотрены следующие формы оплаты: кредит, дебет, оплата при помощи кредитной карты, оплата вызываемой стороной. При этом должны учитываться следующие параметры: тип услуги, длительность связи, время суток.

Технические аспекты защиты. К ним относится первичная защита сети от случайных или умышленных повреждений. Здесь же рассматривается защита информации и доступа, а также связанные с этим вопросы сигнализации, нагрузки, аутентификации, авторизации, шифрования и секретности вызова.

Вопросы качества обслуживания. Конечный пользователь ожидает, что услуга передачи речевой информации будет предоставляться с хорошим качеством и высокой надежностью. Но такие примеры, как предоставление услуг сотовой связи стандарта GSM и микросотовой связи стандарта DECT, показали, что конечного пользователя удовлетворяет качество обслуживания, худшее по сравнению с ТФОП или ISDN, до тех пор, пока он получает выгоду от использования новой услуги. В случае предоставления услуг сотовой связи – это мобильность терминала, а в случае IP-телефонии это могут быть низкая стоимость, возможности интеграции услуг в рамках единой сети.

Вопросы мобильности пользователя. Пользователь должен иметь доступ к услуге передачи речевой информации по IP-сетям в любом месте сети.

Ниже несколько подробнее рассматриваются наиболее интересные, как показалось авторам, направления деятельности групп, работающих над проектом TIPHON. Одним из таких направлений является разработка принципа декомпозиции шлюза.

Взятую за основу рекомендацию ITU-T H.323, спецификации TIPHON дополняют некоторыми обязательными процедурами, а также механизмами взаимодействия IP-сетей с ТФОП. Функциональная модель сети IP-телефонии, разработанная TIPHON, состоит из тех же компонентов, что и модель сети H.323 (привратник, шлюз, терминал), однако в ней предусмотрено разделение шлюза на три функционально-независимых объекта. Это шлюз сигнализации (SG), транспортный шлюз (MG) и контроллер транспортного шлюза (MGC).

Шлюз сигнализации служит промежуточным звеном сигнализации между IP-сетями и ТФОП. В задачи транспортного шлюза входит преобразование и/или перекодирование передаваемой информации. К транспортному шлюзу подключены ИКМ-тракты сети с коммутацией каналов, он также подавляет эхо, воспроизводит различные сообщения для абонентов, принимает и передает сигналы DTMF и т.д. Контроллер транспортного шлюза MGC выполняет процедуры сигнализации H.323, которые определены в рекомендациях ITU-T H.323, H.225 (RAS и Q.931) и H.245, а также преобразует сигнализацию ТФОП в сигнализацию H.323. Основная его задача – управлять работой транспортного шлюза, т.е. осуществлять управление соединениями, использованием ресурсов, преобразованием протоколов и т.п.

Привратник отвечает за управление объектами сети, в частности, выполняет преобразование адресов (например, телефонных номеров в соответствующие IP-адреса) и маршрутизацию сигнальной информации. Привратник в модели сети TIPHON поддерживает все те функции, которые определены для него в рекомендации H.323. Но, помимо этого, он отвечает за начисление платы, взаиморасчеты, составление отчетов об использовании ресурсов и выполняет некоторые другие функции.

Следует особо подчеркнуть, что MGC – это объект, контролирующий работу транспортного шлюза. Управление соединениями в его функции не входит. Это – задача привратника, который выполняет ее в соответствии с рекомендацией ITU-T H.323.

Разработанная в рамках проекта TIPHON модель сети, состоящая из функциональных элементов и интерфейсов (точек доступа) между ними, показана на рис. 2.8. Чтобы соответствовать рекомендациям TIPHON, оборудование должно поддерживать эти интерфейсы. Так, интерфейс D предназначен для организации взаимодействия между привратниками, а интерфейс C – между контроллером шлюза MGC и привратником. Интерфейс N поддерживает взаимодействие между объектами MGC и MG. Они могут общаться на предмет создания, модификации и завершения соединений; определения требуемого формата информации; генерации акустических сигналов и различных речевых уведомлений; запроса отчетов о событиях, связанных с прохождением информационного потока. Показанные на рис. 2.8 функции поддержки (back-end) могут быть использованы для аутентификации, биллинга, преобразования адресов и других задач.

Смоделированный на основе трех описанных элементов распределенный шлюз воспринимается другими элементами сети как единая система.

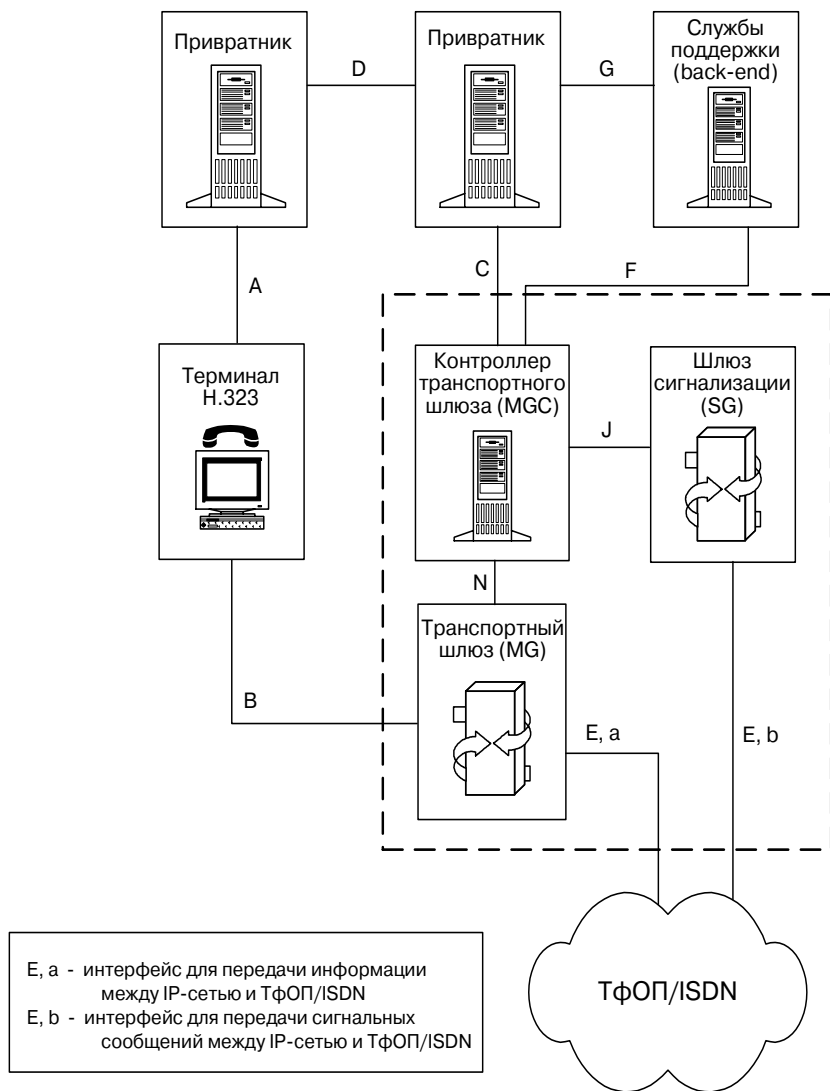


Рис. 2.8 Модель сети TIPON

Три упомянутых элемента (SG, MG, MGC) могут не быть физически разделены, однако такое разделение дает определенные преимущества. Дело в том, что использование трех отдельных объектов позволит обрабатывать больше вызовов, поскольку в этом случае разные функции распределяются по отдельным процессорам. В идеале такие объекты должны иметь стандартные интерфейсы, что даст оператору возможность использовать продукцию разных фирм-производителей. В приведенной выше модели один шлюз сигнализации с целью

более экономичного развертывания сети может быть использован для обслуживания большого числа транспортных шлюзов.

Теперь следует рассмотреть вопрос об адресации в рамках проекта TIPHON. От решения задач адресации во многом зависят удобство пользования услугой, работа алгоритмов маршрутизации, обеспечение мобильности абонентов и т.д. Концепцией телефонной связи предусмотрено, что абонент сети ТфОП должен иметь возможность связаться с другим абонентом со своего телефона путём набора номера вне зависимости от того, к сети какого типа подключён адресат. Формат номера обычно соответствует рекомендации E. 164. В настоящее время органами стандартизации разрабатываются механизмы преобразования телефонных номеров либо в IP-адреса, либо в унифицированные указатели ресурсов (URL).

Отображение телефонных номеров на IP-адреса создаёт проблему управления данными, так как пользователи имеют тенденцию перемещаться по всей сети Интернет и входить в систему из разных мест, поэтому их IP-адрес регулярно изменяется. Если предполагается, что сети IP-телефонии будут обслуживать сотни миллионов пользователей, то гибкое и надёжное решение вопроса о том, каким образом должно выполняться регулярное обновление данных и как должны обрабатываться запросы, со всей очевидностью станет сложной проблемой.

Отображение телефонных номеров на URL немного упрощает проблему преобразования адресов путём использования интернетовского ярлыка для идентификации пользователя. Однако, как только телефонный номер преобразован в ярлык, последний должен быть преобразован в адрес поставщика услуг Интернет, который, в свою очередь, формирует окончательный IP-адрес получателя. Наличие такого большого количества стадий, нужных, чтобы найти вызываемого абонента, будет, очевидно, существенно увеличивать время между набором номера вызывающим абонентом и получением им сигнала КПВ или зуммера «Занято».

В настоящее время органами стандартизации разрабатываются и другие механизмы, обеспечивающие надлежащую адресацию и маршрутизацию номеров E. 164, однако простых и универсальных путей решения этой проблемы пока не видно. Вопрос преобразования номера телефонной сети общего пользования в IP-адрес представляется пока еще довольно сложным, и пути его решения разрабатываются не только рабочей группой 4 в рамках проекта TIPHON, но и другими организациями, например IETF.

Еще одним важным направлением работы TIPHON является вопрос о классах обслуживания. Для операторов очень привлекатель-

на возможность предоставления услуг с разным уровнем качества (и, соответственно, с разными тарифами), причем поддерживаемым не только в пределах сети одного оператора, но и при связи между сетями разных операторов. Для этого в рамках проекта TIPHON определены четыре класса обслуживания, каждый из которых гарантирует определенное качество, как при установлении соединения, так и во время сеанса связи (таблица 2.2).

Таблица 2.2 Характеристики классов обслуживания TIPHON

Характеристика	Классы обслуживания			
	Высший (4)	Высокий (3)	Средний (2)	Низкий (1)
Качество передачи речи в одном направлении	Лучше, чем G.711	Не хуже, чем G.726 (32 Кбит/с)	Не хуже, чем GSM-FR	Не определено
Сквозная задержка, мс	<150	<250	<350	<450
Время установления соединения при прямой IP-адресации, с	<1,5	<4	<7	<7
Время установления соединения при преобразовании номера E.164 в IP-адрес, с *	<2	<5	<10	<10
Время установления соединения при преобразовании номера E.164 в IP-адрес через клиринговый центр или при роуминге, с **	<3	<8	<15	<15
Время установления соединения при преобразовании номера E.164 в IP-адрес, с **	<4	<10	<20	<20
Время установления соединения при преобразовании номера E.164 в IP-адрес через клиринговый центр или при роуминге, с **	<6	<15	<30	<30
Время установления соединения при преобразовании адреса электронной почты в IP-адрес, с	<4	<13	<25	<25

* - пользователь IP-сети вызывает абонента ТфОП.

** - абонент ТфОП вызывает пользователя IP-сети.

Качество обслуживания при установлении соединения характеризуется, прежде всего, временем его установления, т.е. временем между набором абонентом последней цифры номера (или, например, команды ввода при наборе адреса на компьютере) и получением им ответного акустического сигнала. Качество обслуживания во время сеанса связи определяется многими факторами, основными из которых являются сквозная временная задержка и качество сквозной передачи речи (оценивается методами экспертной оценки).

2.3 Установление телефонного соединения в IP-сети

Рассмотрим процедуру установления соединения через сеть IP при вызове с предоплатой или с оплатой после разговора. Для организации такого соединения абонент А набирает местный телефонный номер шлюза своего поставщика услуг IP-телефонии. Абоненту А передается второй сигнал ответа станции и предлагается ввести телефонный номер вызываемого абонента, номер счёта и пароль, если вызов производится не с домашнего, зафиксированного у поставщика телефона. Далее устанавливается соединение со стороны вызываемого абонента В. На рис. 2.9 приведены компоненты IP-телефонии, которые обычно используются в таком соединении.

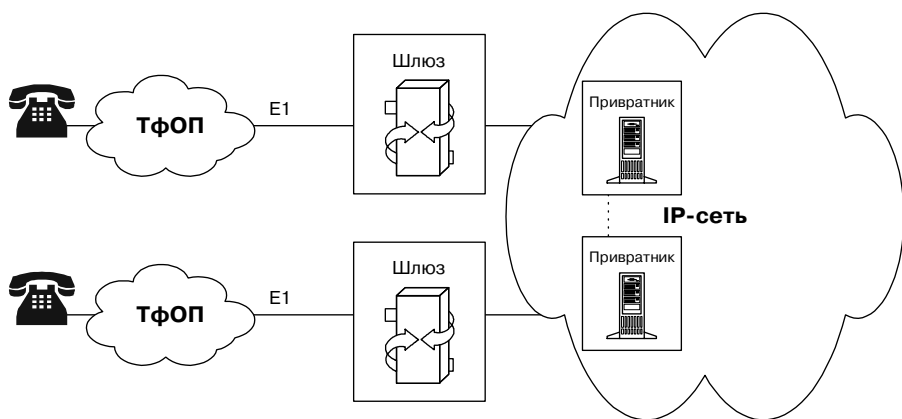


Рис. 2.9 Компоненты IP-телефонии

Одним из этих компонентов является шлюз H.323, который служит средством взаимодействия между ТФОП и IP-сетью. Преобразование адресной информации E.164 в IP-адрес и маршрутизацию вызова осуществляет привратник H.323. Для конкретного сценария могут потребоваться и другие компоненты. Может потребоваться, например, процедура обращения к поставщику услуг урегулирования (settlement provider) для того, чтобы обеспечить телефонные соединения с абонентами в тех местах, где у данного поставщика услуг IP-телефонии нет физического присутствия. Поставщик услуг урегулирования обычно работает с несколькими поставщиками услуг IP-телефонии и следит за тем, какому из них, в каком регионе и по какой стоимости целесообразно перепоручить соединение.

Общим протоколом для услуг урегулирования является *открытый протокол урегулирования* (Open Settlement Protocol). Этот протокол позволяет инфраструктуре динамической маршрутизации и начисления платы выбирать оптимальный маршрут для телефонного со-

единения в зависимости от времени суток, местоположения вызывающего и вызываемого абонентов и многих других факторов.

На рис. 2.10, 2.11 и 2.12 более подробно представлена процедура установления соединения для вызовов с предоплатой или с оплатой после разговора, являющаяся, в известном смысле, уточнением упрощенной процедуры на рис. 1.8 предыдущей главы. Рис. 2.10 отражает следующие стадии установления соединения.

1. Абонент А набирает местный номер доступа к шлюзу.
2. Шлюз запрашивает у специального сервера данные о вызывающем абоненте (по информации АОН или по идентификационному номеру). Сервер может быть совмещен с привратником.
3. Сервер просматривает информацию АОН для того, чтобы убедиться, что абоненту А разрешено пользоваться данной услугой, и затем передает к шлюзу сообщение аутентификации пользователя.



Рис. 2.10 Установление соединения: Часть 1

Рис. 2.11 отражает следующие стадии.

4. Абонент А набирает телефонный номер вызываемого абонента Б.
5. Шлюз консультируется с привратником о возможных способах маршрутизации вызова.
6. Привратник просматривает адрес E.164 на фоне таблицы маршрутизации и передает к исходящему шлюзу IP-адрес встречного (входящего) шлюза. При этом привратнику может понадобиться консультация с привратником другой зоны.



Рис. 2.11 Установление соединения: Часть 2

Финальные стадии установления соединения показаны на рис. 2.12:

7. Исходящий шлюз направляет вызов H.323 по IP-сети к входящему шлюзу.

8. Входящий шлюз направляет вызов по сети ТфОП к вызываемому абоненту.

9. Шлюзы посылают на упоминавшийся ранее специальный сервер данные о начале/окончании установления соединения для начисления платы за связь.

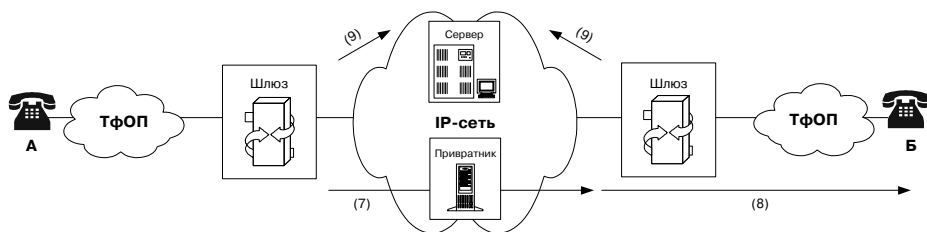


Рис. 2.12 Установление соединения: Часть 3

2.4 Эффективность IP-телефонии

Как уже отмечалось ранее, привлекательность всех алгоритмов сценария «телефон-телефон» для пользователя заключается в значительно более низких, по сравнению с обычной междугородной или международной телефонной связью, тарифах, что является следствием применения технологий, обеспечивающих вторичное уплотнение телефонных каналов. Поэтому многие пользователи согласны терпеть снижение качества передачи речи.

Предоставление телефонных услуг через инфраструктуру IP позволяет поставщику услуг IP получать большую, по сравнению с традиционными операторами, прибыль благодаря тому, что:

- функции предоставления услуг телефонии и передачи данных объединяются в общей инфраструктуре IP; основной объем обслуживаемого трафика приходится на традиционные данные Интернет, а транспортировка относительно невысокого объема трафика IP-телефонии может осуществляться с использованием той же инфраструктуры при очень незначительных дополнительных затратах,
- отсутствует необходимость обеспечивать качество и объем услуг, требуемые от операторов ТфОП, что допускает реализацию услуг IP-телефонии на базе более дешёвого оборудования.

Для традиционных телефонных операторов IP-телефония также достаточно перспективна. Операторы ТфОП в США и Европе вкладывают значительные средства в создание развитой инфраструктуры IP и в привлечение на свою сторону поставщиков услуг Интернет.

Так, например, компания US West Inc. (Инглвуд, Колорадо) объявила о проекте реализации технологии xDSL в масштабе всей страны, компания Worldcom Inc. (Джексон, Миссисипи) уже владеет первым поставщиком услуг Интернет – Uunet Technologies Inc. (Фоллс Черч, Виргиния) – и намеревается приобрести фирму MCI Communications Corp. (Вашингтон, округ Колумбия).

Но мотивы такой тенденции не только в сокращении затрат на обслуживание трафика. В настоящее время минута телефонного разговора по сетям коммутации каналов внутри США обходится местной телефонной компании примерно в 6 центов, а передача речи по Интернет стоит от 1 до 2 центов за минуту. Такая разница вряд ли достаточна для того, чтобы радикально перестроить инфраструктуру дальней связи, использующую технологию 1980-х годов, но потребовавшую в свое время многомиллиардных затрат на цифровизацию сети. В свете этого, сегодняшняя ситуация с расценками на междугородную и международную телефонную связь кратковременна и в ближайшее время перестанет быть столь же важной причиной развития IP-телефонии, как это имело место на начальной стадии ее внедрения. Стратегические преимущества новой технологии заключаются в конвергенции услуг, в создании интегрированных приложений в конечных узлах. Контролируя технологии коммутации каналов и пакетов, можно приобрести гигантское преимущество (во всемирном масштабе) при вступлении в следующее столетие.

Тем не менее, эффективность IP-телефонии ограничивается сегодня неустойчивыми и непредсказуемыми уровнями задержки на передачу пакетов. Другими словами, IP-телефония представляет собой пример классического проектного компромисса между стоимостью и характеристиками качества. Разумеется, в будущем компромиссное решение будет другим, и некоторые способы его оптимизации ясны уже сейчас.

В этом направлении ведется разработка оборудования следующего поколения. Шлюзы (маршрутизаторы) располагаются только на краях сети, где должны приниматься наиболее часто сложные решения и где должны вызываться наиболее используемые процессы, а далее разворачиваются высокоскоростные коммутаторы АТМ, причем, в соответствии с проектными спецификациями, маршрутизаторы и коммутаторы смогут работать со скоростью 1 Тбит/с. Если к этому добавить невероятно высокоскоростные системы оптоволоконной передачи в сети, то перспектива представляется весьма оптимистичной. Каждое оптическое волокно в настоящее время может поддерживать не менее 32 световых волн (оптических частот), причем каждая запускается на скорости не менее 10 Гбит/с и поддерживает приблизительно 130,000 каналов передачи речевой информации при стандартных скоростях 64 кбит/с. Вдоль маршрута укладываются сотни оптических волокон.

Кроме того, будет предусматриваться фиксация маршрутов от каждого шлюза к каждому из остальных шлюзов, чтобы все пакеты от шлюза N к шлюзу M направлялись по тому же самому маршруту.

Стала очевидной также избыточность традиционной передачи речевой информации со скоростью 64 Кбит/с. Современные алгоритмы сжатия позволяют использовать для передачи речи полосу пропускания 5,3 Кбит/с. По мере уменьшения требований к ширине полосы возрастает производительность, за тот же период времени по тем же каналам и через те же коммутаторы передается больше данных, и цены на телефонные разговоры снижаются. Соответствующие стандарты сжатия речи были разработаны уже в середине 90-х гг.

Это – рекомендация G.729, которая предусматривает 8-кратное сжатие речевого сигнала, что дает возможность передавать его в полосе 8 Кбит/с с тем качеством, которое поддерживают обычные телефонные сети. В основу стандарта положен алгоритм сжатия CS-ACELP. Последняя его версия, G.729A, использует тот же алгоритм, но упрощенный кодек, что значительно снижает нагрузку на процессор при обработке речевого потока.

Другая рекомендация – G.723.1 – позволяет сжимать речевой сигнал в 12 раз и транспортировать его со скоростью 5,3 или 6,3 Кбит/с. При этом качество передачи речи немного снижается, но остается вполне достаточным для делового общения. Для сжатия полосы до 5,3 Кбит/с применяется алгоритм ACELP, а до 6,3 Кбит/с – алгоритм MP-MLQ.

Общее правило гласит, что более «плотное» сжатие приводит к снижению качества речи, однако разработка все более сложных алгоритмов компрессии делает это правило спорным. Выбор алгоритма обуславливается тремя основными факторами – распространенностью, поддержкой в имеющемся оборудовании и ожиданиями пользователей. На нынешнем этапе оба алгоритма хорошо себя показали и приняты производителями средств пакетной телефонии.

Отметим, что устройства, поддерживающие G.723.1, не могут «разговаривать» напрямую с устройствами на основе G.729; для их взаимодействия необходим специальный конвертер. Сигнальный процессор DSP, реализующий эти функции, может вносить задержки и искажения, снижающие качество речи до неприемлемого уровня. Кроме того, современные технологии не способны производить такое преобразование в реальном времени. Более подробно эти вопросы рассматриваются в следующей главе.

Глава 3

Передача речи по IP-сетям

3.1 Особенности передачи речевой информации по IP – сетям

Если проблемы ограничения задержки и подавления эха в традиционной телефонии существовали всегда, а при переходе к IP-сетям лишь усугубились, то потери информации (пакетов) и стохастический характер задержки породили совершенно новые проблемы, решение которых сопряжено с большими трудностями. Этим объясняется тот факт, что понадобился длительный период развития сетевых технологий, прежде чем появились коммерческие приложения IP-телефонии, хотя, справедливости ради, нужно отметить, что трудно назвать другую телекоммуникационную технологию, которая смогла «повзрослеть» столь же быстро.

3.1.1 Задержки

При передаче речи по IP-сети возникают намного большие, чем в ТФОП, задержки, которые, к тому же, изменяются случайным образом. Этот факт представляет собой проблему и сам по себе, но кроме того, усложняет обсуждаемую далее в этой главе проблему эха. Задержка (или время запаздывания) определяется как промежуток времени, затрачиваемый на то, чтобы речевой сигнал прошел расстояние от говорящего до слушающего. Покажем, что и как оказывает влияние на количественные характеристики этого промежутка времени.

Влияние сети

Во-первых, неустойчиво и плохо предсказуемо время прохождения пакета через сеть. Если нагрузка сети относительно мала, маршрутизаторы и коммутаторы, безусловно, могут обрабатывать пакеты практически мгновенно, а линии связи бывают доступны почти всегда. Если загрузка сети относительно велика, пакеты могут довольно долго ожидать обслуживания в очередях. Чем больше маршрутизаторов, коммутаторов и линий в маршруте, по которому проходит пакет, тем больше время его запаздывания, и тем больше вариация этого времени, т.е. *джиттер*. В главе 10, посвященной качеству обслуживания (QoS), будет показано, каким образом и с использованием каких протоколов и алгоритмов следует строить сети, чтобы минимизировать задержки и их джиттер.

Влияние операционной системы

Большинство приложений IP-телефонии (особенно клиентских) представляет собой обычные программы, выполняемые в среде какой-либо операционной системы, такой как Windows или Linux. Эти программы обращаются к периферийным устройствам (платам обработки речевых сигналов, специализированным платам систем сигнализации) через интерфейс прикладных программ для взаимодействия с драйверами этих устройств, а доступ к IP-сети осуществляют через Socket-интерфейс.

Большинство операционных систем не может контролировать распределение времени центрального процессора между разными процессами с точностью, превышающей несколько десятков миллисекунд, и не может обрабатывать за такое же время более одного прерывания от внешних устройств. Это приводит к тому, что задержка в продвижении данных между сетевым интерфейсом и внешним устройством речевого вывода составляет, независимо от используемого алгоритма кодирования речи, величину такого же порядка, или даже больше.

Из сказанного следует, что выбор операционной системы является важным фактором, влияющим на общую величину задержки. Чтобы минимизировать влияние операционной системы, некоторые производители шлюзов и IP-телефонов используют так называемые ОС реального времени (VxWorks, pSOS, QNX Neutrino и т.д.), которые используют более сложные механизмы разделения времени процессора, действующие таким образом, чтобы обеспечивать значительно более быструю реакцию на прерывания и более эффективный обмен потоками данных между процессами.

Другой, более плодотворный подход – переложить все функции, которые необходимо выполнять в жестких временных рамках (обмен данными между речевыми кодеками и сетевым интерфейсом, под-

держку RTP и т.д.), на отдельный быстродействующий специализированный процессор. При этом пересылка речевых данных осуществляется через выделенный сетевой интерфейс периферийного устройства, а операционная система рабочей станции поддерживает только алгоритмы управления соединениями и протоколы сигнализации, т.е. задачи, для выполнения которых жестких временных рамок не требуется. Этот подход реализован в платах для приложений IP-телефонии, производимых фирмами Dialogic, Audiocodes, Natural Microsystems. По такой же технологии выполнен и шлюз IP-телефонии в платформе Протей-IP, что позволило обеспечить высокое качество передачи речи.

Влияние джиттер-буфера

Проблема джиттера весьма существенна в пакетно-ориентированных сетях. Отправитель речевых пакетов передает их через фиксированные промежутки времени (например, через каждые 20 мс), но при прохождении через сеть задержки пакетов оказываются неодинаковыми, так что они прибывают в пункт назначения через разные промежутки времени. Это иллюстрирует рис. 3.1.

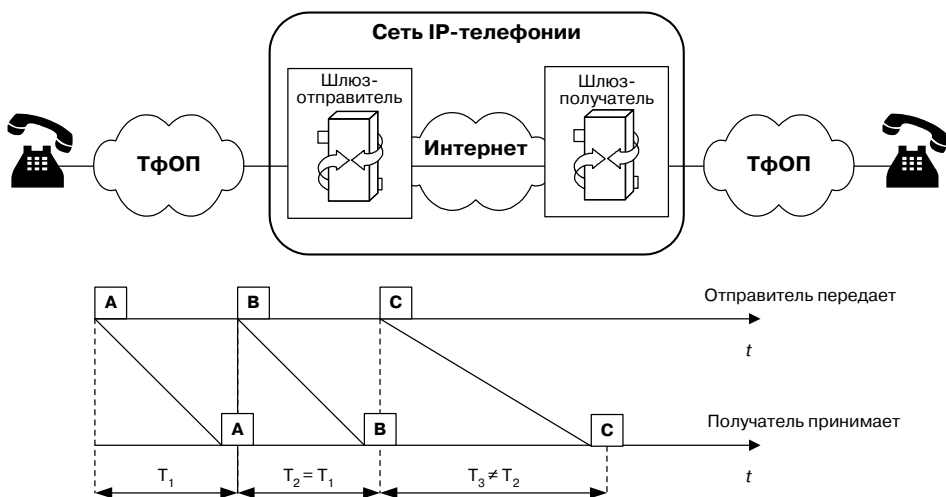


Рис. 3.1 Различие интервалов между моментами прибытия пакетов (джиттер)

Задержка прохождения пакетов по сети T может быть представлена как сумма постоянной составляющей $\bar{T}$ (время распространения плюс средняя длительность задержки в очередях) и переменной величины j , являющейся результатом джиттера: $T = \bar{T} \pm j$.

Для того, чтобы компенсировать влияние джиттера, в терминалах используется т.н. *джиттер-буфер*. Этот буфер хранит в памяти при-

бывшие пакеты в течение времени, определяемого его емкостью (длиной). Пакеты, прибывающие слишком поздно, когда буфер заполнен, отбрасываются. Интервалы между пакетами восстанавливаются на основе значений временных меток RTP-пакетов. В функции джиттер-буфера обычно входит и восстановление исходной очередности следования пакетов, если при транспортировке по сети они оказались «перепутаны».

Слишком короткий буфер будет приводить к слишком частым потерям «опоздавших» пакетов, а слишком длинный – к неприемлемо большой дополнительной задержке. Обычно предусматривается динамическая подстройка длины буфера в течение всего времени существования соединения. Для выбора наилучшей длины используются эвристические алгоритмы.

Влияние кодека и количества передаваемых в пакете кадров

Большинство современных эффективных алгоритмов кодирования/декодирования речи ориентировано на передачу информации кадрами, а не последовательностью кодов отдельных отсчетов. Поэтому в течение времени, определяемого длиной кадра кодека, должна накапливаться определенной длины последовательность цифровых представлений отсчетов. Кроме того, некоторым кодекам необходим предварительный анализ большего количества речевой информации, чем должно содержаться в кадре. Это неизбежное время накопления и предварительного анализа входит в общий бюджет длительности задержки пакета.

На первый взгляд, можно было бы заключить, что чем меньше длина кадра, тем меньше должна быть задержка. Однако, как будет показано ниже, из-за значительного объема служебной информации, передаваемой в RTP/UDP/IP-пакетах, передача маленьких порций данных очень неэффективна, так что при применении кодеков с малой длиной кадра приходится упаковывать несколько кадров в один пакет. Кроме того, кодеки с большей длиной кадра более эффективны, поскольку могут «наблюдать» сигнал в течение большего времени и, следовательно, могут более эффективно моделировать этот сигнал.

ITU-T в рекомендации G.114 определил требования к качеству передачи речи. Оно считается хорошим, если сквозная задержка при передаче сигнала в одну сторону не превышает 150 мс (рис. 3.2). Современное оборудование IP-телефонии при включении «спина к спине» (два устройства – шлюза – соединяются напрямую) вносит задержку порядка 60-70 мс. Таким образом, остается еще около 90 мс на сетевую задержку при передаче IP-пакета от отправителя к пункту назначения, что говорит о возможности обеспечить при современном уровне технологии передачу речи с достаточно хорошим качеством.

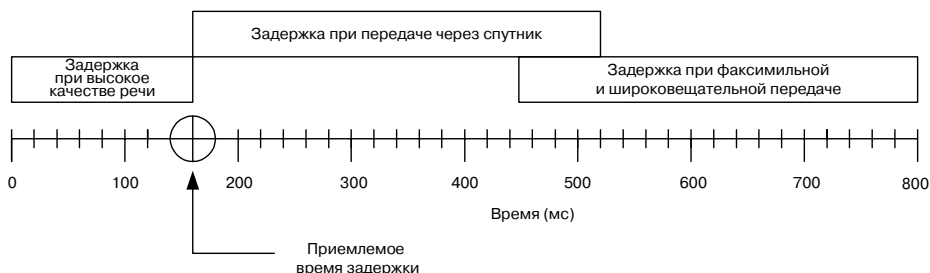


Рис. 3.2 Задержка при передаче

Авторам отнюдь не хотелось бы, чтобы у читателя сложилось впечатление, будто временные задержки – проблема исключительно IP-телефонии. Именно поэтому на рис. 3.2 приведены также характеристики спутниковой передачи, при которой требуется примерно 250 мс для того, чтобы сигнал достиг спутника и вернулся обратно к Земле (без учета затрат времени на обработку сигнала). Таким образом, полное время задержки превышает 250-300 мс. Согласно рекомендации G.114, такая задержка выходит за границы диапазона, приемлемого для передачи речи. Тем не менее, ежедневно значительное количество разговоров ведется по спутниковым линиям связи. Следовательно, *приемлемое* качество речи определяется, прежде всего, требованиями пользователей.

3.1.2 Эхо

Феномен эха вызывает затруднения при разговоре и у говорящего, и у слушающего. Говорящий слышит с определенной задержкой свой собственный голос. Если сигнал отражается дважды, то слушающий дважды слышит речь говорящего (второй раз – с ослаблением и задержкой).

Эхо может иметь электрическую и акустическую природу.

Отражения в дифсистеме являются неотъемлемым свойством ТфОП. Поэтому они проявляются при взаимодействии ТфОП и IP-сетей.

С целью экономии кабеля в ТфОП для подключения абонентских терминалов с давних пор используются двухпроводные линии, по которым речевые сигналы передаются в обоих направлениях. Более того, во многих телефонных сетях передача сигналов обоих направлений по двум проводам используется и в соединительных линиях между электромеханическими АТС [6] (хотя теперь для организации связи между АТС всё чаще используется раздельная передача сигналов разных направлений, т.е. четырехпроводная схема их передачи). Для разделения сигналов разных направлений в терминалах абонентов (телефонных аппаратах) и на АТС применяются простые мостовые схемы, называемые дифсистемами (hybrid). Работа этих мос-

товых схем основывается на согласовании импедансов в плечах моста, одним из плеч которого является двухпроводная абонентская линия. Так как абонентские линии могут очень сильно различаться по своим параметрам (длине, диаметру жил кабеля и т.п.), то достичь точного согласования (тем более, во всей полосе передаваемых частот) невозможно. Вместо этого администрация связи вынуждена ориентироваться на некоторую среднюю величину импеданса для всех абонентских линий своей национальной сети. Это приводит к тому, что сигналы прямого и обратного направления в большинстве случаев не разделяются полностью, и в дифсистеме возникает частичное отражение сигналов.

Если задержка распространения сигнала в сети невелика (что обычно и бывает в местных сетях), такой отраженный сигнал попросту незаметен и не вызывает неприятных ощущений. Если задержка достигает величины 15-20мс, возникает эффект «огромного пустого помещения». При дальнейшем увеличении задержки субъективная оценка качества разговора резко ухудшается, вплоть до полной невозможности продолжать беседу.

В рамках ТфОП проблема такого эха известна с тех самых пор, когда телефонная сеть стала настолько протяженной, что задержки распространения сигналов перестали быть неощутимыми. Были разработаны и методы борьбы с этим феноменом – от минимизации задержек путем соответствующего планирования сети до применения эхозаградителей и эхокомпенсаторов. Как мы уже видели выше, задержки, свойственные процессам передачи речи по IP-сетям, таковы, что не оставляют выбора и делают механизмы, ограничивающие эффект эха, обязательными в любом оборудовании IP-телефонии.

Акустическое эхо возникает при пользовании терминалами громкоговорящей связи, независимо от того, какая технология используется в них для передачи информации. Акустическое эхо может обладать значительной длительностью, а особенно неприятным бывает изменение его характеристик при изменении, например, взаимного расположения терминала и говорящего, или даже других людей в помещении. Эти обстоятельства делают построение устройств эффективного подавления акустического эха очень непростой задачей.

3.1.3 Устройства ограничения эффектов эха

Существуют два типа устройств, предназначенных для ограничения вредных эффектов эха: эхозаградители и эхокомпенсаторы.

Эхозаградители появились в начале 70-х годов. Принцип их работы прост и состоит в отключении канала передачи, когда в канале приема присутствует речевой сигнал. Такая техника широко используется в дешевых телефонных аппаратах с громкоговорящей связью

(speakerphones), однако простота не обеспечивает нормального качества связи – перебить говорящего становится невозможно, т.е. связь, по сути, становится полудуплексной.

Эхокомпенсатор – это более сложное устройство, которое моделирует эхосигнал для последующего его вычитания из принимаемого сигнала (рис. 3.3). Эхо моделируется как взвешенная сумма задержанных копий входного сигнала или, иными словами, как свертка входного сигнала с оцененной импульсной характеристикой канала. Оценка импульсной характеристики происходит в тот момент, когда говорит только удаленный корреспондент, для чего используется детектор одновременной речевой активности. После вычитания синтезированной копии эхосигнала из сигнала обратного направления полученный сигнал подвергается нелинейной обработке для увеличения степени подавления эха (подавление очень слабых сигналов).

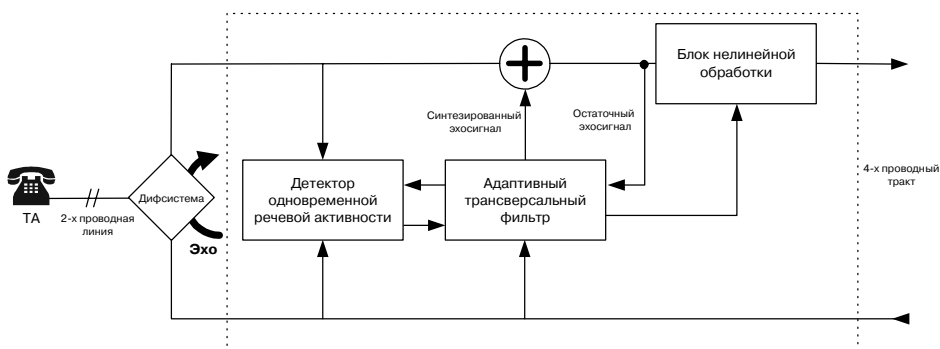


Рис. 3.3 Упрощенная блок-схема эхокомпенсатора

Поскольку эхо моделируется только как линейный феномен, любые нелинейные процессы на пути его возникновения приводят к ухудшению работы эхокомпенсатора. Использование более сложных алгоритмов позволяет подавлять эхо, представляющее собой не только задержанный, но и сдвинутый по частоте сигнал, что часто происходит из-за наличия в ТФОП устаревших частотных систем передачи. Реализация таких алгоритмов необходима для успешного функционирования эхокомпенсаторов в телефонных сетях на территории России и бывшего СССР, и поэтому алгоритмы эхокомпенсации в российском оборудовании IP-телефонии на базе интеллектуальной платформы Протей-IP разработаны именно с учетом сдвига эха по частоте. К проблемам технической реализации оборудования IP-телефонии мы еще вернемся в заключительной главе данной книги.

Эхокомпенсатор должен хранить амплитуды эхосигналов, задержанных на время от нуля до продолжительности самого длительного подавляемого эхосигнала. Это значит, что эхокомпенсаторы, рассчитанные на подавление более длительных эхосигналов, требуют для

своей реализации большего объема памяти и большей производительности процессора. Таким образом, выгодно помещать эхокомпенсаторы «максимально близко», в смысле задержки, к источнику эха.

По изложенным выше причинам эхокомпенсаторы являются неотъемлемой частью шлюзов IP-телефонии. Алгоритмы эхокомпенсации реализуются обычно на базе тех же цифровых сигнальных процессоров, что и речевые кодеки, и обеспечивают подавление эхосигналов длительностью до 32–64 мс. К эхокомпенсаторам терминалов громкоговорящей связи предъявляются гораздо более строгие требования, которые здесь рассматриваться не будут, так как проблема акустического эха не входит в число проблем, специфических для IP-телефонии.

3.2 Принципы кодирования речи

Как стало ясно со времени изобретения Александра Белла, для того, чтобы передать речь через телефонную сеть, речевую информацию нужно преобразовать в аналоговый электрический сигнал. При переходе к цифровым сетям связи возникла необходимость преобразовать аналоговый электрический сигнал в цифровой формат на передающей стороне, то есть закодировать, и перевести обратно в аналоговую форму, то есть декодировать, на приемной стороне.

Процесс преобразования аналогового речевого сигнала в цифровую форму называют *анализом* или *цифровым кодированием речи*, а обратный процесс восстановления аналоговой формы речевого сигнала – *синтезом* или *декодированием речи*.

Цель любой схемы кодирования – получить такую цифровую последовательность, которая требует минимальной скорости передачи и из которой декодер может восстановить исходный речевой сигнал с минимальными искажениями.

При преобразовании речевого сигнала в цифровую форму, так или иначе, имеют место два процесса – *дискретизация* (*sampling*), т.е. формирование дискретных во времени отсчетов амплитуды сигнала, и *квантование*, т.е. дискретизация полученных отсчетов по амплитуде (кодирование непрерывной величины – амплитуды – числом с конечной точностью). Эти две функции выполняются т.н. аналого-цифровыми преобразователями (АЦП), которые размещаются в современных АТС на плате абонентских комплектов, а в случае передачи речи по IP-сетям – в терминале пользователя (компьютере или IP-телефоне).

Так называемая теорема отсчетов гласит, что аналоговый сигнал может быть успешно восстановлен из последовательности выборок с частотой, которая превышает, как минимум, вдвое максимальную

частоту, присутствующую в спектре сигнала. В телефонных сетях полоса частот речевого сигнала намеренно, посредством специальных фильтров, ограничена диапазоном 0.3 – 3.4 кГц, что не влияет на разборчивость речи и позволяет узнавать собеседника по голосу. По этой причине частота дискретизации при аналого-цифровом преобразовании выбрана равной 8кГц, причем такая частота используется во всех телефонных сетях на нашей планете.

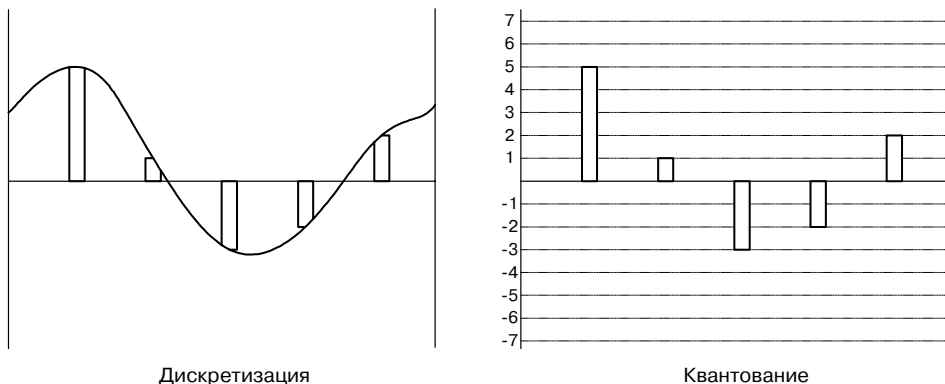


Рис. 3.4 Дискретизация и квантование аналогового речевого сигнала

При квантовании непрерывная величина отображается на множество дискретных значений, что, естественно, приводит к потерям информации. Для того, чтобы обеспечить в такой схеме достаточный динамический диапазон (способность передавать без искажений как сильные, так и слабые сигналы), дискретная амплитуда сигнала кодируется 12/13-ти разрядным двоичным числом по линейному закону.

Процесс аналого-цифрового преобразования получил, применительно к системам связи, название *импульсно-кодовой модуляции* (ИКМ).

Чтобы снизить необходимую скорость передачи битов, применяют нелинейный (логарифмический) закон квантования, т.е. квантованию подвергается не амплитуда сигнала, а ее логарифм. В данном случае имеет место процесс «сжатия» динамического диапазона сигнала, а при восстановлении сигнала происходит обратный процесс.

После длительных и бурных дебатов в отношении законов кодирования сегодня применяются две основные разновидности ИКМ: с кодированием по μ -закону и по А-закону. В результате сжатия сигнал с амплитудой, кодируемой 12 – 13 битами, описывается всего восемью битами. Различаются эти разновидности ИКМ деталями процесса сжатия (μ -закон кодирования предпочтительнее использовать при малой амплитуде сигнала и при малом отношении сиг-

нал/шум). Исторически сложилось так, что в Северной Америке используется кодирование по μ -закону, а в Европе – по А-закону. Поэтому при международной связи во многих случаях требуется преобразование μ -закона в А-закон, ответственность за которое несет страна, в которой используется μ -закон кодирования. В обоих случаях каждый отсчет кодируется 8 битами, или одним байтом, который можно считать звуковым фрагментом. Для передачи последовательности таких фрагментов необходима пропускная способность канала, равная 64 Кбит/с. Это определяется простыми арифметическими действиями:

$$4\,000\text{ Гц} * 2 = 8\,000\text{ отсчетов/с}, 8\,000\text{ отсчетов/с} * 8\text{ битов} = 64\text{ Кбит/с},$$

что составляет основу всей цифровой телефонии. Поскольку ИКМ была первой стандартной технологией, получившей широкое применение в цифровых системах передачи, пропускная способность канала, равная 64 Кбит/с, стала всемирным стандартом для цифровых сетей всех видов, причем – стандартом, который обеспечивает передачу речи с очень хорошим качеством. Соответствующие процедуры кодирования и декодирования стандартизованы ITU-T в рекомендации G.711.

Однако такое высокое качество передачи речевого сигнала (являющееся эталоном при оценке качества других схем кодирования) достигнуто в системах ИКМ за счет явно избыточной, при современном уровне технологии, скорости передачи информации.

Чтобы уменьшить присущую ИКМ избыточность и снизить требования к полосе пропускания, последовательность чисел, полученная в результате преобразования речевого аналогового сигнала в цифровую форму, подвергается математическим преобразованиям, позволяющим уменьшить необходимую скорость передачи. Эти преобразования «сырого» цифрового потока в поток меньшей скорости называют «сжатием» (а часто – кодированием, рассматривая ИКМ как некую отправную точку для дальнейшей обработки информации).

Существует множество подходов к «сжатию» речевой информации; все их можно разделить на три категории: *кодирование формы сигнала* (waveform coding), *кодирование исходной информации* (source coding) и *гибридное кодирование*, представляющее собой сочетание двух предыдущих подходов.

3.2.1 Кодирование формы сигнала

Импульсно-кодовая модуляция, по сути, и представляет собой схему кодирования формы сигнала. Однако нас интересуют более сложные алгоритмы, позволяющие снизить требования к полосе пропускания.

Рассматриваемые методы кодирования формы сигнала используют то обстоятельство, что между случайными значениями несколь-

ких следующих подряд отсчетов существует некоторая зависимость. Проще говоря, значения соседних отсчетов обычно мало отличаются одно от другого. Это позволяет с довольно высокой точностью предсказать значение любого отсчета на основе значений нескольких предшествовавших ему отсчетов.

При построении алгоритмов кодирования названная закономерность используется двумя способами. Во-первых, есть возможность изменять параметры квантования в зависимости от характера сигнала. В этом случае шаг квантования может изменяться, что позволяет до некоторой степени сгладить противоречие между уменьшением числа битов, необходимых для кодирования величины отсчета при увеличении шага квантования, и сужением динамического диапазона кодера, неизбежным без адаптации (о которой речь пойдет ниже). Некоторые алгоритмы предусматривают изменение параметров квантования приблизительно в рамках произносимых слогов, а некоторые изменяют шаг квантования на основе анализа статистических данных об амплитуде сигнала, полученных за относительно короткий промежуток времени.

Во-вторых, существует подход, называемый *дифференциальным кодированием* или *линейным предсказанием*. Вместо того, чтобы кодировать входной сигнал непосредственно, кодируют разность между входным сигналом и «предсказанной» величиной, вычисленной на основе нескольких предыдущих значений сигнала.

Если отсчеты входного сигнала обозначить как $y(i)$, то предсказанное значение в момент времени i представляет собой линейную комбинацию нескольких p предыдущих отсчетов:

$$\hat{y}(i) = a_1 y(i-1) + a_2 y(i-2) + \dots + a_p y(i-p)$$

где множители a_i называются *коэффициентами предсказания*.

Разность $e(i) = y(i) - \hat{y}(i)$ имеет меньший динамический диапазон и может кодироваться меньшим числом битов, что позволяет снизить требования к полосе пропускания.

Описанный метод называется *линейным предсказанием*, так как он использует только линейные функции предыдущих отсчетов. Коэффициенты предсказания выбираются так, чтобы минимизировать среднеквадратическое значение ошибки предсказания $e(i)$, при этом значения коэффициентов изменяются, в среднем, каждые 10-25 мс.

Простейшей (и представляющей сегодня, скорее, исторический интерес) реализацией последнего подхода является так называемая *дельта-модуляция* (ДМ), алгоритм которой предусматривает кодирование разности между соседними отсчетами сигнала только одним информационным битом, обеспечивая передачу, по сути, только знака разности.

Наиболее совершенным алгоритмом, построенным на описанных выше принципах, является алгоритм адаптивной дифференциальной импульсно-кодовой модуляции (АДИКМ), предложенный ITU-T в рекомендации G.726. Алгоритм предусматривает формирование сигнала ошибки предсказания и его последующее адаптивное квантование. Существует версия этого алгоритма, в которой информационные биты выходного цифрового потока организованы по иерархической схеме, что позволяет отбрасывать наименее значимую информацию, не уведомляя об этом кодер, и получать поток меньшей скорости за счет некоторого ухудшения качества. Документ G.726 специфицирует кодирование при скоростях 40, 32, 24 и 16 Кбит/с, что соответствует передаче 5, 4, 3 или 2 битов на отсчет. Качество речи, передаваемой с использованием АДИКМ G.726 при скорости 32 Кбит/с соответствует качеству речи, обеспечиваемому алгоритмом кодирования G.711.

При достаточно хороших характеристиках алгоритма, АДИКМ практически не применяется для передачи речи по сетям с коммутацией пакетов, так как этот алгоритм очень чувствителен к потерям целых блоков отсчетов, происходящим при потерях пакетов в сети. В таких случаях нарушается синхронизация кодера и декодера, что приводит к катастрофическому ухудшению качества воспроизведения речи даже при малой вероятности потерь.

3.2.2 Кодеры исходной информации (вокодеры) и гибридные алгоритмы

Многие методы кодирования используют особенности человеческой речи, связанные со строением голосового аппарата. Кодеры, в которых реализуются такие методы, называют *кодерами исходной информации* или *вокодерами* (voice coding).

Звуки речи образуются при прохождении выдыхаемого воздуха через голосовой аппарат человека, важнейшими элементами которого являются язык, нёбо, губы, зубы и голосовые связки. В формировании того или иного звука участвует та или иная часть этих элементов. Если звук формируется с участием голосовых связок, поток воздуха из легких вызывает их колебание, что порождает звуковой *тон*. Последовательность формируемых таким образом звуков составляет *тоновую* речь (или *тоновый* сегмент речи). Если звук формируется без участия связок, тон в нем отсутствует, и последовательность таких звуков составляет *нетоновую* речь (*нетоновый* сегмент речи). Спектр тонового звука может быть смоделирован путем подачи специальным образом сформированного сигнала возбуждения на вход цифрового фильтра с параметрами, определяемыми несколькими действительными коэффициентами. Спектр нетоновых звуков – практически равномерный, что обусловлено их шумовым характером.

В реальных речевых сигналах не все звуки можно четко разделить на тоновые и нетоновые, а приходится иметь дело с некими переходными вариантами, что затрудняет создание алгоритмов кодирования, обеспечивающих высокое качество передачи речи при низкой скорости передачи информации.

Рис. 3.5 иллюстрирует описанную упрощенную модель функционирования голосового тракта человека. Работа кодера, согласно такой модели, состоит в том, чтобы, анализируя блок отсчетов речевого сигнала, вычислить параметры соответствующего фильтра и параметры возбуждения (тоновый/нетоновый сегмент речи, частота тона, громкость и т.д.).

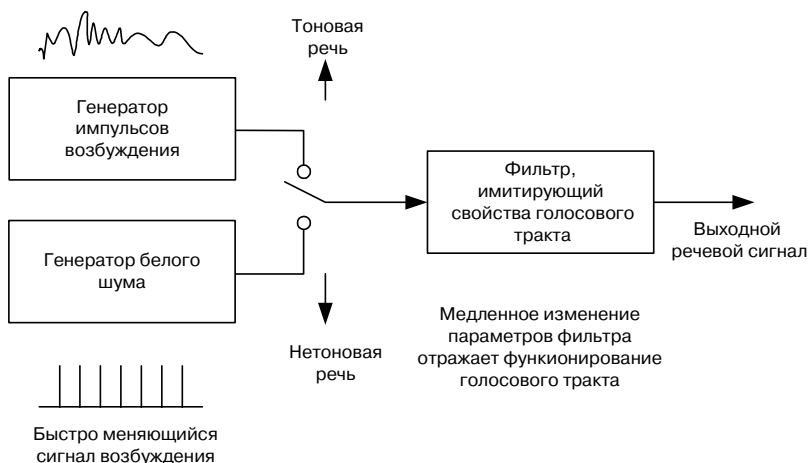


Рис. 3.5 Модель функционирования голосового тракта

Описанный принцип кодирования получил название LPC (Linear Prediction Coding – кодирование с линейным предсказанием), поскольку центральным элементом модели голосового тракта является линейный фильтр. Наиболее известный стандартный алгоритм, построенный по описанному принципу, был стандартизован министерством обороны США под названием LPC-10, где число 10 соответствует количеству коэффициентов фильтра. Данный кодер обеспечивает очень низкую скорость передачи информации 2.4 Кбит/с, однако качество воспроизводимых речевых сигналов оставляет желать лучшего и не удовлетворяет требованиям коммерческой речевой связи – речь носит ярко выраженный «синтетический» характер.

Как уже отмечалось, алгоритмы кодирования формы сигнала основаны на наличии корреляционных связей между отсчетами сигнала, которые дают возможность линейного предсказания. В сочетании с адаптивным квантованием этот подход позволяет обеспечить хорошее качество речи при скорости передачи битов порядка 24-32 Кбит/с. LPC-кодеры (вокодеры) используют простую матема-

тическую модель голосового тракта и позволяют использовать очень низкие скорости передачи информации 1200-2400 бит/с, однако ценой «синтетического» характера речи.

Гибридные алгоритмы кодирования и алгоритмы типа «анализ путем синтеза» (ABS) представляют собой попытки совместить положительные свойства двух описанных выше основных подходов и строить эффективные схемы кодирования с диапазоном скоростей передачи битов 6-16Кбит/с.

Важное отличие кодеров такого типа состоит в том, что в рамках этих алгоритмов нет необходимости принимать решение о типе воспроизводимого звука (тоновый или нетоновый), так как предусматриваются специальные меры для кодирования сигнала ошибки после прохождения возбуждения через LPC-фильтр. Например, сигнал ошибки может быть закодирован по алгоритму, аналогичному АДИКМ, что обеспечит высокую точность его передачи. ABS-кодеры не могут быть строго классифицированы как кодеры формы сигнала, однако реально целью процедуры минимизации ошибки (рис. 3.6), т.е. различия между входным и синтезированным сигналами, является синтез на выходе кодера сигналов, форма которых наиболее близка к форме входных. ABS-декодер является малой частью кодера и очень прост (рис. 3.7).

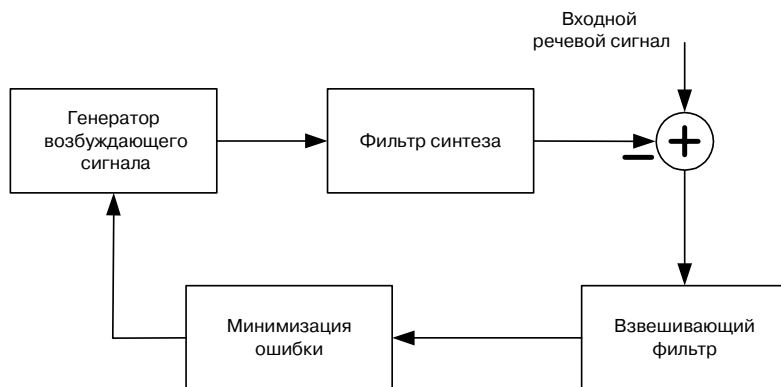


Рис. 3.6 Упрощенная блок-схема ABS-кодера



Рис. 3.7 Упрощенная блок - схема ABS - декодера

3.2.3 Процессоры цифровой обработки сигналов для речевых кодеков

Узкополосному кодированию речевых сигналов дорогу на рынок коммерческих приложений открыло развитие микроэлектроники и, в частности, появление дешевых процессоров цифровой обработки сигналов (DSP – Digital Signal Processor) в интегральном исполнении. До этого цифровая обработка сигналов (в том числе, узкополосное кодирование речи) была уделом разработчиков аппаратуры для нужд армии и спецслужб.

Процессоры DSP имеют архитектуру, оптимизированную для выполнения операций, которые характерны для типичных алгоритмов обработки сигналов. В качестве примеров таких операций можно назвать умножение с накоплением, а также выборку операндов с бит-инверсной адресацией, необходимую для выполнения быстрого преобразования Фурье.

Архитектура процессоров DSP часто характеризуется наличием нескольких вычислительных блоков, обеспечивающих выполнение одновременных операций в одном такте работы процессора. Для загрузки вычислительных блоков данными предусматривается несколько шин передачи данных и многопортовая память данных. Для увеличения производительности память инструкций и память данных разделены, а доступ к ним осуществляется также по отдельным шинам. Для процессоров DSP характерно использование инструкций увеличенной длины, содержащих поля для управления всеми вычислительными блоками.

Физически процессоры DSP выполняются в виде интегральных микросхем, содержащих в одном кристалле ядро процессора, память и периферийные устройства для обмена информацией. Наличие встроенной памяти обеспечивает быстрый доступ ядра к ее содержанию для получения максимальной производительности.

Существует множество модификаций процессоров DSP, различающихся производительностью, объемом памяти, потребляемой мощностью. В оборудовании IP-телефонии используются дешевые процессоры со средней производительностью и малой потребляемой мощностью, ориентированные на реализацию малого числа (единицы) каналов обработки речевой информации и применяемые, в основном, в составе терминальных устройств, или мощные высокопроизводительные процессоры, ориентированные на многоканальные (десятки каналов) приложения и используемые в составе таких групповых устройств как многоканальные шлюзы IP-телефонии, подключаемые к ТФОП по цифровым трактам E1.

Одними из самых известных производителей DSP являются фирмы Texas Instruments (www.ti.com), Analog Devices (www.analog.com), Motorola (www.motorola.com), на сайтах которых можно получить дополнительную информацию о номенклатуре DSP и об их применении.

Оборудование ПРОТЕЙ-IP использует DSP с лицензированным у одной из ведущих в данной области фирм программным обеспечением, реализующим необходимые алгоритмы (речевые кодеки, факс, модем). Это позволило, опираясь на существующий опыт, резко сократить время выхода оборудования на рынок. Кроме того, в данном случае исключается трудоемкая и длительная процедура лицензирования алгоритмов речевых кодеков (G.723.1, G.729), требующая значительных единовременных финансовых затрат. По такому же пути идут и ведущие мировые производители оборудования VoIP (Cisco, Dialogic и др.), лицензируя программное обеспечение DSP у компаний, специализирующихся именно в этой области, и концентрируя свои силы на реализации тех функций, которые традиционно обеспечивают данным производителям оборудования технологическое лидерство.

3.2.4 Основные алгоритмы кодирования речи, используемые в IP-телефонии

В первую очередь необходимо понять, какими критериями нужно руководствоваться при выборе «хорошего» кодека для использования в IP-телефонии.

Использование полосы пропускания канала

Скорость передачи, которую предусматривают имеющиеся сегодня узкополосные кодеки, лежит в пределах 1.2 – 64 Кбит/с. Естественно, что от этого параметра прямо зависит качество воспроизводимой речи. Существует множество подходов к проблеме определения качества. Наиболее широко используемый подход оперирует оценкой MOS (Mean Opinion Score), которая определяется для конкретного кодека как средняя оценка качества большой группой слушателей по пятибалльной шкале. Для прослушивания экспертам предъявляются разные звуковые фрагменты – речь, музыка, речь на фоне различного шума и т.д. Оценки интерпретируют следующим образом:

- **4-5** – высокое качество; аналогично качеству передачи речи в ISDN, или еще выше;
- **3.5-4** – качество ТфОП (toll quality); аналогично качеству речи, передаваемой с помощью кодека АДИКМ при скорости 32 Кбит/с. Такое качество обычно обеспечивается в большинстве телефонных разговоров. Мобильные сети обеспечивают качество чуть ниже toll quality;
- **3-3.5** – качество речи, по-прежнему, удовлетворительно, однако его ухудшение явно заметно на слух;
- **2.5-3** – речь разборчива, однако требует концентрации внимания для понимания. Такое качество обычно обеспечивается в системах связи специального применения (например, в вооруженных силах).

В рамках существующих технологий качество ТфОП (toll quality) невозможно обеспечить при скоростях менее 5 Кбит/с.

Подавление периодов молчания (VAD, CNG, DTX)

При диалоге один его участник говорит, в среднем, только 35 процентов времени. Таким образом, если применить алгоритмы, которые позволяют уменьшить объем информации, передаваемой в периоды молчания, то можно значительно сузить необходимую полосу пропускания. В двустороннем разговоре такие меры позволяют достичь сокращения объема передаваемой информации до 50%, а в децентрализованных многоадресных конференциях (за счет большего количества говорящих) – и более. Нет никакого смысла организовывать многоадресные конференции с числом участников больше 5-6, не подавляя периоды молчания. Технология подавления таких периодов имеет три важные составляющие.

Нужно отметить, что определение границ пауз в речи очень существенно для эффективной синхронизации передающей и приемной сторон: приемник может, незначительно изменяя длительности пауз, производить подстройку скорости воспроизведения для каждого отдельного сеанса связи, что исключает необходимость синхронизации тактовых генераторов всех элементов сети, как это имеет место в ТфОП.

Детектор речевой активности (Voice Activity Detector – VAD) необходим для определения периодов времени, когда пользователь говорит. Детектор VAD должен обладать малым временем реакции, чтобы не допускать потерь начальных слов и не упускать бесполезные фрагменты молчания в конце предложений; в то же время детектор VAD не должен срабатывать от воздействия фонового шума.

Детектор VAD оценивает энергию входного сигнала и, если она превышает некоторый порог, активизирует передачу. Если бы детектор отбрасывал всю информацию до момента, пока энергия сигнала не стала выше порога, то происходило бы отрезание начальной части периода активности. Поэтому реализации VAD требуют сохранения в памяти нескольких миллисекунд информации, чтобы иметь возможность запустить передачу до начала периода активности. Это увеличивает, в некоторой степени, задержку прохождения сигнала, однако ее можно минимизировать или свести к нулю в кодерах, работающих с блоками отсчетов.

Поддержка прерывистой передачи (Discontinuous Transmission – DTX) позволяет кодеку прекратить передачу пакетов в тот момент, когда VAD обнаружил период молчания. Некоторые наиболее совершенные кодеры не прекращают передачу полностью, а переходят в режим передачи гораздо меньшего объема информации (интенсивность, спектральные характеристики), нужной для того, чтобы декодер на удаленном конце мог восстановить фоновый шум.

Генератор комфортного шума (Comfort Noise Generator – CNG) служит для генерации фонового шума. В момент, когда в речи активного участника беседы начинается период молчания, терминалы слушающих могут просто отключить воспроизведение звука. Однако это было бы неразумно. Если в трубке возникает «гробовая тишина», т.е. фоновый шум (шум улицы и т.д.), который был слышен во время разговора, внезапно исчезает, то слушающему кажется, что соединение по каким-то причинам нарушилось, и он обычно начинает спрашивать, слышит ли его собеседник.

Генератор CNG позволяет избежать таких неприятных эффектов. Простейшие кодеки просто прекращают передачу в период молчания, и декодер генерирует какой-либо шум с уровнем, равным минимальному уровню, отмеченному в период речевой активности. Более совершенные кодеки (G.723.1 Annex A, G.729 Annex B) имеют возможность предоставлять удаленному декодеру информацию для восстановления шума с параметрами, близкими к фактически наблюдавшимся.

Размер кадра

Большинство узкополосных кодеков обрабатывает речевую информацию блоками, называемыми кадрами (frames), и им необходимо производить предварительный анализ отсчетов, следующих непосредственно за отсчетами в блоке, который они в данный момент кодируют.

Размер кадра важен, так как минимальная теоретически достижимая задержка передачи информации (алгоритмическая задержка) определяется суммой этого параметра и длины буфера предварительного анализа. В действительности процессоры цифровой обработки сигналов, которые выполняют алгоритм кодирования, имеют конечную производительность, так что реальная задержка сигнала больше теоретической.

Можно, казалось бы, заключить, что кодеки с меньшим размером кадра лучше в смысле такого важного критерия как минимизация задержки. Если, однако, учесть, что происходит при передаче информации по сети, то мы увидим, что к кадру, сформированному кодеком, добавляется множество дополнительной информации – заголовки IP (20 байтов), UDP (8 байтов), RTP (12 байтов). Для кодека с длительностью кадра 30 мс посылка таких кадров по сети привела бы к передаче избыточной информации со скоростью 10.6 кбит/с, что превышает скорость передачи речевой информации у большинства узкополосных кодеков.

Поэтому обычно используется пересылка нескольких кадров в пакете, при этом их количество ограничено максимально допустимой задержкой. В большинстве случаев в одном пакете передается до 60 мс речевой информации. Чем меньше длительность кадра, тем боль-

ше кадров приходится упаковывать в один пакет, т.е. задержка определяется вовсе не длиной кадра, а практически приемлемым объемом полезной нагрузки в пакете.

Кроме того, кодеки с большей длиной кадра более эффективны, так как здесь действует общий принцип: чем дольше наблюдается явление (речевой сигнал), тем лучше оно может быть смоделировано.

Чувствительность к потерям кадров

Потери пакетов являются неотъемлемым атрибутом IP-сетей. Так как пакеты содержат кадры, сформированные кодеком, то это вызывает потери кадров. Но потери пакетов и потери кадров не обязательно напрямую связаны между собой, так как существуют подходы (такие как применение кодов с исправлением ошибок – forward error correction), позволяющие уменьшить число потерянных кадров при данном числе потерянных пакетов. Требуемая для этого дополнительная служебная информация распределяется между несколькими пакетами, так что при потере некоторого числа пакетов кадры могут быть восстановлены.

Однако положительный эффект от введения избыточности для борьбы с потерями пакетов не столь легко достижим, поскольку потери в IP-сетях происходят пачками, т.е. значительно более вероятно то, что будет потеряно сразу несколько пакетов подряд, чем то, что потерянные пакеты распределятся в последовательности переданных пакетов по одному. Так что если применять простые схемы введения избыточности (например, повторяя каждый кадр в двух последовательно передаваемых пакетах), то в реальных условиях они, хотя и увеличат объем избыточной информации, но, скорее всего, окажутся бесполезными.

Кроме того, введение избыточности отрицательно сказывается на задержке воспроизведения сигнала. Например, если мы повторяем один и тот же кадр в четырех пакетах подряд, чтобы обеспечить возможность восстановления информации при потере трех подряд переданных пакетов, то декодер вынужден поддерживать буфер из четырех пакетов, что вносит значительную дополнительную задержку воспроизведения.

Влияние потерь кадров на качество воспроизводимой речи зависит от используемого кодека. Если потерян кадр, состоящий из N речевых отсчетов кодека G.711, то на приемном конце будет отмечен пропуск звукового фрагмента длительностью $N \cdot 125$ мкс. Если используется более совершенный узкополосный кодек, то потеря одного кадра может сказаться на воспроизведении нескольких следующих, так как декодеру потребуется время для того, чтобы достичь синхронизации с кодером – потеря кадра длительностью 20 мс может приводить к слышимому эффекту в течение 150 мс и более.

Кодеры типа G.723.1 разработаны так, что они функционируют без существенного ухудшения качества в условиях некоррелированных потерь до 3% кадров, однако при превышении этого порога качество ухудшается катастрофически.

3.3 Кодеки, стандартизованные ITU-T

3.3.1 Кодек G.711

Кодек G.711 – «дедушка» всех цифровых кодеков речевых сигналов, был одобрен ITU-T в 1965 году. Применяемый в нем способ преобразования аналогового сигнала в цифровой с использованием полулогарифмической шкалы был достаточно подробно описан выше. Типичная оценка MOS составляет 4.2. В первую очередь отметим, что, как и для ТФОП, минимально необходимым для обслуживания VoIP является ИКМ-кодирование G.711. Это означает, что любое устройство VoIP должно поддерживать этот тип кодирования.

3.3.2 Кодек G.723.1

Рекомендация G.723.1 утверждена ITU-T в ноябре 1995 года. Форум IMTC выбрал кодек G.723.1 как базовый для приложений IP-телефонии.

Кодек G.723.1 производит кадры длительностью 30 мс с продолжительностью предварительного анализа 7.5 мс. Предусмотрено два режима работы: 6.3 Кбит/с (кадр имеет размер 189 битов, дополненных до 24 байтов) и 5.3 Кбит/с (кадр имеет размер 158 битов, дополненных до 20 байтов). Режим работы может меняться динамически от кадра к кадру. Оба режима обязательны для реализации.

Оценка MOS составляет 3.9 в режиме 6.3 Кбит/с и 3.7 в режиме 5.3 Кбит/с.

Кодек специфицирован на основе операций как с плавающей точкой, так и с фиксированной точкой в виде кода на языке C. Реализация кодека на процессоре с фиксированной точкой требует производительности около 16 MIPS.

Кодек G.723.1 имеет детектор речевой активности и обеспечивает генерацию комфортного шума на удаленном конце в период молчания. Эти функции специфицированы в приложении A (Annex A) к рекомендации G.723.1. Параметры фонового шума кодируются очень маленькими кадрами размером 4 байта. Если параметры шума не меняются существенно, передача полностью прекращается.

3.3.3 Кодек G.726

Алгоритм кодирования АДИКМ (рекомендация ITU-T G.726, принятая в 1990 г.) описан выше. Он обеспечивает кодирование цифрового потока G.711 со скоростью 40, 32, 24 или 16 Кбит/с, гарантируя оценки MOS на уровне 4.3 (32 Кбит/с), что часто принимается за эталон уровня качества телефонной связи (toll quality). В приложениях IP-телефонии этот кодек практически не используется, так как он не обеспечивает достаточной устойчивости к потерям информации (см. выше).

3.3.4 Кодек G.728

Кодек G.728 использует оригинальную технологию с малой задержкой LD-CELP (low delay code excited linear prediction) и гарантирует оценки MOS, аналогичные АДИКМ G.726 при скорости передачи 16 Кбит/с. Данный кодек специально разрабатывался как более совершенная замена АДИКМ для оборудования уплотнения телефонных каналов, при этом было необходимо обеспечить очень малую величину задержки (менее 5 мс), чтобы исключить необходимость применения эхокомпенсаторов. Это требование было успешно выполнено учеными Bell Labs в 1992 году: кодер имеет длительность кадра только 0.625 мс. Реально задержка может достигать 2.5 мс, так как декодер должен поддерживать синхронизацию в рамках структуры из четырех кадров.

Недостатком алгоритма является высокая сложность – около 20 MIPS для кодера и 13 MIPS для декодера – и относительно высокая чувствительность к потерям кадров.

3.3.5 Кодек G.729

Кодек G.729 очень популярен в приложениях передачи речи по сетям Frame Relay. Он использует технологию CS-ACELP (Conjugate Structure, Algebraic Code Excited Linear Prediction). Кодек использует кадр длительностью 10 мс и обеспечивает скорость передачи 8 Кбит/с. Для кодера необходим предварительный анализ сигнала продолжительностью 5 мс.

Существуют два варианта кодека:

- G.729 (одобрен ITU-T в декабре 1996), требующий около 20 MIPS для кодера и 3 MIPS для декодера.
- Упрощенный вариант G.729A (одобрен ITU-T в ноябре 1995), требующий около 10.5 MIPS для реализации кодера и около 2 MIPS для декодера.

В спецификациях G.729 определены алгоритмы VAD, CNG и DTX. В периоды молчания кодер передает 15-битовые кадры с информацией о фоновом шуме, если только шумовая обстановка изменяется.

3.4 Кодеки, стандартизованные ETSI

В рамках деятельности европейского института ETSI стандартизованы узкополосные кодеки для применения в системах мобильной связи (GSM).

Спецификации кодека *GSM Full Rate*, известного также как GSM 06.10, утверждены в 1987г. Это первый и, скорее всего, наиболее известный из узкополосных кодеков, применяемый в миллионах мобильных телефонов по всему миру. Обеспечивает хорошее качество и устойчивую работу в условиях фоновых шума (оценка MOS порядка 3.7 в условиях без шума). Кодируются кадры длительностью 20 мс, образуя цифровой поток со скоростью 13 Кбит/с. Кодек не требует высокой производительности процессора – необходимо только 4.5 MIPS для дуплексной реализации. Кодек очень важен для некоммерческих проектов в области IP-телефонии, особенно – для проектов, связанных с открытым распространением исходных текстов ПО (open source), благодаря возможности бесплатного лицензирования. Такие проекты сегодня могут использовать только кодеки GSM FR и G.711, а также АДИКМ.

Существуют также спецификации кодеков GSM Half Rate, принятые в 1994 году, и GSM Enhanced Full Rate, принятые в 1995 году. Характеристики этих кодеков превосходят характеристики исходного варианта, описанного выше, однако алгоритмы требуют большей производительности процессора (до 30 MIPS). В приложениях IP-телефонии они, по разным причинам, распространения пока не получили.

Рассмотрение кодеков было бы неполным, если бы, наряду со специфицированными ITU-T и ETSI, не были упомянуты и т.н. *нестандартные кодеки*.

Сегодня в приложениях VoIP, кроме кодеков, прошедших процедуры международной стандартизации в ITU-T и ETSI, в продуктах ряда фирм–производителей применяются также нестандартные внутрифирменные алгоритмы. Такие алгоритмы часто лицензируются для использования в продуктах других компаний. В качестве примеров можно назвать такие кодеки, как Lucent/Elementary SX7003P, имеющий очень хорошие характеристики при умеренной вычислительной сложности, и Voxware RT24, который предусматривает сверхнизкую (2.4 Кбит/с) скорость передачи информации при сохранении достаточно хорошего качества речи (оценка MOS около 3.2).

3.5 Передача сигналов DTMF

Строго говоря, сигналы многочастотного набора номера (DTMF) – это не что иное, как просто звуковые сигналы, передаваемые по те-

лефонному каналу. При передаче их по цифровой телефонной сети не возникает никаких проблем, так как кодирование при помощи алгоритма G.711 не накладывает никаких ограничений на вид звуковых сигналов – это может быть речь, сигналы модема, или тональные сигналы – все они будут успешно воспроизведены на принимающей стороне.

Узкополосные кодеки, чтобы достичь низких скоростей передачи, используют тот факт, что сигнал, который они кодируют, представляет именно речь. Сигналы DTMF при прохождении через такие кодеки искажаются и не могут быть успешно распознаны приемником на приемной стороне.

Когда пользователю ТфОП нужно ввести какую-то дополнительную информацию в удаленную систему при уже установленном соединении (например, номер дебитной карты или номер пункта меню автоинформатора), необходимо обеспечить возможность надежной передачи DTMF-сигналов через сеть IP-телефонии. В случаях, когда система, взаимодействующая с пользователем, просто задает вопрос и ждет ввода, длительность и момент передачи сигнала не важны. В других случаях система зачитывает пользователю список и просит его нажать, например, кнопку «#», как только он услышит нужную информацию; здесь ситуация более сложная, и необходима более точная привязка ко времени.

Существуют два основных метода передачи сигналов DTMF по сетям IP-телефонии.

- Обязательный метод. Специальное сообщение протокола H.245 (UserInputIndication) может содержать символы цифр и «*», «#». В данном случае используется надежное TCP-соединение, так что информация не может быть потеряна. Однако из-за особенностей TCP могут иметь место значительные задержки;
- Нестандартный метод, предложенный Форумом VoIP. Он может быть применен в терминалах H.323v2 при использовании процедуры fastStart и отсутствии канала H.245. Для передачи сигналов DTMF открывается специальная RTP-сессия, в которой передаются кодированные значения принятых цифр, а также данные об амплитуде и длительности сигналов. Может быть использована та же сессия, что и для речи, но со специальным типом полезной нагрузки. Использование RTP позволяет привязать DTMF-сигналы к реальному времени, что является важным преимуществом данного метода.

В принципе, первый метод может быть более предпочтительным, однако в случае международных вызовов и при использовании удаленных систем, требующих жесткой привязки ввода пользователя ко времени, может оказаться необходимым применить второй метод.

Шлюзы IP-телефонии должны обязательно подавлять искаженные сигналы DTMF, прошедшие через основной речевой канал. В противном случае, при восстановлении сигналов, о которых была принята информация, могут возникнуть неприятные эффекты наложения и размножения сигналов.

3.6 Передача факсимильной информации

В становлении IP-телефонии, наряду с телефоном, значительную роль сыграл телефакс. Идею нынешнего телефакса (от греческого «теле» – далеко и латинского «facsimile» – делай подобное) предложил англичанин Александр Байн в 1843 году, то есть за 33 года до появления телефона. В такой же последовательности (начиная с факсов) стали практически использоваться преимущества IP-телефонии с ее весьма низкими тарифами для передачи информации на дальние расстояния. Значительный экономический эффект от такого применения обусловлен чрезвычайно высокой распространенностью факс-машин; в мире их насчитывается много миллионов.

Говоря о распространенности факс-машин, отметим, что имеются в виду аппараты группы 3, специфицированные в рекомендации ITU-T T.30. Именно появление этой технологии и открыло дорогу широкому внедрению услуг факсимильной связи. Оказалось, что функции, реализованные в факсах группы 3, вполне устраивают пользователей, а стандарт практически не требует развития. Об этом свидетельствует тот факт, что более современная технология, т.н. факс группы 4, не получила никакого распространения и практически забыта. На наш взгляд, неуспех этой технологии можно объяснить тем, что, во-первых, все ее потенциальные преимущества (передача цветных изображений, высокая скорость обмена и т.д.) проще и дешевле реализуются на базе компьютерных технологий (обмен файлами по электронной почте, например), а во-вторых, сеть ISDN, на которую были ориентированы факсы группы 4, не получила глобального распространения.

Что же касается необходимости обеспечить возможность обмена факсимильными сообщениями факс-машин группы 3, то, в силу огромного количества последних, без такой функции не имеет смысла даже рассуждать о предоставлении услуг ТфОП на базе IP-сетей. Пересылка факсов через Интернет не является чем-то новым. Очень многие компании предлагают услуги факс-серверов отложенной доставки (Store & Forward). Пользователь отправляет факс на специальный сервер по заранее установленному телефонному номеру, вводя вслед за этим телефонный номер пункта назначения. Сервер, имитирующий работу факса принимающей стороны, принимает сообщение, преобразует его в набор графических файлов и отправляет дан-

ные файлы через Интернет к другому серверу, который находится ближе к месту назначения, например, в другой стране. Сервер-получатель организует связь с пунктом назначения по полученному им телефонному номеру и передает факсимильное сообщение адресату, уведомляя отправителя об успешной (или неуспешной) передаче. Технология Store & Forward Fax описана в рекомендации T.37.

Использование такого принципа пересылки факсов не очень удобно с точки зрения как пользователя, так и оператора сети IP-телефонии. Для пользователя в данном случае теряется одно из важнейших преимуществ факсимильной технологии – возможность сразу же узнать результат пересылки: доставлен ли документ, и с каким качеством он доставлен. Оператора же технология Store&Forward вынуждает принимать на себя дополнительную ответственность за успешную доставку сообщения, в то время как оно может оказаться не доставленным не по вине оператора, а просто потому, что адресат забыл включить свою факс-машину.

Единственным полноценным решением этих проблем является организация передачи факсов по IP-сетям в реальном времени и так, чтобы пользователи двух факсимильных аппаратов не подозревали о том, что связь между их терминалами осуществляется с использованием сети с коммутацией пакетов. К счастью, спецификации протокола передачи факсимильной информации группы 3 позволяют реализовать такое решение. Результатом усилий ITU-T в данном направлении стала рекомендация T.38, определяющая процедуры взаимодействия факсимильных терминалов группы 3 в реальном времени с использованием IP-сетей. Эта рекомендация позволяет обмен факсимильной информацией между факсами с использованием шлюзов, между факсом и компьютером, подключенными к Интернет, или даже между компьютерами, хотя последнее не кажется полезным свойством – просто при установлении соединения мы можем не догадываться, что имеем дело с компьютером, а не с факсом.

Принцип передачи факсов в реальном времени очевиден: на ближнем конце сигналы факса демодулируются и упаковываются в пакеты двоичных данных, а на удаленном конце происходит их восстановление в вид, пригодный для передачи по каналам ТфОП. Кроме собственно информационных пакетов, содержащих управляющие последовательности и графические данные, передается также информация обо всех прочих событиях, связанных с передачей факса, т.е. о тональных сигналах и служебных последовательностях, необходимых для настройки приемников модемных сигналов. Такой подход, по понятным причинам, не требует для передачи факса значительной полосы пропускания. Однако нужно отдавать себе отчет в том, что факсимильные сессии более требовательны к качеству обслуживания, чем речевые, в связи с особенностями протокола передачи факсимильной информации. Действительно, потеря 100 мс ре-

чевой информации может быть воспринята лишь как щелчок, тогда как для факсимильной сессии потеря всего одного информационно-го пакета может обернуться потерей нескольких строк изображения.

Рекомендация T.38 предусматривает использование особого протокола IFP, цель которого – перенос сообщений между шлюзами и/или компьютерами. Сообщения IFP, в свою очередь, могут передаваться внутри TCP-соединения или с использованием UDP, причем в последнем случае предусматривается введение информационной избыточности, обеспечивающей восстановление одиночных потерянных пакетов. Использование протокола T.38 закреплено в рамках рекомендации H.323. Обязательным условием является поддержка протокола TCP для переноса информации IFP, а использование протокола UDP является лишь возможным вариантом. Информация IFP передается по двум логическим каналам (от отправителя к получателю и в обратном направлении). Когда в качестве транспорта применяется протокол TCP, существует два возможных варианта: передавать сообщения IFP, используя их туннелирование в канале H.225.0/Q.931, или использовать для этого выделенное соединение.

Несмотря на то, что согласно ITU-T реализация на основе протокола TCP является обязательной, в шлюзах большинства крупных производителей реализован транспорт IFP поверх протокола UDP. Отчасти это можно объяснить тем, что при таком решении механизм открытия логических каналов выглядит совершенно аналогично механизму, используемому для передачи речевой информации. Кроме того, протокол T.38 обычно реализуется на основе либо тех же DSP, что и речевые кодеки, либо специализированного процессора, обеспечивающего пересылку речевой информации, а для таких процессоров реализация протокола TCP слишком тяжеловесна, и ее стараются избежать. Как бы то ни было, реализации T.38 на базе протокола UDP широко эксплуатируются и доказали работоспособность такого решения. Шлюз IP-телефонии семейства оборудования Протей-IP использует транспорт UDP, а вариант с TCP может быть реализован, если на рынке появится в достаточном количестве оборудование, использующее этот подход.

3.7 О реализации «стандартных» алгоритмов

Как может показаться на первый взгляд, узкополосное кодирование речи, требующее огромной (миллионы операций в секунду) вычислительной мощности, является самой сложной задачей, выполняемой оборудованием IP-телефонии. Однако это отнюдь не так: алгоритмы кодирования речи стандартизованы и отлично документированы, более того, на рынке доступны весьма эффективные их реализации для всех популярных DSP-платформ. С другой стороны,

в оборудовании IP-телефонии должны быть реализованы многие другие функции, способ реализации которых не является объектом стандартизации, а представляет собой «know-how» разработчиков.

На передающей стороне оборудование IP-телефонии работает по принципу «закодировал, передал и забыл». На приемной стороне все гораздо сложнее. Пакеты приходят из сети с задержкой, меняющейся по случайному закону. Более того, пакеты могут придти не в той последовательности, в которой были переданы, а некоторые пакеты могут вообще быть потеряны. Приемник должен справляться со всеми этими трудностями, обеспечивая на выходе нормальный звуковой поток с тактовой синхронизацией, либо генерируемой на основе принимаемого потока данных, либо получаемой из ТФОП по каналам E1. Привязка речевых потоков к местному тактовому синхросигналу производится, как уже отмечалось выше, путем незаметной на слух деформации периодов молчания в воспроизводимом сигнале.

К этому остается добавить необходимость передачи факсимильной информации в реальном времени с автоматическим распознаванием сигналов факсимильных аппаратов и передачу DTMF-сигналов с корректным их восстановлением в приемнике.

На основе данного обзора функций оборудования IP-телефонии можно сделать вывод, о том что, несмотря на существование стандартных алгоритмов кодирования речи, у разработчиков есть огромный простор для деятельности, направленной на дальнейшее совершенствование технологии IP-телефонии.

Глава 4

Протоколы

сети Интернет

4.1 Интернет ab ovo

Общеизвестна дата начала знаменитого проекта сети пакетной коммутации ARPA – прототипа сегодняшней сети Интернет. Это 1971 год. Однако сама идея сети Интернет имеет гораздо более давнюю историю. Чего стоит одно только определение сетевой структуры, данное Буддой: «Как сеть состоит из множества узлов, так и всё на этом свете связано узлами. Если кто-то полагает, что ячейка сети является чем-то независимым, изолированным, то он ошибается. Она называется сетью, поскольку состоит из множества взаимосвязанных ячеек, и у каждой ячейки своё место и свои обязательства по отношению к другим ячейкам».

Реальная история Интернет началась, разумеется, спустя много веков после того, как появилось это определение. Авторы данной книги датируют ее начало 1957 годом – датой запуска первого советского искусственного спутника Земли. Именно в ответ на этот запуск США сформировали в составе Минобороны (Department of Defense – DoD) специальное агентство – Advanced Research Projects Agency (ARPA), создавшее сеть ARPANET – прообраз Интернет – и собравшее вокруг себя коллектив исследователей и ученых, заложивших основы сегодняшней сети Интернет. Таким образом, именно события, связанные с запуском первого советского спутника, стимулировали интеллектуальные усилия тысяч разработчиков и саму Интернет-революцию, которая сейчас сотрясает мир. Уже в июле 1961 года Леонард Клейнрок опубликовал первую статью по теории пакетной коммутации «Information Flow in Large Communication Nets».

В 1964 году последовала работа Пола Барана (Paul Baran, RAND) «On Distributed Communications Networks»

В 1965 году, под эгидой ARPA, компьютеры двух организаций – TX-2 в MIT Lincoln Lab и AN/FSQ-32 в System Development Corporation (Santa Monica, CA) – были связаны выделенной телефонной линией на скорости 1200 бит/с. В октябре 1967 года в Гатлинбурге, штат Теннесси, на симпозиуме ACM собрались представители трех независимых команд – ARPA, RAND и NPL. Последняя из них, Национальная физическая лаборатория (National Physical Laboratory – NPL), построившая экспериментальную сеть пакетной коммутации на скорости 768 Кбит/с, более известна тем, что руководитель разработки Дональд Дэвис является автором термина «пакет».

В 1969 году на основе мини-компьютера DDP-516 фирмы Honeywell с памятью объемом 12К были созданы четыре первых узла сети ARPANET: Калифорнийский университет в Лос-Анжелесе (University of California Los Angeles – UCLA), Стэнфордский НИИ (Stanford Research Institute – SRI), университет Санта-Барбары и университет штата Юта. Компания AT&T предоставила для этой сети линии со скоростью передачи 50 Кбит/с. Первые пакеты данных были переданы Чарли Клайном (Charley Kline) из UCLA, когда он пытался связаться с компьютером SRI. Первая попытка 29 октября 1969 г. закончилась аварийным отказом системы во время ввода буквы G из слова LOGIN. Пикантность ситуации заключалась в том, что ARPANET определялась как полностью отказоустойчивая компьютерная сеть с распределением вычислительной мощности и резервированием устройств коммутации данных и компьютерных линий связи, способная выдержать ядерный удар. Тогда же первым проектом документа RFC (Request for Comments) «Программное обеспечение рабочих станций (hosts)» было положено начало стандартам Интернет.

Первая публикация, относящаяся к сети Интернет, появилась в 1970 году и была посвящена протоколу взаимодействия рабочих станций в составе сети ARPANET: Ч.Кар, С.Крокер, В.Серф. «Протокол связи рабочих станций в сети ARPA».

В 1971 году уже имеется 15 узлов (23 рабочие станции), и Рей Момлинсон изобретает программу электронной почты для передачи сообщений по распределенной сети. Оригинальная программа была создана на базе двух программ: программы внутримашинной электронной почты (SENDMSG) и экспериментальной программы пересылки файлов (CPYNET). Из клавиш пунктуации на телетайпе Model 33 производства Tomlinson в марте 1972 г. выбран знак @ в его значении «в».

Докторская диссертация Боба Меткафа из Гарварда наметила идею сети Ethernet. Эта идея была проверена на компьютерах Alto производства Xerox PARC, и первая сеть Ethernet получила в мае 1973 г. на-

звание Alto Aloha System. А число пользователей ARPANET к марту 1973 достигло 2000. Тогда же в Мичиганском университете создается сеть Merit на базе протокола X.25, а в Стэнфордском университете начинается разработка набора протоколов, которые должны обеспечивать взаимодействие компьютеров, включённых в сеть ARPANET.

В мае 1974 года Винт Серф из Стэнфорда и Боб Кан (Vint Cerf, Bob Kahn) из агентства перспективных научных проектов Министерства обороны США опубликовали в журнале «IEEE Transactions on Communications» статью «A Protocol for Packet Network Intercommunication» со спецификацией протокола TCP, ставшей вскоре основой Интернет. Об истории этой публикации рассказывается в колонке редактора журнала «Сети и системы связи» (№11, 1999). Серф и Кан решали, чье имя поставить первым в спецификации протокола TCP, и бросили монетку. Первым оказался Винсент Серф, и именно он, со временем, стал известен широкой публике, занял должность вице-президента MCI и получил признание как один из основателей сети Интернет.

Тогда же появляется первый список адресов электронной почты – MsgGroup. Самым популярным неофициальным списком становится список любителей научной фантастики – SF-Lovers. Спустя полтора года разрабатывается спецификация электронной почты (RFC 733). В 1976 году в лаборатории Александра Белла (Bell Laboratories) корпорации AT&T разрабатывается протокол UUCP (Копирование Unix-Unix), который начинает распространяться год спустя вместе с операционной системой UNIX.

В 1978 году протокол TCP разделяется на протоколы TCP и IP, а изложенная в статье Серфа и Кана концепция получает название TCP/IP. Работа над концепцией была завершена в 1980 году, а в 1983 году управление Минобороны США утвердило новый набор компьютерных протоколов в качестве стандарта для ARPANET, вместо протокола NCP (Network Control Protocol), использовавшегося с 1970 года. Чтобы поощрить переход колледжей и университетов на технологию TCP/IP, силами ARPA был облегчен процесс внедрения операционной системы Berkeley UNIX, реализующей протоколы TCP/IP. Этот шаг привел к формированию одного из первых определений понятия «интернет» как совокупности соединенных между собой сетей, в частности, сетей с использованием стека протоколов TCP/IP, а понятия «Интернет» как совокупности сетей, реализованных на базе технологии TCP/IP.

Сама же ARPANET в конце 1983 года разделилась на две сети: DARPANET (оборонная сеть) и MILNET (военная сеть). Несмотря на то, что ARPANET официально прекратила своё существование в июне 1990 года, сеть Интернет уцелела. Более того, протокол TCP/IP был усовершенствован и стал чрезвычайно популярен в сферах образования, научно-исследовательских разработок, в коммерции и во мно-

гих, многих других, порой неожиданных применениях, примером чему является эта книга. В 1984 году вводится система доменных имен (DNS). Она увязывает IP-адреса с именами компьютеров в Интернет. И в том же 1984 году Уильям Гибсон, в романе «Неутомансер», вводит термин «гиперпространство». Количество рабочих станций, подключенных к сети, достигает 1000, а к 1987 году их становится уже 10000.

Национальный Фонд Науки США, с целью обеспечить взаимодействие своих суперкомпьютерных центров и доступ к Интернет, создает в 1985 году сеть NSFNET (Сеть Национального фонда науки). NSFNET – это высокоскоростная магистральная сеть, состоящая из двухточечных линий связи в узловой конфигурации. Сеть была полностью развернута в 1988 году и первоначально работала на скорости 56 Кбит/с. В 1986 году NSFNET организовала ряд региональных сетей, объединённых в магистральную сеть. Позже основные части сети NSFNET были модернизированы для работы на скоростях до OC-3 (155 Мбит/с) и выше. NSFNET официально прекратила своё существование в 1995 году, и на смену ей пришла сеть MERIT. Первоначально, MERIT была сетью масштаба штата, эксплуатируемой Университетом Мичигана, и региональным компонентом как сети NSFNET, так и Интернет.

В 1988 году Ажако Ойкаринен (Jarkko Oikari-nen) разрабатывает технологию Internet Relay Chat (IRC). Тогда же появился новый термин «хакер», а 1 ноября 1988 года вирусная программа «Internet Worm» сумела повредить более 6000 рабочих станций.

В 1989 году количество рабочих станций достигает 100000 штук, а в 1990 году – 300000. Появляется первый коммерческий поставщик услуг сети Интернет, а год спустя Тим Бернерс-Ли (Tim Berners-Lee) из организации CERN, Швейцария, выпускает свою разработку Всемирной Паутины – World Wide Web (WWW). Вскоре программы просмотра WWW стали неотъемлемой частью повседневной жизни, и началась эра бизнеса в сети Интернет. В 1992 году количество рабочих станций превысило 1 миллион, а в 1993 году в Интернет появился адрес www.whitehouse.gov и электронная почта president@whitehouse.gov Б. Клинтона. Понадобилось менее 2 лет, чтобы в Интернет появились адреса правительств большинства других стран, включая, например, адрес Ватикана – www.vatican.va. В том же 1995 году коллектив программистов Sun Microsystems под руководством Джеймса Гозлинга (James Gosling) создали язык программирования Java, радикально изменивший сам смысл программирования Интернет-приложений.

В 1996 году сети Интернет исполнилось 25 лет, а число рабочих станций, подключенных к Интернет, составило 10 миллионов.

К концу 2000 года насчитывалось около 300 миллионов пользователей Интернет. Количество рабочих станций сейчас возрастает примерно на 80 процентов за год. В первой главе приведен рисунок, на котором авторы взяли на себя смелость показать, что приблизительно к 2004 году Интернет разрастется до размеров современной телефонной сети.

4.2 Стандарты в сфере Интернет

Из материала предыдущего параграфа видно, что Интернет – самая необычная из всех сетей. Практически любой объект может подключиться к Интернет, чтобы предложить ресурсы, или для доступа к ним. По Интернет может гулять практически любой вид информации без каких-либо ограничений. Отсутствует центральный орган, который регулировал бы работу сети Интернет, хотя существуют организации, устанавливающие определённые фундаментальные принципы и руководящие работой сети. Сеть Интернет по своей философии является автономной и даже анархической; в конечном счёте, в этом и её сила, и её слабость.

Существует ряд организаций, которые участвуют в различных мероприятиях по администрированию и поддержке Интернет. В контексте данной книги среди этих организаций следует упомянуть CERT, IAB, IETF, IESG, IRTF, ICANN и The Internet Society (Общество Интернет, известное также как ISOC).

Группа реагирования на нарушения компьютерной защиты (CERT) – группа экспертов Университета Карнеги-Меллона, которая отвечает за вопросы, связанные с нарушением компьютерной защиты в сети Интернет. CERT была образована ARPA в ноябре 1998 года как реакция на ряд инцидентов, связанных с появлением вирусных программ самотиражирования.

Совет по архитектуре Интернет (IAB), первоначально – *Координационный совет сети Интернет* – добровольный орган, имеющий в своем составе 12 экспертов, которые используют ресурсы своих компаний-спонсоров для того, чтобы способствовать интересам Интернет. IAB контролирует и координирует деятельность двух проблемных (рабочих) групп: IETF и IRTF. В совокупности, эти организации вырабатывают техническую политику и направления работы.

Инженерная проблемная группа Интернет (IETF) определяет, устанавливает приоритеты и вырабатывает решения по краткосрочным вопросам и проблемам, включая протоколы, архитектуру и эксплуатацию. Предложенные стандарты публикуются в Интернет в виде Запросов комментариев и предложений (RFC). После выработки окончательной версии стандарта, он поступает на утверждение в *группу управления инженеров Интернет (IESG)*.

Научно-исследовательская проблемная группа Интернет (IETF) занимается долгосрочными вопросами, включая схемы адресации и технологии.

Корпорация Интернет по присвоению имен и номеров (ICANN) – некоммерческая организация, образованная в 1999 году. ICANN была создана для того, чтобы взять на себя полномочия федерального органа IANA по распределению общеизвестных номеров портов, управлению IP-адресами и присвоению имён доменов. Номера портов представляют собой 16-битовые величины в диапазоне от 0 до 65 536. Общеизвестные порты нумеруются числами из диапазона от 0 до 1 023 и используются системными процессами или прикладными программами. Примерами общеизвестных портов являются: порт 25 для протокола SMTP (Простого протокола пересылки почты), порт 80 для протокола HTTP (Гипертекстового транспортного протокола) и порт 107 для Дистанционной службы Telnet. В среде клиент/сервер Интернет на базе протокола TCP/IP, сервер назначает порты с учётом протокола прикладного уровня, который выполняется на клиентском уровне. ICANN также присваивает IP-адреса организациям, желающим поместить компьютеры в Интернет; количество адресов зависит от размера организации.

Общество Интернет (ISOC) – добровольная организация, которая представляет собой некоторую формальную структуру для администрирования Интернет. Общество Интернет предоставило официальные полномочия IESG принимать решения по стандартам.

4.3 Адресация

Интернет- это совокупность тысяч компьютеров, объединенных в сети, которые, в свою очередь, соединены между собой посредством маршрутизаторов. Для организации линий связи используются практически все возможные технологии – от коммутируемых телефонных соединений до самых современных оптических систем передачи. Активно используются и спутниковые линии связи.

Сеть Интернет имеет иерархическую структуру. Этот подход является эффективным потому, что позволяет идентифицировать компоненты Интернет посредством адресов, также имеющих иерархическую структуру. Например, в телефонной сети полный номер абонента содержит такие составляющие как код страны, код зоны, номер АТС, номер абонента в АТС. Аналогичная концепция была принята и в сети Интернет: старшие биты адреса идентифицируют сеть, в которой находится рабочая станция, а младшие – расположение рабочей станции в этой сети.

Подавляющее большинство сетей сейчас использует протокол *IPv4 (интернет – протокол версии 4)*, хотя уже разработана шестая версия протокола IP, которая применяется в некоторых недавно соз-

данных крупных сетях. Схема адресации протокола IPv4, который был определён в RFC 791, предусматривает размер адресного поля 32 бита, что даёт 2^{32} (или 4 294 967 296) потенциальных адресов.

IP-адрес любой рабочей станции состоит из адреса сети и адреса компьютера в этой сети. В архитектуре адресации предусмотрено пять форматов адреса, каждый из которых начинается с одного, двух, трёх или четырёх битов, идентифицирующих класс сети (класс A, B, C, D или E). Область сетевого идентификатора (Network ID) определяет конкретную сеть в классе, а область Host ID идентифицирует конкретный компьютер в сети¹.

- Адреса класса A идентифицируются начальным битом 0. Следующие семь битов определяют конкретную сеть (число возможных значений – 128 или 2^7). Остальные 24 бита определяют конкретный компьютер в сети, при возможном количестве компьютеров – 16 777 216 (2^{24}). Адреса класса A предназначены для очень крупных сетей с большим количеством рабочих станций. Первые адреса класса A были присвоены таким компаниям как IBM Corporation, Hewlett-Packard Company, Ford Motor Company и др.
- Адреса класса B идентифицируются начальной двухбитовой двоичной последовательностью 10. Следующие 14 битов определяют сеть, при возможном количестве сетей 16 384 (2^{14}). Остальные 16 битов определяют конкретный компьютер, с возможным количеством компьютеров – 65 536 (2^{16}).
- Адреса класса C идентифицируются начальной трёхбитовой последовательностью 110. Следующие 21 бит определяют сеть, с возможным количеством сетей – 2 097 152. Остальные 8 битов определяют конкретный компьютер в сети, с возможным количеством компьютеров – 256 (2^8). Большинство организаций имеют адреса класса C.
- Адреса класса D идентифицируются начальной четырёхбитовой последовательностью 1110. Адреса этого класса предназначены для групповой передачи, и оставшиеся 28 битов определяют групповой адрес.
- Адреса класса E идентифицируются начальной четырёхбитовой двоичной последовательностью 1111. Адреса этого класса зарезервированы для будущего использования.

Способ, при помощи которого записываются все IP-адреса, называется пунктирной десятичной системой обозначений. Каждое 32-битовое адресное поле разделено на четыре поля в виде xxx.xxx.xxx.xxx, и каждому полю даётся десятичное числовое значение от 0 до 255, выраженное в виде одного октета ($2^8 = 256$ или 0-255). Адреса класса A начинаются с 1-127, адреса класса B – с 128-191, и адреса класса C – с 192-223.

¹ Строго говоря, адрес идентифицирует только сетевой интерфейс рабочей станции, т.е. точку подключения к сети.

Как отмечалось выше, *корпорация Интернет по присвоению имен и номеров (ICANN)* присваивает IP-адреса организациям, желающим подключить компьютеры к сети Интернет. Класс IP-адреса и, следовательно, количество возможных адресов компьютеров зависит от размеров организации. Организация, которой присвоены номера, может затем переназначить их на основе либо статической, либо динамической адресации. *Статическая адресация* означает жесткую привязку IP-адреса к конкретному компьютеру. При *динамической адресации* компьютеру присваивается доступный IP-адрес всякий раз при установлении соединения. Например, поставщик услуг Интернет может иметь один или несколько адресных блоков класса C. При ограниченном количестве доступных IP-адресов поставщик присваивает IP-адрес компьютеру пользователя всякий раз, когда пользователь коммутируемой линии получает к нему доступ, чтобы установить соединение с Интернет. После завершения соединения этот IP-адрес может присваиваться другим пользователям. Динамическое присвоение IP-адресов обычно осуществляется через маршрутизатор, работающий по протоколу *DHCP (Протокол динамической конфигурации рабочей станции)*. Наоборот, если доступ к поставщику осуществляется по xDSL, поставщик услуг Интернет обычно присваивает пользователю один или более статических IP-адресов. Так как соединение по xDSL всегда активизировано, динамическая адресация для этой категории пользователей неприменима.

Как уже отмечалось, протокол IP версии 4 предусматривает размер адресного поля 32 бита, что даёт 2^{32} (или 4 294 967 296) потенциальных адресов. Однако возрастающая популярность технологии TCP/IP привела к истощению плана нумерации протокола IPv4 аналогично тому, как популярность подключённых к телефонной сети факсимильных аппаратов, сотовых телефонов, пейджеров, компьютерных модемов и даже копировальных машин привела к истощению плана нумерации ТфОП. Дополнительной проблемой является тот факт, что очень большое количество адресов класса А и класса В было выделено крупным организациям, которые в них на самом деле не нуждались. В связи с тем, что фактически использовался только небольшой процент адресов, огромное количество доступных адресов было потеряно. Это напоминает расточительность, с которой выделялись телефонные номера в городских телефонных сетях (за исключением МГТС) блоками по 10 000 номеров, зачастую вне зависимости от того, сколько их требовалось реально – 100 или 1000.

Чтобы смягчить, по крайней мере – частично, эту проблему, комитет IETF в начале 90-х годов опубликовал в документах RFC 1518 и RFC 1519 положение о *бесклассовой междоменной маршрутизации (CIDR)*. Технология CIDR построена на концепции *суперсети (su-*

pernetting), состоящей из группы подсетей, каждой из которых присваивается адрес подсети. Но в целом совокупность подсетей выглядит, как единая сеть с одним префиксом (например, для Европы выделены префиксы 194 и 195). Благодаря технологии CIDR, сокращается число маршрутов и, следовательно, размер и сложность таблиц маршрутизации, которые должны поддерживать коммутаторы и маршрутизаторы. Несмотря на то, что CIDR приносит известную гибкость в схему IP-адресации, она, тем не менее, не решает главной проблемы – недостатка IP-адресов в обозримом будущем.

Протокол IPv6 решает этот вопрос путём расширения адресного поля до 128 битов, обеспечивая тем самым 2^{128} потенциальных адресов, что составляет величину 340.282.366.920.938.463.463.374.607.431.768.211.456. По расчётам Кристиана Хюйтема, такого адресного пространства достаточно, чтобы присвоить по 32 адреса каждому квадратному дюйму суши на Земле – что, в принципе, должно решить проблему. С учётом предложений о присвоении IP-адресов сетевым кофеваркам, холодильникам, системам обогрева и кондиционирования, автомобилям и вообще всем мыслимым устройствам, ценность и рентабельность протокола IPv6 возрастёт ещё больше. Протокол IPv6 обладает также дополнительными функциональными возможностями, хотя для их реализации потребуется модернизация существующего сетевого программного обеспечения.

Но вернемся к протоколу IPv4. Компьютер, подключенный к сети Интернет, кроме IP-адреса может идентифицироваться доменным именем. Сеть Интернет разделена на логические области (домены). Адреса в *системе имён доменов (DNS)*, администрирование которых лежит на ICANN, имеют стандартный вид, представляющий собой последовательность имен, разделенных точками, например: *компьютер.организация.домен*. Подавляющее большинство из 45 миллионов (или около того) зарегистрированных доменов верхнего уровня (TLD) является коммерческими. Домены TLD, которые идентифицируются как суффикс доменного имени, бывают двух типов: *обобщённые домены верхнего уровня (net, com, org)* и *коды стран (ru, fi, ua)*.

Сам же ICANN получил от IANA полномочия по администрированию Интернет-адресов. При администрировании со стороны IANA, ответственность за присвоение TLD возлагалась на *центр сетевой информации Интернет (InterNIC)* компании Network Solutions Inc. В течение первых десятилетий существования Интернет, присвоение доменов было бесплатным. Позже, InterNIC начал брать плату за домены *.com* в размере \$70 за первые два года и \$35 за каждый следующий год. В 1999 году InterNIC потерял монопольное право на присвоение доменов, так как в апреле 1999 года были утверждены четы-

ре конкурентные организации на испытательный срок до 25 июня 1999 года. ICANN также объявил, что ряд других заявителей удовлетворяют его критериям аккредитации, и они будут аккредитованы по окончании испытательного срока.

Имена доменов гораздо легче запомнить и ввести, но необходимо преобразование для перевода имён доменов в IP-адреса; это необходимо для того, чтобы разные маршрутизаторы и коммутаторы могли направить информацию в нужный пункт назначения.

4.4 Уровни архитектуры Интернет

Функционирование сети Интернет основано на сложном комплексе протоколов, обеспечивающих выполнение различных функций – от непосредственно передачи данных до управления конфигурацией оборудования сети.

Для того, чтобы классифицировать различные протоколы и понять их место в общей структуре технологии межсетевого взаимодействия, удобно воспользоваться так называемым «многоуровневым представлением сетевых протоколов». В рамках такого представления подразумевается, что протоколы более высокого уровня используют функции протоколов более низкого уровня. Классической, хотя и представляющей сейчас, скорее, академический интерес, моделью такого рода является семиуровневая модель взаимодействия открытых систем (Open Systems Interconnection – OSI), разработанная ИТУ-Т в рамках неудавшейся попытки создать международный стандарт семейства сетевых протоколов. Вместе с тем, некоторые результаты данного проекта являются хорошим материалом для учебников, чем мы и воспользуемся.

Рис. 4.1 иллюстрирует взаимоотношения архитектуры Интернет, определенной ARPA, с моделью OSI, а также поясняет функции каждого из уровней.

Архитектура Интернет была разработана агентством ARPA для соединения компьютеров в государственных, военных, академических и других организациях, в основном, на территории США, что обусловило ее практический характер. С другой стороны, модель OSI охватывала более широкий круг вопросов передачи информации, и в ее рамках не был конкретизирован тип взаимодействующих систем, что породило более «дробное» разбиение на уровни. Однако между той и другой архитектурой имеется очевидное соответствие.

Первый уровень модели ARPA – уровень сетевого интерфейса – поддерживает физический перенос информации между устройствами в сети, т.е. объединяет функции двух уровней OSI – физического и звена данных. Уровень сетевого интерфейса обеспечивает физическое соединение со средой передачи, обеспечивает, если

это необходимо, разрешение конфликтов, возникающих в процессе организации доступа к среде (например, используя технологию CSMA/CD в сети Ethernet), упаковывает данные в пакеты. Пакет² – это протокольная единица, которая содержит информацию верхних уровней, и служебные поля (аппаратные адреса, порядковые номера, подтверждения и т.д.), необходимые для функционирования протоколов этого уровня.



Рис. 4.1 Уровни модели OSI и архитектуры Интернет

Сетевой уровень отвечает за передачу информации, упакованной в дейтаграммы (datagram), от одного компьютера к другому. Дейтаграмма – это протокольная единица, которой оперируют протоколы семейства TCP/IP. Она содержит адресную информацию, необходимую для переноса дейтаграммы через сеть, а не только в рамках одного звена данных. Понятие дейтаграммы никак не связано с физическими характеристиками сетей и каналов связи, что подчеркивает независимость протоколов TCP/IP от аппаратуры. Основным протоколом, реализующим функции сетевого уровня, является протокол IP. Этот протокол отвечает за маршрутизацию, фрагментацию и сборку дейтаграмм в рабочей станции.

Обмен между сетевыми узлами информацией о состоянии сети, необходимой для формирования оптимальных маршрутов следования дейтаграмм, обеспечивают протоколы маршрутизации – RIP, EGP, BGP, OSPF и др.

² Иногда при рассмотрении протоколов этого уровня (Ethernet, HDLC) употребляется также термин кадр (frame).

Протокол преобразования адресов (Address Resolution Protocol – ARP) преобразует IP-адреса в адреса, используемые в локальных сетях (например, Ethernet). На некоторых рисунках, изображающих архитектуру и взаимосвязь протоколов, ARP размещают ниже IP, чтобы показать его тесную взаимосвязь с Уровнем Сетевого Интерфейса.

Протокол контрольных сообщений – Internet Control Message Protocol (ICMP) предоставляет возможность программному обеспечению рабочей станции или маршрутизатора обмениваться информацией о проблемах маршрутизации пакетов с другими устройствами в сети. Протокол ICMP – необходимая часть реализации стека протоколов TCP/IP.

Когда дейтаграмма проходит по сети, она может быть потеряна или искажена. Транспортный уровень решает эту проблему и обеспечивает надежную передачу информации от источника к приемнику. Кроме того, реализации протоколов этого уровня образуют универсальный интерфейс для приложений, обеспечивающий доступ к услугам сетевого уровня. Наиболее важными протоколами транспортного уровня являются TCP и UDP.

Конечные пользователи взаимодействуют с компьютером на уровне приложений. Разработано множество протоколов, используемых соответствующими приложениями. Например, приложения передачи файлов используют протокол FTP. Web-приложения используют протокол HTTP. Оба протокола FTP и HTTP базируются на протоколе TCP. Приложение Telnet обеспечивает подключение удаленных терминалов. Протокол эксплуатационного управления сетью SNMP позволяет управлять конфигурацией оборудования в сети и собирать информацию об его функционировании, в том числе, и о аварийных ситуациях. Приложения, созданные для организации речевой связи и видеосвязи, используют протокол RTP для передачи информации, чувствительной к задержкам. X Window – популярный протокол для подключения к интеллектуальному графическому терминалу. Этот список можно продолжать практически бесконечно.

Таким образом, IP-сети используют для передачи информации разнообразные протоколы, причем функции протоколов не зависят от того, какие данные передаются. Иными словами, IP, ARP, ICMP, TCP, UDP и другие элементы стека протоколов TCP/IP предоставляют универсальные средства передачи информации, какой бы она ни была природы (файл по FTP, Web – страница или аудиоданные).

4.5 Протокол IP версии 4

В качестве основного протокола сетевого уровня в стеке протоколов TCP/IP используется протокол IP, который изначально проек-

тировался как протокол передачи пакетов в сетях, состоящих из большого количества локальных сетей. Поэтому протокол IP хорошо работает в сетях со сложной топологией, рационально используя наличие в них подсистем и экономно расходуя пропускную способность низкоскоростных линий связи. Протокол IP организует пакетную передачу информации от узла к узлу IP-сети, не используя процедур установления соединения между источником и приемником информации. Кроме того, Internet Protocol является дейтаграммным протоколом: при передаче информации по протоколу IP каждый пакет передается от узла к узлу и обрабатывается в узлах независимо от других пакетов.

Протокол IP не обеспечивает надежность доставки информации, так как он не имеет механизмов повторной передачи. Он не имеет также и механизмов управления потоком данных (flow-control). Дейтаграммы могут быть потеряны, размножены, или получены не в том порядке, в каком были переданы.

Протокол IP базируется на протоколе уровня звена данных, который обеспечивает передачу данных по физической среде. Программный модуль, реализующий протокол IP, определяет маршрут переноса данных по сети до точки назначения, или до промежуточного маршрутизатора, где дейтаграмма извлекается из кадра локальной сети и направляется в канал, который соответствует выбранному маршруту. Дейтаграммы могут разбиваться на более мелкие фрагменты, или, наоборот, несколько дейтаграмм могут объединяться в одну на стыке разных сетей, если эти сети поддерживают передачу дейтаграмм разной длины.

В каждой рабочей станции, подключенной к IP-сети, обработка IP-дейтаграмм, производится по одним и тем же правилам адресации, фрагментации и маршрутизации. Рабочие станции рассматривают каждую дейтаграмму как независимую протокольную единицу, так как протокол IP не использует логических соединений или каких-либо других средств идентификации виртуальных каналов³.

На рис. 4.2 показана структура протокольной единицы протокола IP – дейтаграммы.

Поле *версия* (version) идентифицирует используемую версию протокола IP, в рассматриваемом случае указывается версия 4. Необходимость этого поля объясняется тем, что в переходный период в сети могут использоваться протоколы разных версий.

Поле *длина заголовка* (header length), состоящее из 4 битов, определяет длину заголовка, причем длина указывается как количество блоков размером 32 бита. В типичном случае значение этого поля равно 5.

³ При рассмотрении протокола IP версии 6 и вопросов обеспечения качества обслуживания, мы увидим некоторые отклонения от этого принципа.

Версия (Version)	Длина заголовка
Тип обслуживания	
Общая длина	
Идентификатор фрагмента	
Флаги	Смещение фрагмента
Время жизни	
Протокол	
Контрольная сумма заголовка	
Адрес отправителя	
Адрес получателя	
Опциональные поля и заполнение	
Данные	

Рис. 4.2 IP-дейтаграмма

Поле *тип обслуживания* (Type of Service) содержит информацию, которая бывает нужна при поддержке сетью разных классов обслуживания. Использование этого поля в Интернет будет возрастать по мере роста в IP-сетях возможностей передачи мультимедийного трафика с задаваемыми параметрами качества обслуживания. Более подробную информацию на эту тему можно найти в главе 10.

Поле *общая длина* (Total Length) определяет общую длину дейтаграммы в октетах (байтах), включая заголовок и полезную нагрузку. Максимальная длина дейтаграммы составляет 65535 октетов, однако, на практике, все рабочие станции и маршрутизаторы работают с длинами, не превышающими 576 байтов. Это объясняется тем, что при превышении указанной длины, снижается эффективность работы сети.

Протокол IP использует 3 поля заголовка для управления фрагментацией/ сборкой дейтаграмм. Как уже упоминалось, фрагментация необходима потому, что разные сети, по которым передаются дейтаграммы, имеют разные максимальные размеры кадра.

Идентификатор фрагмента (Identifier) обозначает все фрагменты одной дейтаграммы, что необходимо для ее успешной сборки на приемной стороне.

Поле *флагов* (Flags) обеспечивает возможность фрагментации дейтаграмм и, при использовании фрагментации, позволяет идентифицировать последний фрагмент дейтаграммы.

Поле *смещение фрагмента* (Fragment Offset) определяет положение фрагмента относительно исходной дейтаграммы в единицах, равных 8 октетам.

Поле *время жизни* (TTL – Time To Live) используется для ограничения времени, в течение которого дейтаграмма находится в сети. Каждый маршрутизатор сети должен уменьшать значение этого поля на единицу, и отбрасывать дейтаграмму, если поле TTL приняло нулевое значение. Наличие поля TTL ограничивает возможность бесконечной циркуляции дейтаграммы по сети, например, в случае, если по какой-либо причине маршрут, по которому она следует, оказался «закольцованным».

Поле *протокол* (Protocol) идентифицирует протокол верхнего уровня (TCP, UDP и т.д.).

Поле *контрольная сумма заголовка* (Header Checksum) обеспечивает возможность контроля ошибок в заголовке. Алгоритм подсчета контрольной суммы весьма прост, поскольку обычно протоколы нижнего уровня имеют более развитые средства контроля ошибок.

IP-дейтаграммы содержат в заголовке два адреса – отправителя (*Source*) и получателя (*Destination*), которые не меняются на протяжении всей жизни дейтаграммы.

Подробнее структура и функции протокола IPv4 описаны в RFC-791.

4.6 Протокол IP версии 6

В начале 90-х годов интенсивное коммерческое использование Интернет привело к резкому росту количества узлов сети, изменению характеристик трафика и ужесточению требований к качеству обслуживания. Сообщество Интернет и весь телекоммуникационный мир начали решать новые задачи путем внедрения новых протоколов в рамках стека протоколов TCP/IP, таких как протокол резервирования ресурсов RSVP, MPLS и т.д. Однако стало ясно, что только таким путем развивать технологию нельзя – нужно идти на модернизацию святого святого стека – протокола IP, так как некоторые проблемы нельзя решить без изменения формата заголовка дейтаграмм и логики его обработки.

Как уже отмечалось выше, самой насущной проблемой становится нехватка адресного пространства, что требует изменения формата адреса.

Другой проблемой является недостаточная масштабируемость процедуры маршрутизации – основы IP-сетей. Быстрый рост сети вызывает перегрузку маршрутизаторов, которые уже сегодня вынуждены поддерживать таблицы маршрутизации с десятками и сотнями тысяч записей, а также решать проблемы фрагментации пакетов. Облегчить работу маршрутизаторов можно, в частности, путем модернизации протокола IP.

Комитет IETF намеревается решить существующие проблемы с помощью межсетевого протокола нового поколения – IPng, известного также как IPv6.

Наряду с вводом новых функций непосредственно в протокол IP, целесообразно обеспечить более тесное взаимодействие его с новыми протоколами, путем введения в заголовок пакета новых полей. Например, работу механизмов обеспечения гарантированного качества обслуживания облегчает внесение в заголовок метки потока, а работу IPSec – внесение в заголовок поля аутентификации.

В результате было решено подвергнуть протокол IP модернизации, преследуя следующие основные цели:

- создание новой расширенной схемы адресации;
- улучшение масштабируемости сетей за счет сокращения функций магистральных маршрутизаторов;
- обеспечение защиты данных.

Работы по модернизации протокола IP начались в 1992 году, когда было предложено несколько альтернативных вариантов спецификаций. С тех пор в рамках IETF была проделана огромная работа, в результате которой в августе 1998 года были приняты окончательные версии стандартов, определяющих как общую архитектуру IPv6 (RFC 2460 «Internet Protocol, Version 6 (IPv6) Specification»), так и отдельные компоненты данной технологии (RFC 2373 «IP Version 6 Addressing Architecture»).

Итак, рассмотрим более подробно особенности IPv6.

Расширение адресного пространства. Протокол IP решает потенциальную проблему нехватки адресов за счет расширения адреса до 128 битов. Однако такое существенное увеличение длины адреса было сделано, в значительной степени, не с целью снять проблему дефицита адресов (для этого было бы достаточно гораздо более скромной размерности), а для повышения эффективности работы сетей на основе этого протокола. Главной целью было структурное изменение системы адресации, расширение ее функциональных возможностей.

Вместо существующих двух уровней иерархии адреса (номер сети и номер узла) в протоколе IPv6 предлагается использовать четыре уровня, что предполагает трехуровневую идентификацию сетей и один уровень для идентификации узлов. За счет увеличения числа уровней иерархии в структуре адреса, новый протокол эффективно поддерживает технологию агрегации адресов (CIDR), которая упоминалась выше. Благодаря этой особенности, а также усовершенствованной системе групповой адресации и введению нового типа адресов (anycast), IPv6 позволяет уменьшить затраты ресурсов оборудования на маршрутизацию.

В 6 версии протокола IP принята новая форма записи адреса, так как при определении адреса сети граница маски часто не совпадает с границей байтов адреса, и десятичная запись в данном случае неудобна. Теперь адрес записывается в шестнадцатеричном виде, причем каждые четыре цифры отделяются друг от друга двоеточием, например:

FEDC:0A96:0:0:0:7733:567A.

Для сетей, поддерживающих обе версии протокола – IPv4 и IPv6, – имеется возможность использовать для младших 4 байтов традиционную десятичную запись, а для старших – шестнадцатеричную:

0:0:0:0:0:FFFF.194.135.75.104.

Типы адресов. Для IPv6 определены следующие основные типы адресов:

- unicast;
- multicast;
- anycast.

Типы адресов определяются содержимым нескольких старших битов адреса, которые получили название *префикса формата*.

Адрес типа unicast представляет собой уникальный идентификатор сетевого интерфейса рабочей станции или маршрутизатора и по смыслу полностью идентичен уникальному адресу IPv4. Однако в версии 6 отсутствует понятие класса сети и фиксированное разбиение адреса на адрес сети и адрес узла по границам байтов.

Адрес типа multicast – групповой адрес, необходимый для многоадресной рассылки. Он характеризуется префиксом формата 11111111 и идентифицирует группу интерфейсов, относящихся к разным рабочим станциям. Пакеты с такими адресами доставляются ко всем интерфейсам, входящим в группу. Существует также предопределенный адрес, обозначающий все интерфейсы подсети. В составе группового адреса IPv6 имеется поле score, которое определяет, входят ли в группу рабочие станции одной подсети, всех подсетей предприятия, или рабочие станции, рассредоточенные по сети Интернет. Кроме того, предусмотрен признак, позволяющий определить, является ли группа постоянной или временной, что также облегчает работу маршрутизаторов.

Адрес типа anycast – новый тип адреса, определяющий, как и multicast, группу интерфейсов. Но пакет с таким адресом доставляется не всем членам группы, а какому-либо одному, как правило, «ближайшему» с точки зрения маршрутизатора. Такой адрес синтаксически никак не отличается от адреса типа unicast и выделяется из того же диапазона. Anycast–адрес может быть присвоен только сетевым интерфейсам маршрутизатора. Интерфейсам маршрутизатора бу-

дут присваиваться индивидуальные unicast-адреса и общий anycast-адрес. Адреса anycast ориентированы на определение маршрута узлом-отправителем. Например, у абонента есть возможность обеспечить прохождение своих пакетов через сеть конкретного поставщика, указав в цепочке адресов маршрута anycast-адрес, присвоенный всем маршрутизаторам в сети этого поставщика. В таком случае пакет будет передан на «ближайший» подходящий маршрутизатор именно этой сети.

В рамках системы адресации IPv6 имеется также выделенное пространство адресов для локального использования, т.е. для сетей, не входящих в Интернет. Существует две разновидности локальных адресов: для «плоских» сетей, не разделенных на подсети (Link-Local), и для сетей, разделенных на подсети (Site-Local), различающиеся значением префикса.

В настоящий момент распределено порядка 15% адресного пространства IPv6, что определяет широкие возможности развития сетей и приложений, их использующих.

Изменение формата заголовков пакетов. Многолетний опыт практического применения протокола показал неэффективность использования некоторых полей заголовка, а также выявил необходимость добавить поля, упрощающие идентификацию пакетов, которые требуют специальной обработки, поля, облегчающие реализацию процедур шифрования, и некоторые другие.

Реализовать это позволяет новая схема организации «вложенных заголовков», обеспечивающая разделение заголовка на основной, который содержит необходимый минимум информации, и дополнительные, которые могут отсутствовать. Такой подход открывает богатые возможности для расширения протокола путем определения новых опциональных заголовков, делая протокол открытым.

Основной заголовок дейтаграммы IPv6 длиной 40 байтов имеет следующий формат (рис. 4.3).

Версия (4 бита)	Класс Трафика (8 битов)	Метка Потока (20 битов)	
Длина (16 битов)		След. Заголовок (8 битов)	Лимит Переходов (8 битов)
Адрес Отправителя (128 битов)			
Адрес Получателя (128 битов)			

Рис. 4.3 Формат основного заголовка дейтаграммы IPv6

Поле *Класс Трафика* (Traffic Class) эквивалентно по назначению полю *Тип Обслуживания* (Type Of Service), а поле *Лимит Переходов*

(Hop Limit) – полю *Время Жизни* (Time To Live) протокола IPv4, рассмотренного в предыдущем параграфе.

Поле *Метка Потока* (Flow Label) позволяет выделять и особым образом обрабатывать отдельные потоки данных без необходимости анализировать содержимое пакетов. Это очень важно с точки зрения снижения нагрузки на маршрутизаторы.

Поле *Следующий Заголовок* (Next Header) является аналогом поля *Протокол* (Protocol) IPv4 и определяет тип заголовка, следующего за основным. Каждый следующий дополнительный заголовок также содержит поле Next Header. Если дополнительные заголовки отсутствуют, то это поле содержит значение, присвоенное тому из протоколов TCP, UDP, OSPF, который используется для переноса полезной нагрузки данной дейтаграммы.

В рамках спецификаций IPv6 определены заголовки следующих типов.

Заголовок Routing – содержит информацию о маршруте, выбранном отправителем дейтаграммы.

Заголовок Fragmentation – содержит информацию о фрагментации дейтаграммы и обрабатывается только конечными узлами сети.

Заголовок Authentication – содержит информацию, необходимую для проверки подлинности отправителя дейтаграммы.

Заголовок Encapsulation – содержит информацию, необходимую для обеспечения конфиденциальности данных путем шифрования.

Заголовок Hop-by-Hop Options – специальные параметры обработки пакетов.

Заголовок Destination Options – дополнительные параметры для узла назначения.

Снижение нагрузки на маршрутизаторы. При переходе к протоколу IPv6 могут быть уменьшены расходы на реализацию функций маршрутизации в сети, а маршрутизаторы могут быть оптимизированы для выполнения их основной функции – продвижения пакетов. Это становится возможным благодаря следующим особенностям нового протокола.

Дополнительные заголовки обрабатываются только конечными узлами и краевыми маршрутизаторам. Это упрощает логику работы маршрутизаторов и позволяет легче реализовать важные функции на аппаратном уровне.

Функции поддержки фрагментации переносятся в конечные узлы или краевые маршрутизаторы. Конечные узлы должны найти минимальный размер пакета вдоль всего пути до узла назначения (эта технология называется Path MTU discovery и уже используется для

протокола IPv4) и не передавать пакеты с размером, превышающим найденное значение. Маршрутизаторы, поддерживающие протокол IPv6, в ядре сети могут не обеспечивать фрагментации, а только передавать сообщение протокола ICMP – «слишком длинный пакет» к конечному узлу, который должен соответственно уменьшить размер пакета.

Агрегация адресов ведет к уменьшению размеров адресных таблиц маршрутизаторов и, соответственно, к уменьшению времени их просмотра.

Широкое использование маршрутизации, управляемой отправителем (например, пограничным маршрутизатором), освобождает маршрутизаторы в ядре сети от просмотра адресных таблиц при выборе следующего маршрутизатора.

В качестве адреса узла в локальной сети можно использовать MAC-адрес сетевого интерфейса, что избавляет от необходимости применять протокол ARP.

Переход к протоколу IP версии 6. Так как IPv6 представляет собой естественное развитие предыдущей версии, он с самого начала спроектирован с учетом возможности поэтапного мягкого перехода к его использованию, что требует обеспечения взаимодействия узлов с разными версиями протоколов. Способы, которые используются для организации совместной работы протоколов IPv6 и IPv4, вполне традиционны:

- Установка на некоторых сетевых узлах сразу двух стеков протоколов, так что при взаимодействии с рабочими станциями, поддерживающими разные версии протокола, используется соответствующий стек протоколов TCP/IP. Маршрутизаторы могут в данном случае обрабатывать оба протокола независимо друг от друга.
- Конвертирование протоколов при помощи специальных шлюзов, которые преобразуют пакеты IPv4 в пакеты IPv6 и обратно. Важнейшая часть этого процесса – преобразование адресов. Для упрощения данной процедуры применяются так называемые «IPv4-совместимые адреса IPv6», которые содержат в четырех младших байтах адрес, используемый в протоколе IPv4.
- Инкапсуляция – туннелирование одного протокола в сетях, построенных на основе другого протокола. При этом пакеты одного протокола помещаются в пакеты другого в пограничных устройствах. Недостаток метода состоит в том, что в данном случае сети никак не взаимодействуют между собой. В настоящее время развита опытная зона эксплуатации IPv6 под названием 6Bone, которая использует технологию инкапсуляции пакетов IPv6 при их транзите через части сети Интернет, не поддерживающие этот протокол.

4.7 Протокол TCP

Протокол управления передачей информации – Transmission Control Protocol (TCP) – был разработан для поддержки интерактивной связи между компьютерами. Протокол TCP обеспечивает надежность и достоверность обмена данными между процессами на компьютерах, входящих в общую сеть.

К сожалению, протокол TCP не приспособлен для передачи мультимедийной информации. Основная причина – обеспечение требуемой достоверности путем повторной передачи потерянных пакетов. Пока передатчик получит информацию о том, что приемник не принял очередной пакет, и передаст его снова, проходит слишком много времени. Приемник вынужден либо ждать прихода повторно переданного пакета, разрушая структуру потоковых данных, либо игнорировать этот пакет, игнорируя одновременно принятый в TCP механизм обеспечения достоверности. Кроме того, TCP предусматривает механизмы управления скоростью передачи с целью избежать перегрузок сети. Аудиоданные и видеоданные требуют, однако, строго определенных скоростей передачи, которые нельзя изменять произвольным образом.

С одной стороны протокол TCP взаимодействует с прикладным протоколом пользовательского приложения, а с другой стороны – с протоколом, обеспечивающим «низкоуровневые» функции маршрутизации и адресации пакетов, которые, как правило, выполняет IP.

В модели межсетевого соединения взаимодействие TCP и протоколов нижнего уровня, вообще говоря, не специфицировано, за исключением того, что должен существовать механизм, который обеспечивал бы асинхронную передачу информации от одного уровня к другому. Результатом работы этого механизма является инкапсуляция протокола более высокого уровня в тело протокола более низкого уровня. Каждый TCP-пакет вкладывается в «пакет» протокола нижележащего уровня, например, IP. Получившаяся таким образом дейтаграмма содержит в себе TCP-пакет так же, как TCP-пакет содержит пользовательские данные.

Простейшая модель работы TCP-протокола выглядит обманчиво гладко, поскольку на самом деле его работа изобилует множеством деталей и тонкостей.

Логическая структура сетевого программного обеспечения, реализующего протоколы семейства TCP/IP в каждом узле сети Internet, изображена на рис. 4.4.

Прямоугольники обозначают модули, обрабатывающие данные, а линии, соединяющие прямоугольники, – пути передачи данных. Горизонтальная линия внизу рисунка обозначает сеть Ethernet, кото-

рая используется в качестве примера физической среды. Понимание этой логической структуры является основой для понимания всей технологии TCP/IP.

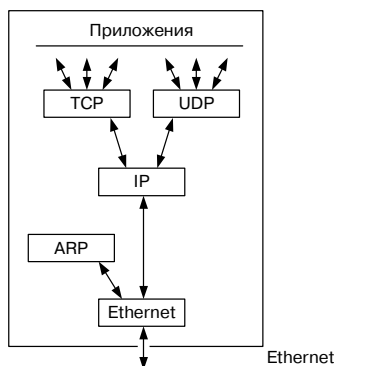


Рис. 4.4 Структура сетевого программного обеспечения стека протоколов TCP/IP

Ниже рассматриваются более подробно возможности, принципы построения и основные функции протокола TCP.

4.7.1 Потоки, стек протоколов, механизм портов и мультиплексирование

Чтобы установить соединение между двумя процессами на разных компьютерах сети, необходимо знать не только Internet-адреса компьютеров, но и номера тех TCP-портов (sockets), которые процессы используют на этих компьютерах. Любое TCP-соединение в сети Internet однозначно идентифицируется двумя IP-адресами и двумя номерами TCP-портов.

Рассмотрим потоки данных, перенос которых обеспечивают протоколы. При использовании протокола TCP данные передаются между прикладным процессом и модулем TCP. Типичным прикладным протоколом, использующим протокол TCP, является FTP (File Transfer Protocol, Протокол переноса файлов). Стек протоколов в этом случае выглядит следующим образом: FTP/TCP/IP/Ethernet. При использовании протокола UDP (User Datagram Protocol, Протокол дейтаграмм пользователя) данные передаются между прикладным процессом и модулем UDP. Транспортными услугами протокола UDP пользуется, например, SNMP (Simple Network Management Protocol, Простой протокол эксплуатационного управления сетью). Его стек протоколов выглядит так: SNMP/UDP/IP/Ethernet.

Один порт компьютера может быть задействован в соединениях с несколькими портами удаленных компьютеров. Таким образом, механизм портов позволяет работать на одном компьютере одно-

временно нескольким приложениям и однозначно идентифицировать каждый поток данных в сети. Это называется мультиплексированием соединений.

Модули TCP, UDP и драйвер Ethernet являются мультиплексорами типа $n \times 1$. Действуя как мультиплексоры, они переключают несколько входов на один выход. Они также являются демультиплексорами типа $1 \times n$. Как демультиплексоры, они переключают один вход на один из многих выходов в соответствии с определенным полем в заголовке протокольного блока данных (в Ethernet-кадре это поле «тип»). Когда Ethernet-кадр попадает в драйвер сетевого интерфейса Ethernet, он может быть направлен либо в модуль ARP, либо в модуль IP. (Значение поля «тип» в заголовке кадра указывает, куда должен быть направлен Ethernet-кадр.)

Если IP-пакет попадает в модуль IP, то содержащиеся в нем данные могут быть переданы либо модулю TCP, либо UDP, что определяется полем «Protocol» в заголовке IP-пакета. Если TCP-сообщение попадает в модуль TCP, то выбор прикладной программы, которой должно быть передано сообщение, производится на основе значения поля «порт» в заголовке TCP-сообщения.

Демультиплексирование данных, передаваемых в обратном направлении, осуществляется довольно просто, так как из каждого модуля существует только один путь «вниз». Каждый протокольный модуль добавляет к пакету свой заголовок, на основании которого машина, принявшая пакет, выполняет демультиплексирование.

Назначение портов для приложений на каждом компьютере производится независимо. TCP может самостоятельно выбирать порт, с которым будет работать приложение, или приложение укажет, с каким портом на данном компьютере оно будет работать. Однако, как правило, часто используемые приложения-сервисы, например, такие как HTTP, FTP, SMTP и др., используют одни и те же номера портов, которые уже стали общеизвестными. Это делается для того, чтобы к данному процессу на компьютере можно было присоединиться, указывая только адрес машины. Например, Internet-браузер, если ему не указать дополнительно, ищет по указанному адресу приложение, работающее с портом 80 (наиболее распространенный порт для серверов WWW). Кроме того, рабочая станция может быть снабжена несколькими сетевыми интерфейсами, тогда она должна осуществлять мультиплексирование типа $n \times m$, т. е. между несколькими прикладными программами и несколькими интерфейсами.

4.7.2 Установление TCP-соединения и передача данных

Режим участия в установлении TCP-соединения может быть активным и пассивным. При пассивном участии рабочая станция ожидает сигнал открытия TCP-канала от встречного оборудования и не пытается открыть TCP-канал сама. Этот режим обычно используется

процессами, которые предоставляют свой сервис через общеизвестный номер своего порта (например, HTTP, SMTP и т. д.). При активном режиме участия рабочая станция сама инициирует открытие TCP-канала. Соединение будет также установлено, если два процесса активно откроют канал навстречу друг другу. Такая гибкость в установлении соединения особенно важна в распределенных сетях, когда компьютеры работают асинхронно.

Процедура установления TCP-соединения выглядит следующим образом. Рабочая станция, инициирующая открытие TCP-канала, передает пакет с флагом SYN, в котором указывается номер порта и начальный порядковый номер пакетов данных. Встречная станция передает на указанный адрес ответ с флагами SYN и ACK, в котором указывается начальный порядковый номер пакетов данных. Сторона, инициирующая установление TCP-соединения, подтверждает получение пакета с флагами SYN и ACK передачей пакета с установленным флагом ACK.

Именно трех тактов квитирования сообщений всегда бывает достаточно, чтобы синхронизировать потоки данных. Соединение считается установленным, когда последовательности передаваемых пакетов в обоих направлениях синхронизируются, т.е. когда и клиент, и сервер «знают», пакет с каким номером поступит с противоположной стороны соединения.

Соединение закрывается, когда порты оборудования обмениваются пакетами, содержащими флаги FIN. При этом все ресурсы системы должны быть освобождены.

4.7.3 Механизмы обеспечения достоверности

Протокол TCP умеет работать с поврежденными, потерянными, дублированными или поступившими с нарушением порядка следования пакетами. Это достигается благодаря механизму присвоения каждому передаваемому пакету порядкового номера и механизму проверки получения пакетов.

Когда протокол TCP передает сегмент данных, копия этих данных помещается в очередь повтора передачи, и запускается таймер ожидания подтверждения. Когда система получает подтверждение (сегмент TCP, содержащий управляющий флаг ACK), что этот сегмент данных получен, она удаляет его из очереди. Сегмент подтверждения получения содержит номер полученного сегмента, на основании которого и происходит контроль доставки данных адресату. Если подтверждение не поступило до срабатывания таймера, сегмент отправляется еще раз. Уведомление о получении сегмента данных еще не означает, что он был доставлен конечному пользователю. Оно только означает, что модуль TCP выполнил возложенные на него функции.

При передаче информации каждому байту данных присваивается порядковый номер, поэтому, в какой бы последовательности эти байты ни достигали точки назначения, они всегда будут собраны в изначальной последовательности. Порядковый номер первого байта данных в передаваемом сегменте называется порядковым номером сегмента. Нумерация проводится «с головы состава», т. е. от заголовка пакета. TCP-пакет содержит также «подтверждающий номер» (acknowledgment number), который представляет собой номер следующего ожидаемого пакета. Иными словами, подтверждающий номер означает: «до сих пор я все получил». Механизм с использованием «подтверждающего номера» исключает дублирование пакетов при повторной отправке не доставленных данных.

Кроме определения порядка следования информационных пакетов, «порядковый номер» играет важную роль в механизме синхронизации соединения и в контроле потерянных пакетов при разрывах соединения.

Стоит сказать несколько слов о механизме, предотвращающем появление в сети пакетов с одинаковыми номерами. Они могут появиться, например, при установлении и быстром сбросе соединения или при сбросе соединения и его быстром восстановлении, т.е. когда номер испорченного пакета может быть сразу использован новым пакетом. Механизм предотвращения подобных ситуаций основан на генерировании начального числа последовательности пакетов, а поскольку счетчик циклический, то все равно, с какого места начинать отсчет.

Так, при установлении нового соединения генерируется 32-битовое число ISN (Initial Sequence Number). Генератор может использовать 32 младших разряда машинного таймера, который меняется каждые 4 микросекунды (полный цикл – 4,55 часа). Это число и служит отсчетом нумератора пакетов. Кроме того, каждая дейтаграмма в сети имеет ограниченное время жизни MSL – Maximum Life Time, которое значительно меньше периода генератора. Таким образом, в сети гарантируется невозможность появления пакетов с одинаковыми номерами.

Поврежденные пакеты отсеиваются механизмом проверки контрольной суммы, которая помещается в каждом передаваемом пакете.

4.7.4 Механизм управления потоком данных

Протокол TCP предоставляет получателю пакетов возможность регулировать передаваемый к нему отправителем поток данных. Этот механизм основан на том, что при передаче флага подтверждения получения пакета (ACK) в TCP-сегменте передается указатель объема данных (так называемое «окно» TCP-соединения), которые могут

быть переданы отправителем, не дожидаясь от получателя разрешения отправить следующую порцию данных. Иными словами, указывается размер свободного места в буферном накопителе, куда записываются только что принятые данные, ожидающие дальнейшей обработки и передачи соответствующим процессам. Этот механизм позволяет избежать «пробок» при обмене данными между системами, обладающими разной производительностью.

«Окно» задается количеством байтов, отсчитываемых от последнего подтвержденного байта (acknowledgment number). Нулевой размер окна означает, что отправитель должен приостановить передачу до тех пор, пока он не будет уведомлен о готовности получателя к приему данных. Необходимо заметить, что в этом случае отправитель передает однобайтовые пакеты.

Безусловно, большой размер окна позволяет передавать данные быстрее, поскольку отправителю пакета не нужно ждать от получателя сигнал о его готовности к приему. Однако в случае сбоя передачи, соответственно, возрастет объем данных, которые нужно отправить заново. При небольшом же размере окна потерянные сегменты данных можно восстановить с минимальными затратами.

Механизм управления потоком данных позволяет протоколу TCP оптимизировать скорость достоверного обмена данными между процессами в сети Интернет.

4.7.5 Состав и назначение полей заголовка

Пакеты протокола TCP переносятся в поле «Данные» IP-дейтаграммы. Заголовок пакета TCP следует за заголовком дейтаграммы. Структура заголовка пакета TCP представлена на рис. 4.5.

Порт отправителя				Порт получателя				
Порядковый номер								
Номер при подтверждении								
Смещение данных	Резерв	U	A	P	R	S	F	Окно
		R	C	S	S	Y	I	
		G	K	H	T	N	N	
Контрольная сумма				Указатель срочности				
Опции						Заполнение		
Данные								

Рис. 4.5 Заголовок пакета TCP

Порт отправителя (Source Port, 16 битов).

Порт получателя (Destination Port, 16 битов).

Порядковый номер (Sequence Number, 32 бита). Если в пакете отсутствует флаг SYN, то это – номер первого октета данных в этом пакете. Если флаг SYN в пакете присутствует, то номер данного пакета становится номером начала последовательности (ISN), и номером первого октета данных становится номер ISN+1.

Номер при подтверждении (Acknowledgment Number, 32 бита) – если пакет содержит установленный флаг ACK, то это поле содержит номер следующего ожидаемого получателем октета данных. При установленном соединении пакет подтверждения отправляется всегда.

Поле величины смещения данных (Data Offset, 4 бита) указывает количество 32-битовых слов заголовка TCP-пакета.

Резерв (Reserved, 6 битов) – зарезервированное поле.

Флаги управления (слева направо):

URG – флаг срочности,

ACK – флаг пакета, содержащего подтверждение получения,

PSH – флаг форсированной отправки,

RST – сброс соединения,

SYN – синхронизация порядковых номеров,

FIN – флаг окончания передачи со стороны отправителя.

Окно (Window, 16 битов) – поле содержит количество байтов данных, которое отправитель данного сегмента может принять, считая от байта с номером, указанным в поле Номер при подтверждении.

Поле контрольной суммы (Checksum, 16 битов).

Поле указателя срочности данных (Urgent Pointer, 16 битов). Это поле содержит номер пакета, начиная с которого следуют пакеты повышенной срочности. Указатель принимается во внимание только в сегментах с установленным флагом URG.

Опции (Options) – поле дополнительных параметров, может быть переменной длины.

Заполнение (Padding) – поле, заполняемое нулями для выравнивания заголовка до размера, кратного 32-битам.

Более подробное описание протокола TCP можно найти в RFC-793, RFC-1180.

4.8 Протокол UDP

Протокол передачи пользовательских дейтаграмм – User Datagram Protocol (UDP) значительно проще рассмотренного в предыдущем параграфе протокола TCP и предназначается для обмена дейтаграммами между процессами компьютеров, расположенных в объединенной системе компьютерных сетей.

Протокол UDP базируется на протоколе IP и предоставляет прикладным процессам транспортные услуги, немногим отличающиеся от услуг протокола IP. Протокол UDP обеспечивает негарантирован-

ную доставку данных, т.е. не требует подтверждения их получения; кроме того, данный протокол не требует установления соединения между источником и приемником информации, т. е. между модулями UDP.

К заголовку IP-пакета протокол UDP добавляет служебную информацию в виде заголовка UDP-пакета (рис. 4.6).

Порт отправителя	Порт получателя
Длина	Контрольная сумма
Данные	
....	

Рис. 4.6 Формат UDP-пакета

Порт отправителя (Source Port) – поле указывает порт рабочей станции, передавшей дейтаграмму. На этот порт следует адресовать ответную дейтаграмму. Если данное поле не используется, оно заполняется нулями.

Порт получателя (Destination Port) – поле идентифицирует порт рабочей станции, на которую будет доставлен пакет.

Длина (Length) – это поле информирует о длине UDP-пакета в октетах, включая как заголовок, так и данные. Минимальное значение длины равно восьми.

Контрольная сумма (Checksum) – поле проверки правильности передачи данных заголовка пакета, псевдозаголовка и поля полезной нагрузки пакета. Если данное поле не используется, оно заполняется нулями.

Модуль IP, реализованный в принимающей рабочей станции, передает поступающий из сети IP-пакет модулю UDP, если в заголовке этого пакета указано, что протоколом верхнего уровня является протокол UDP. При получении пакета от модуля IP модуль UDP проверяет контрольную сумму, содержащуюся в его заголовке. Если контрольная сумма равна нулю, значит, отправитель ее не подсчитал. Протоколы UDP и TCP имеют один и тот же алгоритм вычисления контрольной суммы (RFC-1071), но механизм ее вычисления для UDP-пакета имеет некоторые особенности. В частности, UDP-дейтаграмма может содержать нечетное число байтов, и в этом случае к ней, для унификации алгоритма, добавляется нулевой байт, который никуда не пересылается.

Более подробную информацию о протоколе UDP можно найти в RFC-768.

4.9 Требования к современным IP-сетям

В главе 3 были рассмотрены принципы кодирования речевой информации, используемые для передачи ее по сетям с маршрутиза-

цией пакетов IP. Закодированные при помощи таких алгоритмов данные генерируются с заданной (не обязательно фиксированной) скоростью независимо от загрузки сети. Требуется, чтобы информация была доставлена получателю с точно той же скоростью, с какой ее генерировал отправитель. Аналогичное требование предъявляется и к доставке видеоинформации.

Синхронная передача данных предполагает периодическую генерацию битов, байтов, или пакетов, которые должны быть воспроизведены приемником с точно таким же периодом. В данном случае скорость передачи информации постоянна. ТфОП функционирует на основе синхронной передачи данных, примером может служить цифровой поток E1.

Передача такого рода информации (аудио, видео, синхронные потоки) сама по себе не требует очень малой задержки между источником и приемником, хотя это часто является предпочтительным. Однако принципиально необходимо, чтобы задержка была предсказуема, так как только в этом случае временные параметры переданных сообщений могут быть восстановлены в приемнике.

Требования к скорости передачи информации для разных услуг варьируются очень широко. Например, передача данных телеметрии в реальном времени может требовать скорости несколько бит/с, для речевой информации удовлетворительного качества потребуется от 4 до 32 Кбит/с, для обеспечения качества на уровне ТфОП необходимо до 64 Кбит/с, передача видео требует от десятков Кбит/с до десятков Мбит/с (HDTV), в зависимости от характеристик системы (размер изображения, частота кадров, способ кодирования и т.д.). Требования ко времени доставки тоже могут быть различны. Например, при организации речевой связи допускается сквозная задержка от 12 мс при отсутствии эхокомпенсации (G.164), и до 400 мс при ее наличии. При этом, как отмечалось в главе 3, при стремлении величины задержки к верхнему пределу субъективная оценка качества связи падает, взаимодействие становится полудуплексным. Для не интерактивных приложений (например, предоставление видеоинформации по запросу) могут допускаться задержки более 500 мс, которые ограничиваются только возможностью пользователя нормально управлять процессом воспроизведения и возможностями буферизации данных в приемнике.

Процесс передачи данных по сетям с коммутацией пакетов подвержен влиянию статистически изменяющейся задержки (джиттера), возникающей при обработке очередей в узлах сети. Джиттер компенсируется приемником путем использования буфера воспроизведения. Приемник должен обладать информацией о статистических характеристиках задержки, чтобы предусмотреть необходимое место в буферном накопителе. Например, если допустимы потери 0,1%

пакетов, величина буфера должна поддерживаться на уровне, превышающем переменную составляющую задержки поступающих пакетов в 99,9% случаев. Таким образом, высокий уровень джиттера заставляет мириться либо с большим количеством мест в буферном накопителе и, как следствие, с большими задержками, либо с высоким уровнем потерь информации.

Сеть Интернет была создана для передачи данных на основе адаптивной маршрутизации, предполагающей, что данные могут следовать по разным маршрутам, выбираемым в зависимости от некоторых условий. Кроме того, в сети Интернет не предусматривалось установление соединения между источником и приемником информации, т.е. между компьютерами в сети не устанавливается никаких связей, информация о которых сохранялась бы в сети. Это приводит к тому, что пакеты часто приходят к получателю не в той последовательности, в какой они были переданы.

Интернет – сеть с доставкой по мере возможности. Это значит, что сеть пытается доставить информацию, но если это по каким-либо причинам не получается, то информация будет потеряна. Потери пакетов в Интернет, к сожалению, носят «пачечный» характер, то есть внутри некоторых интервалов времени теряется сразу много пакетов подряд или пакетов, следующих с небольшими промежутками. Эта характеристика сети Интернет затрудняет организацию передачи мультимедийной информации, поскольку такие приложения нормально работают только в условиях случайных независимых потерь.

Интернет – сеть, которая сегодня поддерживает только одноадресную доставку. Многоадресная доставка информации, очень полезная для многих приложений (организация конференций, трансляция телепрограмм и т.д.), поддерживается только в экспериментальном режиме на некоторых участках.

Кроме того, сегодня Интернет предоставляет любым приложениям и любым пользователям одинаковый (и притом, как уже говорилось, не гарантированный и не специфицированный) уровень качества обслуживания. Это не позволяет сравнивать качество услуг IP-телефонии с качеством услуг ТфОП, так как в ТфОП существуют и действуют очень жесткие спецификации качества обслуживания вызовов. Для решения названной проблемы необходимо обеспечить возможность резервирования ресурсов сети в процессе установления соединений.

Скажем несколько слов о надежности. Стремление обеспечить в рамках IP-сетей предоставление услуг телефонии, аналогичных услугам ТфОП, сталкивается с проблемой физической надежности сети. Пользователи ТфОП привыкли, что услуги доступны 24 часа в сутки 7 дней в неделю, т.е. всегда. Эта привычка вполне обоснована, так как АТС и другое оборудование, составляющее основу ТфОП, разрабо-

тано с учетом коэффициента готовности 99.999%, что эквивалентно 3 часам простоя за 40 лет (!) эксплуатации. Так исторически сложилось, что в мире сетей передачи данных действуют совершенно другие стандарты. Большинство людей не смогут ответить, когда последний раз не работал их телефон, однако они без труда припомнят, когда отказала локальная сеть, или не было доступа к Интернет. Создание универсальной сетевой инфраструктуры на базе протокола IP требует пересмотреть требования к надежности IP-сетей.

Подводя итог, отметим, что Интернет в современном состоянии, со всеми отмеченными выше свойствами, вполне удовлетворяет требованиям наиболее популярных приложений (WWW, электронная почта, передача файлов и т.д.), однако, как мы увидим ниже, для поддержки новых услуг, в том числе аналогичных по сути и качеству услугам ТфОП, необходима глубокая поэтапная модернизация сети с внедрением новых протоколов и алгоритмов обслуживания трафика.

4.10 Протоколы RTP и RTCP

Приложения, обеспечивающие передачу речевой и видеоинформации, используют сервис транспортного уровня без установления соединений (например, UDP). При этом каждое приложение может обеспечивать формирование полезной нагрузки пакетов специфическим образом, включая необходимые для функционирования поля и данные. Однако, как показал приведенный в предыдущем параграфе анализ, данные разной природы (речь, видео) имеют общие особенности, которые требуют обеспечения вполне определенной функциональности при их передаче по сети. Это позволяет сформировать некий общий транспортный уровень, объединяющий функции, общие для потоковых данных разной природы, и используемый всеми соответствующими приложениями, придав протоколу этого уровня статус стандарта. Комитетом IETF был разработан протокол транспортировки информации в реальном времени – Realtime Transport Protocol (RTP), который стал базисом практически для всех приложений, связанных с интерактивной передачей речевой и видеоинформации по сети с маршрутизацией пакетов.

Характерные для IP-сетей временные задержки и вариация задержки пакетов (джиттер) могут серьезно исказить информацию, чувствительную к задержке, например, речь и видеоинформацию, сделав ее абсолютно непригодной для восприятия. Отметим, что вариация задержки пакетов гораздо сильнее влияет на субъективную оценку качества передачи, чем абсолютное значение задержки.

Уже длительное время ведется работа по созданию методов уменьшения джиттера и задержек. Для этого могут применяться рассмотренные в главе 10 механизмы, обеспечивающие пользователю

заданный уровень качества обслуживания. Они, конечно, улучшают качество услуг, предоставляемых сетью, но не могут совсем устранить образование очередей в сетевых устройствах и совсем убрать джиттер.

Именно протокол RTP позволяет компенсировать негативное влияние джиттера на качество речевой и видеоинформации. В то же время, он не имеет собственных механизмов, гарантирующих своевременную доставку пакетов или другие параметры качества услуг, – это осуществляют нижележащие протоколы. Он даже не обеспечивает все те функции, которые обычно предоставляют транспортные протоколы, в частности функции исправления ошибок и управления потоком. Обычно протокол RTP базируется на протоколе UDP и использует его функции, но может работать и поверх других транспортных протоколов.

Существует несколько серьезных причин, по которым такой пространственный транспортный протокол, как TCP, плохо подходит для передачи чувствительной к задержкам информации. Во-первых, это алгоритм надежной доставки пакетов. Пока отправитель повторно передаст пропавший пакет, получатель будет ждать, результатом чего может быть недопустимое увеличение задержки. Во-вторых, алгоритм управления при перегрузке в протоколе TCP далеко не оптимален для передачи речи и видеоинформации. При обнаружении потерь пакетов протокол TCP уменьшает размер окна, а затем будет его медленно увеличивать. Однако передача речевой и видеоинформации осуществляется на вполне определенных, фиксированных скоростях, которые нельзя мгновенно уменьшить, не ухудшив качество предоставляемых услуг. Правильной реакцией на перегрузку для информационных потоков этих типов было бы изменение метода кодирования, частоты видеокадров или размера видеоизображения.

Протокол RTP предусматривает индикацию типа полезной нагрузки и порядкового номера пакета в потоке, а также применение временных меток. Отправитель помечает каждый RTP-пакет временной меткой, получатель извлекает ее и вычисляет суммарную задержку. Разница в задержке разных пакетов позволяет определить джиттер и смягчить его влияние – все пакеты будут выдаваться приложению с одинаковой задержкой.

Итак, главная особенность RTP – это вычисление средней задержки некоторого набора принятых пакетов и выдача их пользователю приложению с постоянной задержкой, равной этому среднему значению. Однако следует иметь в виду, что временная метка RTP соответствует моменту кодирования первого дискретного сигнала пакета. Поэтому, если RTP-пакет, например, с видеоинформацией, разбивается на блоки данных нижележащего уровня, то временная метка уже не будет соответствовать истинному

времени их передачи, поскольку они перед передачей могут быть установлены в очередь.

На рис. 4.7 представлен основной заголовок RTP-пакета, содержащий ряд полей, которые идентифицируют такие элементы, как формат пакета, порядковый номер, источник информации, границы и тип полезной нагрузки.



Рис. 4.7 Основной заголовок RTP-пакета

V (2 бита) – поле версии протокола. Текущая версия протокола – вторая.

P (1 бит) – поле заполнения. Сигнализирует о наличии заполнения в конце поля полезной нагрузки. Заполнение применяется, когда приложение требует, чтобы размер полезной нагрузки был кратен, например, 32 битам.

X (1 бит) – поле расширения заголовка. Служит для индикации того, что за основным заголовком следует дополнительный заголовок, используемый в экспериментальных расширениях протокола RTP.

CC (4 бита) – поле отправителей. Содержит идентификаторы отправителей, чьи данные находятся в пакете, причем сами идентификаторы следуют за основным заголовком.

M (1 бит) – поле маркера. Обычно используется для указания границ потока данных. Смысл бита маркера зависит от типа полезной нагрузки. В случае передачи видеоинформации он определяет конец кадра. При передаче речевой информации маркер указывает начало периода активности после периода молчания.

PT (7 битов) – поле типа полезной нагрузки. Идентифицирует тип полезной нагрузки и формат данных, включая сжатие и шифрование. В стационарном состоянии отправитель использует только один тип полезной нагрузки в течение сеанса, но он может его изменить в ответ на изменение условий, если об этом сигнализирует протокол управления транспортировкой информации в реальном времени (Real-Time Transport Control Protocol).

Порядковый номер пакета (Sequence Number, 16 битов). Каждый источник начинает нумеровать пакеты с произвольного номера, увеличиваемого затем на единицу с каждым переданным пакетом RTP.

Это позволяет обнаруживать потери пакетов и определять порядок пакетов с одинаковым временным штампом. Несколько последовательных пакетов могут иметь один и тот же штамп, если логически они порождены в один и тот же момент, как, например, пакеты, принадлежащие одному и тому же видеокадру.

Временной штамп (Timestamp, 32 бита). Момент времени, в который был создан первый октет данных полезной нагрузки. Единицы, в которых время указывается в этом поле, зависят от типа полезной нагрузки. Значение определяется по локальным часам отправителя.

Идентификатор SSRC (Synchronization Source Identifier, 32 бита) – поле идентификатора источника синхронизации. Псевдослучайное число, которое уникальным образом идентифицирует источник в течение сеанса и не зависит от сетевого адреса. Это число играет важную роль при обработке порции данных, поступившей от одного источника.

Идентификатор CSRC (Contributing Source Identifier, 32 бита) – список полей идентификаторов источников, участвующих в создании RTP-пакета. Устройство смешивания информации (миксер) вставляет целый список SSRC идентификаторов источников, которые участвовали в построении данного RTP-пакета. Количество элементов в списке: от 0 до 15. Если число участников более 15, выбираются первые 15. Примером может служить речевая конференция, в которой передаются RTP-пакеты с речью всех участников – каждый со своим идентификатором SSRC. Они-то и образуют список идентификаторов CSRC. Вся конференция имеет общий идентификатор SSRC.

Доставка RTP-пакетов контролируется специальным протоколом RTCP (Real Time Control Protocol).

Основной функцией протокола RTCP является организация обратной связи приемника с отправителем информации для отчета о качестве получаемых данных. Протокол RTCP передает сведения (как от приемника, так и от отправителя) о числе переданных и потерянных пакетов, значении джиттера, задержке и т.д. Эта информация может быть использована отправителем для изменения параметров передачи, например для уменьшения коэффициента сжатия информации с целью улучшения качества ее передачи. Более подробное описание протоколов RTP и RTCP можно найти в RFC-1889.

4.11 Многоадресная рассылка

Основной целью группового вещания является создание эффективного механизма передачи данных по схеме «один-ко-многим» и «многие-ко-многим».

Традиционные механизмы доставки пакетов стека TCP/IP мало пригодны для поддержки группового вещания. Например, использование уникальных адресов (unicast) приводит к необходимости установления многочисленных двухточечных соединений между отправителем и каждым из получателей.

Другим способом передачи данных является широковещательная передача, когда станция направляет пакеты, используя широковещательные адреса (broadcast). Пакеты с такими адресами передаются ко всем конечным узлам указанной сети, независимо от того, нужны ли они каждому из них. Во многих ситуациях такой способ передачи также оказывается неэффективным вследствие своей избыточности, которая ведет к чрезмерному росту трафика, особенно в крупных сетях.

В случае использования групповых адресов отправитель передает сообщение только один раз, затем оно тиражируется и доставляется только к тем узлам, которые являются членами соответствующей группы. Такой режим экономит пропускную способность за счет передачи только того трафика, который необходим. Номера группы задаются с использованием IP-адреса типа multicast.

Основными протоколами, на базе которых реализуется многоадресная рассылка в IP-сетях, являются протоколы IGMP (Internet Group Management Protocol), DVMRP – (Distance Vector Multicast Routing Protocol), PIM (Protocol Independent Multicast).

Глава 5

Архитектура

H.323

5.1 Стандарты мультимедийной связи

Работа над рекомендациями ITU-T серии H, уже упоминавшимися в первых четырех главах, началась отнюдь не для IP-телефонии. Более 10 лет тому назад Международный союз электросвязи начал разработку рекомендаций для будораживших умы связистов того поколения систем видеотелефонной и мультимедийной связи. Термин «мультимедийная связь» обозначает связь двух или более пользователей, обменивающихся одновременно речью, видеоинформацией и данными.

Первая рекомендация из этой серии, H.320, была выпущена в 1990 году и относилась к системам видеотелефонии, ориентированным на работу в узкополосной ISDN. Следующие рекомендации ITU-T разрабатывались для систем мультимедийной связи, работающих в разном сетевом окружении.

В рекомендациях серии H описываются архитектура и функциональные элементы систем, алгоритмы кодирования речи и видеоинформации, организация передачи данных, протоколы сигнализации и управления информационными каналами (таблица 5.1). За истекшее время ITU-T выпустил для таких систем несколько новых версий рекомендаций, в которых учитывались пожелания и замечания ведущих фирм-производителей оборудования, разрабатываемого на базе этих рекомендаций.

Основу рекомендаций Международного союза электросвязи составляет единая базовая архитектура систем мультимедийной связи, функционирующих в разном сетевом окружении. Все рекомен-

дации предусматривают использование для поддержки передачи речи, видеоинформации и данных, а также для управления системой, идентичных функциональных элементов. Различие заключается в упаковке пользовательской и сигнальной информации – выбирается оптимальный способ упаковки информации для той сети, в которой система будет функционировать.

Таблица 5.1. Рекомендации ITU-T по видеотелефонии и мультимедийной связи

Рекомендации ITU-T серии H	H.320	H.321	H.322	H.323 V1/V2/V3	H.324
Дата утверждения	1999	1998	1996	1996/1998/ 1999	1998
Сетевое окружение	ТфОП, N-ISDN	ТфОП, B-ISDN, ATM, ЛВС	ЛВС, поддерживающие гарантированное качество обслуживания: IsoEthernet	Сеть с коммутацией пакетов без гарантированного качества обслуживания: Интернет, Token Ring, Ethernet	ТфОП
Алгоритмы кодирования видеоинформации	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263
Алгоритмы кодирования речевой информации	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728 G.723 G.729	G.723
Мультиплексирование	H.221	H.221	H.221	H.225.0	H.223
Управление информационными каналами	H.230 H.242	H.242	H.230 H.242	H.245	H.245
Данные	T.120	T.120	T.120	T.120	T.120
Интерфейсы и протоколы	I.400	AAL I.363 ATM I.361 PHY I.400	I.400 TCP/IP	TCP/IP	V.34

Оборудование мультимедийной связи содержит оконечные устройства, то есть устройства конечных пользователей (терминалы), и сетевые устройства, которые предоставляют пользователям услуги. Среди этих услуг – организация конференций, преобразование протоколов сигнализации и пользовательской информации, согласование скоростей передачи и дополнительные услуги, такие как переадресация вызовов и переключение связи.

Представляется полезным кратко рассмотреть особенности каждой из систем мультимедийной связи, предложенных ITU-T.

Рекомендация H.320 специфицирует системы видеотелефонной связи по В-каналам узкополосной ISDN со скоростью 64 Кбит/с и со скоростями до 1.920 Мбит/с, кратными 64 Кбит/с. Видеоинформа-

ция кодируется по алгоритму, предложенному ITU-T в рекомендации Н.261. Алгоритм поддерживает два формата изображений: необязательный формат CIF с разрешением 352 x 288 пикселей и обязательный формат QCIF с разрешением 176 x 144 пикселей. Частота кадров – 30 кадров/с или ниже. Для кодирования аудиоинформации используются алгоритм G.711 – импульсно-кодовая модуляция со скоростью передачи 64 Кбит/с, алгоритм G.722 – адаптивно-дифференциальная импульсно-кодовая модуляция, использующая полосу частот 7кГц и скорости передачи 48, 56, 64 Кбит/с, и G.728 – алгоритм кодирования LD-CELP со скоростью передачи 16 Кбит/с (об этих алгоритмах уже упоминалось в главе 3). Процедуры установления и разрушения соединений, формирования кадра данных и мультиплексирования, а также ряд эксплуатационных и административных функций описываются в рекомендациях Н.221, Н.230 и Н.242.

Рекомендация Н.321 определяет механизм адаптации узкополосных терминалов видеотелефонной связи Н.320 к работе в широкополосной ISDN. Следует отметить, что широкополосные ISDN базируются на транспортной технологии ATM, которая обеспечивает гарантированное качество обслуживания. В терминалах Н.321 реализована часть требований, предъявляемых к широкополосным терминалам видеотелефонной связи Н.310. При этом обязательным требованием Международного союза электросвязи является совместимость терминалов Н.310, Н.321 и Н.320.

В *рекомендации Н.322* представлены технические требования к узкополосным системам видеотелефонии, работающим в локальных вычислительных сетях с гарантированным качеством обслуживания, эквивалентным качеству обслуживания ISDN. Применение данной спецификации ограничено ЛВС IsoEthernet.

Рекомендация Н.323 специфицирует системы мультимедийной связи, которые ориентированы на работу в сетях с коммутацией пакетов, не обеспечивающих гарантированное качество обслуживания. К таким сетям относятся локальные вычислительные сети Ethernet и Token Ring, глобальная сеть Интернет и другие сети, поддерживающие технологию маршрутизации пакетов IP или IPX.

Рекомендация Н.323 предусматривает применение различных алгоритмов сжатия речевой информации, что позволяет использовать полосу пропускания гораздо более эффективно, чем в сетях с коммутацией каналов. Оконечные устройства Н.323 поддерживают передачу информации в режиме многоадресной рассылки, что позволяет организовывать конференции без дорогостоящих устройств управления конференциями (MCU), хотя на сегодняшний день без MCU не обойтись, т.к. режим многоадресной рассылки, как правило, IP-сетями не поддерживается. В приложении D к рекомендации Н.323 описан механизм передачи факсимильной информации в реальном времени по IP-сетям.

В *Рекомендации Н.324* Международный союз электросвязи представил технические требования к системам низкоскоростной мультимедийной связи, ориентированным на работу по аналоговым линиям ТфОП с использованием модемов. С учетом того, что скорость модемной передачи ограничена, в системах реализуются алгоритмы кодирования речи, видеоинформации и данных с высокой степенью сжатия.

Рекомендация Т.120 специфицирует механизм передачи данных в системах мультимедийной связи. Основным телематическим приложением, поддерживаемым рекомендацией Т.120, является обмен текстовыми сообщениями между участниками конференции в реальном времени. Другое приложение – коллективное редактирование растровых изображений. Система Т.120 является полностью платформо-независимой и может работать в широком диапазоне сетевых технологий с надёжной или ненадёжной передачей данных. Следует отметить, что механизм передачи данных Т.120 может функционировать как отдельно, так и совместно с вышеописанными системами мультимедийной связи. При этом поддерживаются режимы передачи данных с адресацией конкретному устройству и с многоадресной рассылкой.

Этот более чем краткий обзор деятельности ITU-T в области стандартизации систем мультимедийной связи, функционирующих в разном сетевом окружении, для целей данной книги представляется достаточным. Но прежде чем приступить к выполнению основной задачи данной главы – анализу системы мультимедийной связи, базирующейся на рекомендации Н.323, – будет полезно также ознакомиться с системами видеотелефонии Н.320, являющимися предшественницами систем мультимедийной связи Н.323.

5.2 Архитектура систем видеотелефонии в узкополосных ISDN

В 1990 году Международный союз электросвязи разработал рекомендацию Н.320, принятую впоследствии всеми ведущими производителями оборудования в качестве стандарта для реализации систем видеоконференций в узкополосных ISDN.

Система видеоконференций Н.320 включает в себя две основные группы компонентов – терминалы и устройства управления конференциями (Multipoint Control Units – MCU). Терминал представляет собой оборудование конечного пользователя, в то время как устройство управления конференциями является сетевым оборудованием, позволяющим организовывать видеоконференции с участием множества пользователей. Разрешается каскадное подключение друг к другу нескольких устройств MCU в рамках одной конференции. На рис. 5.1 показана архитектура системы видеоконференций, функционирующей в узкополосной ISDN.



Рис. 5.1 Система видеоконференций в узкополосной ISDN

Терминал Н.320 состоит из следующих функциональных модулей.

Блок видеоаппаратуры включает в себя видеокамеру, монитор и блок обработки видеоинформации, необходимый для реализации такой функции как разделение изображения в мониторе на несколько частей.

Блок аудиоаппаратуры содержит микрофон, громкоговоритель и блок обработки речевой информации, реализующий функции компенсации эха.

Телематические приложения обеспечивают передачу неподвижных изображений, коллективное редактирование растровых изображений, передачу файлов и др. Услуги передачи данных в системах видеотелефонии Н.320 реализуются на базе набора протоколов Т.120.

Модуль управления системой отвечает за выполнение таких функций, как организация доступа к сетевым ресурсам при помощи абонентской сигнализации и управление соединением – организация общего режима функционирования на основе сквозной (end-to-end) сигнализации.

Для установления, поддержания и разрушения соединений системы Н.320 используют протокол сигнализации Q.931 [7]. Обмен сигнальными сообщениями между терминалом и опорной АТС производится по D-каналу. Следует отметить также, что сигнальные сообщения не мультиплексируются с пользовательской и управляющей информацией.

Управление соединением производится на основе протоколов Н.242 и Н.243. Протокол Н.242 используется для обмена информацией о функциональных возможностях терминалов, выбора режима передачи речи, видео и данных в соединении между двумя пользо-

вателями и для изменения режима. Протокол H.243 используется для организации конференций, в которых несколько участников соединяются через устройство управления конференциями.

Аудиокодеки кодируют и декодируют речевую информацию. Рекомендация H.320 определила в качестве основного алгоритма кодирования речевой информации алгоритм G.711, рассмотренный в главе 3, но на практике, чаще всего, при скорости передачи информации 128 Кбит/с в конференциях используется алгоритм кодирования G.728, а при скорости 384 Кбит/с – алгоритм G.722.

Видеокодеки сжимают видеоинформацию и выполняют обратное преобразование. Рекомендация H.320 определяет видеокодек H.261 как обязательный; может также использоваться кодек H.263.

Блок синхронизации обеспечивает задержку речевых сигналов при передаче для синхронизации движения губ говорящего с его речью. Необходимость задержки связана с тем, что обработка видеоинформации занимает значительно больше времени, чем кодирование речи. Внесение задержки при передаче речевой информации не является обязательным требованием рекомендации H.320, но большинство производителей оборудования реализуют эту возможность, поскольку покупатели предпочитают видеть изображения, согласованные со звуком. Можно отметить, что в системах мультимедийной связи H.324, функционирующих в ТфОП, передающая сторона вместо задержки речевых сигналов информирует приемную сторону о средней разности фаз между звуковым и видеосигналом, благодаря чему приемная сторона может внести требуемую задержку при воспроизведении речи и изображения.

Система мультиплексирования, специфицированная в рекомендации H.221, обеспечивает возможность совместной передачи сигналов управления, речи, видео и данных. В-каналы разделяются на октеты, каждый бит которых в действительности представляет отдельный подканал. Например, в одном октете могут находиться биты управляющей информации, речи, видеоинформации и данных.

Сетевой интерфейс выполняет функции адаптации терминала, необходимые для его подключения к сети в соответствии с требованиями рекомендации I.400, рассмотренными в главе 3 [7].

Следующим элементом систем видеотелефонии H.320 является устройство управления конференциями (MCU).

Рекомендация H.320 определяет **устройство управления конференциями** как сетевое устройство, обеспечивающее участие пользователя в соединении одновременно с несколькими другими пользователями. Это устройство выполняет такие функции, как согласование скоростей передачи информации, смешивание речевых сигналов, переключение и смешивание видеосигналов, передача

данных от одного пользователя ко многим другим пользователям и управление конференциями. В части управления конференциями MCU реализует функцию согласования функциональных возможностей терминалов для того, чтобы выбрать общий для всех терминалов режим работы.

Таким образом, функции MCU можно разделить на две основные части: управление конференциями и обработка сигналов информационных каналов.

Управление конференциями включает в себя согласование алгоритмов, используемых каждым из участников конференции для кодирования речи, видеоинформации и данных, причем MCU может транскодировать информацию любого вида, если терминалы, участвующие в конференции, используют разные алгоритмы ее кодирования. В процессе согласования функциональных возможностей терминалов устройство управления конференциями выбирает режим конференции (selected communication mode).

Обработка сигналов информационных каналов является неотъемлемой функцией устройства управления конференциями. От каждого терминала, участвующего в конференции, MCU принимает речь, видеоинформацию и данные. Речевую информацию, получаемую от всех участников конференции, устройство управления конференциями смешивает и направляет суммарный речевой сигнал к каждому из участников. Для предотвращения эха MCU может не включать в суммарный речевой сигнал голос того участника, которому этот сигнал передается.

Видеосигналы обычно коммутируются устройством управления конференциями на основе анализа уровня речевого сигнала. Ко всем терминалам, участвующим в конференции, направляется видеосигнал, связанный с речевым сигналом, имеющим самый высокий уровень. Проще говоря, всем участникам конференции передается изображение участника, который в данный момент времени говорит наиболее громко. Помимо этого, устройство управления конференциями может работать в режиме смешивания видеосигналов для получения видеоизображения, содержащего информацию сразу от нескольких источников.

Кроме обработки речи и видеоинформации MCU может выполнять обработку данных в соответствии с рекомендацией T.120.

5.3 Мультимедийная связь в IP-сетях

Рекомендация Международного союза электросвязи H.323 является первой зонтичной спецификацией систем мультимедийной связи для работы в сетях с коммутацией пакетов, не обеспечивающих

гарантированное качество обслуживания. В рекомендациях, входящих в семейство H.323, определены протоколы, методы и сетевые элементы, необходимые для организации мультимедийной связи между двумя или более пользователями.

Наиболее востребованной из услуг, специфицированных в рекомендации H.323, в силу разных обстоятельств оказалась услуга передачи речевой информации по сетям с маршрутизацией пакетов IP. Самым распространенным подходом к построению сетей IP-телефонии сегодня является именно подход, предложенный ITU-T в рекомендации H.323.

Сети, построенные на базе протоколов H.323, ориентированы на интеграцию с телефонными сетями и могут рассматриваться как сети ISDN, наложенные на сети передачи данных. В частности, процедура установления соединения в таких сетях IP-телефонии базируется на рекомендации ITU-T Q.931 и практически идентична той же процедуре в сетях ISDN.

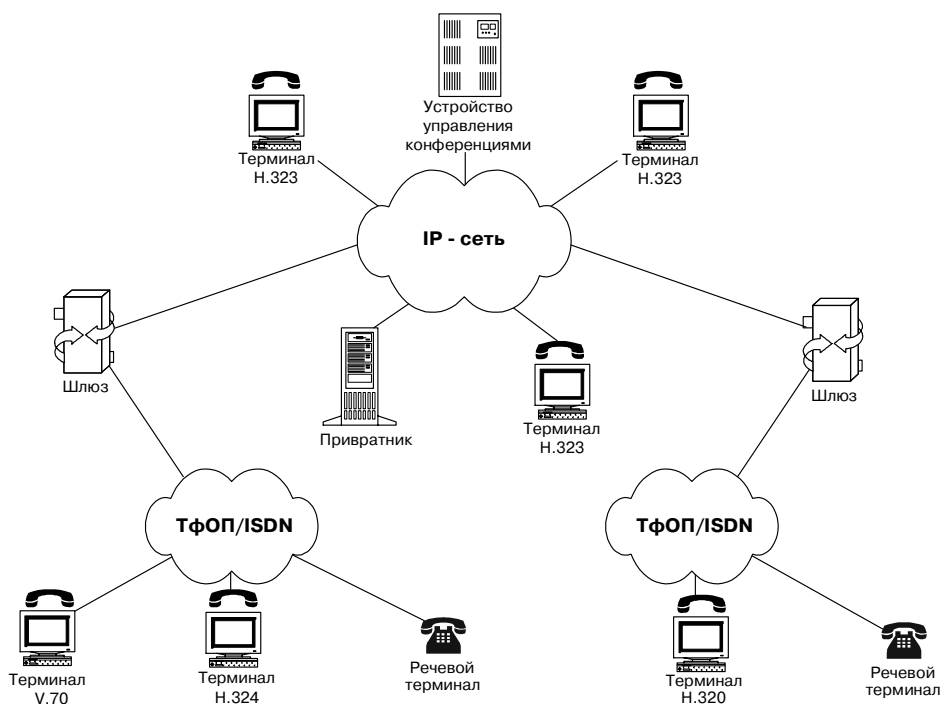


Рис. 5.2 Архитектура сети H.323

Этот вариант построения сетей IP-телефонии ориентирован на операторов местной телефонной связи (или на компании, владеющие транспортными сетями), которые желают использовать сети с маршрутизацией пакетов IP для предоставления услуг междугородной и международной связи. Протокол RAS, входящий в семейство

протоколов H.323, предоставляет операторам высокий уровень контроля использования сетевых ресурсов и обеспечивает поддержку аутентификации пользователей и начисления платы за предоставленные услуги.

На рис. 5.2 изображена архитектура сети, построенной на базе рекомендации H.323.

Основными устройствами сети являются: терминал, шлюз, привратник и устройство управления конференциями.

5.4 Терминал H.323

Терминал H.323 – это оконечное устройство сети IP-телефонии, обеспечивающее двухстороннюю речевую или мультимедийную связь с другим терминалом, шлюзом или устройством управления конференциями. Структурная схема терминала H.323 приведена на рис. 5.3.

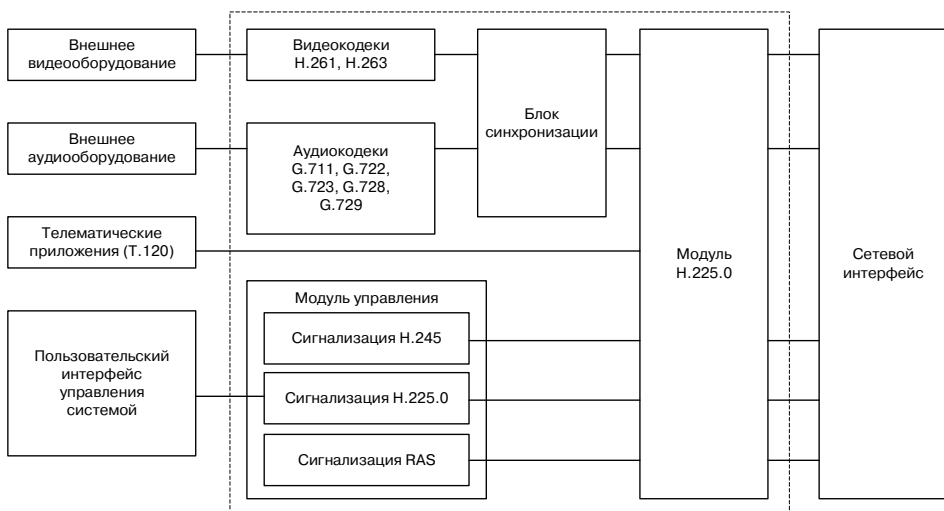


Рис. 5.3 Терминал H.323

Пользовательский интерфейс управления системой дает пользователю возможность создавать и принимать вызовы, а также конфигурировать систему и контролировать ее работу.

Модуль управления поддерживает три вида сигнализации: H.225, H.245 и RAS. Этот модуль обеспечивает регистрацию терминала у привратника, установление и завершение соединения, обмен информацией, необходимой для открытия разговорных каналов, предоставление дополнительных услуг и техобслуживание.

Телематические приложения обеспечивают передачу пользовательских данных, неподвижных изображений и файлов, доступ к базам данных и т.п. Стандартным протоколом для поддержки таких приложений является протокол T.120.

Модуль H.225.0 отвечает за преобразование видеоинформации, речи, данных и сигнальной информации в вид, пригодный для передачи по сетям с маршрутизацией пакетов IP, и за обратное преобразование. Кроме того, функциями модуля являются разбиение информации на логические кадры, нумерация последовательно передаваемых кадров, выявление и коррекция ошибок.

Сетевой интерфейс обеспечивает гарантированную передачу управляющих сообщений H.245, сигнальных сообщений H.225.0 (Q.931) и пользовательских данных при помощи протокола TCP и негарантированную передачу речевой и видеоинформации, а также сообщений RAS, при помощи протокола UDP.

Блок синхронизации вносит задержку на приемной стороне с целью обеспечить синхронизацию источника информации с ее приемником, согласование речевых и видеоканалов или сглаживание вариации задержки информации.

Видеокодеки кодируют видеоинформацию, поступающую от внешнего источника видеосигналов (видеокамеры или видеомagni-тофона), для ее передачи по сети с маршрутизацией пакетов IP и декодируют сигналы, поступающие из сети, для последующего отображения видеоинформации на мониторе или телевизоре.

Аудиокодеки кодируют аудиоинформацию, поступающую от микрофона (или других источников аудиоинформации), для ее передачи по сети с маршрутизацией пакетов IP и декодируют сигналы, поступающие из сети, для последующего воспроизведения. Любое терминальное оборудование H.323 должно иметь аудиокодеки. Обязательным для реализации является кодек, выполняющий преобразование речевой информации в соответствии с рекомендацией G.711. Реализация остальных алгоритмов кодирования, показанных на рис. 5.3, не обязательна. В том случае, когда в терминалах реализовано несколько алгоритмов кодирования речевой информации, желательно, чтобы терминалы могли работать в асимметричном режиме, например, принимать речь, закодированную по алгоритму G.711, и передавать речь, закодированную по алгоритму G.728.

Следует отметить, что при организации децентрализованной конференции терминал H.323 может принимать более чем один поток речевой информации. В этом случае терминал должен уметь смешивать или переключать пакетированную речь, поступающую от остальных участников конференции.

5.5 Шлюз Н.323

Основной функцией шлюза является преобразование речевой (мультимедийной) информации, поступающей со стороны ТФОП с постоянной скоростью, в вид, пригодный для передачи по IP-сетям, т.е. кодирование информации, подавление пауз в разговоре, упаковка информации в пакеты RTP/UDP/IP, а также обратное преобразование.

Кроме того, шлюз должен уметь поддерживать обмен сигнальными сообщениями как с коммутационным или терминальным оборудованием ТФОП, так и с привратником или оконечным устройством сети Н.323. Таким образом, шлюз должен преобразовывать аналоговую абонентскую сигнализацию, сигнализацию по 2ВСК, сигнальные сообщения систем сигнализации DSS1 и ОКС7 [6,7] в сигнальные сообщения Н.323. Спецификации преобразования сигнализации Q.931 (DSS1, QSIG) и ОКС7 в сигнализацию Н.225.0, основанную на Q.931, приведены в рекомендации ITU-T Н.246. Для поддержки дополнительных услуг в шлюзе может быть обеспечена прозрачная передача сигнальных сообщений Q.932 и Н.450.

Желательно, чтобы шлюз мог генерировать и распознавать сигналы DTMF на стороне ТФОП и передавать сигналы DTMF в сообщениях Н.245 **userInputIndication** по сети с маршрутизацией пакетов IP.

При отсутствии в сети привратника должна быть реализована еще одна функция шлюза – преобразование номера ТФОП в транспортный адрес IP-сети.

Со стороны сетей с маршрутизацией пакетов IP, так же, как и со стороны ТФОП, шлюз может участвовать в соединениях в качестве терминала или устройства управления конференциями (рис. 5.4).

Примечательно, что шлюз может изначально участвовать в соединении в качестве терминала, а затем, при помощи сигнализации Н.245, продолжить работу в качестве устройства управления конференциями.

В случае, когда терминал Н.323 связывается с другим терминалом Н.323, расположенным в той же самой IP-сети, шлюз в этом соединении не участвует.

Шлюз, в совокупности с привратником сети IP-телефонии, образует универсальную платформу для предоставления всего спектра услуг мультимедийной связи. На рис. 5.5 представлены некоторые услуги, которые могут быть реализованы в прикладном программном обеспечении на базовой платформе аппаратных и программных средств шлюза IP-телефонии.

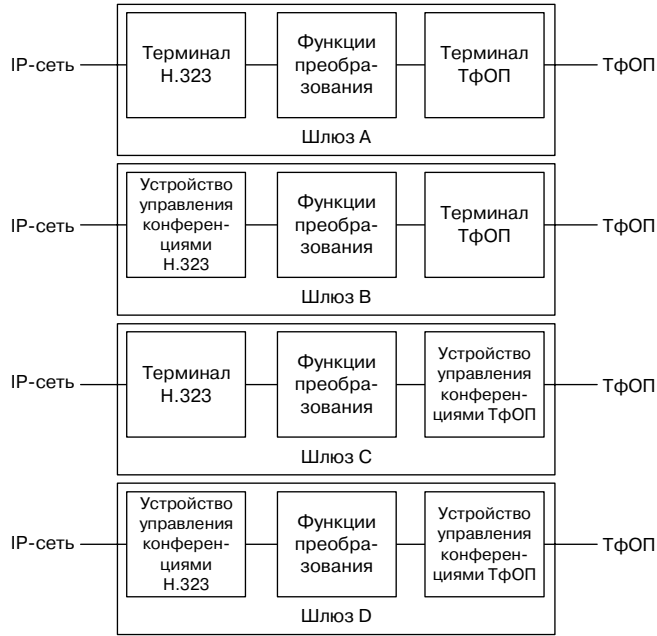


Рис. 5.4 Возможные конфигурации шлюза

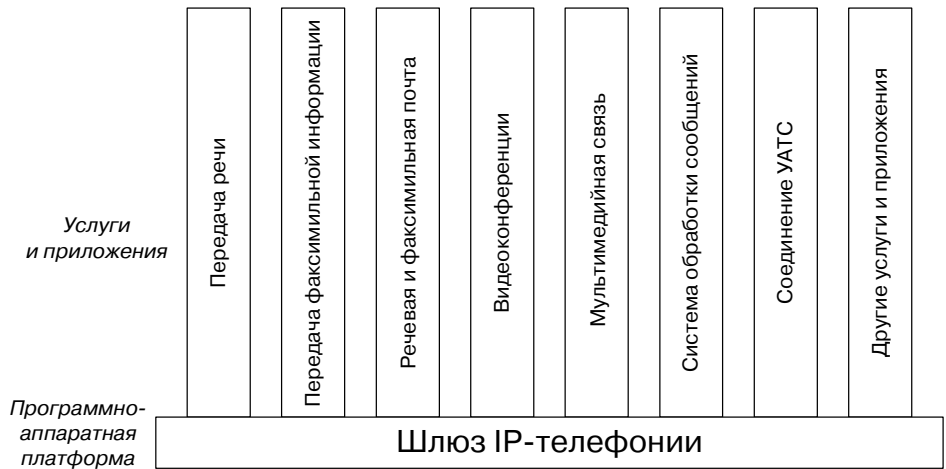


Рис. 5.5 Платформа и услуги шлюза IP-телефонии

5.6 Привратник

В привратнике сосредоточен весь интеллект сетей IP-телефонии, базирующихся на рекомендации ITU H.323. Сеть H.323 имеет зонную архитектуру (рис. 5.6). Привратник выполняет функции управления зоной сети IP-телефонии, в которую входят терминалы, шлюзы

и устройства управления конференциями, зарегистрированные у этого привратника. Разные участки зоны сети H.323 могут быть территориально разнесены и соединяться друг с другом через маршрутизаторы. Следует обратить внимание на то, что коммутаторы кадров Ethernet и маршрутизаторы пакетов IP не являются сетевыми элементами H.323, так как они работают на звеньевом или сетевом уровнях соответственно, в то время как оборудование H.323 работает на прикладном уровне стека протоколов TCP/IP.

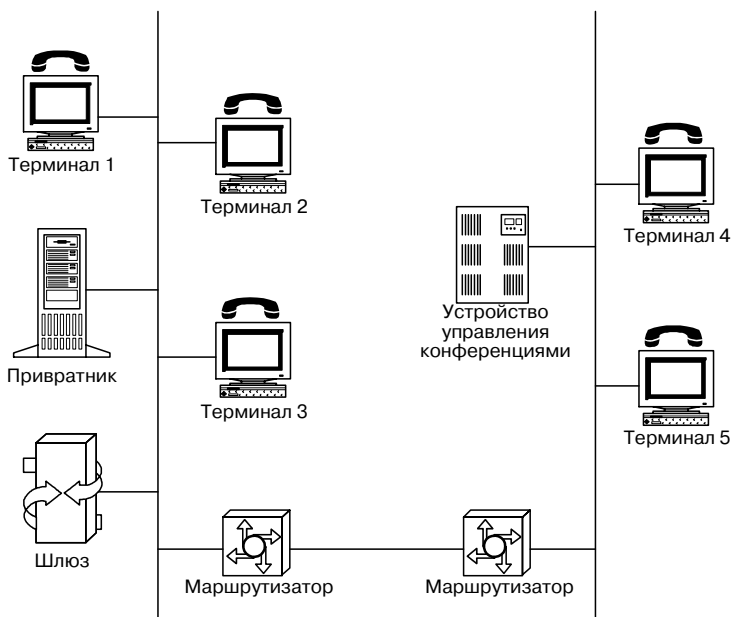


Рис. 5.6 Зона сети H.323

В число наиболее важных функций, выполняемых привратником, входят:

- преобразование так называемого *alias*-адреса (имени абонента, телефонного номера, адреса электронной почты и др.) в транспортный адрес сети с маршрутизацией пакетов IP (IP адрес и номер порта TCP);
- контроль доступа пользователей системы к услугам IP-телефонии при помощи сигнализации RAS (используются сообщения ARQ/ACF/ARJ);
- контроль, управление и резервирование пропускной способности сети;
- маршрутизация сигнальных сообщений между терминалами, расположенными в одной зоне; привратник может организовывать сигнальный канал непосредственно между терминалами или

ретранслировать сигнальные сообщения от одного терминала к другому. В последнем случае привратник в любое время знает состояние конечных пользователей и может предоставлять дополнительные услуги, такие как переадресация, переключение связи, установка вызова на ожидание, перехват вызова и т.д. Хотя, справедливости ради, надо отметить, что эти услуги могут быть реализованы (согласно рекомендациям ITU H.450.x) в терминалах пользователей и предоставляться без участия привратника.

В том случае, когда вызывающий абонент знает IP-адрес терминала вызываемого абонента, соединение между двумя устройствами может быть установлено без помощи привратника. Это означает, что наличие в сети H.323 привратника не обязательно. Но, в то же время, следует отметить, что при наличии привратника обеспечивается мобильность абонентов, т.е. способность пользователя получить доступ к услугам с любого терминала в любом месте сети и способность сети идентифицировать пользователей при их перемещении из одного места в другое.

При отсутствии в сети привратника преобразование адреса вызываемого абонента, поступающего со стороны ТФОП в формате E.164, в транспортный адрес IP-сети должно выполняться шлюзом.

В одной сети может находиться несколько привратников, которые должны взаимодействовать между собой. Следует особо отметить, что хотя привратник является отдельным логическим элементом сети, он может быть реализован в терминале, в шлюзе, в устройстве управления конференциями или в устройствах, не специфицированных в рекомендации H.323. Примерами таких устройств могут быть система распределения вызовов, учрежденческая телефонная станция, система обработки телефонных карт, система речевой почты и другие приложения.

5.7 Устройство управления конференциями

Рекомендация H.323 предусматривает три вида конференций (рис. 5.7).

Первый вид – централизованная конференция, в которой конечные устройства соединяются в режиме точка-точка с устройством управления конференциями (MCU), контролирующим процесс создания и завершения конференции, а также обрабатывающим потоки пользовательской информации.

Второй вид – децентрализованная конференция, в которой каждый ее участник соединяется с остальными участниками в режиме точка – группа точек, и конечные устройства сами обрабатывают (переключают или смешивают) потоки информации, поступающие от других участников конференции.

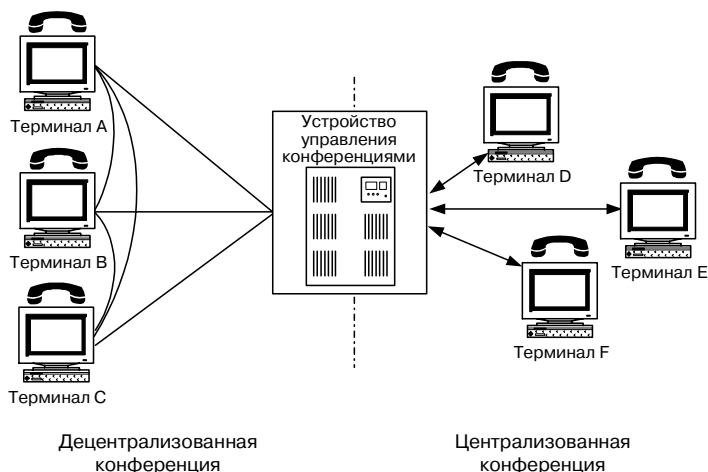


Рис. 5.7 Разные виды конференции в сети H.323

Третий вид – смешанная конференция, т.е. комбинация двух предыдущих видов.

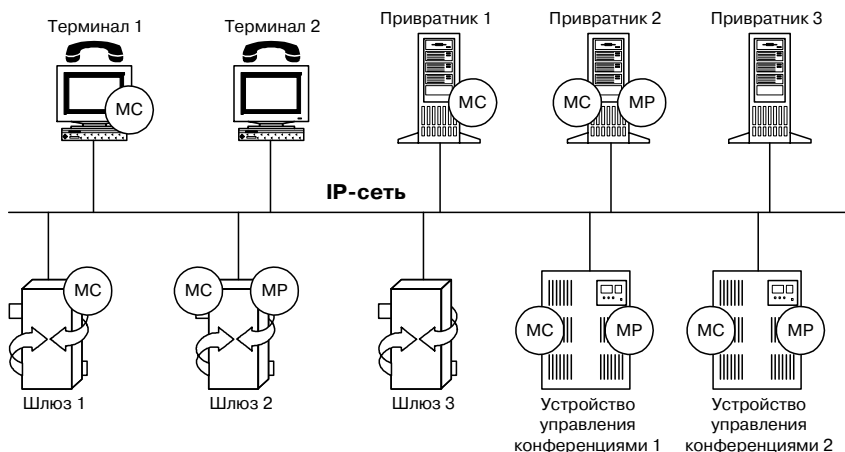
Преимущество централизованной конференции – сравнительно простые требования к терминальному оборудованию, недостаток – большая стоимость устройства управления конференциями.

Для децентрализованной конференции требуется более сложное терминальное оборудование, кроме того, желательно, чтобы в сети поддерживалась передача пакетов IP в режиме многоадресной рассылки (IP multicasting). Если сеть не поддерживает этот режим, терминал может передавать информацию к каждому из остальных терминалов, участвующих в конференции, в режиме точка-точка, но это становится неэффективным при числе участников более четырех.

Устройство управления конференциями MCU содержит один обязательный элемент – контроллер многоточечных соединений – Multipoint controller (MC). Кроме того, MCU может содержать один или более процессоров для обработки информации пользователей при многоточечных соединениях – Multipoint processor (MP). Следует отметить, что контроллер MC и процессор MP являются самостоятельными логическими устройствами H.323 и что контроллер может существовать независимо от процессора (обратное неверно). Контроллер может быть физически совмещен с привратником, со шлюзом или с MCU, а MCU, в свою очередь, может быть совмещено со шлюзом или с привратником (рис. 5.8).

Контроллер конференций должен использоваться для организации конференции любого вида. Он организует обмен между участниками конференции данными о функциональных возможностях (capabilities) их терминалов, указывает, в каком режиме (с использова-

нием каких кодеков) участники конференции могут передавать информацию, причем этот режим может изменяться в ходе конференции, например, при подключении к ней нового участника. Таким образом, контроллер MC определяет режим конференции (selected communication mode – SCM), который может быть общим для всех участников конференции или отдельным для каждого из них.



Примечание: шлюз, привратник и устройство управления конференциями могут представлять собой единое устройство

Рис. 5.8 Возможная реализация MC и MP в оборудовании H.323

Так как в сети может быть несколько контроллеров MC, то для каждой вновь создаваемой конференции должна проводиться процедура определения ведущего и ведомого оборудования, с тем, чтобы выявить тот из контроллеров MC, который будет управлять данной конференцией.

При организации централизованной конференции, кроме контроллера MC, должен использоваться процессор MP, обрабатывающий пользовательскую информацию и отвечающий за переключение или смешивание речевых потоков, видеоинформации и данных. При организации децентрализованной конференции процессор MP не используется.

5.8 Реализация оборудования H.323

В 2000 г. успешно прошел сертификацию комплекс оборудования IP-телефонии компании Lucent Technologies типа Packet-Star IP 1000. В состав этого комплекса входят шлюз, привратник и устройство управления сетью (Manager). Шлюз Packet-Star IP Gateway 1000 поддерживает до 3360 одновременных соединений и может, совместно с привратником, обслуживать до нескольких миллионов телефонных

соединений и факсимильных сессий в день. Основные особенности оборудования состоят в том, что со стороны опорной АТС принимается множество потоков речевой информации со скоростью 2 048 Кбит/с (Е1), речь кодируется при помощи алгоритмов G.729a и G.723.1, поддерживаются все распространенные системы сигнализации, внутренняя шина системы функционирует на базе технологии ATM со скоростью 5 Гбит/с, вносимая шлюзом задержка при использовании алгоритма кодирования речи G.729a составляет до 80 мс. Помимо поддержки второй версии протокола H.323, обеспечена совместимость с оборудованием других фирм-производителей по профилю iNOW! В минимальной комплектации стоимость комплекса составляет несколько сотен тысяч долларов. Близкий по функциям комплекс оборудования IP-телефонии, рассчитанный на поддержку до 2000 одновременных разговорных соединений, разработала фирма Nokia. Масштабы и стоимость этой аппаратуры ориентированны на крупных операторов связи.

Lucent Technologies выпускает также платы IP-телефонии для учрежденческих АТС Definity. Они позволяют направлять телефонные вызовы по сети Интернет или Intranet с помощью функции маршрутизации в IP-сетях – DEFINITY World Class Routing. Модуль IP-Trunk представляет собой встроенный шлюз IP-телефонии, принимающий речевой сигнал из ТФОП и преобразующий его в пакетную форму. Одна плата IP-Trunk может обработать 30 речевых каналов, то есть полный цифровой поток Е1. Отдельный системный вход в модуль IP-Trunk позволяет администратору АТС устанавливать соответствие между номерами телефонной сети и IP-адресами, задавать правила маршрутизации и параметры обслуживания. Аналогичный по функциям модуль IP-card поддерживает подключение 24 IP-телефонов (серия 6600), работающих по протоколу H.323.

Lucent Technologies выпускает также программный продукт Definity Soft Phone, который работает совместно с широко распространенным приложением Microsoft NetMeeting и позволяет, подсоединившись к АТС Definity через Интернет как к серверу, стать ее внутренним абонентом. Качество связи при этом будет, разумеется, зависеть от скорости передачи пакетов по сети Интернет. Кроме того, компания Lucent Technologies предлагает отдельное решение в области IP-УАТС под названием IP ExchangeComm, по структуре и компонентам сходное с решением CCN компании Cisco. Отдельно поставляется инструментальный разработчика программ (Software Developer's Kit) для TAPI 2.1-3.0, позволяющий создавать новые приложения для IP-сети и добавлять новые функции к уже существующим продуктам. Компания Lucent начала выпуск и IP-телефонов – IP Exchange. Аппаратное и программное обеспечение телефонов IP Exchange совместимо с версией 2 стандарта H.323 и поддерживает алгоритмы компрессии речи G.711, G.723 и G.729.

Интерес представляет решение компании Cisco – использовать сервера доступа и маршрутизаторы, например, Cisco 5300 и 3600, в которые добавлены речевые модули. Маршрутизаторы играют роль шлюзов, упаковывая речевую информацию в IP-пакеты. В этом направлении, кроме Cisco, активно работают фирмы Nortel и Ericsson (аппаратура IPTC). Фирмой OKI Network Technologies разработана аппаратура BS 1200 Internet Voice Gateway, предназначенная для установки непосредственно на АТС (УАТС). Помимо сокращения объема оборудования такое решение упрощает синхронизацию и сигнализацию. Программное обеспечение Cisco CallManager, работающее под операционной системой Windows NT, предназначено для управления соединениями между IP-телефонами Cisco и телефонными аппаратами ТфОП. Внешне IP-телефоны фирмы Cisco выглядят как цифровые телефонные аппараты с жидкокристаллическим экраном и встроенным 2-х портовым Ethernet-концентратором, позволяющим не занимать для телефона отдельную розетку RJ-45. Поддерживается компрессия речи G.711 или G.723.1, в зависимости от выбора ПО CallManager. IP-адрес может присваиваться телефону статически или динамически, в последнем случае используется протокол DHCP.

Продукт IP-телефонии компании Ericsson состоит из следующих компонентов: шлюза, преобразующего формат речевой информации и сигнализацию, Sitekeeper – системы управления, выполняющей все функции управления, такие как маршрутизация, сбор информации о вызовах и т.д. В систему входит база данных, которая хранит информацию о вызовах, а также биллинговую информацию и, при необходимости, информацию об абонентах сети (пароль, тип услуги, текущее состояние). Сервер управления выполняет функции глобального управления всей сетью IP-телефонии, хранит информацию о топологии и конфигурации этой сети, а также таблицы маршрутизации. Следуя общим принципам технологии IP-телефонии, продукт Ericsson имеет, в то же время, некоторые особенности. Применено дополнительное технологическое решение – Phone Doubler, позволяющее создавать в Интернет вторую виртуальную телефонную линию. Точнее, оператор получает возможность предоставить клиентам новую услугу: работать в Интернет и разговаривать по телефону одновременно, занимая всего лишь одну аналоговую линию. Обычно для этого нужна либо вторая аналоговая линия, либо линия ISDN. В данном случае вызывающий абонент набирает номер вызываемого абонента, и если этот номер занят, то вызов переадресуется телефонной станцией к шлюзу. В свою очередь, шлюз передает запрос установления соединения вызываемому абоненту через IP-сеть. У этого абонента на экране появляется сообщение о входящем вызове. Далее устанавливается телефонное соединение. Кроме того, воспользовавшись продуктом Phone Doubler, пользователь Интернет

может, например, произвести телефонный вызов, не прерывая своей Интернет-сессии, используя клиентское ПО. Более подробное описание реализации данной услуги в рамках другой платформы – комплекса Протей-IP – приведено в главе 11.

Nortel Networks анонсировала целую серию новых продуктов IP-телефонии под общим названием Inca. Семейство Inca предоставляет потребителям портфель открытых решений на базе стандарта H.323 для объединения сетей IP и телефонных сетей общего пользования. I2004 Internet Telephone – первый из этой серии Интернет-телефонов – совместим со стандартом H.323 и использует алгоритмы сжатия речи G.711, G.723.1 и G.729A перед ее упаковкой в IP-пакеты. Еще в 1999 года корпорация Nortel Networks создала платы IP-телефонии для своей базовой модели ATC Meridian. Плата, называемая Meridian Integrated IP Telephony Gateway, способна обрабатывать и упаковывать в IP-пакеты 8 речевых каналов. Она устанавливается в периферийный модуль ATC, поддерживающий интерфейс 10Base-T с корпоративной сетью. Поддерживаются стандарты сжатия речи G.711 (64 Кбит/с), G.723 (до 5.3 Кбит/с) и G.729 (8 Кбит/с). Архитектура Integrated IP Telephony Gateway близка к рассмотренному в параграфе 11.8 модулю IPU, поэтому подробное рассмотрение этих решений отложим до главы 11.

IP-ATC RC3000 и IP-телефон LP 5100 IP, выпускаемые фирмой Siemens с начала 1999 года, относятся к сетевому оборудованию HiNet. Система рассчитана максимум на 50 абонентов. HiNet LP 5100 IP поддерживает стандарт H.323, алгоритмы сжатия речи G.711 (64 Кбит/с) и G.723.1 (6.3 или 5.3 Кбит/с), а также функцию подавления эха.

В таблицу 5.2 сведены основные технические и функциональные характеристики учрежденческих ATC, реализованных на базе технологии маршрутизации речевой информации по сетям с маршрутизацией пакетов IP (IP-PBX) и поддерживающих протокол H.323.

Подводя итог данной главы, нужно сказать о том, что большинство экспертов считает основной тенденцией мирового телекоммуникационного рынка конца 90-х годов движение бизнес-телефонии от традиционной коммутации каналов к коммутации пакетов. Эта тенденция тесно связана с повсеместным переходом к сетям с интеграцией в одном канале всех видов информации – данных, речи, видео и факсимиле. Новые продукты, о которых мы говорили, – первый отклик крупных телекоммуникационных компаний на интерес потенциальных потребителей к этой части рынка. Большинство разработок пока предполагается использовать в корпоративных сетях или в небольших офисах, а не в глобальных сетях компаний-операторов связи. Однако в ближайшее время ожидается появление более мощного оборудования IP-телефонии.

Таблица 5.2. Сравнительные характеристики IP-RBX

Название	WebSwitch 2000	Phoneware SBX	IP ExchangeComm System	HiNetRC3000	CallManager
Фирма	Ericsson	Tedas	Lucent Technologies	Siemens	Cisco
Емкость	Модель M2 - 32 пользователей; модель M4 - 64 пользователей; для увеличения емкости несколько PBX объединяются	Одновременно 8/30 внешних соединений и до 240 внутренних соединений	От 2 до 96 пользователей; для увеличения емкости несколько PBX объединяются	Нет данных	10000 пользователей на 1 кластер; до 10 кластеров, т.е. для увеличения емкости несколько PBX объединяются
Версия протокола H.323	H.323v2	H.323v1 (версия 2 готовится)	H.323v1	Нет данных	Нет данных
Интерфейсы	Аналоговые абонентские линии и линии к ATC, E1, 10BaseT, WAN-интерфейсы;	4 BRI или 1 PRI (DSS1)	аналоговые и T1	4 BRI или 1 PRI и 4 10BaseT	интерфейсы ISDN
Шлюз	интегрированный	интегрированный	интегрированный	необязательный	интегрированный
Функциональные возможности оборудования	поддержка IP-телефонов и аналоговых телефонов, совместимость с Netmeeting; эхокомпенсация G.168; видеореχεία почта, поддерживается доступ к речевой почте извне; wireless VoIP: Symbol Technologies access point и 802.11-беспроводные телефоны (или DECT, когда не нужно передавать данные); объединение PBX в сеть через SL, общий план нумерации при нескольких PBX в сети и сокращенная нумерация, Web-телефония	SDK; PBX на базе Gatekeeper; MCU для смешивания и речевой информации и проигрывания музыки при удержании; речевая почта с аутентификацией, доступом извне, индикация приема почты или передача аудио файла; поддержка беспроводных терминалов; видеоконференция	SDK; администрирование из любого места в сети через web-интерфейс, Windows NT; начисление платы; автоответчик; подключение аналоговых устройств через адаптер; поддержка качества обслуживания (приоритеты трафика); поддержка секретности; SNMP с индикацией аварий и статуса; контроль использования ресурсов; речевая почта: 96 почтовых ящиков с неограниченным числом сообщений максимальной длительности 5 мин; PBX соединяются через IP между собой, образуя сеть, и вызов приходит в ТФОП из IP в наилучшей точке	администрируется локально через RS232 или удаленно через SNMP; разработан терминальный адаптер для подключения аналоговых устройств; интегрированный привратник; речевые кодеки только G.711 и G.723; передача данных T.128; совместная работа с приложениями (Application shared - полное и неполное); Windows NT; поддержка DNS; подробные записи о вызовах; эксплуатационное управление через интуитивно понятный Web-интерфейс; персональная адресная книга; возможность ограничения пользования услугами	автоинформатор; поддержка протокола MGCP; передача сигналов DTMF через H.245; блокирование вызовов; анализ и обработка номера (ввод префиксов, удаление цифр); начисление платы и статистика; передача факсимильной информации (G.711); генерация комфортного шума; PBX соединяются между собой через WAN

СТП	ТАРІ	Outlook contacts как телефонная книга; создание и завершение соединения по расписанию, журнал событий в формате MS Access	ТАРІ 2.1; протокол LDAP для работы с директориями	Unified message: e-mail с приложенными текстовыми, речевыми и аудио файлами	ТАРІ 2.1 и JTAR 1.3
Функции ступени распределения вызова	ACD, IVR с автоответчиком	ACD, IVR с автоответчиком; поддержка не только IP-телефонов: внешние ТФОП телефоны также могут считаться телефонами РВХ; начисление платы; аутентификация; менеджмент через web; сокращенный набор и интеграция с WWW; группа серийного искания	Нет данных	ACD; обратный вызов после запроса через web; IVR; воспроизведение извещений и подсказок; статистика	Нет данных
Дополнительные услуги	конференция до 8 участников; переключение связи и переадресация внутри станции и вне ее; режим удержания; перехват вызова; ответ с другого аппарата (pick up), постановка входящего вызова в очередь при занятости, группы серийного искания, извещение о новом вызове при разговоре, музыка при удержании и при ожидании, основная группа дополнительных услуг доступна с IP-аналоговых и IP-телефонов	режим удержания, ответ с другого аппарата (Pickup), переключение связи, переадресация (любая), конференция между терминалами LAN и ТФОП, CLIP/CLIR, COLP/COLR, MSN/DDI, DTMF - детектирование/ генерация (N.245)	отказ принять вызов, переключение связи, режим удержания, переключение связи, установка на ожидание, извещение о новом вызове во время разговора, идентификация имени/номера вызывающего абонента, конференция, повторение номера, набранного последним, блокировка всех входящих вызовов; сокращенный набор	режим удержания; переключение связи; переадресация (любая); детектирование и генерация DTMF (RTP и N.245); повторение номера, набранного последним	автоматический прием/отказ от приема вызова; переключение связи внутри сети и во внешнюю сеть; переадресация (любая); режим удержания с наведением справки; парковка/ответ с другого аппарата; постановка на ожидание; CLIP/CLIR; COLP/COLR; DID; извещение о новом вызове во время связи; конференция; набор номера и заказ переадресации через WEB

Глава 6

Сигнализация

Н.323

6.1 Семейство протоколов Н.323

Семейство протоколов Н.323 включает в себя три основных протокола: протокол взаимодействия оконечного оборудования с привратником – RAS, протокол управления соединениями – Н.225 и протокол управления логическими каналами – Н.245.

Эти три протокола, совместно с Интернет-протоколами TCP/IP, UDP, RTP и RTCP, а также с описанным в [6] протоколом Q.931, представлены на рис.6.1. Суть изображенной на этом рисунке иерархии заключается в следующем. Для переноса сигнальных сообщений Н.225 и управляющих сообщений Н.245 используется протокол с установлением соединения и с гарантированной доставкой информации – TCP. Сигнальные сообщения RAS переносятся протоколом с негарантированной доставкой информации – UDP. Для переноса речевой и видеоинформации используется протокол передачи информации в реальном времени – RTP. Контроль переноса пользовательской информации производится протоколом RTCP.

С учетом того, что стек протоколов TCP/IP и протоколы UDP, RTP и RTCP уже рассматривались в главе 4, материал данной главы будет посвящен протоколам RAS, Н.225 и Н.245. Степень детализации их рассмотрения определяется конечной целью – изложению сценария базового процесса обслуживания вызова, в упрощенном виде уже упоминавшегося в главах 1 и 2.

Гарантированная доставка информации по протоколу TCP		Негарантированная доставка информации по протоколу UDP		
H.245	H.225		Потоки речи и видеoinформации	
	Управление соединением (Q.931)	RAS	RTCP	RTP
TCP		UDP		
IP				
Канальный уровень				
Физический уровень				

Рис. 6.1 Семейство протоколов H.323

6.2 Протокол RAS

Международный союз электросвязи в рекомендации H.225.0 определил протокол взаимодействия рассмотренных в предыдущей главе компонентов сети H.323: оконечного оборудования (терминалов, шлюзов, устройств управления конференциями) с привратником. Этот протокол получил название RAS (Registration, Admission and Status).

Основными процедурами, выполняемыми оконечным оборудованием и привратником с помощью протокола RAS, являются:

1. Обнаружение привратника.
2. Регистрация оконечного оборудования у привратника.
3. Контроль доступа оконечного оборудования к сетевым ресурсам.
4. Определение местоположения оконечного оборудования в сети.
5. Изменение полосы пропускания в процессе обслуживания вызова.
6. Опрос и индикация текущего состояния оконечного оборудования.
7. Оповещение привратника об освобождении полосы пропускания, ранее занимавшейся оборудованием.

Выполнение первых трех процедур, предусмотренных протоколом RAS, является начальной фазой установления соединения с использованием сигнализации H.323. Далее следуют фаза сигнализации H.225.0 (Q.931) и обмен управляющими сообщениями H.245. Разъединение происходит в обратной последовательности: в первую очередь закрывается управляющий канал H.245 и сигнальный канал H.225.0, после чего по каналу RAS привратник оповещается об освобождении ранее занимавшейся оконечным оборудованием полосы пропускания.

Для переноса сообщений протокола RAS используется протокол негарантированной доставки информации UDP. В связи с этим ITU-T рекомендовал передавать повторно те сообщения RAS, получение которых не было подтверждено в течение установленного промежутка времени. Оконечное оборудование или привратник, не имеющие возможности в текущий момент времени ответить на полученный запрос, могут передавать сообщение RIP (Request in Progress) для индикации того, что запрос находится в стадии обработки. При приеме сообщения RIP привратник и оконечное оборудование должны перезапустить свои таймеры.

Важно отметить, что в сети без привратника сигнальный канал RAS вообще не используется.

6.2.1 Обнаружение привратника

Для взаимодействия оконечного оборудования с привратником нужно, чтобы устройству стал известен сетевой адрес подходящего привратника. Процесс определения этого адреса называется обнаружением привратника. Определены два способа обнаружения – ручной и автоматический.

Ручной способ заключается в том, что привратник, обслуживающий данное устройство, определяется заранее при конфигурации этого устройства. Первая фаза установления соединения начинается сразу с запроса регистрации устройства, который передается на уже известный сетевой адрес привратника и на UDP-порт 1719, а в случае взаимодействия с привратником, поддерживающим первую версию протокола Н.323, – на порт 1718.

При автоматическом способе обнаружения привратника устройство передает запрос Gatekeeper Request (GRQ) в режиме многоадресной рассылки (multicasting), используя IP-адрес 224.0.1.41 – Gatekeeper UDP Discovery Multicast Address – и UDP порт 1718 – Gatekeeper UDP Discovery Port. Ответить оконечному оборудованию могут один или несколько привратников, передав на адрес, указанный в поле **rasAddress** запроса GRQ, сообщение Gatekeeper Confirmation (GCF) с предложением своих услуг и с указанием транспортного адреса канала RAS (рис.6.2.). Если привратник не имеет возможности зарегистрировать оконечное оборудование, он отвечает на запрос сообщением Gatekeeper Reject (GRJ).

Если на GRQ отвечает несколько привратников, оконечное оборудование может выбрать по своему усмотрению любой из них, после чего инициировать процесс регистрации. Если в течение 5 секунд ни один привратник не ответит на GRQ, оконечное оборудование может повторить запрос. Если ответ опять не будет получен, необходимо прибегнуть к ручному способу обнаружения привратника.

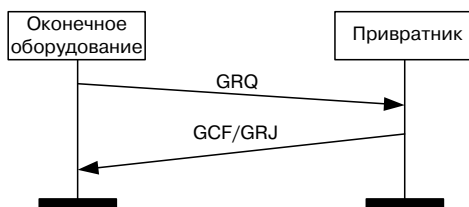


Рис. 6.2 Автоматическое обнаружение привратника

При возникновении ошибки в процессе регистрации у своего привратника, т.е. при получении отказа в регистрации или при отсутствии ответа на запрос регистрации, оконечное оборудование должно провести процедуру обнаружения привратника снова.

С точки зрения простоты технического обслуживания сети автоматический способ обнаружения предпочтительнее ручного, так как при возникновении каких-либо неисправностей в работе привратника для переключения к новому привратнику не надо будет вручную менять конфигурацию оборудования зоны: переключение устройств к другому привратнику произойдет автоматически. Чтобы облегчить эту задачу и повысить надежность работы сети, привратник может предоставлять в поле **alternateGatekeeper** сообщений GCF и RCF перечень альтернативных привратников, к которым устройство может переключиться в случае выхода из строя собственного привратника.

В то же время, следует сказать о том, что режим многоадресной рассылки в IP-сетях не очень распространен, поэтому, скорее всего, автоматическое обнаружение привратника найдет применение только в корпоративных сетях. Следует также отметить, что привратник должен уметь принимать и обрабатывать множество запросов от одного и того же оборудования, так как процедура обнаружения может периодически повторяться, например, при включении питания или при входе в сеть.

6.2.2 Регистрация оконечного оборудования

После выполнения процедуры обнаружения привратника оконечное оборудование должно быть присоединено к зоне сети, обслуживаемой данным привратником. Для этого оборудование должно сообщить привратнику свою адресную информацию: список *alias*-адресов и транспортных адресов. Этот процесс называется регистрацией оконечного оборудования у привратника.

В предыдущей главе уже упоминалось, что если в качестве оконечного оборудования выступают шлюз или устройство управления конференциями, то они могут зарегистрировать у привратника несколько транспортных адресов для каналов сигнализации RAS и H.225.0 (Q.931). Кроме того, для повышения надежности работы

сети окончному оборудованию разрешается иметь дополнительные транспортные адреса, что дает возможность иметь в одном оборудовании два сетевых интерфейса или предусматривать дублирующее оборудование. Дополнительные транспортные адреса указываются в параметре **alternateEndpoint** некоторых сообщений сигнализации RAS.

Процесс регистрации представлен на рис.6.3. Оконечное оборудование передает запрос регистрации Registration Request (RRQ) на сетевой адрес привратника, либо полученный при выполнении процедуры его автоматического обнаружения, либо известный априори. Стоит отметить, что запрос направляется на общеизвестный номер UDP-порта 1719. Этот порт имеет соответствующее название – Gatekeeper UDP Registration and Status Port. Привратник отвечает на запрос подтверждением Registration Confirmation (RCF) или отказом в регистрации Registration Reject (RRJ). Напомним, что окончное оборудование может зарегистрироваться только у одного привратника.

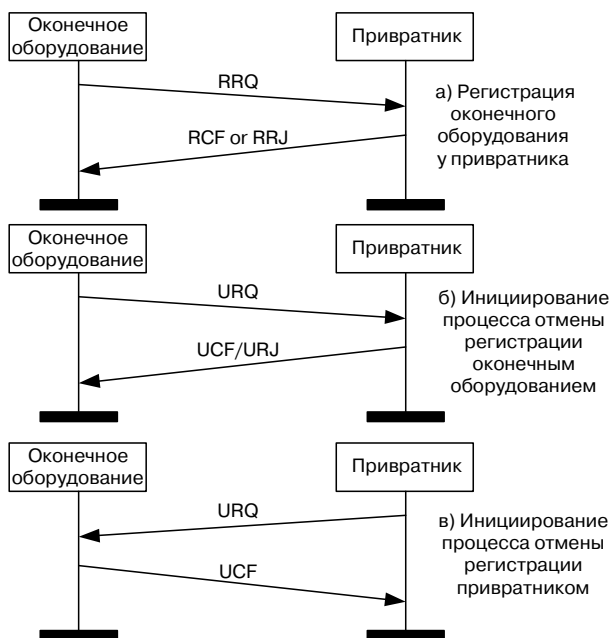


Рис. 6.3 Процесс регистрации и отмены регистрации

Если окончное оборудование не указывает свой *alias*-адрес в запросе RRQ, привратник может сам назначить такой адрес и вернуть его в сообщении RCF.

Регистрация окончного оборудования должна быть проведена перед началом установления первого соединения с любым другим

оборудованием. Этот процесс может периодически повторяться, например, при включении питания оборудования, поэтому привратник должен уметь обрабатывать множество запросов регистрации от одного и того же оборудования.

Если привратник получает запрос RRQ, содержащий те же самые alias-адрес и транспортный адреса окончного оборудования, что и в предыдущем RRQ, он должен ответить подтверждением RCF. Если привратник получает запрос RRQ с тем же, что и в предыдущем RRQ, alias-адресом, но с другим транспортным адресом, он может либо подтвердить регистрацию, либо отказать в ней, в зависимости от внутренней политики сети. При приеме запроса RRQ, содержащего тот же, что и предыдущий RRQ, транспортный адрес, но другой alias-адрес оборудования, привратник должен закрепить за принятым транспортным адресом тот alias-адрес, который был принят последним, и подтвердить запрос. Заметим, что привратник может проверять наличие права пользователей на проведение вышеуказанных изменений.

Оконечное оборудование может регистрироваться на определенный промежуток времени, указывая в параметре **timeToLive** сообщения RRQ длительность этого промежутка в секундах. Привратник может подтвердить регистрацию сообщением RCF с параметром **timeToLive**, имеющим то же или меньшее значение.

В течение указанного промежутка времени окончное оборудование может продлить регистрацию, передав сообщение RRQ с параметром **keepAlive**. Получив это сообщение, привратник должен перезапустить таймер.

По истечении назначенного промежутка времени регистрация считается недействительной. В этом случае привратник может передать сообщение об отмене регистрации, и окончное оборудование должно пройти повторную регистрацию.

Оконечное оборудование может отменить регистрацию у привратника, передав сообщение Unregister Request (URQ); привратник должен ответить подтверждением Unregister Confirmation (UCF). Такая процедура позволяет оборудованию изменить свой alias-адрес или транспортный адрес. Если оборудование не было зарегистрировано у привратника, последний должен ответить на требование URQ отказом Unregister Reject (URJ).

Привратник может отменить регистрацию оборудования, передав сообщение Unregister Request (URQ), при получении которого окончное оборудование должно ответить подтверждением Unregister Confirmation (UCF). Теперь, чтобы получить возможность участия в любом соединении, окончное оборудование должно перерегистрироваться у того же привратника или зарегистрироваться у нового.

Оборудование, не зарегистрированное у привратника, не может требовать от него допуск к участию в любых соединениях. Привратник не выполняет для этого оборудования такие функции как управление полосой пропускания, преобразование адресов и другие предусмотренные рекомендацией Н.323 функции. Кроме того, привратник может запретить окончному оборудованию своей зоны принимать вызовы от оборудования, которое у него не зарегистрировано.

6.2.3 Доступ к сетевым ресурсам

В начальной фазе установления соединения, а также после получения запроса соединения (сообщения Setup), оборудование обращается к привратнику при помощи запроса Admission Request (ARQ) с просьбой разрешить соединение с другим оборудованием (рис. 6.4), что является началом процедуры доступа к сетевым ресурсам. Важно отметить, что процедура доступа выполняется всеми участниками соединения.

В сообщении ARQ обязательно содержится идентификатор оборудования, пославшего сообщение ARQ, и контактная информация того оборудования, с которым желает связаться оборудование, пославшее сообщение ARQ. Контактная информация оборудования включает в себя alias-адрес и/или транспортный адрес сигнального канала, но, как правило, в запрос ARQ помещается только alias-адрес вызываемого оборудования.

В сообщении ARQ указывается также верхний предел суммарной скорости передачи и приема пользовательской информации по всем речевым и видеоканалам без учета заголовков RTP/UDP/IP и другой служебной информации. Во время связи средняя за секунду суммарная скорость передачи и приема информации окончным оборудованием не должна превышать этот верхний предел. Отметим, что суммарная скорость не включает в себя скорость передачи и приема информации по каналу передачи данных, по управляющему и сигнальному каналам.

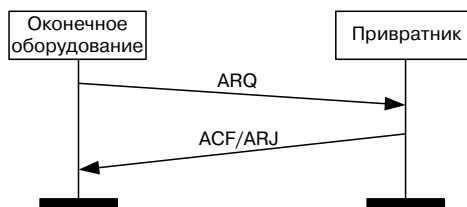


Рис. 6.4 Управление доступом к сетевым ресурсам

Как показано в примере на рис.6.4, привратник может выделить требуемую полосу пропускания или снизить предел суммарной скорости, передав сообщение Admission Confirm (ACF). В этом же сооб-

щении, кроме суммарной скорости, указывается транспортный адрес сигнального канала встречного оборудования, если сигнальный канал будет организован непосредственно между тем и другим оборудованием, или адрес привратника, если он будет маршрутизировать сигнальные сообщения.

Если процедура доступа инициируется вызывающим оборудованием, то после получения ответа ACF, на указанный в этом сообщении адрес передается сообщение Setup и делается попытка установить сигнальное соединение H.225.0. Следует отметить, что инициирование процедуры доступа к сетевым ресурсам вызываемым оборудованием начинается уже после установления сигнального канала и получения по нему сообщения Setup.

Если требуемая полоса недоступна, привратник передает сообщение Admission Reject (ARJ).

6.2.4 Определение местоположения оборудования в сети

Оконечное оборудование или привратник, которые имеют alias-адрес некоторого оборудования и желают узнать его контактную информацию (адреса сигнального канала и канала RAS), могут послать запрос Location Request (LRQ) по адресу канала RAS отдельного привратника или по общему адресу всех привратников (режим Gatekeeper's Discovery Multicast). Привратник, у которого зарегистрировано указанное оборудование, должен ответить сообщением Location Confirmation (LCF), содержащим требуемую контактную информацию. Эта процедура называется определением местоположения окончного оборудования в сети (рис.6.5).

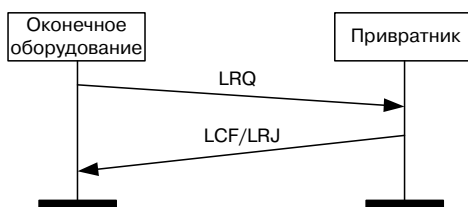


Рис. 6.5 Определение местоположения оборудования в сети

Привратник, получивший на транспортный адрес своего канала RAS запрос LRQ, должен ответить отказом Location Reject (LRJ), если искомое оборудование у него не зарегистрировано. Те же привратники, у которых искомое оборудование не зарегистрировано, а сообщение LRQ было получено в режиме многоадресной рассылки Gatekeeper's Discovery Multicast, вообще не должны отвечать на запрос.

Вышеописанная процедура используется, в частности, тогда, когда в сети имеется несколько зон и вызов выходит за пределы одной

зоны. Привратник, у которого зарегистрировано вызывающее оборудование, передает запрос адреса сигнального канала вызываемого оборудования.

Кроме того, оконечное оборудование или привратник могут передавать в поле **destinationInfo** запроса LRQ номер абонента ТФОП в формате E.164 с целью определить местонахождение шлюза, посредством которого может быть установлено соединение.

6.2.5 Изменение полосы пропускания

В процессе обслуживания вызова оконечное оборудование или привратник могут предпринять попытку изменить в ту или иную сторону суммарную скорость передачи информации. Данная процедура называется изменением полосы пропускания.

Оконечное оборудование может изменять суммарную скорость, не обращая за разрешением к привратнику, если после этого изменения средняя суммарная скорость не превысит предела, определенного при получении доступа к сетевым ресурсам.

Оконечное оборудование, которому нужно превысить указанный предел, должно передать привратнику запрос Bandwidth Change Request (BRQ), но до получения ответа средняя суммарная скорость должна быть не выше этого предела. Если привратник может выделить требуемую полосу пропускания, он отвечает сообщением Bandwidth Change Confirm (BCF). Далее речевые и видеоканалы закрываются, а затем при помощи управляющих сообщений H.245 открываются каналы с новой скоростью передачи и приема информации. Если же привратник по каким-либо причинам не может удовлетворить требование оборудования, он отклоняет это требование и передает сообщение Bandwidth Change Reject (BRJ). Сценарий процедуры представлен на рис.6.6.

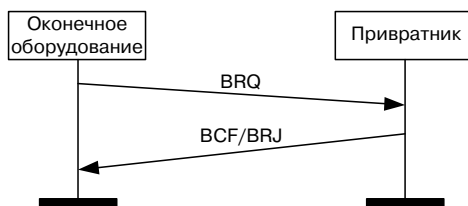


Рис. 6.6 Изменение полосы пропускания в процессе обслуживания вызова

В процессе обслуживания вызова привратник может изменить в ту или иную сторону выделенную оборудованию полосу пропускания, передав сообщение BRQ. Если это требование предписывает снизить скорость, оконечное оборудование обязано подчиниться, т.е. передать подтверждение BCF и переустановить логические каналы.

Если сообщением BRQ привратник предлагает увеличить скорость, то решение принять или не принимать это предложение остается за окончательным оборудованием.

6.2.6 Опрос текущего состояния оборудования

Привратник в любой момент времени может определить текущее состояние оборудования, т.е. установить, доступно ли ему это оборудование. Данный процесс называется опросом текущего состояния оборудования (рис. 6.7). Очевидно, что если питание оборудования выключено, или если в его работе возникла какая-либо неисправность, то оборудование становится недоступным.

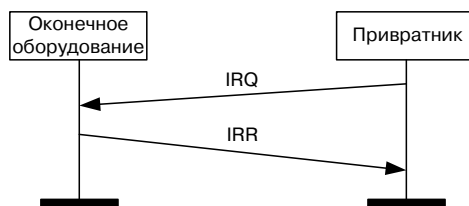


Рис. 6.7 Опрос текущего состояния оборудования

Запрос информации о текущем состоянии (статусе) оборудования производится привратником при помощи сообщения Information Request (IRQ). Интервал между посылками IRQ оставлен на усмотрение производителя, но должен быть не меньше 10 с. Получив запрос IRQ, окончательное оборудование должно передать запрашиваемую информацию в сообщении Information Request Response (IRR).

Привратник может дать окончательному оборудованию предписание передавать сообщения IRR без запросов с его стороны. Для этого привратник использует сообщение ACF, в поле **irrFrequency** которого указывается частота, с какой окончательное оборудование должно выдавать информацию о своем текущем состоянии. Получив такое предписание, окончательное оборудование должно передавать сообщения IRR с указанной частотой в течение всего времени обслуживания вызова, причем привратник может запрашивать дополнительную информацию, используя сообщения IRQ, как было описано выше.

Оконечное оборудование, желающее убедиться в том, что сообщения IRR, посылаемые без предварительных запросов со стороны привратника, достигают адресата, может требовать от привратника подтверждений получения сообщений IRR. Наличие поля **willRespondToIRR** в сообщениях RCF или ACF, получаемых от привратника, означает его согласие удовлетворить данное требование. Привратник может подтверждать получение сообщения IRR сообщением IACK или сообщать о потере или задержке сообщения IRR с помощью сообщения INAK. Оба сообщения IACK и INAK используются,

когда сообщения IRR переданы (привратникам версии 2 или выше) с полем `needResponse`, которому присвоено значение `TRUE`.

Существует еще один вариант использования сообщений IRR. Привратник может потребовать от оконечного оборудования присылать копии всех или некоторых сигнальных сообщений, передаваемых и принимаемых этим оборудованием. Если оборудование может удовлетворить данное требование, оно передает запрашиваемую информацию в сообщениях IRR сразу же после того, как получит или отправит сигнальное сообщение.

6.2.7 Освобождение полосы пропускания

Как уже упоминалось ранее, процедура завершения соединения выглядит следующим образом: сначала закрываются логические каналы, затем управляющий и сигнальный каналы. В конечной фазе завершения соединения оборудование извещает привратник об освобождении ранее занимавшейся полосы пропускания (рис.6.8). Оконечное оборудование передает своему привратнику сообщение `Disengage Request (DRQ)`, на которое тот должен ответить подтверждением `Disengage Confirm (DCF)`. Следует отметить, что после того, как полоса пропускания освобождена, оконечное оборудование не должно передавать незапрашиваемые сообщения IRR.

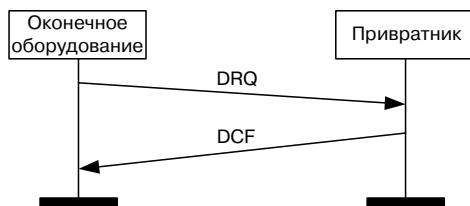


Рис. 6.8 Освобождение полосы пропускания

Привратник может сам инициировать освобождение сетевых ресурсов, т.е. разрушение существующего соединения, передав сообщение `DRQ`. Получив сообщение `DRQ`, оконечное оборудование должно закрыть логические каналы, управляющий и сигнальный каналы, а затем ответить подтверждением `DCF`.

В случае, если привратник инициирует завершение конференции, сообщение `DRQ` должно передаваться каждому ее участнику.

6.2.8 Метка доступа

Метка доступа передается в некоторых сообщениях сигнализации `RAS` и в сообщении `Setup`, причем имеются два основных варианта ее использования.

Первый вариант служит для сокрытия транспортного адреса и `alias`-адреса оконечного оборудования. Пользователь, желающий со-

хранить в тайне свои адреса, сообщает каким-либо образом вызывающему пользователю метку доступа, о наличии которой привратник заранее оповещен в процессе регистрации. Вызывающий абонент использует метку доступа для установления соединения с вызываемым абонентом, причем сигнальные каналы непременно должны проходить через привратник, который маршрутизирует сигнальные сообщения от одного абонента к другому.

Во втором варианте использования метки доступа она назначается привратником и должна передаваться во всех сообщениях, служащих для установления соединения. Примером такого использования метки доступа может служить установление соединения со шлюзом. По наличию метки шлюз определяет, что устанавливать соединение с его участием абоненту разрешено.

В заключение этого параграфа приведем итоговую таблицу (табл. 6.1) сообщений протокола RAS, рассмотренных выше. В этой и следующих аналогичных таблицах для удобства читателей, работающих также с рекомендациями ITU-T, используются следующие обозначения: О (options) – необязательное, М (mandatory) – обязательное.

Таблица 6.1 Сообщения RAS

Сообщение RAS	Передача оконечным оборудованием	Приём оконечным оборудованием	Передача привратником	Приём привратником	Примечания
GRQ	О			М	<i>Gatekeeper Request</i> (Запрос привратника) Любой привратник, принявший это сообщение, должен на него ответить
GCF		О	М		<i>Gatekeeper Confirm</i> (Подтверждение привратника) Привратник идентифицирует себя
GRJ		О	М		<i>Gatekeeper Reject</i> (Отказ привратника) Указывается причина
RRQ	М			М	<i>Registration Request</i> (Запрос регистрации)
RCF		М	М		<i>Registration Confirm</i> (Подтверждение регистрации)
RRJ		М	М		<i>Registration Reject</i> (Отказ в регистрации) Указывается причина
URQ	О	М	О	М	<i>Unregistration Request</i> (Запрос отмены регистрации) Терминал желает отменить регистрацию у привратника

UCF	M	O	M	O	<i>Unregistration Confirm</i> (Регистрация отменена)
URJ	O	O	M	O	<i>Unregistration Reject</i> (Отказ в отмене регистрации) Указывается причина.
ARQ	M			M	<i>Admission Request</i> (Запрос доступа)
ACF		M	M		<i>Admission Confirm</i> (Подтверждение доступа)
ARJ		M	M		<i>Admission Reject</i> (Отказ в доступе) Указывается причина
BRQ	M	M	O	M	<i>Bandwidth Request</i> (Запрос изменения полосы пропускания)
BCF	M	M	M	O	<i>Bandwidth Confirm</i> (Подтверждение изменения полосы пропускания)
BRJ	M	M	M	O	<i>Bandwidth Reject</i> (Отказ в предоставлении полосы) Указывается причина
IRQ		M	M		<i>Information Request</i> (Запрос информации)
IRR	M			M	<i>Information Response</i> (Ответ на запрос информации)
IACK		O	Условное		<i>InfoRequestAck</i> (Подтверждение получения сообщения IRR)
INAK		O	Условное		<i>InfoRequestNak</i> (Индикация потери или задержки сообщения IRR)
DRQ	M	M	O	M	<i>Disengage Request</i> (Запрос разъединения). Информирует привратник, что оконечное оборудование освобождает ранее занимавшуюся полосу пропускания, или оборудование о том, что ему необходимо освободить занимаемую полосу пропускания
DCF	M	M	M	M	<i>Disengage Confirm</i> (Подтверждение получения сообщения DRQ)
DRJ	M	M	M	M	<i>Disengage Reject</i> (Отклонение запроса/разъединения) Передаётся привратником, если оконечное оборудование не было зарегистрировано у данного привратника

LRQ	O		O	M	<i>Location Request</i> (Запрос местоположения) Запрос предоставления транспортного адреса оконечного оборудования
LCF		O	M	O	<i>Location Confirm</i> (Сообщение о местоположе- нии оборудования) Сообщается транспортный адрес искомого оконечного оборудования
LRJ		O	M	O	<i>Location Reject</i> (Отказ дать сведения о местоположении оборудования) Указывается причина, вероятнее всего - "искмое оборудование не зарегистрировано у привратника"
NSM	O	O	O	O	<i>Non-Standard Message</i> (Нестандартное сообщение) Передаются данные, не специфицированные в рекомендации H.225.0
XRS	M	M	M	M	<i>Unknown Message Response</i> (Ответ на неизвестное сообщение) Передаётся оконечным оборудованием всякий раз, когда оно получает нераспознанное сообщение
RIP	Условное	M	Условное	M	<i>Request in Progress</i> (Запрос обрабатывается) Передаётся оконечным оборудованием, если ответ на сообщение не может быть послан до срабатывания таймера RAS
RAI	O			M	<i>Resource Availability Indication</i> (Индикация доступности ресурсов) Передаётся шлюзом привратнику, чтобы уведомить его о ресурсе шлюза для каждого из протоколов серии H и о соответствующих скоростях передачи
RAC		O	M		<i>Resource Availability Confirm</i> (Подтверждение индикации доступности ресурсов) Ответ привратника на сообщение RAI

6.3 Сигнальный канал Н.225.0

Процедуры управления соединениями в сетях Н.323 специфицированы Международным союзом электросвязи в рекомендации Н.225.0. Данные процедуры предусматривают использование в базовом процессе обслуживания вызова ряда сигнальных сообщений Q.931 [7], причем должен быть реализован симметричный обмен сигнальными сообщениями в соответствии с приложением D к рекомендации Q.931. Это требование не распространяется на взаимодействие шлюза с сетью коммутации каналов.

Для реализации дополнительных услуг в соответствии с рекомендацией Н.450 в сетях, построенных по рекомендации Н.323, привлекаются сигнальные сообщения Q.932. В данном параграфе рассматриваются наиболее часто используемые сигнальные сообщения.

Сообщение *Setup* передается вызывающим оборудованием с целью установить соединение. Это сообщение передается на общеизвестный TCP порт 1720 вызываемого оборудования (см. рис.1.4 главы 1).

Сообщение *Call Proceeding* передается вызываемому оборудованию, чтобы известить его о том, что вызов принят к обслуживанию.

Сообщение *Alerting* передается вызываемому оборудованию и информирует его о том, что вызываемое оборудование не занято, и что пользователю подается сигнал о входящем вызове.

Сообщение *Connect* передается вызываемому оборудованию и информирует его о том, что вызываемый пользователь принял входящий вызов. Сообщение *Connect* может содержать транспортный адрес управляющего канала Н.245.

Сообщение *Release Complete* передается вызывающим или вызываемым оборудованием с целью завершить соединение. Это сообщение передается только в том случае, когда открыт сигнальный канал.

Сообщение Q.932 *Facility* используется для обращения к дополнительным услугам в соответствии с Рекомендациями ITU Н.450.х.

Транспортировку сигнальных сообщений обеспечивает протокол с установлением соединения и с гарантированной доставкой информации – Transport Control Protocol (TCP). В соответствии с первой и второй версиями рекомендации Н.323 для каждого нового вызова открывается отдельный сигнальный канал. Начиная с третьей версии рекомендации Н.323, один сигнальный канал Н.225.0 может переносить сообщения, относящиеся к разным вызовам и имеющие разные метки соединения (call reference). Наличие такой возможности позволяет значительно уменьшить время установления соединения с участием шлюзов и объем передаваемой служебной информации.

Оборудование, поддерживающее управление множеством сигнальных соединений в одном сигнальном канале, присваивает в сигнальных сообщениях значение **TRUE** информационному полю **multipleCalls**. Оборудование может ограничивать количество сигнальных соединений, использующих один сигнальный канал, назначая определенный порог. Если этот порог достигнут, оборудование передает отказ в попытке установить соединение – сообщение **Release Complete** – с указанием причины **newConnectionNeeded** (требуется открыть новый сигнальный канал).

Кроме того, в версии 3 рекомендации H.323 говорится о том, что сигнальный канал H.225.0 может быть организован перед тем, как по нему потребуется передавать сигнальную информацию, и оставаться открытым после завершения соединения. Оборудование, поддерживающее постоянно открытый сигнальный канал, должно присваивать в сигнальных сообщениях значение **TRUE** информационному полю **maintainConnection**. Желательно также указывать на эту возможность при регистрации у привратника, что позволит привратнику (в случае маршрутизации им сигнальной информации) подключаться к оборудованию в любой момент после регистрации.

В сетях, не имеющих привратника, открывается сигнальный канал H.225.0, непосредственно связывающий вызывающее оконечное оборудование с вызываемым. В этом случае вызывающий пользователь должен знать транспортный адрес сигнального канала (**Call Signalling Transport Address**) оборудования вызываемого пользователя.

В сетях с привратником вызывающее оборудование передает по транспортному адресу канала RAS привратника сообщение ARQ с указанием **alias**-адреса вызываемого пользователя. Если сигнальные сообщения будет маршрутизировать привратник (**Gatekeeper Routed Call Signalling**), то в ответном сообщении он передает транспортный адрес своего сигнального канала, что представлено на рис. 6.9. Если же сигнальный канал будет, согласно рис. 6.10, устанавливаться непосредственно между вызывающим и вызываемым оборудованием (**Direct Endpoint Call Signalling**), то передается транспортный адрес сигнального канала вызываемого оборудования. Выбор варианта передачи сигнальных сообщений оставлен за привратником, хотя оконечное оборудование может указывать, какой вариант для него предпочтителен. И в первом, и во втором случае сигнальный канал H.225 выполняет одни и те же функции и переносит одни и те же сообщения.

При маршрутизации сигнальных сообщений привратником сигнальный канал может закрываться сразу после установления соединения или оставаться открытым в течение всего соединения для предоставления дополнительных услуг. Закрывать сигнальный канал мо-

жет только привратник, но если в соединении участвует шлюз, то сигнальный канал должен оставаться открытым до окончания соединения. При закрытии сигнального канала окончательным оборудованием должно сохраняться текущее состояние соединения. Привратник может в любой момент соединения снова открыть сигнальный канал.

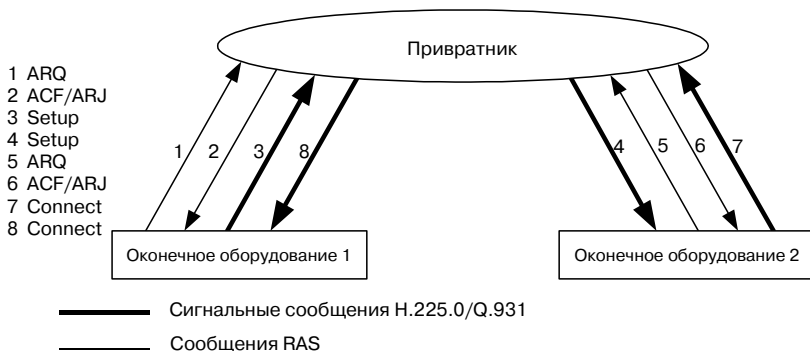


Рис. 6.9 Маршрутизация сигнальной информации привратником

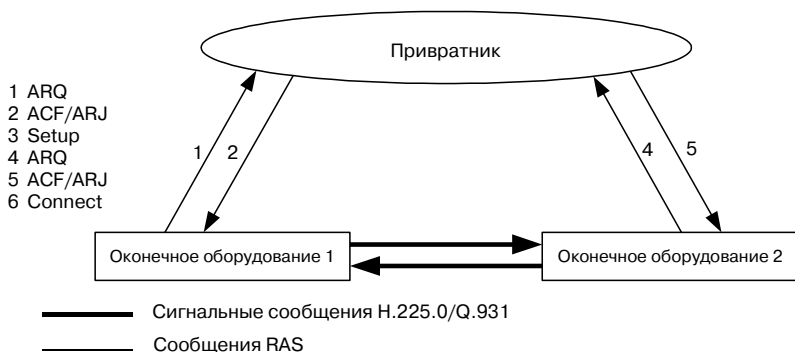


Рис. 6.10 Передача сигнальной информации напрямую

После обмена с привратником сообщениями ARQ и ACF по каналу RAS вызывающее оборудование передает запрос соединения Setup либо по транспортному адресу сигнального канала привратника (если сигнальные сообщения будут маршрутизировать привратник), либо по транспортному адресу сигнального канала вызываемого оборудования (если сигнальный канал будет связывать вызывающее и вызываемое оборудование непосредственно). В ответ на сообщение Setup вызываемое оборудование может передать сообщение Call Proceeding, означающее, что вся информация, необходимая для установления соединения, получена, и вызов принят к обслуживанию. Далее от вызываемого оборудования может поступить сообщение

Alerting, означающее, что вызываемому пользователю подается вызывной сигнал. После того как пользователь принимает вызов, вызывающему оборудованию передается сообщение Connect с транспортным адресом управляющего канала Н.245 вызываемого оборудования, если управляющий канал связывает вызывающее и вызываемое оборудование напрямую (рис.6.11), или транспортный адрес канала Н.245 привратника, если управляющие сообщения маршрутизирует привратник (рис.6.12). В некоторых случаях, например, для проключения разговорных каналов в предответном состоянии, транспортный адрес управляющего канала Н.245 включается в сообщения Call Proceeding или Alerting.

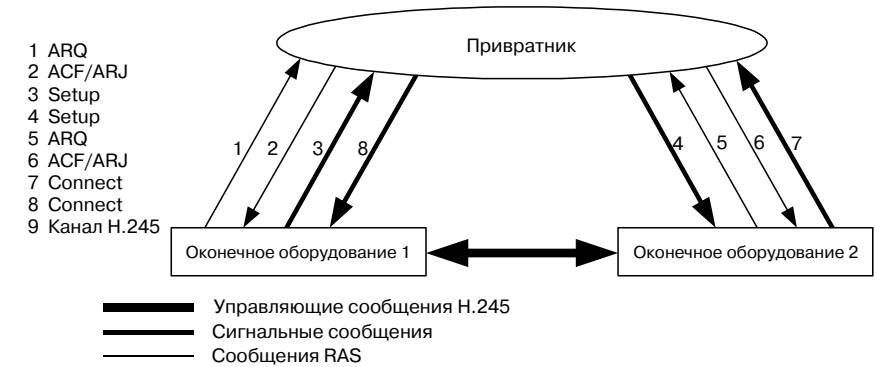


Рис. 6.11 Управляющий канал, связывающий оконечное оборудование напрямую

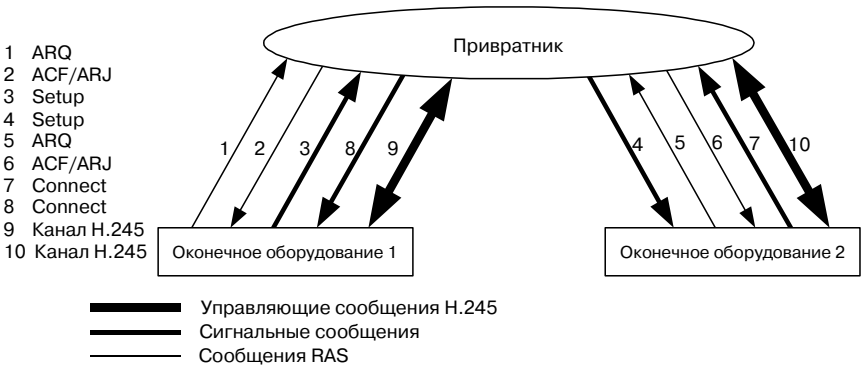


Рис. 6.12 Маршрутизация управляющих сообщений привратником

И, по аналогии с рассмотрением в предыдущем параграфе протокола RAS, завершим краткое описание сигнального канала Н.225.0 перечнем сообщений, сведенным в таблицу 6.2.

Таблица 6.2 Сообщения протокола Н.225.0

Сообщение Q.931/Q.932	Передача	Прием
Alerting (Аналог "КПВ")	М	М
Call Proceeding (Соединение устанавливается)	О	Условное
Connect (Соединение установлено)	М	М
Connect Acknowledge (Подтверждение установления соединения)	Не разрешено	Не разрешено
Progress (Особенности маршрута)	О	О
Setup (Запрос соединения)	М	М
Setup Acknowledge (Подтверждение приема запроса Setup)	О	О
Disconnect (Разъединение)	Не разрешено	Не разрешено
Release (Освободить ресурсы)	Не разрешено	Не разрешено
Release Complete (Ресурсы освобождены)	М	М
Resume (Возобновить соединение)	Не разрешено	Не разрешено
Resume Acknowledge (Соединение возобновлено)	Не разрешено	Не разрешено
Resume Reject (Отказ возобновить соединение)	Не разрешено	Не разрешено
Suspend (Прервать соединение)	Не разрешено	Не разрешено
Suspend Acknowledge (Соединение прервано)	Не разрешено	Не разрешено
Suspend Reject (Отказ прервать соединение)	Не разрешено	Не разрешено
User Information (Информация пользователя)	О	О
Congestion Control (Управление потоком сообщений USER INFORMATION)	Не разрешено	Не разрешено
Information (Информация)	О	О
Notify (Уведомление)	О	О
Status (Статус)	М	М
Status Inquiry (Запрос статуса)	О	М
Facility (Дополнительная услуга)	М	М
Hold (Удерживать соединение)	Не разрешено	Не разрешено
Hold Acknowledge (Соединение удерживается)	Не разрешено	Не разрешено
Hold Reject (Отказ удерживать соединение)	Не разрешено	Не разрешено
Retrieve (Снять с удержания)	Не разрешено	Не разрешено
Retrieve Acknowledge (С удержания снято)	Не разрешено	Не разрешено
Retrieve Reject (Отказ снять с удержания)	Не разрешено	Не разрешено

6.4 Управляющий канал Н.245

Ранее в книге уже упоминалось, что в рекомендации ITU-T Н.245 определен ряд независимых процедур, которые должны выполняться для управления информационными каналами. К ним относятся процедуры:

- определения ведущего и ведомого устройств (Master/slave determination);
- обмена данными о функциональных возможностях (Capability Exchange);
- открытия и закрытия однонаправленных логических каналов (Logical Channel Signalling);
- открытия и закрытия двунаправленных логических каналов (Bidirectional Logical Channel Signalling);
- закрытия логических каналов (Close Logical Channel Signalling);
- определения задержки, возникающей при передаче информации от источника к приемнику и в обратном направлении (Round Trip Delay Determination);
- выбора режима обработки информации (Mode Request);
- сигнализации по петле, создаваемой для целей технического обслуживания оборудования (Maintenance Loop Signalling).

Для выполнения вышеуказанных процедур между оконечными устройствами или между оконечным оборудованием и устройством управления конференциями или привратником организуется управляющий канал Н.245. При этом оконечное оборудование должно открывать один (и только один) управляющий канал для каждого соединения, в котором оно участвует. Примечательно, что терминалы, устройства управления конференциями, шлюзы и привратники могут участвовать одновременно в нескольких соединениях и, следовательно, открывать несколько управляющих каналов.

Перенос управляющей информации Н.245 осуществляется протоколом TCP по нулевому логическому каналу, который должен быть постоянно открытым с момента организации канала Н.245 и вплоть до его ликвидации. Следует отметить, что нормальные процедуры открытия и закрытия логических каналов, описываемые в этой главе, для управления нулевым логическим каналом не применяются.

По управляющему каналу Н.245 передаются сообщения четырех категорий: запросы, ответы, команды и индикации. Получив сообщение-запрос, оборудование должно выполнить определенное действие и немедленно передать обратно сообщение-ответ. Получив сообщение-команду, оборудование также должно выполнить определенное действие, но отвечать на команду не должно. Сообще-

ние-индикация служит для того, чтобы информировать о чем-либо получателя, но не требует от него ни ответа, ни каких бы то ни было действий.

Ниже в этом параграфе дается краткое описание основных процедур Н.245, выполняемых в процессе управления логическими каналами.

6.4.1 Определение ведущего и ведомого

Процедура определения ведущего и ведомого оборудования используется для разрешения конфликтов, возникающих между двумя устройствами при организации конференции, когда ведущим в ней может быть любое из этих устройств, или между двумя устройствами, которые одновременно пытаются открыть двунаправленный логический канал. Устройства обмениваются сообщениями **masterSlaveDetermination** (рис.6.13), в поле **terminalType** которых помещается значение, соответствующее типу данного оборудования (таблица 6.3), а в поле **statusDeterminationNumber** – случайное число из интервала $[0 - (2^{24}-1)]$. Ведущим становится оборудование, поместившее большее число в поле **terminalType**, а при совпадении типов оборудования – большее число в поле **statusDeterminationNumber**.

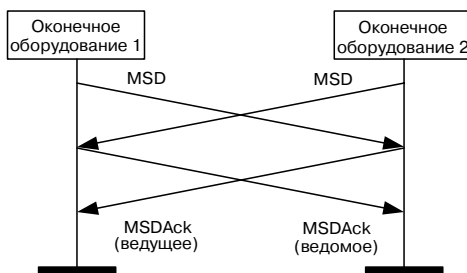


Рис. 6.13 Первый вариант определения ведущего и ведомого оборудования

В ответ на полученные сообщения **masterSlaveDetermination** оба устройства передают сообщения **masterSlaveDeterminationAck**, в которых указывается, какое оборудование является для данного соединения ведущим, а какое – ведомым. При этом любое оборудование стандарта Н.323 должно быть способно работать и в качестве ведущего, и в качестве ведомого.

Следует отметить, что активный МС в конференции должен использовать значение 240. При этом в конференции может быть только один активный контроллер. В ходе конференции активный контроллер не должен меняться.

Таблица 6.3 Значения поля TerminalType для разных типов оборудования

Тип оборудования стандарта H.323	Значение в поле TerminalType			
	Терминал	Шлюз	Привратник	MCU
Оборудование, не содержащее MC	50	60	–	–
Оборудование, содержащее MC, но без MP	70	80	120	160
Оборудование, содержащее MC и MP для данных	–	90	130	170
Оборудование, содержащее MC и MP для данных и речи	–	100	140	180
Оборудование, содержащее MC и MP для данных, для речи и для видеoinформации	–	110	150	190

Существует вариант процедуры Master-Slave Determination, предусматривающий сокращение числа передаваемых сообщений (рис.6.14).

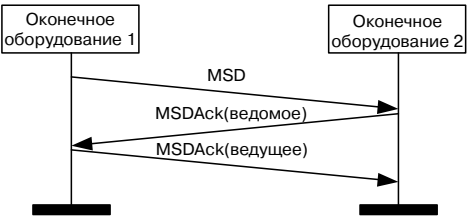


Рис. 6.14 Второй вариант определения ведущего и ведомого оборудования

В этом варианте оборудование, передавшее сообщение **masterSlaveDetermination** и получившее в ответ сообщение **masterSlaveDeterminationAck**, передает сообщение **masterSlaveDeterminationAck**.

6.4.2 Обмен данными о функциональных возможностях

Оборудование стандарта H.323, в общем случае, способно принимать и передавать речь, видеoinформацию и данные. Это означает, что оборудование обычно содержит приемник и передатчик информации. Как правило, устройства поддерживают несколько алгоритмов кодирования и декодирования информации каждого вида, которые подробно обсуждались в главе 3. Для согласования режимов работы передающей и принимающей сторон используется процедура, называемая обменом данными о функциональных возможностях оборудования (рис.6.15).

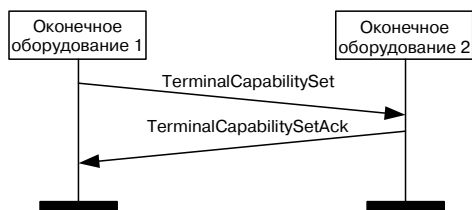


Рис. 6.15 Обмен данными о функциональных возможностях оборудования

Терминалы обмениваются сообщениями **TerminalCapabilitySet**, в которых каждый из них указывает алгоритмы, используемые для декодирования принимаемой и кодирования передаваемой информации, то есть режимы, в которых оборудование может функционировать.

Следует подчеркнуть, что оборудование должно указывать поддерживаемые им алгоритмы декодирования принимаемой информации, а передающая сторона должна использовать для кодирования передаваемой информации только те кодеки, которые имеет принимающая сторона. Оборудование, которое не указывает алгоритмы, используемые им для декодирования принимаемой информации, может только передавать информацию.

Кроме того, оборудование может указывать режимы, которые оно поддерживает при передаче информации, и предоставлять возможность выбора режима приемной стороне. Оборудование, не указывающее алгоритмы, используемые для кодирования передаваемой информации, не оставляет возможности выбора принимающей стороне, но оно может передавать информацию, кодируя ее в соответствии с любым из алгоритмов, поддерживаемых приемной стороной. Таким образом, алгоритмы, которые используются для кодирования передаваемой информации, указывать не обязательно, и в существующих продуктах IP-телефонии, реализованных на базе Н.323, для речи и видеоинформации обычно указываются только алгоритмы, которые используются для декодирования принимаемой информации.

В сообщении **TerminalCapabilitySet** включается поле **capabilityTable** – таблица функциональных возможностей, где каждому алгоритму кодирования/декодирования присвоен порядковый номер. Например, возможности приема речевой информации, закодированной по алгоритму G.723.1, соответствует номер 1, возможности приема речевой информации, закодированной по алгоритму G.728, – номер 2, возможности приема видеосигналов, закодированных по алгоритму H.263, – номер 3 и т. д.

Указанные порядковые номера объединяются в список альтернативных режимов **alternativeCapabilitySet**. Оборудование может ис-

пользовать любой (но только один) из режимов, указанных в списке. Например, список альтернативных режимов {G.711, G.723.1, G.728} означает, что оборудование может функционировать в любом из указанных режимов обработки речи, но только в одном.

В свою очередь, альтернативные режимы объединяются в наборы одновременно возможных режимов функционирования **simultaneousCapabilities**. Например, набор одновременно возможных режимов, содержащий список альтернативных режимов обработки видеоинформации {H.261, H.263} и список альтернативных режимов обработки речевой информации {G.711, G.723.1, G.728}, означает, что оборудование может использовать любой из указанных алгоритмов кодирования видеоинформации совместно с любым из списка алгоритмов кодирования речевой информации.

Другой пример: набор одновременно возможных режимов, содержащий два списка альтернативных режимов обработки видеоинформации {H.261}, {H.261, H.263} и один список альтернативных режимов обработки аудиоинформации {G.711, G.723.1, G.728}, означает, что оборудование может одновременно использовать два алгоритма кодирования видеоинформации (первый – H.261, второй – H.261 или H.263) и один алгоритм декодирования речи (либо G.711, либо G.723.1, либо G.728).

Функциональные возможности терминала описываются набором дескрипторов (**capabilityDescriptor**), каждый из которых состоит из одного набора одновременно возможных режимов функционирования оборудования и номера дескриптора (**capabilityDescriptorNumber**). Если при обмене данными о функциональных возможностях оборудование указывает более чем один дескриптор, то это означает, что оборудование поддерживает несколько режимов функционирования. Например, наличие в сообщении **TerminalCapabilitySet** двух дескрипторов: первого – как и в предыдущем примере, т.е. {H.261, H.263} и {G.711, G.723.1, G.728}, а второго – {H.262} и {G.711}, означает, что оборудование, кроме описанного выше режима, поддерживает обработку видеоинформации, закодированной по алгоритму H.262, совместно с обработкой речи, закодированной по менее сложному, по сравнению с остальными, алгоритму кодирования G.711.

Заметим, что функциональные возможности оборудования, не определенные рекомендацией ITU H.245, могут быть указаны в поле **nonStandardParameter**.

Оборудование может в любое время передать сообщение **TerminalCapabilitySet** с дескриптором, добавляющим новые функциональные возможности, или с дескриптором, обеспечивающим исключение некоторых из ранее указанных возможностей. Любое оборудование стандарта H.323 должно включать в сообщение **Terminal-**

CapabilitySet, по крайней мере, один дескриптор. Исключение составляет сообщение **EmptyCapabilitySet** (пустой набор функциональных возможностей), которое используется для реализации дополнительных возможностей системы.

Оборудование, которое получило от другого оборудования сообщение **TerminalCapabilitySet**, может подтвердить его получение передачей сообщения **TerminalCapabilitySetAck**.

При получении сообщения с некорректным набором возможностей оборудование отвечает сообщением **TerminalCapabilitySetReject**. При срабатывании таймера, запущенного после отправки сообщения **TerminalCapabilitySet**, оборудование, его пославшее, передает сообщение **TerminalCapabilitySetRelease**.

6.4.3 Открытие и закрытие логических каналов

Информация, передаваемая источником к одному или более приемникам в сетях, базирующихся на рекомендации Н.323, переносится по логическим каналам, которые идентифицируются уникальным для каждого направления передачи номером канала.

Рекомендацией Н.245 предусмотрена возможность открытия логических каналов двух видов: однонаправленных (uni-directional), т.е. открывающихся в направлении от источника к приемнику информации, и двунаправленных (bi-directional), открывающихся сразу в двух направлениях – от источника к приемнику информации и в обратном направлении.

Однонаправленные логические каналы открываются при помощи процедуры Uni-directional Logical Signalling (рис.6.16).

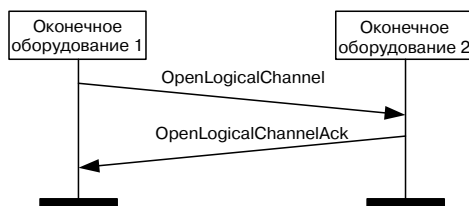


Рис. 6.16 Процедура открытия однонаправленных логических каналов

В требовании открыть логический канал **openLogicalChannel** оборудование указывает вид информации, который будет передаваться по этому каналу, и алгоритм кодирования информации. Если логический канал предназначается для переноса речи или видеoinформации, упакованной в пакеты рассмотренного в главе 4 протокола RTP (Real Time Protocol), то в сообщении **openLogicalChannel** должен включаться параметр **mediaControlChannel** с указанием транспорт-

ного адреса канала протокола RTCP (Real Time Control Protocol), при помощи которого ведется контроль передачи RTP пакетов.

Оборудование, получившее запрос открыть логический канал для приема данных, вид которых не поддерживается или не распознан, должно ответить сообщением **openLogicalChannelReject**. Получение корректного сообщения **openLogicalChannel** оборудование должно подтвердить сообщением **openLogicalChannelAck**.

Если логический канал открывается для переноса речи или видеоинформации, то принимающая сторона указывает в параметре **mediaTransportChannel** сообщения **openLogicalChannelAck** транспортный адрес, на который передающая сторона должна передавать RTP пакеты, а в параметре **mediaControlChannel**, – транспортный адрес канала RTCP.

При открытии каналов для передачи данных, например для приложений T.120 параметр **mediaControlChannel** в сообщениях **openLogicalChannel** и **openLogicalChannelAck** отсутствует.

Когда оборудование открывает однонаправленный логический канал, то, чтобы организовать дуплексную связь, встречное оборудование также должно открыть однонаправленный канал в обратном направлении, используя для этого вышеописанную процедуру Uni-directional Logical Signalling.

Для передачи речи или видеоинформации, как правило, открывается однонаправленный канал от источника к приемнику информации и, независимо, канал в обратном направлении. Поэтому допускается асимметричный режим работы, когда в разных направлениях передачи открывается разное количество каналов и используются разные алгоритмы кодирования информации одного и того же вида.

Если приемная сторона способна работать только в симметричном режиме, она может указать на это ограничение при выполнении процедуры Capabilities exchange.

Следует отметить, что прямой и обратный каналы не должны иметь один и тот же номер, так как номера логических каналов присваиваются независимо для каждого направления передачи. Кроме того, для прямого и обратного логических каналов, относящихся к одной RTP-сессии и имеющих один и тот же идентификатор сессии (**sessionID**), открывается только один канал RTCP.

В некоторых случаях, например, для обмена данными по протоколу T.120, оборудование, иницирующее такой обмен, должно открывать сразу и прямой, и обратный каналы. Делается это с помощью процедуры Bi-directional Logical Signalling, которая практически идентична вышеописанной процедуре Uni-directional Logical Signalling и также предусматривает обмен сообщениями **openLogicalChan-**

nel и **openLogicalChannelAck**. Добавляется сообщение – **openLogicalChannelConfirm**, – которое передается в ответ на сообщение **openLogicalChannelAck** и подтверждает, что двунаправленный логический канал открыт (см. сценарий на рис.6.17). Заметим, что если процедура Uni-directional Logical Signalling для организации дуплексной связи должна выполняться два раза, то процедура Bi-directional Logical Signalling выполняется только один раз.

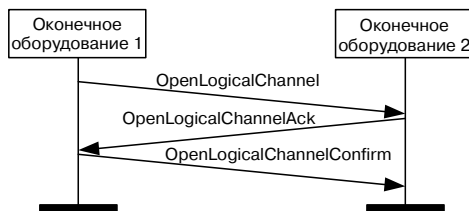


Рис. 6.17 Процедура открытия двунаправленного логического канала

Закрытие логических каналов может производиться с помощью процедуры **CloseLogicalChannel**, но она используется, в основном, для поддержки предоставления дополнительных услуг, в первую очередь, – перевода в режим удержания. Для нормального разрушения соединения стороны обмениваются сообщениями **endSessionCommand**. После обмена этими сообщениями закрываются не только логические каналы, но и управляющий канал Н.245.

6.4.4 Выбор режима обработки информации

Оконечное оборудование в ходе процедуры **Capabilities exchange** может объявить поддерживаемые им режимы передачи информации. Встречное оборудование, получив список возможных режимов передачи информации, может, передав сообщение **requestMode**, запросить передачу в одном из этих режимов. Устройство, получившее сообщение **requestMode**, должно, если это возможно, выполнить содержащееся в нем требование (рис.6.18). Оборудование, не желающее находиться под контролем другого оборудования в части выбора режима передачи информации, может просто не указывать, каким способом оно будет ее передавать.

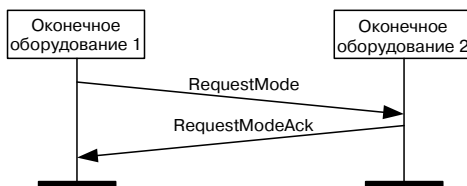


Рис. 6.18 Выбор режима обработки информации

Терминальное оборудование, участвующее в конференции и получившее от контроллера конференций сообщение **multipointModeCommand**, должно выполнять требования, содержащиеся в сообщениях **requestMode**, если эти требования не выходят за пределы возможностей оборудования. Примечательно, что в централизованных и децентрализованных конференциях, все сообщения **requestMode**, передаваемые терминалами, поступают на контроллер конференций, и он принимает решение, удовлетворить полученные требования или нет.

В таблице 6.4 приведены сообщения Н.245, которые оборудование стандарта Н.323 обязано принимать и передавать, а также необязательные и запрещенные сообщения Н.245. Приняв нераспознаваемое сообщение, оборудование Н.323 должно передать в ответ сообщение **functionNotSupported**. Следует отметить, что описание наиболее часто используемых сообщений было приведено в данном параграфе.

Таблица 6.4 Управляющие сообщения Н.245

Сообщения Н.245	Прием	Передача
Determination	М	М
Determination Acknowledge	М	М
Determination Reject	М	М
Determination Release	М	М
Capability Set	М	М
Capability Set Acknowledge	М	М
Capability Set Reject	М	М
Capability Set Release	М	М
Open Logical Channel	М	М
Open Logical Channel Acknowledge	М	М
Open Logical Channel Reject	М	М
Open Logical Channel Confirm	М	М
Close Logical Channel	М	М
Close Logical Channel Acknowledge	М	М
Request Channel Close	М	О
Request Channel Close Acknowledge	О	О
Request Channel Close Reject	О	М
Request Channel Close Release	О	М
Multiplex Entry Send	Не разрешено	Не разрешено
Multiplex Entry Send Acknowledge	Не разрешено	Не разрешено
Multiplex Entry Send Reject	Не разрешено	Не разрешено
Multiplex Entry Send Release	Не разрешено	Не разрешено

Request Mode	M	O
Request Mode Acknowledge	M	O
Request Mode Reject	O	M
Request Mode Release	O	M
Round Trip Delay Request	M	O
Round Trip Delay Response	O	M
Maintenance Loop Request		
System Loop	Не разрешено	Не разрешено
Media Loop	Обязательно только для шлюзов	
Logical Channel Loop	Не разрешено	Не разрешено
Maintenance Loop Acknowledge	O	O
Maintenance Loop Reject	O	M
Maintenance Loop Command Off	M	O
Terminal List Request	O	O
Drop Terminal	O	O
Make Me Chair	O	O
Cancel Make Me Chair	O	O
Enter H.243 Password	O	O
Enter H.243 Terminal Id	O	O
Enter H.243 Conference ID	O	O
Request Terminal ID	O	O
Terminal ID Response	O	O
MC Terminal ID Response	O	O
Enter Extension Address	O	O
Enter Address Response	O	O
Terminal List Response	O	O
Make Me Chair Response	O	O
Conference ID Response	O	O
Password Response	O	O
Send Terminal Capability Set	M	M
Encryption	O	O
Flow Control	M	O
End Session	M	M
Equalize Delay	O	O
Zero Delay	O	O
Multipoint Mode Command	M	O
Cancel Multipoint Mode Command	M	O
Video Freeze Picture	M	O
Video Fast Update Picture	M	O
Video Fast Update GOB	M	O
Video Fast Update MB	M	O
Video Temporal Spatial Trade Off	O	O

Video Send Sync Every GOB	O	O
Video Send Sync Every GOB Cancel	O	O
Terminal ID Request	O	O
Video Command Reject	O	O
Make Me Chair Response	O	O
Broadcast My Logical Channel Me	O	O
Cancel Broadcast My Logical Channel Me	O	O
Make Terminal Broadcaster	O	O
Cancel Make Terminal Broadcaster	O	O
Send This Source	O	O
Cancel Send This Source	O	O
Drop Conference	O	O
Communication Mode Command	M	O
Communication Mode Request	O	O
Communication Mode Response	O	O
Function Not Understood	M	M
Function Not Supported	M	M
Logical Channel Active	O	O
Logical Channel Inactive	O	O
Multipoint Conference	M	O
Cancel Multipoint Conference	M	O
Multipoint Zero Comm	O	O
Cancel Multipoint Zero Comm	O	O
Multipoint Secondary Status	O	O
Cancel Multipoint Secondary Status	O	O
Video Indicate Ready to Activate	O	O
Video Temporal Spatial Trade Off	O	O
Video Not Decoded MBs	O	O
SBE Number	O	O
Terminal Number Assign	M	O
Terminal Joined Conference	O	O
Terminal Left Conference	O	O
Seen By At Least One Other	O	O
Cancel Seen By At Least One Other	O	O
Seen By All	O	O
Cancel Seen By All	O	O
Terminal You Are Seeing	O	O
Request For Floor	O	O
Vendor Indications	O	O
MC Location Indication	M	O
Jitter Indication	O	O
H.223 Skew Indication	Не разрешено	Не разрешено
H2250MaximumSkewIndication	O	M
New ATM Virtual Channel Indication	Не разрешено	Не разрешено
User Input	M (0-9, * и #)	M (0-9, * и #)

6.5 Алгоритмы установления, поддержания и разрушения соединения

Процедура установления, поддержания и разрушения соединений в IP-сети с использованием семейства протоколов Н.323 на страницах этой книги обсуждалась уже дважды, в главах 1 и 2, с разной степенью детализации. Теперь, вооружившись материалами глав 4, 5 и 6, пора продолжить эту цепочку последовательных приближений и рассмотреть алгоритмы установления, поддержания и разрушения соединений по протоколу Н.323 более подробно.

В силу важности этих алгоритмов авторам хотелось бы сохранить легкость восприятия материала. Поэтому они приняли решение рассмотреть наиболее часто применяемые на практике примеры базового соединения в сети, базирующейся на рекомендации Н.323. В качестве примеров взяты случаи:

- вызываемый и вызывающий пользователи зарегистрированы в одном и том же привратнике, который маршрутизирует сигнальную и управляющую информацию;
- вызываемый и вызывающий пользователи соединяются непосредственно друг с другом, привратник в сети отсутствует.

Прежде чем рассматривать эти два сценария, отметим, что в общем случае алгоритмы установления, поддержания и разрушения соединений по Н.323 включают в себя следующие фазы:

- Фаза А. Установление соединения;
- Фаза В. Определение ведущего/ведомого оборудования и обмен данными о функциональных возможностях;
- Фаза С. Установление аудиовизуальной связи между вызывающим и вызываемым оборудованием;
- Фаза D. Изменение полосы пропускания, запрос текущего состояния оборудования, создание конференций и обращение к дополнительным услугам;
- Фаза Е. Завершение соединения.

6.5.1 Базовое соединение с участием привратника

Сценарий этого первого случая приведен на рис.6.19. Вызывающее оборудование передает сообщение ARQ с alias-адресом вызываемого абонента, в ответ на которое привратник передает сообщение ACF с уведомлением, что именно он будет маршрутизировать сигнальные сообщения (Gatekeeper routed call signaling), и с указанием транспортного адреса своего сигнального канала. Далее вызывающее оборудование передает на этот транспортный адрес запрос соединения Setup. Привратник пересылает сообщение Setup вызываемому оборудованию и передает вызывающему оборудованию

сообщение Call Proceeding, означающее, что полученной информации достаточно для обслуживания поступившего вызова. Вызываемое оборудование также отвечает на Setup сообщением Call Proceeding. Если оборудование имеет возможность принять вызов, оно передает запрос допуска к ресурсам сети ARQ, на который привратник может ответить подтверждением ACF или отказом в допуске к ресурсам сети ARJ. В первом случае вызываемое оборудование передает сообщение Alerting, и привратник маршрутизирует его к вызываемому оборудованию. Вызываемому пользователю подается визуальный или акустический сигнал о входящем вызове, а вызывающему дается индикация того, что вызываемый пользователь не занят и ему подается вызывной сигнал. При отказе в допуске к ресурсам сети вызываемое оборудование закрывает сигнальный канал путем передачи привратнику сообщения Release Complete.

После того как вызываемый пользователь примет входящий вызов, привратнику передается сообщение Connect с транспортным адресом управляющего канала H.245 вызываемого оборудования. Привратник заменяет этот адрес транспортным адресом своего управляющего канала H.245 и пересылает Connect вызываемому оборудованию, после чего открывается управляющий канал H.245.

Чтобы ускорить открытие разговорной сессии, управляющий канал может быть открыт вызываемым оборудованием после получения сообщения Setup с транспортным адресом управляющего канала H.245 вызываемого оборудования или привратника, или вызывающим пользователем после получения сообщения Call Proceeding или Alerting, содержащего транспортный адрес управляющего канала H.245 вызываемого пользователя или привратника.

После открытия управляющего канала H.245 начинается обмен данными о функциональных возможностях оборудования. В рассматриваемом нами случае все управляющие сообщения, передаваемые от одного оконечного оборудования к другому, маршрутизируются привратником. Терминалы обмениваются сообщениями **TerminalCapabilitySet**, в которых указываются возможные алгоритмы декодирования принимаемой информации. Следует отметить, что сообщение **TerminalCapabilitySet** должно быть первым сообщением, передаваемым по управляющему каналу. Оборудование, принявшее сообщение **TerminalCapabilitySet** от другого оборудования, подтверждает его получение передачей сообщения **TerminalCapabilitySetAck**.

Затем инициируется процедура определения ведущего/ведомого оборудования, необходимая для разрешения конфликтов, возникающих между двумя устройствами при организации конференции, когда оба они могут быть активными контроллерами конференций, или между двумя устройствами, пытающимися одновременно открыть двунаправленные логические каналы. В ходе процедуры устройства обмениваются сообщениями **masterSlaveDetermination**.

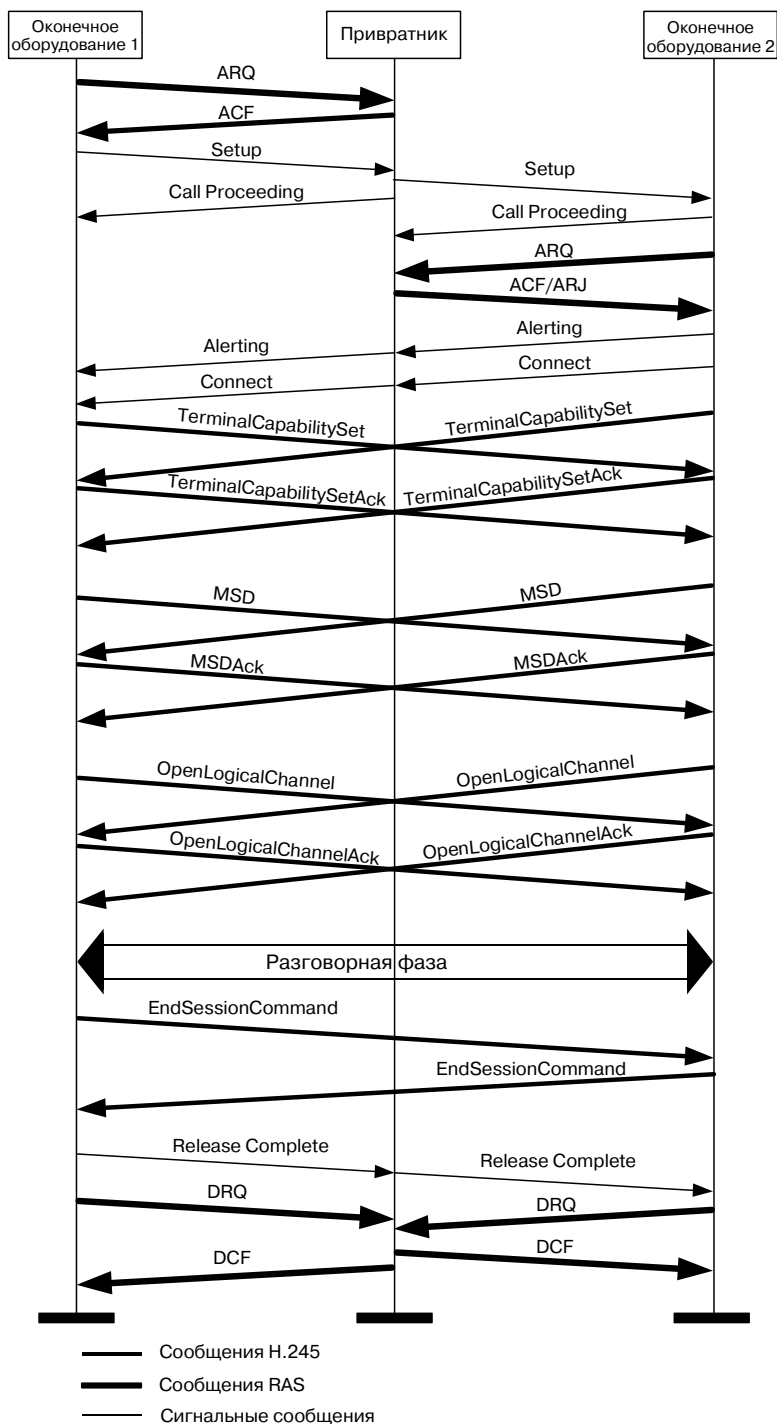


Рис. 6.19 Пример соединения с участием привратника

В ответ на полученные сообщения **masterSlaveDetermination** оба устройства передают сообщения **masterSlaveDeterminationAck**, в которых указывается, какое из этих устройств является для данного соединения ведущим, а какое – ведомым.

Напомним, что возможен сценарий процедуры Master-Slave Determination, предусматривающий сокращение количества передаваемых сообщений: оборудование, передавшее сообщение **masterSlaveDetermination** и получившее в ответ сообщение **masterSlaveDeterminationAck**, передает сообщение **masterSlaveDeterminationAck**.

После обмена данными о функциональных возможностях и определения ведущего и ведомого оборудования может выполняться процедура открытия однонаправленных логических каналов. В требовании открыть логический канал (в нашем случае – прямой логический канал) **openLogicalChannel** оборудование указывает вид информации, который будет передаваться по этому каналу, и алгоритм кодирования. В нашем случае логический канал предназначается для переноса речи, поэтому в сообщении **openLogicalChannel** включается параметр **mediaControlChannel** с указанием транспортного адреса канала RTCP, при помощи которого производится контроль передачи RTP пакетов. В ответ на сообщение **openLogicalChannel** оборудование должно передать подтверждение **openLogicalChannelAck**, в котором указывается транспортный адрес, на который передающей стороне следует посылать RTP пакеты, а также транспортный адрес канала RTCP.

Далее открывается разговорная сессия. Оборудование вызывающего пользователя передает речевую информацию, упакованную в пакеты RTP/UDP/IP, на транспортный адрес RTP-канала оборудования вызванного пользователя, а вызванный пользователь передает пакетированную речевую информацию на транспортный адрес RTP-канала оборудования вызывающего пользователя. При помощи канала RTCP ведется контроль передачи информации по RTP каналам.

После окончания разговорной фазы начинается фаза разрушения соединения. Оборудование пользователя, инициирующего разъединение, должно прекратить передачу речевой информации, закрыть логические каналы и передать по управляющему каналу сообщение H.245 **endSessionCommand**, означающее, что пользователь хочет завершить соединение. Далее от встречного оборудования ожидается сообщение **endSessionCommand**, после приема которого управляющий канал H.245 закрывается. Следующим шагом, если сигнальный канал еще открыт, передается сообщение Release Complete.

Пользователь, получивший команду **endSessionCommand** от пользователя, инициировавшего разрушение соединения, должен прекратить передачу речевой информации, закрыть логические ка-

налы и передать сообщение **endSessionCommand**. Далее, если сигнальный канал остался открытым, передается сообщение Release Complete, и сигнальный канал закрывается.

После вышеописанных действий окончательное оборудование извещает привратник об освобождении зарезервированной полосы пропускания. С этой целью каждый из участников соединения передает по каналу RAS запрос выхода из соединения DRQ, на который привратник должен ответить подтверждением DCF, после чего обслуживание вызова считается завершенным.

6.5.2 Базовое соединение без участия привратника

Теперь рассмотрим случай, когда вызываемое и вызывающее оборудование взаимодействуют непосредственно друг с другом, привратник в сети отсутствует (рис.6.20). Вызывающее оборудование посылает запрос соединения Setup на известный транспортный адрес сигнального канала вызываемого оборудования. Вызываемое оборудование отвечает на Setup сообщением Call Proceeding, а затем – Alerting. Вызываемому пользователю дается визуальный или акустический сигнал о входящем вызове, а вызывающему – индикация того, что вызываемый пользователь не занят и получает вызывной сигнал.

Как только вызываемый пользователь примет входящий вызов, передается сообщение Connect с указанием транспортного адреса управляющего канала Н.245 вызываемого оборудования, после чего открывается управляющий канал Н.245.

И здесь, чтобы ускорить открытие разговорной сессии, управляющий канал тоже может быть открыт вызываемым оборудованием после получения сообщения Setup с транспортным адресом управляющего канала Н.245 вызывающего оборудования, или вызывающим пользователем после получения сообщения Call Proceeding или Alerting, в котором содержится транспортный адрес управляющего канала Н.245 вызываемого оборудования.

После открытия управляющего канала выполняются все процедуры, описанные в первом случае: обмен данными о функциональных возможностях, определение ведущего/ведомого оборудования, открытие однонаправленных логических каналов.

Далее открывается разговорная сессия. Оборудование вызывающего пользователя передает речевую информацию, упакованную в пакеты RTP/UDP/IP, на транспортный адрес RTP-канала оборудования вызываемого пользователя, а оно, в свою очередь, передает пакетированную речевую информацию на транспортный адрес RTP-канала оборудования вызывающего пользователя.

После окончания разговорной фазы начинается фаза разрушения соединения. Оборудование пользователя, иницирующего разъедин-

нение, прекращает передачу речевой информации, закрывает логические каналы и передает по управляющему каналу сообщение H.245 **endSessionCommand**, означающее, что пользователь хочет завершить соединение. Ожидается сообщение **endSessionCommand** от встречного оборудования, после чего управляющий канал H.245 закрывается. Следующим шагом передается сообщение Release Complete, и сигнальный канал закрывается.

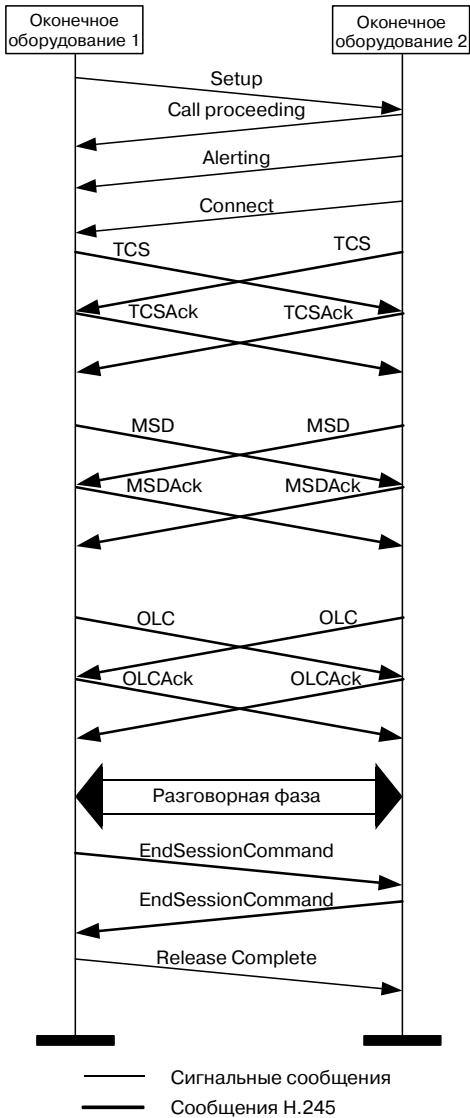


Рис. 6.20 Пример соединения без участия привратника

Пользователь, получивший команду **endSessionCommand** от пользователя, инициировавшего разъединение, должен прекратить передачу речевой информации, закрыть логические каналы и передать сообщение **endSessionCommand**. Далее, если сигнальный канал остался открытым, передается сообщение **Release Complete**, сигнальный канал закрывается, и обслуживание вызова считается завершенным.

6.5.3 Туннелирование управляющих сообщений

Для ускорения установления соединения может использоваться процесс, известный как инкапсуляция или туннелирование управляющих сообщений Н.245. При этом передача сообщений Н.245 осуществляется по сигнальному, а не по отдельному управляющему каналу. Одно или несколько сообщений Н.245 переносятся в элементе **h245Control** информационного поля **h323_uu_pdu** в любом из разрешенных сообщений Q.931.

Чтобы применить инкапсуляцию сообщений Н.245, вызывающее оборудование должно присвоить значение **TRUE** элементу **h245Tunnelling**, передаваемому в сообщении **Setup** и в последующих сообщениях Q.931. Вызываемое оборудование, получившее в сообщении **Setup** элемент **h245Tunnelling** со значением **TRUE** и желающее использовать инкапсуляцию управляющих сообщений, также должно присвоить значение **TRUE** элементу **h245Tunnelling** в сообщении, передаваемом в ответ на сообщение **Setup**, и в последующих сообщениях Q.931.

Вызываемое оборудование, не поддерживающее туннелирование управляющих сообщений, присваивает элементу **h245Tunnelling**, передаваемому в ответе на сообщение **Setup**, значение **FALSE**. В этом случае для передачи управляющих сообщений открывается отдельный канал Н.245.

6.5.4 Процедура быстрого установления соединения

Самый быстрый способ установления соединения в сети, базирующейся на рекомендации Н.323, – это использование процедуры **Fast Connect**. Чтобы инициировать процедуру **Fast Connect**, вызывающее оборудование передает сообщение **Setup** с элементом **fastStart**. Этот элемент может включать в себя одну или несколько структур **OpenLogicalChannel**. Одна из структур **OpenLogicalChannel** обязательно должна содержать элемент **forwardLogicalChannelParameters** и может содержать элемент **reverseLogicalChannelParameters**, но, в то же время, структура **OpenLogicalChannel** описывает точно один однонаправленный логический канал. Это означает, что когда описывается прямой логический канал, то в структуре присутствует только элемент **forwardLogicalChannelParameters**. Элемент содер-

жит информацию об алгоритме, который используется вызывающим оборудованием для кодирования передаваемой информации, и адрес канала RTCP. При описании канала обратного направления в элементе **forwardLogicalChannelParameters** не содержится никакой информации, хотя сам он обязательно присутствует, а в элементе **reverseLogicalChannelParameters** содержатся сведения об алгоритме декодирования принимаемой информации, транспортный адрес RTP, на который следует передавать информацию, и адрес канала RTCP.

В элементе **fastStart** может присутствовать несколько альтернативных структур **OpenLogicalChannel**, различающихся алгоритмами кодирования передаваемой информации или декодирования принимаемой информации, причем наиболее предпочтительные алгоритмы указываются в первую очередь.

Вызываемое оборудование может отклонить процедуру Fast Connect, либо если оно ее не поддерживает, либо если существует потребность в использовании процедур H.245 с открытием отдельного канала H.245 или с туннелированием управляющих сообщений. В этом случае элемент **fastStart** не включается ни в одно из сообщений, передаваемых после приема Setup, до сообщения Connect включительно. Открытие логических каналов для передачи речевой информации производится с использованием процедур H.245.

Вызываемое оборудование, получившее сообщение Setup с элементом **fastStart** и могущее поддержать процедуру Fast Connect, должно включить элемент **fastStart** в любое из сообщений Q.931, передаваемых после приема Setup, до сообщения Connect включительно. Элемент **fastStart** содержит структуры **OpenLogicalChannel**, которые выбраны вызываемым оборудованием из структур, предложенных вызывающим оборудованием. И снова одна из структур **OpenLogicalChannel** содержит элемент **forwardLogicalChannelParameters** со сведениями об алгоритме кодирования информации, с транспортными адресами каналов RTP и RTCP вызываемого оборудования. Вторая структура **OpenLogicalChannel** включает в себя элемент **forwardLogicalChannelParameters**, не содержащий никакой информации, и элемент **reverseLogicalChannelParameters** со сведениями об алгоритме кодирования информации и с транспортным адресом канала RTCP вызываемого оборудования.

Вызываемое оборудование может начинать передачу информации сразу вслед за любым сообщением Q.931 с элементом **fastStart**. Это означает, что вызывающее оборудование должно быть готовым к приему информации, закодированной любым из указанных в сообщении Setup способов. Сообщение Q.931 с элементом **fastStart**, переданное вызываемым оборудованием после получения сообщения Setup, может прийти после начала передачи пользовательской

информации. Если вызывающее оборудование не желает принимать речевую информацию до прихода сообщения Connect, оно присваивает значение TRUE элементу **mediaWaitForConnect**, передаваемому в сообщении Setup.

Вызывающее оборудование, инициировавшее процедуру Fast Connect, может начать передачу речевой информации сразу после приема любого из разрешенных сообщений Q.931, содержащего элемент **fastStart**.

При разрушении соединения одним из участников передается сообщение Release Complete, после чего закрывается сигнальный канал и соединение считается завершенным.

Следует отметить, что при использовании процедуры Fast Connect или при туннелировании управляющих сообщений как одна, так и другая сторона может открыть управляющий канал Н.245, для чего оборудование этой стороны должно включить в любое сообщение Q.931 элемент **h245Address**. При этом процедура Fast Connect или туннелирование прерывается.

6.5.5 Установление соединения с участием шлюза

В главе 2 обсуждались основные сценарии установления соединения в IP-телефонии. Напомним приведенные там варианты, предполагающие участие шлюза – элемента сети Н.323, который был рассмотрен в предыдущей главе. Первый вариант – это случай, когда абонент ТфОП вызывает пользователя IP-сети, второй – когда пользователь IP-сети вызывает абонента ТфОП, а в третьем варианте абонент ТфОП вызывает абонента ТфОП, но соединение проходит через IP-сеть.

В первом варианте с точки зрения протоколов Н.323 соединение устанавливается так же, как соединение участников, включенных в сеть с маршрутизацией пакетов IP. Рассмотрим ситуацию с точки зрения ТфОП. Существует два способа набора номера вызываемого абонента: одноступенчатый и двухступенчатый.

При одноступенчатом способе вызывающий абонент сразу набирает номер вызываемого абонента, и шлюз устанавливает с ним соединение. Пока устанавливается соединение в IP-сети, шлюз может передать вызываемому абоненту ТфОП сообщение Call Proceeding, чтобы перезапустить таймеры. Данный способ может использоваться в корпоративной сети для организации связи между абонентами учреждений телефонных станций.

В сетях связи общего пользования применяется двухступенчатый способ, при котором вызывающий абонент сначала набирает телефонный номер шлюза и устанавливает с ним соединение. Затем абонент вводит свой персональный код для идентификации и номер вы-

зываемого абонента; эта информация передается по проклученному разговорному тракту сигналами DTMF. Следует отметить, что необходимость в наборе персонального кода возникает не всегда, так как номер вызывающего абонента может содержаться в сигнальных сообщениях систем сигнализации DSS1 и OKC7, а при использовании систем сигнализации 2BCK или аналоговых систем сигнализации – определяться при помощи АОН.

Существует несколько способов идентификации абонентов. В первом случае alias-адрес абонента (PIN-код или телефонный номер) шлюз передает привратнику в сообщении ARQ. Во втором случае идентификационный номер вызывающего абонента, набранный с помощью DTMF, передается специальному серверу. Кроме того, в ТфОП вызов может обрабатываться системой обработки телефонных карт, которая отвечает за идентификацию пользователей и начисление платы. Существует еще один вариант когда функции идентификации абонентов и начисления оплаты возложены на опорную АТС, к которой подключен шлюз.

Во втором сценарии, когда пользователь IP-сети вызывает абонента ТфОП при помощи шлюза, с точки зрения протоколов H.323 соединение устанавливается так же, как описанное соединение участников, включенных в сеть с маршрутизацией пакетов IP. Вызываемое оборудование организует сигнальный канал H.225.0 со шлюзом (при участии или без участия привратника). Далее передается требование на установление соединения Setup, которое содержит телефонный номер вызываемого абонента в формате E.164. Пока устанавливается соединение в ТфОП, шлюз может передать вызывающему абоненту IP-сети сообщение Call Proceeding, чтобы перезапустить таймеры, если в течение 4 секунд после приема сообщения Setup он не передал сообщения Alerting, Connect или Release Complete. Чтобы указать, что вызов выходит за пределы IP-сети, в сообщения Alerting, Call Proceeding, Progress и Connect должен включаться информационный элемент Progress Indicator.

Сценарий вызова абонента ТфОП абонентом ТфОП является комбинацией двух предыдущих сценариев и с технической точки зрения не содержит никаких новых процедур. О социальном и экономическом аспектах обслуживания вызовов по этому сценарию уже говорилось в главе 2.

Глава 7

Протокол инициирования сеансов связи – SIP

7.1 Принципы протокола SIP

Протокол инициирования сеансов – Session Initiation Protocol (SIP) является протоколом прикладного уровня и предназначается для организации, модификации и завершения сеансов связи: мультимедийных конференций, телефонных соединений и распределения мультимедийной информации. Пользователи могут принимать участие в существующих сеансах связи, приглашать других пользователей и быть приглашенными ими к новому сеансу связи. Приглашения могут быть адресованы определенному пользователю, группе пользователей или всем пользователям.

Протокол SIP разработан группой MMUSIC (Multiparty Multimedia Session Control) комитета IETF (Internet Engineering Task Force), а спецификации протокола представлены в документе RFC 2543 [54]. В основу протокола рабочая группа MMUSIC заложила следующие принципы:

Персональная мобильность пользователей. Пользователи могут перемещаться без ограничений в пределах сети, поэтому услуги связи должны предоставляться им в любом месте этой сети. Пользователю присваивается уникальный идентификатор, а сеть предоставляет ему услуги связи вне зависимости от того, где он находится. Для этого пользователь с помощью специального сообщения – **REGISTER** – информирует о своих перемещениях сервер определения местоположения.

Масштабируемость сети. Она характеризуется, в первую очередь, возможностью увеличения количества элементов сети при её расширении. Серверная структура сети, построенной на базе протокола SIP, в полной мере отвечает этому требованию.

Расширяемость протокола. Она характеризуется возможностью дополнения протокола новыми функциями при введении новых услуг и его адаптации к работе с различными приложениями.

В качестве примера можно привести ситуацию, когда протокол SIP используется для установления соединения между шлюзами, взаимодействующими с ТфОП при помощи сигнализации ОКС7 или DSS1. В настоящее время SIP не поддерживает прозрачную передачу сигнальной информации телефонных систем сигнализации. Вследствие этого дополнительные услуги ISDN оказываются недоступными для пользователей IP-сетей.

Расширение функций протокола SIP может быть произведено за счет введения новых заголовков сообщений, которые должны быть зарегистрированы в уже упоминавшейся ранее организации IANA. При этом, если SIP-сервер принимает сообщение с неизвестными ему полями, то он просто игнорирует их и обрабатывает лишь те поля, которые он знает.

Для расширения возможностей протокола SIP могут быть также добавлены и новые типы сообщений.

Интеграция в стек существующих протоколов Интернет, разработанных IETF. Протокол SIP является частью глобальной архитектуры мультимедиа, разработанной комитетом Internet Engineering Task Force (IETF). Эта архитектура включает в себя также протокол резервирования ресурсов (Resource Reservation Protocol – RSVP, RFC 2205), транспортный протокол реального времени (Real-Time Transport Protocol – RTP, RFC 1889), протокол передачи потоковой информации в реальном времени (Real-Time Streaming Protocol – RTSP, RFC 2326), протокол описания параметров связи (Session Description Protocol – SDP, RFC 2327). Однако функции протокола SIP не зависят ни от одного из этих протоколов.

Взаимодействие с другими протоколами сигнализации. Протокол SIP может быть использован совместно с протоколом H.323 (Главы 5 и 6). Возможно также взаимодействие протокола SIP с системами сигнализации ТфОП – DSS1 и ОКС7 [6,7]. Для упрощения такого взаимодействия сигнальные сообщения протокола SIP могут переносить не только специфический SIP-адрес, но и телефонный номер формата E.164 или любого другого формата. Кроме того, протокол SIP, наравне с протоколами H.323 и ISUP/IP, может применяться для синхронизации работы устройств управления шлюзами, о чем пойдет речь в следующей главе (рис. 8.2); в этом случае он должен взаи-

модействовать с протоколом MGCP. Другой важной особенностью протокола SIP является то, что он приспособлен к организации доступа пользователей сетей IP-телефонии к услугам интеллектуальных сетей [8], и существует мнение, что именно этот протокол станет основным при организации связи между указанными сетями.

7.2 Интеграция протокола SIP с IP-сетями

Одной из важнейших особенностей протокола SIP является его независимость от транспортных технологий. В качестве транспорта могут использоваться протоколы X.25, Frame Relay, AAL5/ATM, IPX и др. Структура сообщений SIP не зависит от выбранной транспортной технологии. Но, в то же время, предпочтение отдается технологии маршрутизации пакетов IP и протоколу UDP. При этом, правда, необходимо создать дополнительные механизмы для надежной доставки сигнальной информации. К таким механизмам относятся повторная передача информации при ее потере, подтверждение приема и др.

Здесь же следует отметить то, что сигнальные сообщения могут переноситься не только протоколом транспортного уровня UDP, но и протоколом TCP. Протокол UDP позволяет быстрее, чем TCP, доставлять сигнальную информацию (даже с учетом повторной передачи неподтвержденных сообщений), а также вести параллельный поиск местоположения пользователей и передавать приглашения к участию в сеансе связи в режиме многоадресной рассылки. В свою очередь, протокол TCP упрощает работу с межсетевыми экранами (fire-wall), а также гарантирует надежную доставку данных. При использовании протокола TCP разные сообщения, относящиеся к одному вызову, либо могут передаваться по одному TCP-соединению, либо для каждого запроса и ответа на него может открываться отдельное TCP-соединение. На рисунке 7.1 показано место, занимаемое протоколом SIP в стеке протоколов TCP/IP.

Протокол инициирования сеансов связи (SIP)	Прикладной уровень
Протоколы TCP и UDP	Транспортный уровень
Протоколы IPv4 и IPv6	Сетевой уровень
PPP, ATM, Ethernet	Уровень звена данных
UTP5, SDH, DDH, V.34 и др.	Физический уровень

Рис. 7.1 Место протокола SIP в стеке протоколов TCP/IP

По сети с маршрутизацией пакетов IP может передаваться пользовательская информация практически любого вида: речь, видео и данные, а также любая их комбинация, называемая мультимедийной информацией. При организации связи между терминалами пользователей необходимо известить встречную сторону, какого рода информация может приниматься (передаваться), алгоритм ее кодирования и адрес, на который следует передавать информацию. Таким образом, одним из обязательных условий организации связи при помощи протокола SIP является обмен между сторонами данными об их функциональных возможностях. Для этой цели чаще всего используется протокол описания сеансов связи – SDP (Session Description Protocol). Поскольку в течение сеанса связи может производиться его модификация, предусмотрена передача сообщений SIP с новыми описаниями сеанса средствами SDP. Более подробно протокол SDP рассмотрен в главе 8.

Для передачи речевой информации комитет IETF предлагает использовать протокол RTP, рассмотренный в главе 4 настоящей книги, но сам протокол SIP не исключает возможность применения для этих целей других протоколов.

В протоколе SIP не реализованы механизмы управления потоками информации и предоставления гарантированного качества обслуживания. Кроме того, протокол SIP не предназначен для передачи пользовательской информации, в его сообщениях может переноситься информация лишь ограниченного объема. При переносе через сеть слишком большого сообщения SIP не исключена его фрагментация на уровне IP, что может повлиять на качество передачи информации.

В глобальной информационной сети Интернет уже довольно давно функционирует экспериментальный участок Mbone, который образован из сетевых узлов, поддерживающих режим многоадресной рассылки мультимедийной информации. Важнейшей функцией Mbone является поддержка мультимедийных конференций, а основным способом приглашения участников к конференции стал протокол SIP.

Протокол SIP предусматривает организацию конференций трех видов:

- в режиме многоадресной рассылки (multicasting), когда информация передается на один multicast-адрес, а затем доставляется сетью конечным адресатам;
- при помощи устройства управления конференции (MCU), к которому участники конференции передают информацию в режиме точка-точка, а оно, в свою очередь, обрабатывает ее (т.е. смешивает или коммутрует) и рассылает участникам конференции;

- путем соединения каждого пользователя с каждым в режиме точка-точка.

Протокол SIP дает возможность присоединения новых участников к уже существующему сеансу связи, т.е. двусторонний сеанс может перейти в конференцию.

И, в заключение рассказа об интеграции протокола SIP с IP-сетями, следует отметить то, что разработаны методы совместной работы этого протокола с преобразователем сетевых адресов – Network Address Translator (NAT).

7.3 Адресация

Для организации взаимодействия с существующими приложениями IP-сетей и для обеспечения мобильности пользователей протокол SIP использует адрес, подобный адресу электронной почты. В качестве адресов рабочих станций используются специальные универсальные указатели ресурсов – URL (Universal Resource Locators), так называемые SIP URL.

SIP-адреса бывают четырех типов:

- *имя@домен*;
- *имя@хост*;
- *имя@IP-адрес*;
- *№телефона@шлюз*.

Таким образом, адрес состоит из двух частей. Первая часть – это имя пользователя, зарегистрированного в домене или на рабочей станции. Если вторая часть адреса идентифицирует какой-либо шлюз, то в первой указывается телефонный номер абонента.

Во второй части адреса указывается имя домена, рабочей станции или шлюза. Для определения IP-адреса устройства необходимо обратиться к службе доменных имен – Domain Name Service (DNS). Если же во второй части SIP-адреса размещается IP-адрес, то с рабочей станцией можно связаться напрямую.

В начале SIP-адреса ставится слово «sip:», указывающее, что это именно SIP-адрес, т.к. бывают и другие (например, «mailto:»). Ниже приводятся примеры SIP-адресов:

sip: als@rts.loniis.ru

sip: user1@192.168.100.152

sip: 294-75-47@gateway.ru

7.4 Архитектура сети SIP

В некотором смысле прародителем протокола SIP является протокол переноса гипертекста – HTTP (Hypertext Transfer Protocol, RFC 2068). Протокол SIP унаследовал от него синтаксис и архитектуру «клиент-сервер», которую иллюстрирует рис. 7.2.

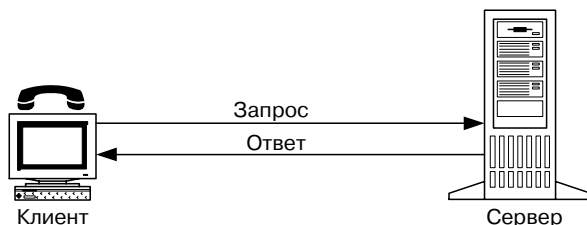


Рис. 7.2 Архитектура "клиент-сервер"

Клиент выдает запросы, в которых указывает, что он желает получить от сервера. Сервер принимает запрос, обрабатывает его и выдает ответ, который может содержать уведомление об успешном выполнении запроса, уведомление об ошибке или информацию, затребованную клиентом.

Управление процессом обслуживания вызова распределено между разными элементами сети SIP. Основным функциональным элементом, реализующим функции управления соединением, является терминал. Остальные элементы сети отвечают за маршрутизацию вызовов, а в некоторых случаях предоставляют дополнительные услуги.

7.4.1 Терминал

В случае, когда клиент и сервер взаимодействуют непосредственно с пользователем (т.е. реализованы в оконечном оборудовании пользователя), они называются, соответственно, клиентом агента пользователя – User Agent Client (UAC) – и сервером агента пользователя – User Agent Server (UAS).

Следует особо отметить, что сервер UAS и клиент UAC могут (но не обязаны) непосредственно взаимодействовать с пользователем, а другие клиенты и серверы SIP этого делать не могут. Если в устройстве присутствуют и сервер UAS, и клиент UAC, то оно называется агентом пользователя – User Agent (UA), а по своей сути представляет собой терминальное оборудование SIP.

Кроме терминалов определены два основных типа сетевых элементов SIP: прокси-сервер (proxy server) и сервер переадресации (redirect server).

7.4.2 Прокси-сервер

Прокси-сервер (от английского *proxy* – представитель) представляет интересы пользователя в сети. Он принимает запросы, обрабатывает их и, в зависимости от типа запроса, выполняет определенные действия. Это может быть поиск и вызов пользователя, маршрутизация запроса, предоставление услуг и т.д. Прокси-сервер состоит из клиентской и серверной частей, поэтому может принимать вызовы, инициировать собственные запросы и возвращать ответы. Прокси-сервер может быть физически совмещен с сервером определения местоположения (в этом случае он называется *registrar*) или существовать отдельно от этого сервера, но иметь возможность взаимодействовать с ним по протоколам LDAP (RFC 1777), rwhois (RFC 2167) и по любым другим протоколам.

Предусмотрено два типа прокси-серверов – с сохранением состояний (*stateful*) и без сохранения состояний (*stateless*).

Сервер первого типа хранит в памяти входящий запрос, который явился причиной генерации одного или нескольких исходящих запросов. Эти исходящие запросы сервер также запоминает. Все запросы хранятся в памяти сервера только до окончания транзакции, т.е. до получения ответов на запросы.

Сервер первого типа позволяет предоставить большее количество услуг, но работает медленнее, чем сервер второго типа. Он может применяться для обслуживания небольшого количества клиентов, например, в локальной сети. Прокси-сервер должен сохранять информацию о состояниях, если он:

- использует протокол TCP для передачи сигнальной информации;
- работает в режиме многоадресной рассылки сигнальной информации;
- размножает запросы.

Последний случай имеет место, когда прокси-сервер ведет поиск вызываемого пользователя сразу в нескольких направлениях, т.е. один запрос, который пришел к прокси-серверу, размножается и передается одновременно по всем этим направлениям.

Сервер без сохранения состояний просто ретранслирует запросы и ответы, которые получает. Он работает быстрее, чем сервер первого типа, так как ресурс процессора не тратится на запоминание состояний, вследствие чего сервер этого типа может обслужить большее количество пользователей. Недостатком такого сервера является то, что на его базе можно реализовать лишь наиболее простые услуги. Впрочем, прокси-сервер может функционировать как сервер с сохранением состояний для одних пользователей и как сервер без сохранения состояний – для других.

Алгоритм работы пользователей с прокси-сервером выглядит следующим образом. Поставщик услуг IP-телефонии сообщает ад-

рес прокси-сервера своим пользователям. Вызывающий пользователь передает к прокси-серверу запрос соединения. Сервер обрабатывает запрос, определяет местоположение вызываемого пользователя и передает запрос этому пользователю, а затем получает от него ответ, подтверждающий успешную обработку запроса, и транслирует этот ответ пользователю, передавшему запрос. Прокси-сервер может модифицировать некоторые заголовки сообщений, которые он транслирует, причем каждый сервер, обработавший запрос в процессе его передачи от источника к приемнику, должен указать это в SIP-запросе для того, чтобы ответ на запрос вернулся по такому же пути.

7.4.3 Сервер переадресации

Сервер переадресации предназначен для определения текущего адреса вызываемого пользователя. Вызывающий пользователь передает к серверу сообщение с известным ему адресом вызываемого пользователя, а сервер обеспечивает переадресацию вызова на текущий адрес этого пользователя. Для реализации этой функции сервер переадресации должен взаимодействовать с сервером определения местоположения.

Сервер переадресации не терминирует вызовы как сервер RAS и не инициирует собственные запросы как прокси-сервер. Он только сообщает адрес либо вызываемого пользователя, либо прокси-сервера. По этому адресу инициатор запроса передает новый запрос. Сервер переадресации не содержит клиентскую часть программного обеспечения.

Но пользователю не обязательно связываться с каким-либо SIP-сервером. Он может сам вызвать другого пользователя при условии, что знает его текущий адрес.

7.4.4 Сервер определения местоположения пользователей

Пользователь может перемещаться в пределах сети, поэтому необходим механизм определения его местоположения в текущий момент времени. Например, сотрудник предприятия уезжает в командировку, и все вызовы, адресованные ему, должны быть направлены в другой город на его временное место работы. О том, где он находится, пользователь информирует специальный сервер с помощью сообщения **REGISTER**. Возможны два режима регистрации: пользователь может сообщить свой новый адрес один раз, а может регистрироваться периодически через определенные промежутки времени. Первый способ подходит для случая, когда терминал, доступный пользователю, включен постоянно, и его не перемещают по сети, а второй – если терминал часто перемещается или выключается.

Для хранения текущего адреса пользователя служит *сервер определения местоположения пользователей*, представляющий собой базу данных адресной информации. Кроме постоянного адреса пользователя, в этой базе данных может храниться один или несколько текущих адресов.

Этот сервер может быть совмещен с прокси-сервером (в таком случае он называется registrar) или быть реализован отдельно от прокси-сервера, но иметь возможность связываться с ним.

В RFC 2543 сервер определения местоположения представлен как отдельный сетевой элемент, но принципы его работы в этом документе не регламентированы. Стоит обратить внимание на то, что вызывающий пользователь, которому нужен текущий адрес вызываемого пользователя, не связывается с сервером определения местоположения напрямую. Эту функцию выполняют SIP-серверы при помощи протоколов LDAP (RFC 1777), rwhois (RFC 2167), или других протоколов.

7.4.5 Пример SIP- сети

Резюмируя все сказанное выше, отметим, что сети SIP строятся из элементов трех основных типов: терминалов, прокси-серверов и серверов переадресации. На рис. 7.3 приведен пример возможного построения сети SIP.

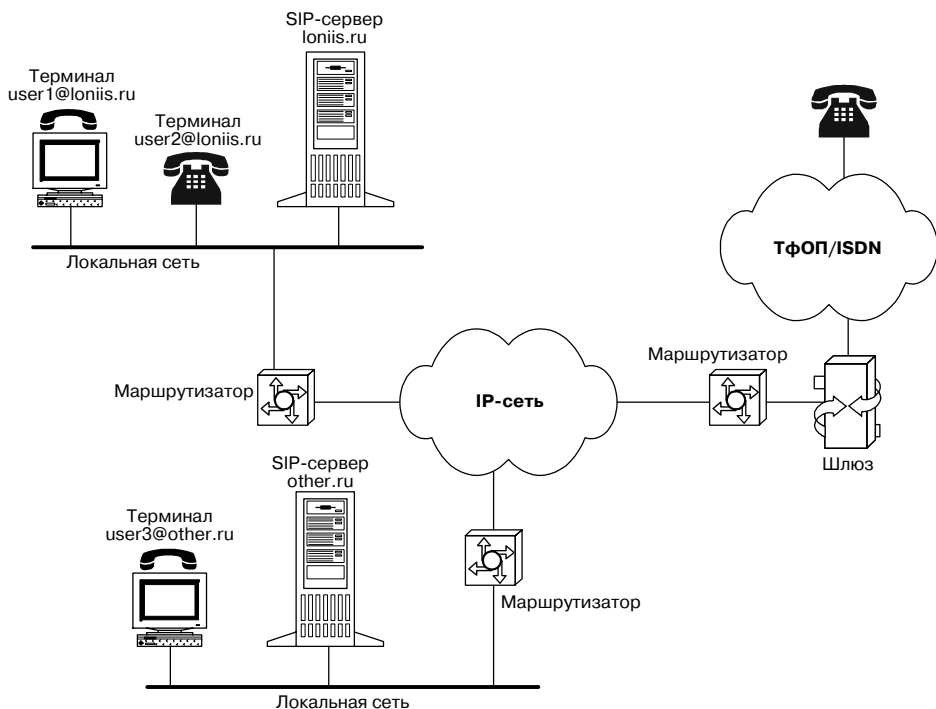


Рис. 7.3 Пример построения сети SIP

Стоит обратить внимание на то что, что SIP-серверы, представленные на рис. 7.3, являются отдельными функциональными сетевыми элементами. Физически они могут быть реализованы на базе серверов локальной сети, которые, помимо выполнения своих основных функций, будут также обрабатывать SIP-сообщения. Терминалы же могут быть двух типов: персональный компьютер со звуковой платой и программным обеспечением SIP-клиента (UA) или SIP-телефон, подключающийся непосредственно к ЛВС Ethernet (SIP-телефоны, производимые компанией Cisco Systems, недавно появились на российском рынке). Таким образом, пользователь локальной вычислительной сети передает все запросы к своему SIP-серверу, а тот обрабатывает их и обеспечивает установление соединений. Путем программирования сервер можно настроить на разные алгоритмы работы: он может обслуживать часть пользователей (например, руководство предприятия или особо важных лиц) по одним правилам, а другую часть – по иным. Возможно также, что сервер будет учитывать категорию и срочность вызовов, а также вести начисление платы за разговоры.

Структурная схема организации услуг SIP-сервера представлена на рисунке 7.4.

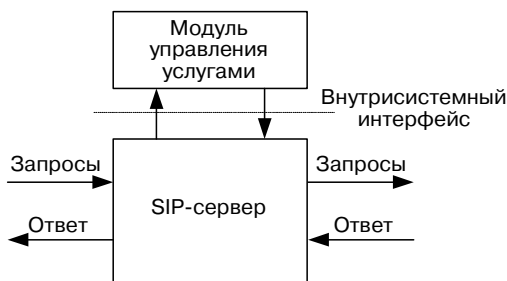


Рис. 7.4 Структурная схема организации услуг SIP-сервера

Модуль управления услугами отвечает за предоставление услуг и за общее управление сервером. Принятые сервером запросы и ответы поступают в модуль управления услугами и обрабатываются им, на основании чего определяется реакция на полученные сообщения. Интерфейс человек-машина позволяет гибко менять настройки сервера и вести мониторинг сети.

7.5 Сообщения протокола SIP

7.5.1 Структура сообщений

Согласно архитектуре «клиент-сервер» все сообщения делятся на запросы, передаваемые от клиента к серверу, и на ответы сервера клиенту.

Например, чтобы инициировать установление соединения, вызывающий пользователь должен сообщить серверу ряд параметров, в частности, адрес вызываемого пользователя, параметры информационных каналов и др. Эти параметры передаются в специальном SIP-запросе. От вызываемого пользователя к вызывающему передается ответ на запрос, также содержащий ряд параметров.

Все сообщения протокола SIP (запросы и ответы), представляют собой последовательности текстовых строк, закодированных в соответствии с документом RFC 2279. Структура и синтаксис сообщений SIP, как уже упоминалось ранее, идентичны используемым в протоколе HTTP. На рисунке 7.5 представлена структура сообщений протокола SIP.

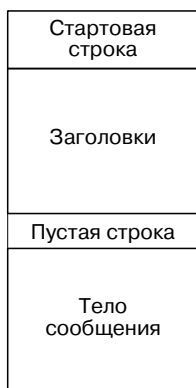


Рис. 7.5 Структура сообщений протокола SIP

Стартовая строка представляет собой начальную строку любого SIP-сообщения. Если сообщение является запросом, в этой строке указываются тип запроса, адресат и номер версии протокола. Если сообщение является ответом на запрос, в стартовой строке указываются номер версии протокола, тип ответа и его короткая расшифровка, предназначенная только для пользователя.

Заголовки сообщений содержат сведения об отправителе, адресате, пути следования и др., в общем, переносят информацию, необходимую для обслуживания данного сообщения. О типе заголовка можно узнать по его имени. Оно не зависит от регистра (т.е. буквы могут быть прописные и строчные), но обычно имя пишут с большой буквы, за которой идут строчные.

Сообщения протокола SIP могут содержать так называемое тело сообщения. В запросах **ACK**, **INVITE** и **OPTIONS** тело сообщения содержит описание сеансов связи, например, в формате протокола SDP. Запрос **BYE** тела сообщения не содержит, а ситуация с запро-

сом **REGISTER** подлежит дальнейшему изучению. С ответами дело обстоит иначе: любые ответы могут содержать тело сообщения, но содержимое тела в них бывает разным.

7.5.2 Заголовки сообщений

В протоколе SIP определено четыре вида заголовков (Таблица 7.1):

- *Общие заголовки*, присутствующие в запросах и ответах;
- *Заголовки содержания*, переносят информацию о размере тела сообщения или об источнике запроса (начинаются со слова «**Content**»);
- *Заголовки запросов*, передающие дополнительную информацию о запросе;
- *Заголовки ответов*, передающие дополнительную информацию об ответе.

Заголовок содержит название, за которым, отделенное двоеточием, следует значение заголовка. В поле значения содержатся передаваемые данные. Следует отметить, что если сервер принимает сообщения, заголовки которых ему не известны, то эти заголовки игнорируются.

Ниже представлены наиболее часто используемые заголовки.

Заголовок **Call-ID** – уникальный идентификатор сеанса связи или всех регистраций отдельного клиента, он подобен метке соединения (call reference) в сигнализации DSS-1 [7]. Значение идентификатору присваивает сторона, которая инициирует вызов. Заголовок **Call-ID** состоит из буквенно-числового значения и имени рабочей станции, которая присвоила значение этому идентификатору. Между ними должен стоять символ @, например, **2345call@rts.loniis.ru**. Возможна следующая ситуация: к одной мультимедийной конференции относятся несколько соединений, тогда все они будут иметь разные идентификаторы **Call-ID**.

Заголовок **To** – определяет адресата. Кроме SIP-адреса здесь может стоять параметр «**tag**» для идентификации конкретного терминала пользователя (например, домашнего, рабочего или сотового телефона) в том случае, когда все его терминалы зарегистрированы под одним адресом SIP URL. Запрос может множиться и достичь разных терминалов пользователя; чтобы их различать, необходимо иметь метку **tag**. Ее вставляет в заголовок терминальное оборудование вызванного пользователя при ответе на принятый запрос.

Если необходим визуальный вывод имени пользователя, например, на дисплей, то имя пользователя также размещается в поле **To**.

Заголовок **From** – идентифицирует отправителя запроса; по структуре аналогичен полю **To**.

Таблица 7.1 Виды заголовков сообщений SIP

Общие заголовки	Заголовки содержания	Заголовки запросов	Заголовки ответов
Call-ID (идентификатор сеанса связи)	Content-Encoding (кодирование тела сообщения)	Accept (принимается)	Allow (разрешение)
Contact (контактировать)	Content-Length (размер тела сообщения)	Accept-Encoding (метод кодирования поддерживается)	Proxy-Authenticate (подтверждение подлинности прокси-сервера)
CSeq (последовательность)	Content-Type (тип содержимого)	Accept-Language (язык поддерживается)	Retry-After (повторить через некоторое время)
Date (Дата)		Authorization (авторизация)	Server (сервер)
Encryption (шифрование)			Unsupported (не поддерживается)
Expires (срабатывание таймера)		Hide (скрыть)	Warning (предупреждение)
From (источник запроса)		Max-Forwards (максимальное количество переадресаций)	WWW-Authenticate (подтверждение подлинности WWW-сервера)
Record-Route (запись маршрута)		Organization (организация)	
Timestamp (метка времени)		Priority (приоритет)	
To (Адресат)		Proxy-Authorization (авторизация прокси-сервера)	
Via (через)		Proxy-Require (требуется прокси-сервер)	
		Route (маршрут)	
		Require (требуется)	
		Response-Key (ключ кодирования ответа)	
		Subject (тема)	
		User-Agent (агент пользователя)	

Заголовок **CSeq** – уникальный идентификатор запроса, относящегося к одному соединению. Он служит для корреляции запроса с ответом на него. Заголовок состоит из двух частей: натурального числа из диапазона от 1 до 2^{32} и типа запроса. Сервер должен проверять значение **CSeq** в каждом принимаемом запросе и считать запрос новым, если значение **CSeq** больше предыдущего. Пример заголовка: **CSeq: 2 INVITE**.

Заголовок **Via** служит для того, чтобы избежать ситуации, в которых запрос пойдет по замкнутому пути, а также для тех случаев, когда необходимо, чтобы запросы и ответы обязательно проходили по одному и тому же пути (например, в случае использования межсетевого экрана – firewall). Дело в том, что запрос может проходить через несколько прокси-серверов, каждый из которых принимает, обрабатывает и переправляет запрос к следующему прокси-серверу, и так до тех пор, пока запрос не достигнет адресата. Таким образом, в заголовке **Via** указывается весь путь, пройденный запросом: каждый прокси-сервер добавляет поле со своим адресом. При необходимости (например, чтобы обеспечить секретность) действительный адрес может скрываться.

Например, запрос на своем пути обрабатывался двумя прокси-серверами: сначала сервером loniis.ru, потом sip.telecom.com. Тогда в запросе появятся следующие поля:

Via: SIP/2.0/UDP sip.telecom.com:5060;branch=721e418c4.1

Via: SIP/2.0/UDP loniis.ru:5060,

где параметр «**branch**» означает, что на сервере sip.telecom.com запрос был размножен и направлен одновременно по разным направлениям, и наш запрос был передан по направлению, которое идентифицируется следующим образом: **721e418c4.1**.

Содержимое полей **Via** копируется из запросов в ответы на них, и каждый сервер, через который проходит ответ, удаляет поле **Via** со своим именем.

В заголовок **Record-route** прокси-сервер вписывает свой адрес – SIP URL, – если хочет, чтобы последующие запросы прошли через него.

Заголовок **Content-Type** определяет формат описания сеанса связи. Само описание сеанса, например, в формате протокола SDP, включается в тело сообщения.

Заголовок **Content-Length** указывает размер тела сообщения.

После того, как мы рассмотрели наиболее часто встречающиеся заголовки сообщений протокола SIP, следует обратить внимание на то, что запросы и ответы на них могут включать в себя лишь определенный набор заголовков (Таблица 7.2). Здесь опять буква «М» означает обязательное присутствие заголовка в сообщении, буква «О» – необязательное присутствие, буква «F» запрещает присутствие заголовка.

Таблица 7.2 Связь заголовков с запросами и ответами протокола SIPv2.0

Название заголовка	Место использования заголовка	ACK	BYE	CAN	INV	OPT	REG
Accept	Заголовок в запросах	F	F	F	O	O	O
Accept	Заголовок в ответе 415	F	F	F	O	O	O
Accept-Encoding	Заголовок в запросах	F	F	F	O	O	O
Accept-Encoding	Заголовок в ответе 415	F	F	F	O	O	O
Accept-Language	Заголовок в запросах	F	O	O	O	O	O
Accept-Language	Заголовок в ответе 415	F	O	O	O	O	O
Allow	Заголовок в ответе 200	F	F	F	F	M	F
Allow	Заголовок в ответе 405	O	O	O	O	O	O
Authorization	Заголовок в запросах	O	O	O	O	O	O
Call-ID	Общий заголовок - копируется из запросов в ответы	M	M	M	M	M	M
Contact	Заголовок в запросах	O	F	F	O	O	O
Contact	Заголовок в ответах 1xx	F	F	F	O	O	F
Contact	Заголовок в ответах 2xx	F	F	F	O	O	O
Contact	Заголовок в ответах 3xx	F	O	F	O	O	O
Contact	Заголовок в ответе 485	F	O	F	O	O	O
Content-Encoding	Заголовки содержания	O	F	F	O	O	O
Content-Length	Заголовки содержания	O	F	F	O	O	O
Content-Type	Заголовки содержания	*	F	F	*	*	*
Cseq	Общий заголовок - копируется из запросов в ответы	M	M	M	M	M	M
Date	Заголовок в ответах	O	O	O	O	O	O
Encryption	Заголовок в ответах	O	O	O	O	O	O
Expires	Заголовок в ответах	F	F	F	O	F	O
From	Общий заголовок - копируется из запросов в ответы	M	M	M	M	M	M
Hide	Заголовок в запросах	O	O	O	O	O	O
Max-Forwards	Заголовок в запросах	O	O	O	O	O	O
Organization	Общий заголовок	F	F	F	O	O	O
Proxy-Authenticate	Заголовок в ответе 407	O	O	O	O	O	O
Proxy-Authorization	Заголовок в запросах	O	O	O	O	O	O
Proxy-Require	Заголовок в запросах	O	O	O	O	O	O
Priority	Заголовок в запросах	F	F	F	O	F	F
Require	Заголовок в запросах	O	O	O	O	O	O
Retry-After	Заголовок в запросах	F	F	F	F	F	O
Retry-After	Заголовок в ответах 404, 480, 486, 503, 600 и 603	O	O	O	O	O	O
Response-Key	Заголовок в запросах	F	O	O	O	O	O
Record-Route	Заголовок в запросах	O	O	O	O	O	O
Record-Route	Заголовок в ответах 2xx	O	O	O	O	O	O
Route	Заголовок в запросах	O	O	O	O	O	O
Server	Заголовок в ответах	O	O	O	O	O	O
Subject	Заголовок в запросах	F	F	F	O	F	F
Timestamp	Общий заголовок	O	O	O	O	O	O
To	Общий заголовок - копируется из запросов в ответы	M	M	M	M	M	M
Unsupported	Заголовок в ответе 420	O	O	O	O	O	O
User-Agent	Общий заголовок	O	O	O	O	O	O
Via	Общий заголовок - копируется из запросов в ответы	M	M	M	M	M	M
Warning	Заголовок в ответах	O	O	O	O	O	O
WWW-Authenticate	Заголовок в ответе 401	O	O	O	O	O	O

* Примечание – поле необходимо только в случае, когда тело сообщения содержит какую-либо информацию, т.е. не является пустым.

7.5.3 Запросы

В настоящей версии протокола SIP определено шесть типов запросов. Каждый из них предназначен для выполнения довольно широкого круга задач, что является явным достоинством протокола SIP, так как благодаря этому число сообщений, которыми обмениваются терминалы и серверы, сведено к минимуму. С помощью запросов клиент сообщает о текущем местоположении, приглашает пользователей принять участие в сеансах связи, модифицирует уже установленные сеансы, завершает их и т.д. Сервер определяет тип принятого запроса по названию, указанному в стартовой строке. В той же строке в поле **Request-URI** указан SIP-адрес оборудования, которому этот запрос адресован. Содержание полей **To** и **Request-URI** может различаться, например, в поле **To** может быть указан публикуемый адрес абонента, а в поле **Request-URI** – текущий адрес пользователя.

Запрос **INVITE** приглашает пользователя принять участие в сеансе связи. Он обычно содержит описание сеанса связи, в котором указывается вид принимаемой информации и параметры (список возможных вариантов параметров), необходимые для приема информации, а также может указываться вид информации, которую вызываемый пользователь желает передавать. В ответе на запрос типа **INVITE** указывается вид информации, которая будет приниматься вызываемым пользователем, и, кроме того, может указываться вид информации, которую вызываемый пользователь собирается передавать (возможные параметры передачи информации).

В этом сообщении могут содержаться также данные, необходимые для аутентификации абонента, и, следовательно, доступа клиентов к SIP-серверу. При необходимости изменить характеристики уже организованных каналов передается запрос **INVITE** с новым описанием сеанса связи. Для приглашения нового участника к уже установленному соединению также используется сообщение **INVITE**.

Запрос **ACK** подтверждает прием ответа на запрос **INVITE**. Следует отметить, что запрос **ACK** используется только совместно с запросом **INVITE**, т.е. этим сообщением оборудование вызывающего пользователя показывает, что оно получило окончательный ответ на свой запрос **INVITE**. В сообщении **ACK** может содержаться окончательное описание сеанса связи, передаваемое вызывающим пользователем.

Запрос **CANCEL** отменяет обработку ранее переданных запросов с теми же, что и в запросе **CANCEL**, значениями полей **Call-ID**, **To**, **From** и **CSeq**, но не влияет на те запросы, обработка которых уже завершена. Например, запрос **CANCEL** применяется тогда, когда прокси-сервер размножает запросы для поиска пользователя по нескольким направлениям и в одном из них его находит. Обработку

запросов, разосланных во всех остальных направлениях, сервер отменяет при помощи сообщения **CANCEL**.

Запросом **BYE** оборудование вызываемого или вызывающего пользователя завершает соединение. Сторона, получившая запрос **BYE**, должна прекратить передачу речевой (мультимедийной) информации и подтвердить его выполнение ответом **200 OK**.

При помощи запроса типа **REGISTER** пользователь сообщает свое текущее местоположение. В этом сообщении содержатся следующие поля:

- Поле **To** содержит адресную информацию, которую надо сохранить или модифицировать на сервере;
- Поле **From** содержит адрес инициатора регистрации. Зарегистрировать пользователя может либо он сам, либо другое лицо, например, секретарь может зарегистрировать своего начальника;
- Поле **Contact** содержит новый адрес пользователя, по которому должны передаваться все дальнейшие запросы **INVITE**. Если в запросе **REGISTER** поле **Contact** отсутствует, то регистрация остается прежней. В случае отмены регистрации здесь помещается символ «*»;
- В поле **Expires** указывается время в секундах, в течение которого регистрация действительна. Если данное поле отсутствует, то по умолчанию назначается время – 1 час, после чего регистрация отменяется. Регистрацию можно также отменить, передав сообщение **REGISTER** с полем **Expires**, которому присвоено значение 0, и с соответствующим полем **Contact**.

Запросом **OPTIONS** вызываемый пользователь запрашивает информацию о функциональных возможностях терминального оборудования вызываемого пользователя. В ответ на этот запрос оборудование вызываемого пользователя сообщает требуемые сведения. Применение запроса **OPTIONS** ограничено теми случаями, когда необходимо узнать о функциональных возможностях оборудования до установления соединения. Для установления соединения запрос этого типа не используется.

После испытаний протокола SIP в реальных сетях оказалось, что для решения ряда задач вышеуказанных шести типов запросов недостаточно. Поэтому возможно, что в протокол будут введены новые сообщения. Так, в текущей версии протокола SIP не предусмотрен способ передачи информации управления соединением или другой информации во время сеанса связи. Для решения этой задачи был предложен новый тип запроса – **INFO**. Он может использоваться в следующих случаях:

- для переноса сигнальных сообщений ТфОП/ISDN/сотовых сетей между шлюзами в течение разговорной сессии;

- для переноса сигналов DTMF в течение разговорной сессии;
- для переноса биллинговой информации.

Завершив описание запросов протокола SIP, рассмотрим, в качестве примера, типичный запрос типа **INVITE** (рис. 7.6).

```
INVITE sip: watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip: a.g.bell@bell-tel.com>
To: T. Watson <sip: watson@bell-tel.com>
Call-ID: 3298420296@kton.bell-tel.com
Cseq: 1 INVITE
Content-Type: application/sdp
Content-Length: ...

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
C=IN IP4 kton.bell-tel.com
m=audio 3456 RTP/AVP 0 3 4 5
```

Рис. 7.6 Пример запроса INVITE

В этом примере пользователь Bell (a.g.bell@bell-tel.com) вызывает пользователя Watson (watson@bell-tel.com). Запрос передается к прокси-серверу (boston.bell-tel.com). В полях **To** и **From** перед адресом стоит запись, которую вызывающий пользователь желает вывести на дисплей вызываемого пользователя. В теле сообщения оборудование вызываемого пользователя указывает в формате протокола SDP, что оно может принимать в порту 3456 речевую информацию, упакованную в пакеты RTP и закодированную по одному из следующих алгоритмов кодирования: 0 – PCMU, 3 – GSM, 4 – G.723 и 5 – DVI4.

При передаче сообщений протокола SIP, упакованных в сигнальные сообщения протокола UDP, существует вероятность того, что размер запроса или ответа окажется больше максимально допустимого для данной сети, и произойдет фрагментация пакета. Чтобы избежать этого, используется сжатый формат имен основных заголовков, подобно тому, как это делается в протоколе SDP. Ниже приведен список таких заголовков (Таблица 7.3).

Таблица 7.3 Сжатые имена заголовков

Сжатая форма имени	Полная форма имени
c	Content-Type
e	Content-Encoding
f	From
i	Call-ID
m	Contact (от "moved")
l	Content-Length
s	Subject
t	To
v	Via

При написании имен заголовков в сжатом виде сообщение **INVITE**, показанное ранее на рисунке 6, будет выглядеть следующим образом (рис. 7.7):

```
INVITE sip: watson@boston.bell-tel.com SIP/2.0
v: SIP/2.0/UDP kton.bell-tel.com
f: A. Bell <sip: a.g.bell@bell-tel.com>
t: T. Watson <sip: watson@bell-tel.com>
i: 3298420296@kton.bell-tel.com
Cseq: 1 INVITE
c: application/sdp
l: ...

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
C=IN IP4 kton.bell-tel.com
m=audio 3456 RTP/AVP 0 3 4 5
```

Рис. 7.7 Пример запроса INVITE с сокращенными заголовками

В заключение параграфа, как и в предыдущих главах, сведем все запросы, с их кратким описанием, в таблицу 7.4.

Таблица 7.4 Запросы SIP

Тип запроса	Описание запроса
INVITE	Приглашает пользователя к сеансу связи. Содержит SDP-описание сеанса
ACK	Подтверждает прием окончательного ответа на запрос INVITE
BYE	Завершает сеанс связи. Может быть передан любой из сторон, участвующих в сеансе
CANCEL	Отменяет обработку запросов с теми же заголовками Call-ID , To , From и CSeq , что и в самом запросе CANCEL
REGISTER	Переносит адресную информацию для регистрации пользователя на сервере определения местоположения
OPTION	Запрашивает информацию о функциональных возможностях терминала

7.5.4 Ответы на запросы

После приема и интерпретации запроса, адресат (прокси-сервер) передает ответ на этот запрос. Содержание ответов бывает разным: подтверждение установления соединения, передача запрошенной информации, сведения о неисправностях и т.д. Структуру ответов и их виды протокол SIP унаследовал от протокола HTTP.

Определено шесть типов ответов, несущих разную функциональную нагрузку. Тип ответа кодируется трехзначным числом. Самой важной является первая цифра, которая определяет класс ответа, остальные две цифры лишь дополняют первую. В некоторых случаях оборудование даже может не знать все коды ответов, но оно обязательно должно интерпретировать первую цифру ответа.

Все ответы делятся на две группы: информационные и финальные. Информационные ответы показывают, что запрос находится в стадии обработки. Они кодируются трехзначным числом, начинающимся с единицы, – **1xx**. Некоторые информационные ответы, например, **100 Trying**, предназначены для установки на нуль таймеров, которые запускаются в оборудовании, передавшем запрос. Если к моменту срабатывания таймера ответ на запрос не получен, то считается, что этот запрос потерян и может (по усмотрению производителя) быть передан повторно. Один из распространенных ответов – **180 Ringing**; по назначению он идентичен сигналу «Контроль посылки вызова» в ТфОП и означает, что вызываемый пользователь получает сигнал о входящем вызове.

Финальные ответы кодируются трехзначными числами, начинающимися с цифр 2, 3, 4, 5 и 6. Они означают завершение обработки запроса и содержат, когда это нужно, результат обработки запроса. Назначение финальных ответов каждого типа рассматривается ниже.

Ответы **2xx** означают, что запрос был успешно обработан. В настоящее время из всех ответов типа **2xx** определен лишь один – **200 OK**. Его значение зависит от того, на какой запрос он отвечает:

- ответ **200 OK** на запрос **INVITE** означает, что вызываемое оборудование согласно на участие в сеансе связи; в теле ответа указываются функциональные возможности этого оборудования;
- ответ **200 OK** на запрос **BYE** означает завершение сеанса связи, в теле ответа никакой информации не содержится;
- ответ **200 OK** на запрос **CANCEL** означает отмену поиска, в теле ответа никакой информации не содержится;
- ответ **200 OK** на запрос **REGISTER** означает, что регистрация прошла успешно;
- ответ **200 OK** на запрос **OPTION** служит для передачи сведений о функциональных возможностях оборудования, эти сведения содержатся в теле ответа.

Ответы **3xx** информируют оборудование вызывающего пользователя о новом местоположении вызываемого пользователя или переносят другую информацию, которая может быть использована для нового вызова:

- в ответе **300 Multiple Choices** указывается несколько SIP-адресов, по которым можно найти вызываемого пользователя, и вызывающему пользователю предлагается выбрать один из них;
- ответ **301 Moved Permanently** означает, что вызываемый пользователь больше не находится по адресу, указанному в запросе, и направлять запросы нужно на адрес, указанный в поле Contact;
- ответ **302 Moved Temporary** означает, что пользователь временно (промежуток времени может быть указан в поле Expires) находится по другому адресу, который указывается в поле Contact.

Ответы **4xx** информируют о том, что в запросе обнаружена ошибка. После получения такого ответа пользователь не должен передавать тот же самый запрос без его модификации:

- ответ **400 Bad Request** означает, что запрос не понят из-за наличия в нем синтаксических ошибок;
- ответ **401 Unauthorized** означает, что запрос требует проведения процедуры аутентификации пользователя. Существуют разные варианты аутентификации, и в ответе может быть указано, какой из них использовать в данном случае;
- ответ **403 Forbidden** означает, что сервер понял запрос, но отказался его обслуживать. Повторный запрос посылать не следует. Причины могут быть разными, например, запросы с этого адреса не обслуживаются и т.д.;
- ответ **485 Ambiguous** означает, что адрес в запросе не определяет вызываемого пользователя однозначно;
- ответ **486 Busy Here** означает, что вызываемый пользователь в настоящий момент не может принять входящий вызов по данному адресу. Ответ не исключает возможности связаться с пользователем по другому адресу или, к примеру, оставить сообщение в речевом почтовом ящике.

Ответы **5xx** информируют о том, что запрос не может быть обработан из-за отказа сервера:

- ответ **500 Server Internal Error** означает, что сервер не имеет возможности обслужить запрос из-за внутренней ошибки. Клиент может попытаться повторно послать запрос через некоторое время;
- ответ **501 Not Implemented** означает, что в сервере не реализованы функции, необходимые для обслуживания этого запроса. Ответ передается, например в том случае, когда сервер не может распознать тип запроса;
- ответ **502 Bad Gateway** информирует о том, что сервер, функционирующий в качестве шлюза или прокси-сервера, принял некорректный ответ от сервера, к которому он направил запрос;
- ответ **503 Service Unavailable** говорит о том, что сервер не может в данный момент обслужить вызов вследствие перегрузки или проведения технического обслуживания.

Ответы **6xx** информируют о том, что соединение с вызываемым пользователем установить невозможно:

- ответ **600 Busy Everywhere** сообщает, что вызываемый пользователь занят и не может принять вызов в данный момент ни по одному из имеющихся у него адресов. Ответ может указывать время, подходящее для вызова пользователя;

- ответ **603 Decline** означает, что вызываемый пользователь не может или не желает принять входящий вызов. В ответе может быть указано подходящее для вызова время;
- ответ **604 Does Not Exist Anywhere** означает, что вызываемого пользователя не существует.

Напомним, что запросы и ответы на них образуют SIP-транзакцию. Она осуществляется между клиентом и сервером и включает в себя все сообщения, начиная с первого запроса и заканчивая финальным ответом. При использовании в качестве транспорта протокола TCP все запросы и ответы, относящиеся к одной транзакции, передаются по одному TCP-соединению.

На рисунке 7.8 представлен пример ответа на запрос **INVITE**.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: <sip:watson@bell-tel.com>;
Call-ID: 3298420296@kton.bell-tel.com
Cseq: 1 INVITE
Content-Type: application/sdp
Content-Length: ...

v=0
o=watson 4858949 4858949 IN IP4 192.1.2.3
t=3149329600 0
c=IN IP4 boston.bell-tel.com
m=audio 5004 RTP/AVP 0 3
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
```

Рис. 7.8 Пример SIP-ответа 200 OK

В этом примере приведен ответ пользователя Watson на приглашение принять участие в сеансе связи, полученное от пользователя Bell. Наиболее вероятный формат приглашения рассмотрен нами ранее (рис. 7.7). Вызываемая сторона информирует вызывающую о том, что она может принимать в порту 5004 речевую информацию, закодированную в соответствии с алгоритмами кодирования PCMU, GSM. Поля **From**, **To**, **Via**, **Call-ID** взяты из запроса, показанного на рисунке 7.7. Из примера видно, что это ответ на запрос **INVITE** с полем **CSeq: 1**.

После того, как мы рассмотрели запросы и ответы на них, можно отметить, что протокол SIP предусматривает разные алгоритмы установления соединения. При этом стоит обратить внимание, что одни и те же ответы можно интерпретировать по-разному в зависимости от конкретной ситуации. В таблице 7.5 сведены все ответы на запросы, определенные протоколом SIP.

Таблица 7.5 Ответы SIP

Код ответа	Пояснение	Назначение
100	Trying	Запрос обрабатывается, например, сервер обращается к базам данных, но местоположение вызываемого пользователя в настоящий момент не определено
180	Ringing	Местоположение вызываемого пользователя определено. Ему дается сигнал о входящем вызове
181	Call Is Being Forwarded	Прокси-сервер переадресует вызов к другому пользователю
182	Queued	Вызываемый пользователь временно не доступен, но входящий вызов поставлен в очередь. Когда вызываемый пользователь станет доступным, он передаст финальный ответ
200	OK	Команда успешно выполнена
300	Multiple Choices	Вызываемый пользователь доступен по нескольким адресам. Вызывающий пользователь может выбрать любой из них
301	Moved Permanently	Пользователь изменил свое местоположение, его новый адрес указан в поле Contact
302	Moved Temporarily	Пользователь временно изменил свое местоположение, его новый адрес указан в поле Contact
305	Use Proxy	Вызываемая сторона может принять входящий вызов только в том случае, когда он проходит через прокси-сервер. Вызывающей стороне рекомендуется обратиться к прокси-серверу, адрес которого указан в поле Contact. Ответ передается только терминальным оборудованием (UAS)
380	Alternative Service	Вызов не достиг адресата, но существует альтернативный вариант обслуживания, который указан в теле ответа. Например, вызов может быть переадресован к речевому почтовому ящику
400	Bad Request	В запросе обнаружена синтаксическая ошибка
401	Unauthorised	Требуется проведение процедуры авторизации пользователя
402	Payment Required	Требуется предварительная оплата услуг
403	Forbidden	Запрос не будет обслуживаться сервером и не должен передаваться повторно
404	Not Found	Сервер не обнаружил вызываемого пользователя в домене, указанном в поле Request-URI
405	Method Not Allowed	Не разрешается передавать запрос этого типа на адрес, указанный в поле Request-URI. В поле Allow ответа указываются разрешенные типы запросов
406	Not Acceptable	Ответы, генерируемые вызываемой стороной, не будут поняты вызывающей стороной
407	Proxy Authentication Required	Клиент должен подтвердить свое право доступа к прокси-серверу
408	Request Timeout	Сервер не может передать ответ, например, указать местоположение вызываемого пользователя, в течение промежутка времени, специфицированного в поле Expires запроса. Вызывающий пользователь может повторно передать запрос через некоторое время
409	Conflict	Обработка запроса REGISTER не может быть завершена из-за конфликта между действием, определенным в параметре action запроса, и текущим состоянием ресурсов
410	Gone	Сервер больше не имеет доступа к запрашиваемому ресурсу и не знает, куда переадресовать запрос
411	Length Required	Требуется указать длину тела сообщения в поле Content-Length

413	Request Entity Too Large	Размер запроса слишком велик для обработки
414	Request-URI Too Large	Адрес, указанный в поле Request-URI, оказался слишком большим, поэтому его интерпретация невозможна
415	Unsupported Media Type	Запрос содержит не поддерживаемый формат тела сообщения
420	Bad Extension	Сервер не понял расширение протокола, специфицированное в поле Require
480	Temporarily not available	Вызываемый пользователь временно недоступен
481	Call Leg/Transaction Does Not Exist	Посылается в ответ на получение запроса BYE, не относящегося к текущим соединениям, или запроса CANCEL, не относящегося к текущим запросам
482	Loop Detected	Сервер обнаружил, что принятый им запрос передается по замкнутому маршруту (в поле Via уже имеется адрес этого сервера)
483	Too Many Hops	Сервер обнаружил в поле Via, что принятый им запрос прошел через большее количество прокси-серверов, чем разрешено в поле Max-Forwards
484	Address Incomplete	Сервер принял запрос с неполным адресом в поле To или Request-URI. Требуется дополнительная адресная информация
485	Ambiguous	Адрес вызываемого пользователя неоднозначен. В заголовке Contact ответа может содержаться список адресов, по которым этот запрос можно передать
486	Busy Here	В настоящий момент вызываемый пользователь не желает или не может принять вызов на этот адрес. Ответ не исключает возможности связаться с пользователем по другому адресу
500	Internal Server Error	Внутренняя ошибка сервера
501	Not Implemented	В сервере не реализованы функции, необходимые для обслуживания запроса. Ответ передается в том случае, когда сервер не может распознать тип полученного им запроса
502	Bad Gateway	Сервер, функционирующий в качестве шлюза или прокси-сервера, принимает некорректный ответ от сервера, к которому он направил запрос
503	Service Unavailable	Сервер не может в данный момент обслужить вызов вследствие перегрузки или проведения технического обслуживания
504	Gateway Time-out	Сервер, функционирующий в качестве шлюза или прокси-сервера, в течение установленного интервала времени не получил ответ от сервера (например, от сервера определения местоположения), к которому он обратился для завершения обработки запроса
505	SIP Version not supported	Сервер не поддерживает данную версию протокола SIP
600	Busy Everywhere	Вызываемый пользователь занят и не желает принимать вызов в данный момент. Ответ может указывать подходящее для вызова время
603	Decline	Вызываемый пользователь не может или не желает принимать входящие вызовы. В ответе может быть указано подходящее для вызова время
604	Does not exist anywhere	Вызываемого пользователя не существует
606	Not Acceptable	Вызываемый пользователь не может принять входящий вызов из-за того, что вид информации, указанный в описании сеанса связи в формате SDP, полоса пропускания и т.д. неприемлемы

7.6 Алгоритмы установления соединения

Протоколом SIP предусмотрены 3 основных сценария установления соединения: с участием прокси-сервера, с участием сервера переадресации и непосредственно между пользователями. Различие между перечисленными сценариями заключается в том, что по-разному осуществляется поиск и приглашение вызываемого пользователя. В первом случае эти функции возлагает на себя прокси-сервер, а вызывающему пользователю необходимо знать только постоянный SIP-адрес вызываемого пользователя. Во втором случае вызывающая сторона самостоятельно устанавливает соединение, а сервер переадресации лишь реализует преобразование постоянного адреса вызываемого абонента в его текущий адрес. И, наконец, в третьем случае вызывающему пользователю для установления соединения необходимо знать текущий адрес вызываемого пользователя.

Перечисленные сценарии являются простейшими. Ведь прежде чем вызов достигнет адресата, он может пройти через несколько прокси-серверов, или сначала направляется к серверу переадресации, а затем проходит через один или несколько прокси-серверов. Кроме того, прокси-серверы могут размножать запросы и передавать их по разным направлениям и т.д. Но, все же, как уже было уже отмечено в начале параграфа, эти три сценария являются основными. Здесь мы рассмотрим подробно два первых сценария; третий сценарий в данной главе рассматриваться не будет.

7.6.1 Установление соединения с участием сервера переадресации

В этом параграфе описан алгоритм установления соединения с участием сервера переадресации вызовов. Администратор сети сообщает пользователям адрес сервера переадресации. Вызывающий пользователь передает запрос **INVITE** (1) на известный ему адрес сервера переадресации и порт 5060, используемый по умолчанию (Рисунок 7.9). В запросе вызывающий пользователь указывает адрес вызываемого пользователя. Сервер переадресации запрашивает текущий адрес нужного пользователя у сервера определения местоположения (2), который сообщает ему этот адрес (3). Сервер переадресации в ответе **302 Moved temporarily** передает вызывающей стороне текущий адрес вызываемого пользователя (4), или он может сообщить список зарегистрированных адресов вызываемого пользователя и предложить вызывающему пользователю самому выбрать один из них. Вызывающая сторона подтверждает прием ответа **302** посылкой сообщения **ACK** (5).

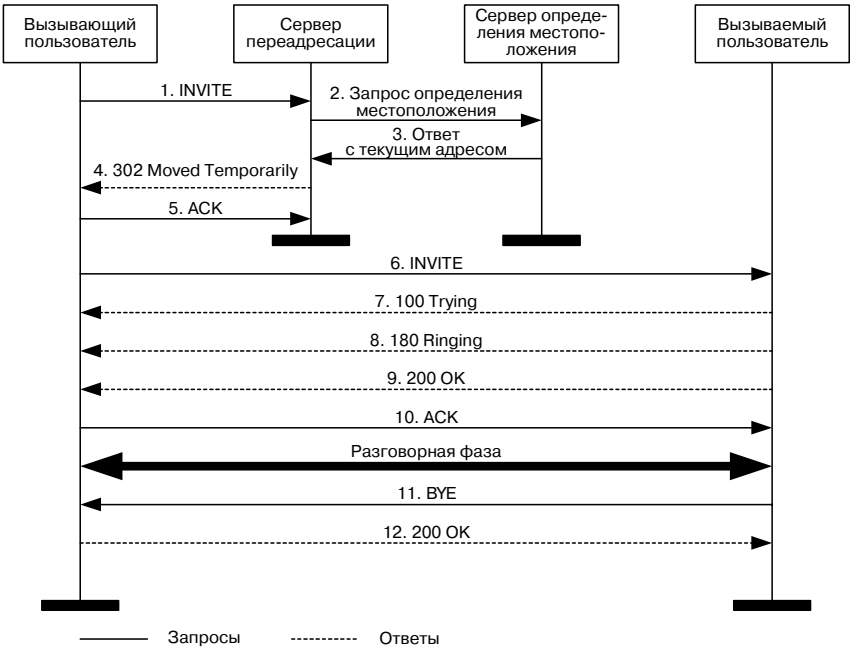


Рис. 7.9 Сценарий установления соединения через сервер переадресации

Теперь вызывающая сторона может связаться непосредственно с вызываемой стороной. Для этого она передает новый запрос **INVITE** (6) с тем же идентификатором **Call-ID**, но другим номером **CSeq**. В теле сообщения **INVITE** указываются данные о функциональных возможностях вызывающей стороны в формате протокола SDP. Вызываемая сторона принимает запрос **INVITE** и начинает его обработку, о чем сообщает ответом **100 Trying** (7) встречному оборудованию для перезапуска его таймеров. После завершения обработки поступившего запроса оборудование вызываемой стороны сообщает своему пользователю о входящем вызове, а встречной стороне передает ответ **180 Ringing** (8). После приема вызываемым пользователем входящего вызова удаленной стороне передается сообщение **200 OK** (9), в котором содержатся данные о функциональных возможностях вызываемого терминала в формате протокола SDP. Терминал вызывающего пользователя подтверждает прием ответа запросом **ACK** (10). На этом фаза установления соединения закончена и начинается разговорная фаза.

По завершении разговорной фазы любой из сторон передается запрос **BYE** (11), который подтверждается ответом **200 OK** (12).

7.6.2. Установление соединения с участием прокси-сервера

В этом параграфе описан алгоритм установления соединения с участием прокси-сервера. Администратор сети сообщает адрес

этого сервера пользователям. Вызывающий пользователь передает запрос **INVITE** (1) на адрес прокси-сервера и порт 5060, используемый по умолчанию (Рисунок 7.10). В запросе пользователь указывает известный ему адрес вызываемого пользователя. Прокси-сервер запрашивает текущий адрес вызываемого пользователя у сервера определения местоположения (2), который и сообщает ему этот адрес (3). Далее прокси-сервер передает запрос **INVITE** непосредственно вызываемому оборудованию (4). Опять в запросе содержатся данные о функциональных возможностях вызывающего терминала, но при этом в запрос добавляется поле **Via** с адресом прокси-сервера для того, чтобы ответы на обратном пути шли через него. После приема и обработки запроса вызываемое оборудование сообщает своему пользователю о входящем вызове, а встречной стороне передает ответ **180 Ringing** (5), копируя в него из запроса поля **To, From, Call-ID, CSeq** и **Via**. После приема вызова пользователем встречной стороне передается сообщение **200 OK** (6), содержащее данные о функциональных возможностях вызываемого терминала в формате протокола SDP. Терминал вызывающего пользователя подтверждает прием ответа запросом **ACK** (7). На этом фаза установления соединения закончена и начинается разговорная фаза.

По завершении разговорной фазы одной из сторон передается запрос **BYE** (8), который подтверждается ответом **200 OK** (9).

Все сообщения проходят через прокси-сервер, который может модифицировать в них некоторые поля.

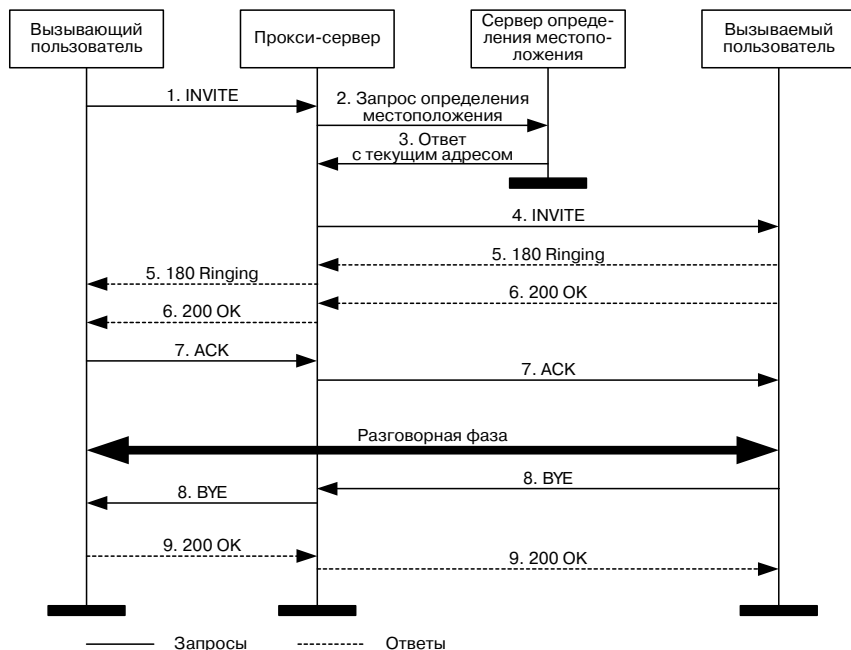


Рис. 7.10 Сценарий установления соединения через прокси-сервер

7.7 Реализация дополнительных услуг на базе протокола SIP

В этом параграфе рассматриваются примеры реализации дополнительных услуг на базе протокола SIP.

Дополнительная услуга «Переключение связи» позволяет пользователю переключить установленное соединение к третьей стороне. На рисунке 7.11 приведен пример реализации этой услуги. Пользователь В устанавливает связь с пользователем А, который, переговорив с В, переключает эту связь к пользователю С, а сам отключается.

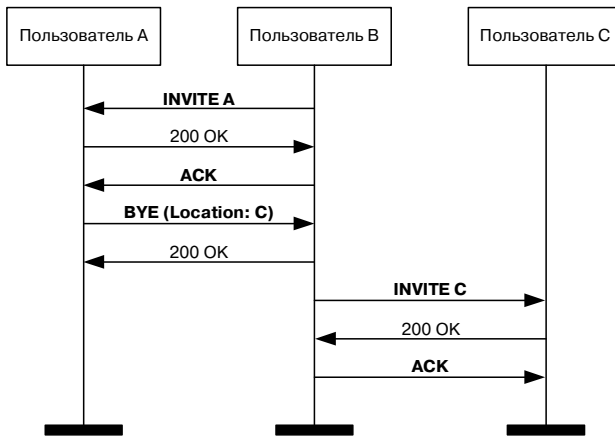


Рис. 7.11 Дополнительная услуга "Переключение связи"

Дополнительная услуга «Переадресация вызова» позволяет пользователю назначить адрес, на который, при определенных условиях, следует направлять входящие к нему вызовы. Такими условиями могут быть занятость пользователя, отсутствие его ответа в течение заданного времени или и то, и другое; возможна также безусловная переадресация. Оборудование пользователя, заказавшего эту услугу, получив сообщение **INVITE B**, проверяет условия, в которых оно получено, и если условия требуют переадресации, передает сообщение **INVITE** с заголовком **Also**, указывая в нем адрес пользователя, к которому следует направить вызов. Терминал вызывающего пользователя, получив сообщение **INVITE** с таким заголовком, инициирует новый вызов по адресу, указанному в поле **Also**. В нашем случае пользователь А вызывает пользователя В, а терминал последнего переадресует вызов к пользователю С (Рисунок 7.12).

Дополнительная услуга «Уведомление о вызове во время связи» позволяет пользователю, участвующему в телефонном разговоре, получить уведомление о том, что к нему поступил входящий вызов (Рисунок 7.13).

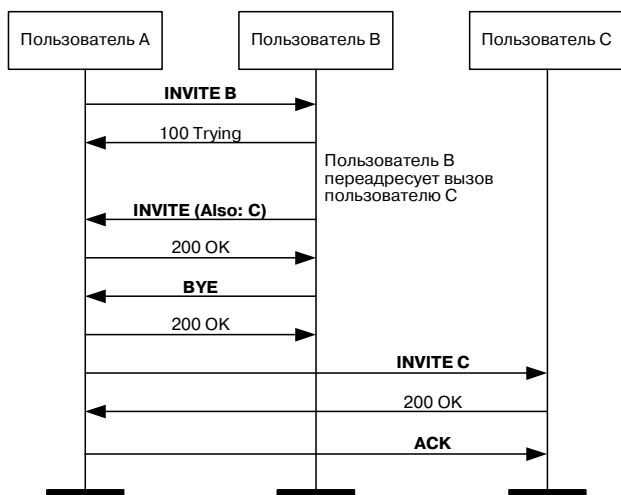


Рис. 7.12 Дополнительная услуга "Переадресация вызова"

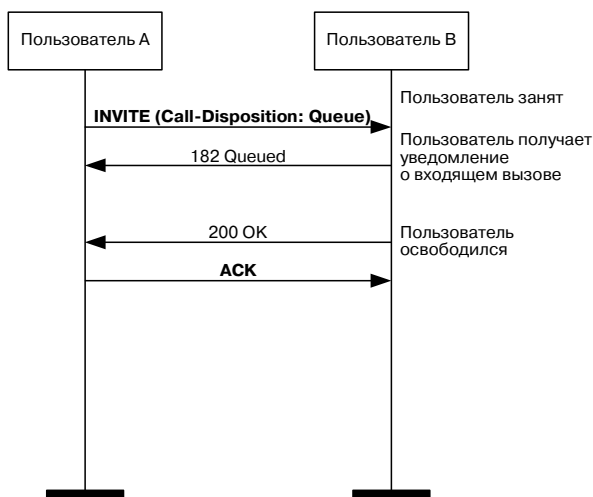


Рис. 7.13 Дополнительная услуга "Уведомление о вызове во время связи"

Услуга реализуется с помощью заголовка **Call-Disposition**, в котором содержится инструкция по обслуживанию вызова. Вызывающий пользователь передает запрос **INVITE** с заголовком **Call-Disposition: Queue**, который интерпретируется следующим образом: вызывающий пользователь хочет, чтобы вызов был поставлен в очередь, если вызываемый пользователь будет занят. Вызываемая сторона подтверждает исполнение запроса ответом **182 Queued**, который может передаваться неоднократно в течение периода ожидания. Вызываемый пользователь получает уведомление о входящем вызове, а когда он освобождается, вызывающей стороне передается финальный ответ **200 OK**.

7.8 Сравнительный анализ протоколов H.323 и SIP

Прежде чем начать сравнение функциональных возможностей протоколов SIP и H.323, напомним, что протокол SIP значительно моложе своего соперника, и опыт его использования в сетях связи несопоставим с опытом использования протокола H.323. Существует еще один момент, на который следует обратить внимание. Интенсивное внедрение технологии передачи речевой информации по IP-сетям потребовало постоянного наращивания функциональных возможностей как протокола H.323 (к настоящему времени утверждена уже третья версия протокола), так и протокола SIP (утверждена вторая версия протокола). Этот процесс приводит к тому, что достоинства одного из протоколов перенимаются другим.

И последнее. Оба протокола являются результатом решения одних и тех же задач специалистами ITU-T и комитета IETF. Естественно, что решение ITU-T оказалось ближе к традиционным телефонным сетям, а решение комитета IETF базируется на принципах, составляющих основу сети Internet.

Перейдем непосредственно к сравнению протоколов, которое будем проводить по нескольким критериям.

Дополнительные услуги. Набор услуг, поддерживаемых обоими протоколами, примерно одинаков.

Дополнительные услуги, предоставляемые протоколом H.323, стандартизированы в серии рекомендаций ITU-T H.450.x. Протоколом SIP правила предоставления дополнительных услуг не определены, что является его серьезным недостатком, так как вызывает проблемы при организации взаимодействия оборудования разных фирм-производителей. Некоторые специалисты предлагают решения названных проблем, но эти решения пока не стандартизированы.

Примеры услуг, предоставляемых обоими протоколами:

- Перевод соединения в режим удержания (Call hold);
- Переключение связи (Call Transfer);
- Переадресация (Call Forwarding);
- Уведомление о новом вызове во время связи (Call Waiting);
- Конференция.

Рассмотрим последнюю услугу несколько более подробно. Протокол SIP предусматривает три способа организации конференции: с использованием устройства управления конференциями MCU, режима многоадресной рассылки и соединений участников друг с другом. В последних двух случаях функции управления конференциями могут быть распределены между терминалами, т.е. центральный контроллер конференций не нужен. Это позволяет организовывать конференции с практически неограниченным количеством участников.

Рекомендация H.323 предусматривает те же три способа, но управление конференцией во всех случаях производится централизованно контроллером конференций MC (Multipoint Controller), который обрабатывает все сигнальные сообщения. Поэтому для организации конференции, во-первых, необходимо наличие контроллера MC у одного из терминалов, во-вторых, участник с активным контроллером MC не может выйти из конференции. Кроме того, при большом числе участников конференции MC может стать «узким местом». Правда, в третьей версии рекомендации ITU-T H.323 принято положение о каскадном соединении контроллеров, однако производители эту версию в своем оборудовании пока не реализовали. Преимуществом протокола H.323 в части организации конференций являются более мощные средства контроля конференций.

Протокол SIP изначально ориентирован на использование в IP-сетях с поддержкой режима многоадресной рассылки информации (примером может служить сеть Mbone, имеющая тысячи постоянных пользователей). Этот механизм используется в протоколе SIP не только для доставки речевой информации (как в протоколе H.323), но и для переноса сигнальных сообщений. Например, в режиме многоадресной рассылки может передаваться сообщение **INVITE**, что облегчает определение местоположения пользователя и является очень удобным для центров обслуживания вызовов (Call-center) при организации групповых оповещений.

В то же время, протокол H.323 предоставляет больше возможностей управления услугами, как в части аутентификации и учета, так и в части контроля использования сетевых ресурсов. Возможности протокола SIP в этой части беднее, и выбор оператором этого протокола может служить признаком того, что для оператора важнее техническая интеграция услуг, чем возможности управления услугами.

Протокол SIP предусматривает возможность организации связи третьей стороной (third-party call control). Эта функция позволяет реализовать такие услуги, как набор номера секретарем для менеджера и сопровождение вызова оператором центра обслуживания вызовов. Подобные услуги предусмотрены и протоколом H.323, но реализация их несколько сложнее.

В протоколе SIP есть возможность указывать приоритеты в обслуживании вызовов, поскольку во многих странах существуют требования предоставлять преимущества некоторым пользователям. В протоколе H.323 такой возможности нет. Кроме того, пользователь SIP-сети может регистрировать несколько своих адресов и указывать приоритетность каждого из них.

Персональная мобильность пользователей. Протокол SIP имеет хороший набор средств поддержки персональной мобильности пользователей, в число которых входит переадресация вызова к новому местоположению пользователя, одновременный поиск по не-

скольким направлениям (с обнаружением заикливания маршрутов) и т.д. В протоколе SIP это организуется путем регистрации на сервере определения местоположения, взаимодействие с которым может поддерживаться любым протоколом. Персональная мобильность поддерживается и протоколом H.323, но менее гибко. Так, например, одновременный поиск пользователя по нескольким направлениям ограничен тем, что привратник, получив запрос определения местоположения пользователя LRQ, не транслирует его к другим привратникам.

Расширяемость протокола. Необходимой и важной в условиях эволюционирующего рынка является возможность введения новых версий протоколов и обеспечение совместимости различных версий одного протокола. Расширяемость (extensibility) протокола обеспечивается:

- согласованием параметров;
- стандартизацией кодеков;
- модульностью архитектуры.

Протокол SIP достаточно просто обеспечивает совместимость разных версий. Поля, которые не понятны оборудованию, просто игнорируются. Это уменьшает сложность протокола, а также облегчает обработку сообщений и внедрение новых услуг. Клиент может запросить какую-либо услугу с помощью заголовка **Require**. Сервер, получивший запрос с таким заголовком, проверяет, поддерживает ли он эту услугу, и если не поддерживает, то сообщает об этом в своем ответе, содержащем список поддерживаемых услуг.

В случае необходимости, в организации IANA (Internet Assigned Numbers Authority) могут быть зарегистрированы новые заголовки. Для регистрации в IANA отправляется запрос с именем заголовка и его назначением. Название заголовка выбирается таким образом, чтобы оно говорило об его назначении. Указанным образом разработчик может внедрять новые услуги.

Для обеспечения совместимости версий протокола SIP определено шесть основных видов запросов и 6 классов ответов на запросы. Так как определяющей в кодах ответов является первая цифра, то оборудование может указывать и интерпретировать только ее, а остальные цифры кода только дополняют смысл и их анализ не является обязательным.

Более поздние версии протокола H.323 должны поддерживать более ранние версии. Но возможна ситуация, когда производители поддерживают только одну версию, чтобы уменьшить размер сообщений и облегчить их декодирование.

Новые функциональные возможности вводятся в протокол H.323 с помощью поля **NonStandardParameter**. Оно содержит код произ-

водителя и, следом за ним, код услуги, который действителен только для этого производителя. Это позволяет производителю расширять услуги, но сопряжено с некоторыми ограничениями. Во-первых, невозможно запросить у вызываемой стороны информацию о поддерживаемых ею услугах, во-вторых, невозможно добавить новое значение уже существующего параметра. Существуют также проблемы, связанные с обеспечением взаимодействия оборудования разных производителей.

На расширение возможностей протокола, как и на совместимость оборудования, его реализующего, оказывает влияние и набор кодеков, поддерживаемый протоколом. В протоколе SIP для передачи информации о функциональных возможностях терминала используется протокол SDP. Если производитель поддерживает какой-то особенный алгоритм кодирования, то этот алгоритм просто регистрируется в организации IANA, неоднократно упоминавшейся в этой главе.

В протоколе H.323 все кодеки должны быть стандартизированы. Поэтому приложения с нестандартными алгоритмами кодирования могут столкнуться с проблемами при реализации их на базе протокола H.323.

Протокол SIP состоит из набора законченных компонентов (модулей), которые могут заменяться в зависимости от требований и могут работать независимо друг от друга. Этот набор включает в себя модули поддержки сигнализации для базового соединения, для регистрации и для определения местоположения пользователя, которые не зависят от модулей поддержки качества обслуживания (QoS), работы с директориями, описания сеансов связи, развертывания услуг (service discovery) и управления конфигурацией.

Архитектура протокола H.323 монолитна и представляет собой интегрированный набор протоколов для одного применения. Протокол состоит из трех основных составляющих, и для создания новой услуги может потребоваться модификация каждой из этих составляющих.

Масштабируемость сети (scalability). Сервер SIP, по умолчанию, не хранит сведений о текущих сеансах связи и поэтому может обработать больше вызовов, чем привратник H.323, который хранит эти сведения (statefull). Вместе с тем, отсутствие таких сведений, по мнению некоторых специалистов, может вызвать трудности при организации взаимодействия сети IP-телефонии с ТФОП.

Необходимо также иметь в виду зоновую архитектуру сети H.323, позволяющую обеспечить расширяемость сети путем увеличения количества зон.

Время установления соединения. Следующей существенной характеристикой протоколов является время, которое требуется, чтобы установить соединение. В запросе **INVITE** протокола SIP содержится вся необходимая для установления соединения информация, включая описание функциональных возможностей терминала. Таким образом, в протоколе SIP для установления соединения требуется одна транзакция, а в протоколе H.323 необходимо производить обмен сообщениями несколько раз. По этим причинам затраты времени на установление соединения в протоколе SIP значительно меньше затрат времени в протоколе H.323. Правда, при использовании инкапсуляции сообщений H.245 в сообщения H.225 или процедуры Fast Connect время установления соединения значительно уменьшается.

Кроме того, на время установления соединения влияет также и нижележащий транспортный протокол, переносящий сигнальную информацию. Ранние версии протокола H.323 предусматривали использование для переноса сигнальных сообщений H.225 и H.245 только протокол TCP, и лишь третья версия протокола предусматривает возможность использования протокола UDP. Протоколом SIP использование протоколов TCP и UDP предусматривалось с самого начала.

Оценка времени установления соединения производится в условных единицах – RTT (round trip time) – и составляет для протокола SIP $1,5 \div 2,5$ RTT, а для протокола H.323 $6 - 7$ RTT.

Адресация. К числу системных характеристик, несомненно, относится и предусматриваемая протоколами адресация. Использование URL является сильной стороной протокола SIP и позволяет легко интегрировать его в существующую систему DNS-серверов и внедрять в оборудование, работающее в IP-сетях. Пользователь получает возможность переправлять вызовы на Web-страницы или использовать электронную почту. Адресом в SIP может также служить телефонный номер с адресом используемого шлюза.

В протоколе H.323 используются транспортные адреса и alias-адреса. В качестве последнего может использоваться телефонный номер, имя пользователя или адрес электронной почты. Для преобразования alias-адреса в транспортный адрес обязательно участие привратника.

Сложность протокола. Протокол H.323, несомненно, сложнее протокола SIP. Общий объем спецификаций протокола H.323 составляет примерно 700 страниц. Объем спецификаций протокола SIP составляет 150 страниц. Протокол H.323 использует большое количество информационных полей в сообщениях (до 100), при нескольких десятках таких же полей в протоколе SIP. При этом для организации базового соединения в протоколе SIP достаточно использовать всего три типа запросов (**INVITE**, **BYE** и **ACK**) и несколько полей (**To**, **From**, **Call-ID**, **CSeq**).

Протокол SIP использует текстовый формат сообщений, подобно протоколу HTTP. Это облегчает синтаксический анализ и генерацию кода, позволяет реализовать протокол на базе любого языка программирования, облегчает эксплуатационное управление, дает возможность ручного ввода некоторых полей, облегчает анализ сообщений. Название заголовков SIP-сообщений ясно указывает их назначение.

Протокол H.323 использует двоичное представление своих сообщений на базе языка ASN.1, поэтому их непосредственное чтение затруднительно. Для кодирования и декодирования сообщений необходимо использовать компилятор ASN.1. Но, в то же время, обработка сообщений, представленных в двоичном виде, производится быстрее.

Довольно сложным представляется взаимодействие протокола H.323 с межсетевым экраном (firewall). Кроме того, в протоколе H.323 существует дублирование функций. Так, например, оба протокола H.245 и RTCP имеют средства управления конференцией и осуществления обратной связи.

Выводы. На основе проведенного выше сравнения можно сделать вывод о том, что протокол SIP больше подходит для использования Internet-поставщиками, поскольку они рассматривают услуги IP-телефонии лишь как часть набора своих услуг.

Операторы телефонной связи, для которых услуги Internet не являются первостепенными, скорее всего, будут ориентироваться на протокол H.323, поскольку сеть, построенная на базе рекомендации H.323, представляется им хорошо знакомой сетью ISDN, наложенной на IP-сеть.

Не стоит также забывать, что к настоящему времени многие фирмы-производители и поставщики услуг уже вложили значительные средства в оборудование H.323, которое успешно функционирует в сетях.

Таким образом, ответ на вопрос, какой из протоколов предпочтительнее использовать, будет зависеть от целей бизнеса и требуемых функциональных возможностей. Скорее всего, эти варианты не следует рассматривать как конкурирующие, а как предназначенные для разных областей рынка услуг, поскольку они могут работать параллельно и взаимодействовать через специальный шлюз. Проиллюстрируем это утверждение следующим примером. В настоящее время рынок услуг все больше нацеливается на услуги с доплатой за дополнительные возможности (value added), и простота их предоставления дает реальные преимущества. Так, использование SIP в каком-либо частном домене дает возможность более гибкого предоставления услуг, а наличие средств, обеспечивающих переход от прото-

кола SIP к протоколу H.323, гарантирует взаимодействие с областями, использующими другие решения. В таблице 7.6 приведен вариант возможного обмена сообщениями.

Таблица 7.6 Алгоритм установления соединения с участием шлюза H.323/SIP

Шаг	H.323-сторона шлюза	SIP-сторона шлюза	Комментарии
1	→ Setup (с процедурой FastStart)		Содержит описание возможностей приема информации
2	← Call proceeding		Подтверждение прокси-сервером приема сообщения SETUP
3		INVITE →	Содержит описание возможностей приема информации в формате SDP
4		180 Ringing ←	Уведомление вызывающего пользователя о том, что вызываемому пользователю передается сигнал о входящем вызове
5	← Alerting		
6		200 OK ←	Вызываемый пользователь принял входящий вызов, сообщение содержит описание возможностей приема информации
7	← Connect		
8		ACK →	
•			
•	Телефонный разговор		
•			
N		BYE ←	Разговор завершен
N+1	← Release complete		
N+2		200 OK →	

Если в течение разговорной фазы оборудованию H.323 необходимо открыть новые логические каналы, шлюз передает новое сообщение **INVITE** терминалу SIP, как это показано в таблице 7.7.

Таблица 7.7 Открытие новых логических каналов

Шаг	H.323-сторона шлюза	SIP-сторона шлюза	Комментарии
	→ OpenLogicalChannel		
		INVITE →	Тот же идентификатор соединения, что и в предыдущем сообщении INVITE (но номер Cseq -увеличен) Описание нового канала в формате SDP
		200 OK ←	Содержит описание нового канала в формате SDP
	← OpenLogicalChannelAck		

Глава 8

Протокол управления шлюзами MGCP

8.1 Принцип декомпозиции шлюза

В недавнем прошлом рабочая группа MEGACO комитета IETF разработала протокол управления шлюзами – Media Gateway Control Protocol (MGCP). Ранее подобный протокол под названием SGCP – Simple Gateway Control Protocol (простой протокол управления шлюзами) – был разработан компанией Telecordia (бывшая компания Bellcore). Фирма Level 3 предложила сходный протокол управления оборудованием, реализующим технологию маршрутизации пакетов IP, – IDCP (IP Device Control Protocol). Оба они впоследствии были объединены в протокол MGCP.

При разработке протокола управления шлюзами рабочая группа MEGACO опиралась на принцип декомпозиции, согласно которому шлюз разбивается на отдельные функциональные блоки (рис. 8.1):

- транспортный шлюз – Media Gateway, который выполняет функции преобразования речевой информации, поступающей со стороны ТфОП с постоянной скоростью, в вид, пригодный для передачи по сетям с маршрутизацией пакетов IP: кодирование и упаковку речевой информации в пакеты RTP/UDP/IP, а также обратное преобразование;
- устройство управления – Call Agent, выполняющее функции управления шлюзом;

- шлюз сигнализации – Signaling Gateway, который обеспечивает доставку сигнальной информации, поступающей со стороны ТфОП, к устройству управления шлюзом и перенос сигнальной информации в обратном направлении.

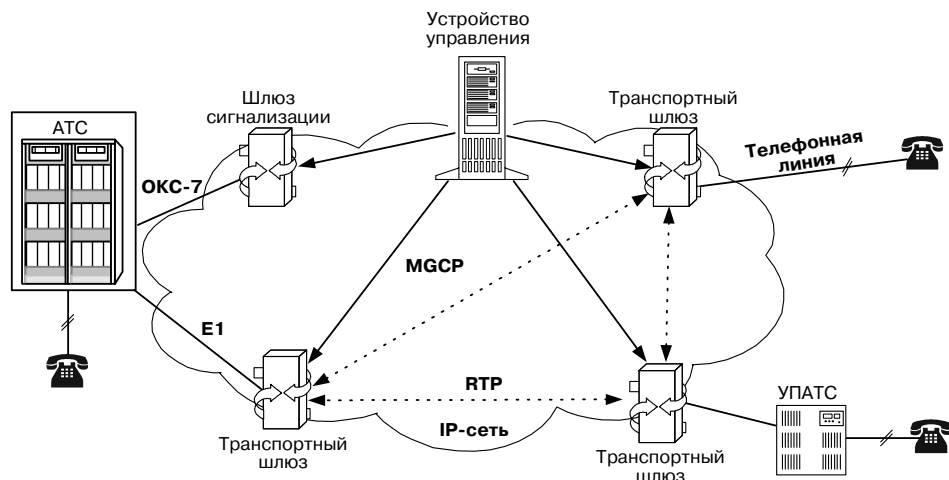


Рис. 8.1 Архитектура сети, базирующейся на протоколе MGCP

Таким образом, весь интеллект функционально распределенного шлюза размещается в устройстве управления, функции которого, в свою очередь, могут быть распределены между несколькими компьютерными платформами. Шлюз сигнализации выполняет функции STP – транзитного пункта системы сигнализации по общему каналу – ОКС7. Транспортные шлюзы выполняют только функции преобразования речевой информации. Одно устройство управления обслуживает одновременно несколько шлюзов. В сети может присутствовать несколько устройств управления. Предполагается, что эти устройства синхронизованы между собой и согласованно управляют шлюзами, участвующими в соединении. Рабочая группа MEGACO не определяет протокол синхронизации работы устройств управления, однако в ряде работ, посвященных исследованию возможностей протокола MGCP, для этой цели предлагается использовать протоколы H.323, SIP или ISUP/IP (рис. 8.2).

Перенос сообщений протокола MGCP обеспечивает протокол не гарантированной доставки – UDP. Кроме того, рабочая группа SIGTRAN комитета IETF в настоящее время разрабатывает механизм взаимодействия устройства управления и шлюза сигнализации. Последний должен принимать поступающие из ТфОП сигнальные единицы подсистемы МТР системы сигнализации ОКС7 и передавать сигнальные сообщения верхнего, пользовательского уровня к устройству управления. Основное внимание рабочей группы SIGTRAN уделено вопросам разработки наиболее эффективного механизма

передачи сигнальной информации по IP-сетям. Следует отметить, что существует несколько причин, уже упоминавшихся ранее, по которым пришлось отказаться от использования для этой цели протокола TCP. Вместо него рабочая группа SIGTRAN предлагает использовать протокол Stream Control Transport Protocol (SCTP), который имеет ряд преимуществ перед протоколом TCP. Основным из этих преимуществ является значительное снижение времени доставки сигнальной информации и, следовательно, времени установления соединения – одного из важнейших параметров качества обслуживания.

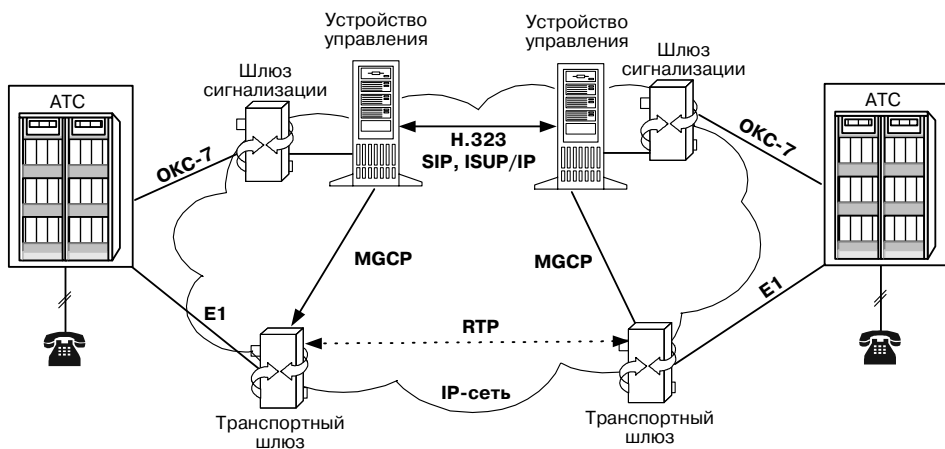


Рис. 8.2 Синхронизация работы устройств управления

Если распределенный шлюз подключается к ТфОП при помощи сигнализации по выделенным сигнальным каналам (ВСК), то сигнальная информация вместе с пользовательской информацией сначала поступает в транспортный шлюз, а затем передается в устройство управления без посредничества шлюза сигнализации.

Одно из основных требований, предъявляемых к протоколу MGCP, состоит в том, что устройства, реализующие этот протокол, должны работать в режиме без сохранения информации о последовательности транзакций между устройством управления и транспортным шлюзом, т.е. в устройствах не требуется реализации конечного автомата для описания этой последовательности. Однако не следует распространять подобный подход на последовательность состояний соединений, сведения о которых хранятся в устройстве управления.

Отметим, что протокол MGCP является внутренним протоколом, поддерживающим обмен информацией между функциональными блоками распределенного шлюза. Протокол MGCP использует принцип master/slave (ведущий/ведомый), причем устройство управления шлюзами является ведущим, а транспортный шлюз – ведомым устройством, выполняющим команды, поступающие от устройства управления.

Такое решение обеспечивает масштабируемость сети и простоту эксплуатационного управления ею через устройство управления шлюзами. К тому же, не интеллектуальные шлюзы требуют меньшей производительности процессоров и, как следствие, оказываются менее дорогими. Кроме того, обеспечивается возможность быстро добавлять новые протоколы сигнализации и новые дополнительные услуги, так как нужные для этого изменения затрагивают только устройство управления шлюзами, а не сами шлюзы.

Основной недостаток этого подхода – незаконченность стандартов. Функциональные блоки распределенных шлюзов, разработанные разными фирмами-производителями телекоммуникационного оборудования, практически несовместимы. Функции устройства управления шлюзами точно не определены. Не стандартизированы механизмы переноса сигнальной информации от шлюза сигнализации (Signalling Gateway) к устройству управления и в обратном направлении. К недостаткам можно отнести также отсутствие стандартизированного протокола взаимодействия между устройствами управления. Кроме того, протокол MGCP, являясь протоколом управления шлюзами, не предназначен для управления соединениями с участием терминального оборудования пользователей (IP-телефонами). Это означает, что в сети, построенной на базе протокола MGCP, для управления терминалами должен присутствовать привратник или сервер SIP (рис. 8.3).

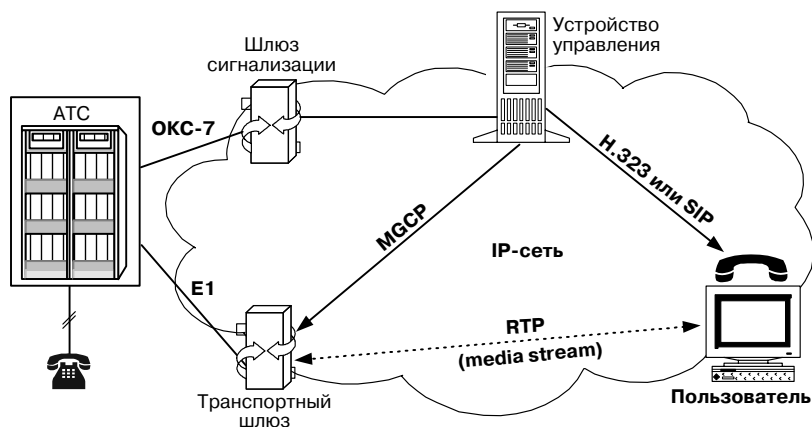


Рис. 8.3 Управление терминалами в сети, базирующейся на протоколе MGCP

8.2 Классификация шлюзов

Рабочей группой MEGACO предложена следующая классификация транспортных шлюзов (Media Gateways):

- Trunking Gateway – шлюз между ТФОП и сетью с маршрутизацией пакетов IP, ориентированный на подключение к телефонной сети посредством большого количества цифровых трактов (от 10 до нескольких тысяч) с использованием системы сигнализации ОКС 7;
- Voice over ATM Gateway – шлюз между ТФОП и ATM-сетью, который также подключается к телефонной сети посредством большого количества цифровых трактов (от 10 до нескольких тысяч);
- Residential Gateway – шлюз, подключающий к IP-сети аналоговые, кабельные модемы, линии xDSL и широкополосные устройства беспроводного доступа;
- Access Gateway – шлюз для подключения к сети IP-телефонии небольшой учрежденческой АТС через аналоговый или цифровой интерфейс;
- Business Gateway – шлюз с цифровым интерфейсом для подключения к сети с маршрутизацией IP-пакетов учрежденческой АТС при использовании, например, системы сигнализации DSS1;
- Network Access Server – сервер доступа к IP-сети для передачи данных;
- Circuit switch или packet switch – коммутационные устройства с интерфейсом для управления от внешнего устройства.

8.3 Модель организации связи

Для описания процесса обслуживания вызова с использованием протокола MGCP рабочей группой MEGACO разработана модель организации соединения – Connection model. Базой модели являются компоненты двух основных видов: порты (Endpoints) и подключения (Connections).

Endpoints – это порты оборудования, являющиеся источниками и приемниками информации. Существуют порты двух видов: физические и виртуальные. Физические порты – это аналоговые интерфейсы, поддерживающие каждый одно телефонное соединение, или цифровые каналы, также поддерживающие одно телефонное соединение и мультиплексированные по принципу временного разделения каналов в тракт Е1. Примером виртуального порта является источник речевой информации в интерактивном речевом сервере, т.е. некое программное средство.

Connection – означает подключение порта к одному из двух концов соединения, которое создается между ним и другим портом. Такое соединение будет установлено после подключения другого порта к его второму концу. Соединение может связывать порты разных шлюзов через сеть с маршрутизацией пакетов IP или порты внутри одного шлюза.

На рисунке 8.4 представлены примеры использования этих двух компонентов. Отметим, что порты некоторых видов могут участвовать в нескольких соединениях одновременно.

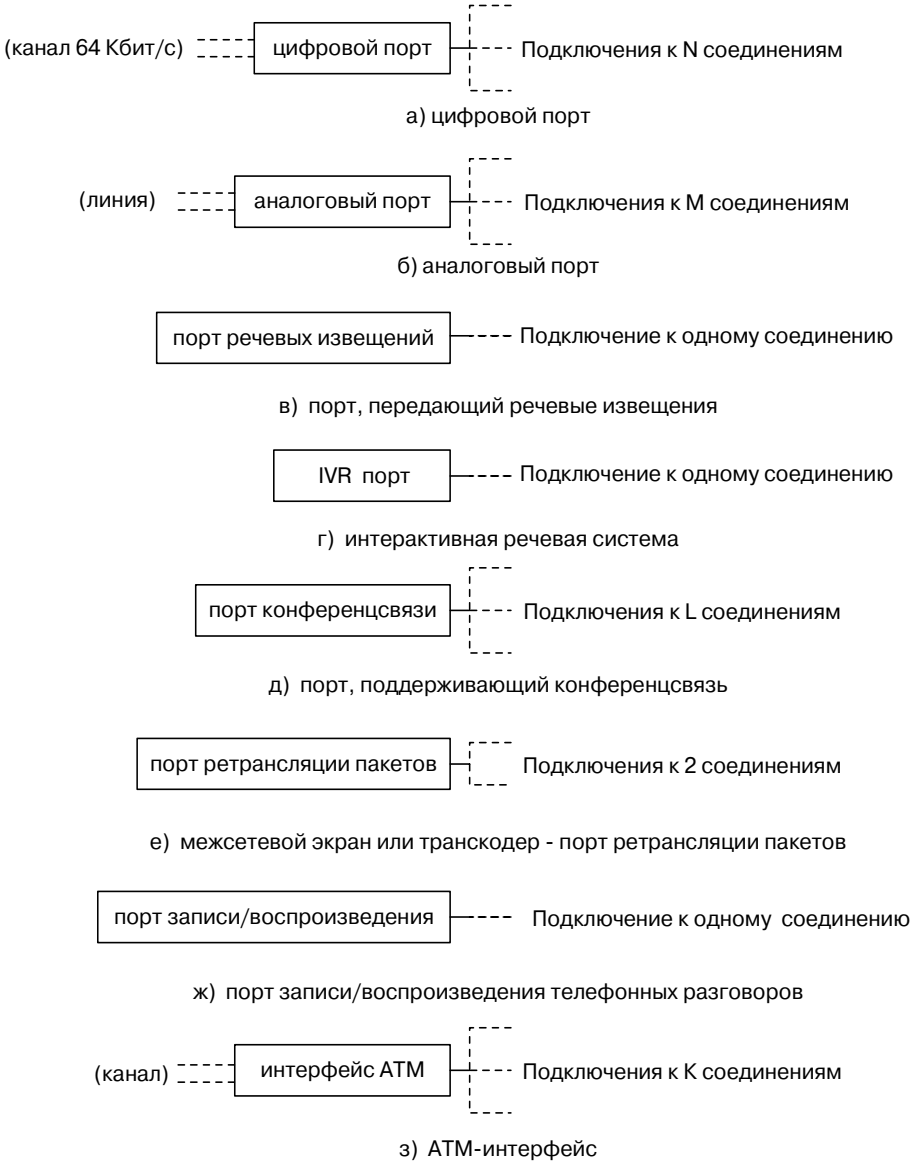


Рис. 8.4 Примеры использования компонентов модели

Подключения создаются устройством управления Call Agent для каждого порта, участвующего в соединении. На рисунке 8.5 показана ситуация, когда одно устройство управления контролирует рабо-

ту двух портов разных шлюзов при организации соединения между этими портами.

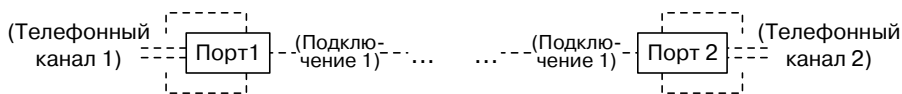


Рис. 8.5 Соединение в сети, построенной на базе протокола MGCP

8.4 Команды протокола MGCP

В ходе установления, поддержания и разрушения соединения при помощи протокола MGCP устройство управления и шлюз обмениваются командами и ответами, которые представляют собой набор текстовых строк. В этом параграфе дается краткое описание команд протокола MGCP, среди которых определены команды управления соединением и команды управления портами оборудования.

При помощи команды EndpointConfiguration устройство управления инструктирует шлюз, каким образом он должен обрабатывать получаемые речевые сигналы, например, использовать для преобразования цифрового сигнала в аналоговую форму закон А или закон μ .

Команда EndpointConfiguration содержит ряд параметров:

```
ReturnCode
    <- EndpointConfiguration( EndpointId,
                             BearerInformation),
```

где EndpointId – идентификатор порта шлюза, к которому относится данная команда;

BearerInformation – параметр, определяющий закон (А или μ) декодирования принятой речевой информации.

ReturnCode – параметр, возвращаемый шлюзом устройству управления, чтобы информировать его о выполнении команды. Данный параметр представляет собой целое число, за которым могут следовать комментарии.

Call Agent при помощи команды Notification Request может дать указание шлюзу выявлять определенные события или сигналы и информировать о них устройство управления. В число детектируемых событий (сигналов) входит изменение сопротивления абонентского шлейфа, происходящее, когда абонент поднимает или кладет трубку, а также получение сигналов факсимильных аппаратов и сигналов DTMF.

Команда NotificationRequest включает в себя следующие параметры (в квадратных скобках указаны те из них, которые не являются обязательными).

```
ReturnCode
  <- NotificationRequest( EndpointId,
                        [NotifiedEntity,]
                        [RequestedEvents,]
                        RequestIdentifier,
                        [DigitMap,]
                        [SignalRequests,]
                        [QuarantineHandling,]
                        [DetectEvents,]
                        [encapsulated EndpointConfiguration])
```

Здесь *NotifiedEntity* – идентификатор устройства, которому должен быть передан ответ на команду. При отсутствии этого параметра ответ передается тому устройству, от которого получен запрос *NotificationRequest*.

RequestedEvents – список событий, о которых следует оповестить управляющее устройство. Кроме того, в этом параметре может быть указано, как шлюз должен реагировать на событие. Определены следующие действия шлюза: оповестить *Call Agent* о событии немедленно; ожидать дальнейших событий; если событие состоит в получении сигнала DTMF, то накапливать такие сигналы в соответствии с требованиями параметра *DigitMap*; в определенных ситуациях передавать в телефонный канал акустические или вызывные сигналы; обработать инкапсулированную команду *EndpointConfiguration*; игнорировать событие и т.д.

RequestIdentifier – идентификатор запроса, в ответ на который передается команда.

DigitMap – необязательный параметр, специфицирующий правила обработки сигналов DTMF. В этом параметре указывает количество сигналов, которые шлюз должен накопить для передачи их устройству управления.

SignalRequests – сигналы, которые должны быть переданы в канал, например, сигнал послышки вызова.

QuarantineHandling – необязательный параметр, определяющий правила обработки событий, которые были обнаружены до момента получения данной команды в период так называемого карантина (*quarantine period*) и о которых *Call Agent* еще не был оповещен.

DetectEvents – необязательный параметр, определяющий события, которые нужно выявить в период карантина, но не оповещать о них *Call Agent* до получения следующей команды *NotificationRequest* с включенным в нее параметром *QuarantineHandling*.

Encapsulated EndpointConfiguration – команда *EndpointConfiguration*, инкапсулированная в команду *NotificationRequest*.

Остальные параметры команды тождественны описанным выше.

При помощи команды *Notify* шлюз информирует устройство управления о том, что произошло событие из числа указанных в команде *NotificationRequest*. Команда *Notify* содержит следующие параметры:

```
ReturnCode  
  <- Notify (EndpointId,  
            [NotifiedEntity,]  
            RequestIdentifier,  
            ObservedEvents)
```

Здесь ObservedEvents – параметр, в котором описываются произошедшие события, например, передаются набранные цифры номера. Остальные параметры были описаны ранее.

При помощи команды CreateConnection управляющее устройство может дать шлюзу указание создать соединение двух портов одного и того же шлюза или разных шлюзов.

Структура этой команды приведена ниже.

```
ReturnCode,  
ConnectionId,  
[SpecificEndPointId,]  
[LocalConnectionDescriptor,]  
[SecondEndPointId,]  
[SecondConnectionId]  
<- CreateConnection(CallId,  
                    EndpointId,  
                    [NotifiedEntity,]  
                    [LocalConnectionOptions,]  
                    Mode,  
                    [{RemoteConnectionDescriptor |  
                     SecondEndPointId}, ]  
                    [Encapsulated NotificationRequest,]  
                    [Encapsulated EndpointConfiguration])
```

CallId – уникальный параметр, идентифицирующий сессию, к которой относится данное соединение.

NotifiedEntity – необязательный параметр, идентифицирующий устройство, к которому должны быть переданы команды Notify или DeleteConnection.

LocalConnectionOptions – параметр, используемый Call Agent, чтобы дать шлюзу указания в отношении характеристик подключения порта к соединению. В параметр могут входить следующие поля: метод кодирования, размер речевых пакетов, полоса пропускания, тип обслуживания, использование эхокомпенсатора, использование режима подавления пауз в разговоре, использование режима подавления шума, использование резервирования ресурсов и другие поля. Кодирование трех первых полей должно производиться в соответствии с протоколом описания сессий SDP (Session Description Protocol), причем Call Agent может указать только полосу пропускания и оставить за шлюзом право выбора метода кодирования и размеров речевых пакетов.

Mode – параметр, определяющий режим работы для данного конца соединения. Определены следующие режимы: передача, прием, прием/передача, конференция, данные, отсутствие активности, петля, тестовый режим и другие.

`RemoteConnectionDescriptor` – описание подключения к соединению на другом его конце. Данный параметр содержит те же поля, что и параметр `LocalConnectionOptions`. Эти поля также должны кодироваться в соответствии с протоколом SDP. Стоит отметить, что при создании соединения между двумя шлюзами, при передаче первой команды `CreateConnection` параметр `RemoteConnectionDescriptor` отсутствует (ему присваивается нулевое значение), так как информация о подключении к соединению на другом его конце в этот момент отсутствует. Не имея такой информации, т.е. не получив команду `ModifyConnection`, шлюз может только принимать информацию (работать в режиме `receive only`).

`SecondEndPointId` – этот параметр может включаться в команду `CreateConnection` вместо параметра `RemoteConnectionDescriptor` при установлении соединения между двумя портами одного и того же шлюза.

`Encapsulated NotificationRequest` – инкапсулированная команда `NotificationRequest`.

В ответ на команду `CreateConnection`, кроме описанного выше параметра `ReturnCode`, шлюз возвращает следующие параметры:

`ConnectionId` – уникальный идентификатор подключения данного порта к соединению.

`SpecificEndPointId` – необязательный параметр, идентифицирующий порт, который отвечает на команду `CreateConnection`, если он не был специфицирован устройством управления.

`LocalConnectionDescriptor` – параметр, содержащий информацию об IP-адресе и номере порта RTP в соответствии с протоколом SDP.

`SecondEndPointId` – параметр, означающий, что команда `CreateConnection` создала два соединения.

`SecondConnectionId` – идентификатор подключения для второго соединения.

Устройство управления может изменить параметры существующего соединения при помощи команды `ModifyConnection`, которая включает в себя следующие параметры.

```
ReturnCode,
[LocalConnectionDescriptor]
← ModifyConnection (CallId,
                    EndpointId,
                    ConnectionId,
                    [NotifiedEntity],
                    [LocalConnectionOptions],
                    [Mode],
                    [RemoteConnectionDescriptor],
                    [Encapsulated NotificationRequest],
                    [Encapsulated EndpointConfiguration])
```

Здесь используются такие же параметры, как и в команде `CreateConnection`, но добавляется обязательный параметр `ConnectionId`,

который идентифицирует подключение к соединению данного порта оборудования, так как один порт может одновременно иметь подключения к нескольким соединениям.

Данная команда может использоваться для передачи информации о другом конце соединения в параметре `RemoteConnectionDescriptor`, для активизации/деактивизации соединения при помощи параметра `Mode`, для изменения алгоритма кодирования, периода пакетизации передаваемой информации или для управления подавлением эха.

Таким образом, если первоначально порт мог только принимать информацию, так как не имел описания функциональных возможностей и адреса порта на другом конце соединения, то описываемая команда создает возможность передавать информацию.

Если параметры соединения на ближнем конце были изменены, например, был изменен номер порта RTP, то в ответе на команду `ModifyConnection` может возвращаться параметр `LocalConnectionDescriptor`.

Устройство управления может разрушить существующее соединение при помощи команды `DeleteConnection`. Кроме того, при помощи этой команды шлюз может передать к `Call Agent` индикацию того, что существующее соединение больше поддерживаться не может.

Команда `DeleteConnection`, передаваемая устройством управления, имеет следующий вид:

```
ReturnCode,
Connection-parameters
<- DeleteConnection (CallId,
                      EndpointId,
                      ConnectionId,
                      [Encapsulated NotificationRequest,]
                      [Encapsulated EndpointConfiguration])
```

Все параметры были описаны ранее, однако следует отметить, что в параметр `NotificationRequest` может включаться инструкция, например, о действиях шлюза при детектировании размыкания абонентского шлейфа (абонент положил трубку): в этом случае шлюз должен разрушить соединение и ждать замыкания шлейфа (следующего вызова).

В общем случае, команда `DeleteConnection` передается обоим шлюзам, подключенным к соединению. После завершения соединения в ответ на команду `DeleteConnection` шлюз возвращает статистические данные, собранные за время соединения – `connection-parameters`:

- количество переданных RTP-пакетов,
- количество переданных байтов информации, не считая служебной информации (заголовков IP/UDP/RTP),

- количество полученных RTP-пакетов,
- количество принятых байтов информации, не считая служебной информации (заголовков IP/UDP/RTP),
- количество потерянных RTP-пакетов,
- вариация времени между поступлениями RTP-пакетов,
- средняя задержка RTP-пакетов.

В некоторых случаях, таких как неисправность порта, участвующего в соединении, или отсутствие ресурсов для поддержания существующего соединения, шлюз должен сам инициировать разрушение соединения при помощи команды `DeleteConnection`, которая имеет следующий вид:

```
ReturnCode,
    <- DeleteConnection (CallId,
                        EndpointId,
                        ConnectionId,
                        Reason-code,
                        Connection-parameters)
```

В параметре `Reason-code` указывается причина, по которой шлюз передает данное сообщение. Остальные параметры были описаны ранее.

Чтобы получить информацию о статусе какого-либо порта шлюза, управляющее устройство может передать запрос `Audit EndPoint`, который имеет следующий вид:

```
ReturnCode,
    EndPointIdList | {
        [RequestedEvents,]
        [DigitMap,]
        [SignalRequests,]
        [RequestIdentifier,]
        [NotifiedEntity,]
        [ConnectionIdentifiers,]
        [DetectEvents,]
        [ObservedEvents,]
        [EventStates,]
        [BearerInformation,]
        [RestartReason,]
        [RestartDelay,]
        [ReasonCode,]
        [Capabilities]}
    <- AuditEndPoint (EndpointId,
        [RequestedInfo])
```

`RequestedInfo` – необязательный параметр, описывающий информацию, которую запрашивает устройство управления.

В ответ на команду `AuditEndPoint` шлюз возвращает требуемую информацию (если никакой информации не запрашивается, но указанный в команде порт существует, то шлюз просто возвращает подтверждение). В ответе могут содержаться следующие параметры:

SignalRequests – необязательный параметр, в котором указывается список сигналов, обрабатываемых в настоящий момент;

ObservedEvents – необязательный параметр, в котором приводится текущий список обнаруженных событий;

RestartReason – необязательный параметр, в котором содержится причина рестарта порта, указанная в последней переданной шлюзом команде RestartInProgress;

RestartDelay – необязательный параметр, в котором содержится величина задержки рестарта, указанная в последней переданной шлюзом команде RestartInProgress;

Capabilities – необязательный параметр, содержащий такую же информацию, как и параметр LocalConnectionOptions.

При помощи команды AuditConnection устройство управления запрашивает параметры соединения, в котором участвует порт шлюза.

Команда имеет следующий вид:

```
ReturnCode,
[CallId,]
[NotifiedEntity,]
[LocalConnectionOptions,]
[Mode,]
[RemoteConnectionDescriptor,]
[LocalConnectionDescriptor,]
[ConnectionParameters]
    <- AuditConnection (EndpointId,
                        ConnectionId,
                        RequestedInfo)
```

Все параметры команды уже были описаны ранее. Если никакой информации не требуется и указанный порт существует, то шлюз проверяет, что соединение существует, и возвращает подтверждение.

Команда RestartInProgress передается шлюзом для индикации того, что один или группа портов выводятся из рабочего состояния или возвращаются в рабочее состояние. Данная команда имеет следующий вид:

```
ReturnCode,
[NotifiedEntity]
<— RestartInProgress (EndPointId,
                      RestartMethod,
                      [RestartDelay,]
                      [Reason-code])
```

Параметр RestartMethod специфицирует вид рестарта. Определено несколько видов рестарта:

- Graceful restart – постепенный рестарт, при котором порты оборудования выводятся из обслуживания после определенной задержки

ки. Установленные соединения не разрушаются, но и новые не создаются.

- **Forced restart** – принудительный рестарт, при котором разрушаются установленные соединения.
- **Restart** – рестарт, при котором порт оборудования возвращается в обслуживание после определенной задержки. При этом порт в момент рестарта не участвует ни в каких соединениях.
- **Disconnected** – данное значение присваивается параметру **RestartMethod**, когда порт находился вне обслуживания, но в данный момент пытается вернуться в обслуживание.
- **Cancel-graceful** – данное значение присваивается параметру **RestartMethod**, когда шлюз отменяет предшествовавшую команду **Restart** с параметром **RestartMethod**, которому было присвоено значение **Graceful**.

Параметр **RestartDelay** определяет задержку рестарта в секундах.

По аналогии с предыдущими главами в таблицу 8.1 сведены все команды протокола MGCP.

Таблица 8.1 Команды протокола MGCP

Команда	Направление передачи	Назначение
EndpointConfiguration (Конфигурация порта)	CA → MG	Call Agent инструктирует шлюз, каким образом ему нужно обрабатывать получаемые речевые сигналы
CreateConnection (Создать соединение)	CA → MG	Call Agent дает указание шлюзу создать соединение
ModifyConnection (Модифицировать соединение)	CA → MG	Call Agent дает указание шлюзу изменить параметры существующего соединения
DeleteConnection (Завершить соединение)	CA → MG, MG → CA	Call Agent и шлюзы завершают соединение
NotificationRequest (Запрос уведомления)	CA → MG	Call Agent инструктирует шлюз, какие события необходимо обнаруживать.
Notify (Уведомить)	MG → CA	Шлюз информирует Call Agent о том, что произошло событие из числа тех, которые были специфицированы в команде NotificationRequest
AuditEndpoint (Проверить порт)	CA → MG	Call Agent запрашивает информацию о каком-либо порте шлюза
AuditConnection (Проверить соединение)	MGC → MG	Call Agent запрашивает параметры соединения
ReStartInProgress (Идёт рестарт)	MG → MGC	Шлюз информирует Call Agent о том, что один или несколько портов выводятся из рабочего состояния или возвращаются в рабочее состояние

8.5 Структура команд

Команда протокола MGCP обязательно содержит заголовок, за которым может следовать описание сеанса связи (session description). Заголовок команды и описание сеанса связи представляют собой набор текстовых строк. Описание сеанса отделено от заголовка команды пустой строкой.

Заголовок содержит список параметров и командную строку вида **CRCX 1204 ts/1@protei.loniis.net MGCP 0.1**. Командная строка, в свою очередь, состоит из нескольких информационных полей:

1. *Название команды* представлено в виде кода из четырех букв (табл.8.2)

Таблица 8.2 Кодировка команд протокола MGCP

Команда	Код
EndpointConfiguration	EPCF
CreateConnection	CRCX
ModifyConnection	MDCX
DeleteConnection	DLCX
NotificationRequest	RQNT
Notify	NTFY
AuditEndpoint	AUEP
AuditConnection	AUCX
ReStartInProgress	RSIP

2. *Идентификатор транзакции*. Протокол MGCP предусматривает корреляцию команд и ответов. Команда и ответ на нее образуют транзакцию, имеющую уникальный идентификатор (Transaction-Identifier). Идентификатор транзакции включается в заголовок и команды, и ответа. Значения идентификаторов выбираются из диапазона чисел 1 – 999999999, причем значение идентификатора текущей транзакции на единицу больше идентификатора предыдущей транзакции.
3. *Идентификатор порта* определяет тот порт шлюза, которому надлежит выполнить команду, за исключением команд Notify и ReStartInProgress, в которых идентификатор определяет порт, передавший команду. Идентификаторы портов кодируются так же, как кодируются адреса электронной почты в соответствии с документом RFC 821 комитета IETF. Например, возможен идентификатор ts/1@protei.loniis.net, который идентифицирует первый порт (временной канал) шлюза «protei», расположенного в доме лониис.
4. *Версия протокола* кодируется следующим образом: MGCP 1.0.

Выше указывалось, что заголовок команды, кроме командной строки, содержит список параметров. Параметры команд протокола MGCP сведены в таблицу 8.3.

Таблица 8.3 Параметры команд протокола MGCP

Название параметра	Код	Описание и значение параметра
ResponseAck (Подтверждение транзакции)	K	Подтверждает завершение одной или нескольких транзакций. Например, параметр K: 6234-6255, 6257, 19030-19044 подтверждает завершение транзакций, имеющих идентификаторы с 6234 по 6255, 6257 и с 19030 по 19044.
BearerInformation (Сведения о виде информации)	B	Служит для доставки информации о законе кодирования речевой информации A или m
ReasonCode (Код причины)		Определены следующие коды причины: 000 – номинальное состояние порта, передается только в ответе на запрос о состоянии порта 900 – неисправность порта 901 – порт выведен из обслуживания 902 – отказ на физическом уровне (например, потеря синхронизации)
CallID (Идентификатор сеанса связи)	C	Идентифицирует сеанс связи, в котором может использоваться одно или несколько соединений. Идентификатор кодируется шестнадцатеричной последовательностью символов длиной не более 32 символом.
ConnectionID (Идентификатор подключения)	I	Идентифицирует подключение данного порта к одному соединению, так как один порт может быть одновременно подключен к нескольким соединениям
NotifiedEntity (Уведомляемый объект)	N	Идентификатор объекта, к которому следует передавать уведомления об обнаруженных событиях. Если данный параметр опущен, порт передает эту информацию к объекту, от которого была получена команда. Идентификатор объекта кодируется так же, как кодируются адреса электронной почты согласно RFC 821, например, MGC@ca.any.net.com:5625 или Joe@[128.23.0.4]. При использовании IP-адреса, он должен быть заключен в квадратные скобки.
RequestIdentifier (Идентификатор запроса)	X	Согласует требование уведомить о событии, полученное от Call Agent, с уведомлением, передаваемым шлюзом в команде Notify.
LocalConnection Options (Параметры подключения порта к соединению)	L	Данные об алгоритме кодирования информации, размере речевых пакетов в мс, используемой полосе пропускания в Кбит/с, типе обслуживания, использовании эхокомпенсации и другие сведения. Передается от Call Agent к шлюзу, обычно в команде CRCX.
ConnectionMode (Режим соединения)	M	Определены следующие режимы соединения: передача, прием, прием/передача, конференция, передача данных, отсутствие активности, петля, тест и другие режимы. Значение параметру присваивает Call Agent.
RequestedEvents (Запрашиваемые события)	R	Список событий, о которых следует оповестить Call Agent, и действия шлюза при обнаружении события. Определены следующие действия: оповестить Call Agent о событии немедленно; ожидать дальнейших событий; если событием является прием сигнала DTMF, то накапливать цифры в соответствии с требованиями параметра DigitMap; в определенных ситуациях передавать в канал акустические или вызывные сигналы; обработать инкапсулированную команду Endpoint Configuration, игнорировать событие и др.

SignalRequests (Требование передать сигнал)	S	Специфицируется сигнал, который должен быть передан абоненту, например, акустический сигнал "Ответ станции".
DigitMap (План нумерации)	D	Специфицирует правила обработки сигналов DTMF. При накоплении количества цифр, указанного в данном параметре, шлюз должен передать их устройству управления.
ObservedEvents (Обнаруженные события)	O	Список обнаруженных событий.
ConnectionParameters (Параметры соединения)	P	Статистические данные о соединении, передаваемые шлюзом после его завершения.
SpecifiedEndPointID (Идентификатор порта)	Z	Идентификатор порта в формате RFC821, например, EndPoint@hub1.any.net.com:5625,
RequestedInfo (Запрашиваемая информация)	F	Описывает информацию, которую Call Agent запрашивает у шлюза, например, цифры номера вызываемого абонента, набранные вызывающим абонентом.
QuarantineHandling (Карантинная обработка)	Q	Определяет правила обработки событий, которые были обнаружены до получения данной команды в период так называемого карантина (quarantine period), и о которых Call Agent еще не был оповещен.
DetectEvents (Выявляемые события)	T	Перечень событий, которые порт должен отслеживать, а при их обнаружении - извещать об этом Call Agent.
EventStates (Состояния, обусловленные событиями)	ES	Перечень состояний порта, обусловленных, например, тем, что абонент снял или положил трубку; информация об этих состояниях должна передаваться к Call Agent в ответ на команду AuditEndpoint.
RestartMethod (Метод рестарта)	RM	Способ индикации шлюзом вывода порта из обслуживания или ввода его в обслуживание. Поддерживаются несколько вариантов рестарта: "graceful", "forced", "restart", "disconnected" or "cancel-graceful".
RestartDelay (Задержка рестарта)	RD	Определяет время в секундах, после которого производится производится порта. Если этот параметр отсутствует, задержка рестарта равна нулю. При получении от Call Agent требования о принудительном рестарте порта команда выполняется незамедлительно.
Capabilities (Функциональные возможности порта)	A	Информацию о функциональных возможностях порта запрашивает Call Agent при помощи команды AuditEndpoint. Эти возможности порта включают в себя: поддерживаемые алгоритмы кодирования, период пакетизации, полосу пропускания, эхокомпенсацию, подавление пауз речи, режимы соединения, тип обслуживания, совокупность событий и др.

Не все параметры, приведенные в таблице 8.3, должны обязательно присутствовать во всех командах протокола MGCP. В таблице 8.4 представлены возможные комбинации параметров в командах протокола MGCP. Буква «М» означает обязательное присутствие параметра в команде, буква «О» – не обязательное присутствие, буква «F» запрещает присутствие параметра.

Таблица 8.4 Комбинации параметров
в командах протокола MGCP

Имя параметра	EP CF	CR CX	MD CX	DL CX	RQ NT	NT FY	AU EP	AU CX	RS IP
ResponseAck	O	O	O	O	O	O	O	O	O
BearerInformation	M	O	O	O	O	F	F	F	F
CallId	F	M	M	O	F	F	F	F	F
ConnectionId	F	F	M	O	F	F	F	M	F
RequestIdIdentifier	F	O**	O**	O**	M	M	F	F	F
LocalConnection	F	O	O	F	F	F	F	F	F
Options									
Connection Mode	F	M	M	F	F	F	F	F	F
RequestedEvents	F	O	O	O	O*	F	F	F	F
SignalRequests	F	O	O	O	O*	F	F	F	F
NotifiedEntity	F	O	O	O	O	O	F	F	F
ReasonCode	F	F	F	O	F	F	F	F	O
ObservedEvents	F	F	F	F	F	M	F	F	F
DigitMap	F	O	O	O	O	F	F	F	F
Connection	F	F	F	O	F	F	F	F	F
Parameters									
Specific Endpoint ID	F	F	F	F	F	F	F	F	F
Second Endpoint ID	F	O	F	F	F	F	F	F	F
RequestedInfo	F	F	F	F	F	F	M	M	F
QuarantineHandling	F	O	O	O	O	F	F	F	F
DetectEvents	F	O	O	O	O	F	F	F	F
EventStates	F	F	F	F	F	F	F	F	F
RestartMethod	F	F	F	F	F	F	F	F	M
RestartDelay	F	F	F	F	F	F	F	F	O
SecondConnectionID	F	F	F	F	F	F	F	F	F
Capabilities	F	F	F	F	F	F	F	F	F
RemoteConnection	F	O	O	F	F	F	F	F	F
Descriptor									
LocalConnection	F	F	F	F	F	F	F	F	F
Descriptor									

** – параметр RequestIdentifier не обязателен для команд Create-Connection, ModifyConnection и DeleteConnection, но если эти команды содержат инкапсулированную команду NotificationRequest, присутствие в них параметра RequestIdentifier становится обязательным;

* – параметры RequestedEvents и SignalRequests не обязательны для команды NotificationRequest.

8.6 Структура ответов на команды

Протокол MGCP предусматривает подтверждение получения всех команд. Структура ответов на команды в протоколе MGCP идентична вышеописанной структуре самих команд. Ответ на команду также представляет собой набор текстовых строк и обязательно содержит заголовок ответа, за которым (после пустой строки) может следовать описание сеанса связи.

В этом параграфе речь пойдет, главным образом, о заголовке ответа. Заголовок состоит из ответной строки, например, **200 1203 OK**, и списка параметров. Ответная строка, в свою очередь, состоит из нескольких информационных полей: кода ответа, идентификатора транзакции и необязательного комментария.

В таблице 8.5 приведены возможные варианты кода ответа на команды протокола MGCP.

Таблица 8.5 Коды ответов на команды протокола MGCP

Код	Значение кода
100	Полученная команда в данный момент обрабатывается, сообщение о выполнении команды будет передано позже
200	Полученная команда выполнена
250	Соединение разрушено
400	Транзакция не может быть выполнена из-за временной ошибки
401	Трубка телефона уже снята
402	Трубка телефона уже повешена
403	Команда не может быть выполнена из-за отсутствия в данный момент необходимых ресурсов
404	В настоящий момент отсутствует необходимая полоса пропускания
500	Команда не может быть выполнена, потому что порт неизвестен
501	Команда не может быть выполнена, потому что порт не готов к ее выполнению
502	Команда не может быть выполнена, потому что порт не имеет необходимой полосы пропускания
510	Команда не может быть выполнена из-за ошибки в протоколе
511	Команда не может быть выполнена, так как в ней содержится нераспознанное расширение
512	Команда не может быть выполнена, потому что шлюз не имеет средств детектирования одного из запрашиваемых сигналов

513	Команда не может быть выполнена, потому что шлюз не имеет средств генерирования одного из запрашиваемых сигналов
514	Команда не может быть выполнена, потому что шлюз не может передать необходимое речевое уведомление или подсказку
515	Команда имеет некорректный идентификатор соединения, например, идентификатор уже завершённого соединения
516	Команда имеет некорректный идентификатор сеанса связи
517	Неподдерживаемый или некорректный режим
518	Неподдерживаемая или неизвестная совокупность сигналов или событий
519	Порт не имеет сведений о плане нумерации
520	Команда не может быть выполнена, потому что идет рестарт порта
521	Порт передан другому Call Agent
522	Нет такого события или сигнала
523	Неизвестное действие или неразрешённая комбинация действий
524	Внутреннее несоответствие в параметре LocalConnectionOptions
525	Неизвестное расширение параметра LocalConnectionOptions
526	Недостаточная полоса пропускания
527	Отсутствует параметр LocalConnectionOptions
528	Несовместимая версия протокола
529	Отказ в аппаратном обеспечении
530	Ошибка в сигнальном протоколе CAS
531	Отказ группы каналов или трактов

Из представленного в таблице 8.5 перечня кодов ответов видно, что их основная роль заключается в защите от ошибок протокола, конфигурации или функциональных возможностей. На основании информации, предоставляемой этими кодами ошибок, невозможно реализовать осмысленный механизм диагностики. Для получения диагностической информации от шлюзов и портов шлюза нужны другие методы. Одним из возможных методов является упоминавшийся в главе 4 протокол SNMP (простой протокол эксплуатационного управления сетью), который, безусловно, найдёт применение в транспортных шлюзах IP-телефонии.

В заключение рассмотрения структуры ответов на команды протокола MGCP приведем возможные комбинации параметров в ответах (таблица 8.6).

Таблица 8.6 Возможные комбинации параметров в ответах протокола MGCP

Имя параметра	EP CF	CR CX	MD CX	DL CX	RQ NT	NT FY	AU EP	AU CX	RS IP
ResponseAck	F	F	F	F	F	F	F	F	F
BearerInformation	F	F	F	F	F	F	O	F	F
CallId	F	F	F	F	F	F	F	O	F
ConnectionId	F	O	F	F	F	F	F	F	F
RequestIdIdentifier	F	F	F	F	F	F	O	F	F
LocalConnection	F	F	F	F	F	F	O	O	F
Options									
Connection Mode	F	F	F	F	F	F	F	O	F
RequestedEvents	F	F	F	F	F	F	O	F	F
SignalRequests	F	F	F	F	F	F	O	F	F
NotifiedEntity	F	F	F	F	F	F	F	F	O
ReasonCode	F	F	F	F	F	F	O	F	F
ObservedEvents	F	F	F	F	F	F	O	F	F
DigitMap	F	F	F	F	F	F	O	F	F
Connection	F	F	F	O	F	F	F	O	F
Parameters									
Specific Endpoint ID	F	O	F	F	F	F	F	F	F
RequestedInfo	F	F	F	F	F	F	F	F	F
QuarantineHandling	F	F	F	F	F	F	O	F	F
DetectEvents	F	F	F	F	F	F	O	F	F
EventStates	F	F	F	F	F	F	O	F	F
RestartMethod	F	F	F	F	F	F	O	F	F
RestartDelay	F	F	F	F	F	F	O	F	F
Capabilities	F	F	F	F	F	F	O	F	F
SecondConnectionId	F	O	F	F	F	F	F	F	F
SecondEndpointID	F	O	F	F	F	F	F	F	F
LocalConnection	F	M	O	F	F	F	F	O	F
Descriptor									
RemoteConnection	F	F	F	F	F	F	F	O	F
Descriptor									

8.7 Описания сеансов связи

При установлении соединений Call Agent предоставляет портам шлюзов, участвующим в этих соединениях, необходимую информацию друг о друге – описание сеансов связи. Описание сеанса связи вводится в состав некоторых команд и ответов протокола MGCP и включает в себя IP-адрес, UDP/RTP порт, вид информации, алгоритм кодирования информации, размер речевых пакетов и т.д. Синтаксис описания сеанса связи в протоколе MGCP соответствует син-

таксису протокола описания сеансов связи – session description protocol (SDP), предложенному для использования в вышеуказанных целях комитетом IETF в документе RFC 2327 [53].

Протокол SDP может применяться для описания мультимедийных конференций, но текущая версия протокола MGCP использует протокол SDP только для описания параметров передачи речи и данных.

Так как книга посвящена анализу технологии передачи речевой информации по сетям с маршрутизацией пакетов IP, в данном параграфе мы рассмотрим синтаксис протокола SDP только в части описания сеанса речевой связи. Для описания такого сеанса в протоколе SDP предусмотрено несколько информационных полей:

- *Версия протокола SDP.* Текущая версия протокола – нулевая. Поле кодируется следующим образом: v=0.
- *IP-адрес шлюза.* Это поле содержит IP-адрес, который будет использоваться для обмена пакетами RTP. Если это поле включено в команды протокола MGCP, то оно означает адрес удаленного шлюза, если поле включено в ответы, то – адрес шлюза, передающего ответ.
- *Поле описания речевого канала.* Данное поле содержит индикацию вида передаваемой или принимаемой информации (в нашем случае – речи), номер порта, используемого для приема RTP пакетов удаленным шлюзом (если поле описания речевого канала включено в команды протокола MGCP) или локальным шлюзом (если это поле включено в ответы), индикацию использования протокола RTP для передачи речи и алгоритмы кодирования речевой информации. Поле кодируется буквой «m».
- *Режим соединения.* Режимы соединений представлены в таблице 8.7.

Таблица 8.7 Режимы соединения

Кодировка режима	Функционирование шлюза
sendonly	Шлюзу надлежит только передавать информацию
recvonly	Шлюзу надлежит только принимать информацию
sendrecv	Шлюзу надлежит передавать и принимать информацию
inactive	Шлюз не должен ни передавать, ни принимать информацию
loopback	Шлюз должен передавать принимаемую информацию в обратном направлении
conttest	Шлюзу надлежит перевести порт в режим тестирования

Кроме вышеуказанных полей, для описания сеанса речевой связи в протоколе SDP предусмотрено еще несколько необязательных информационных полей. Отметим, что если в команду или в ответ

протокола MGCP включены описания нескольких сеансов связи, то они отделяются друг от друга строкой с указанием версии протокола SDP. Типичный пример описания сеанса речевой связи с использованием протокола SDP приведён ниже:

```
v = 0
c = IN IP4 128.96.41.1
m = audio 3456 RTP/AVP 0
```

Данный пример заслуживает краткого комментария. Для описания сеанса связи используется протокол SDP, версия 0, в сети используется протокол IP, версия 4, IP адрес шлюза – 128.96.41.1, передается или принимается речевая информация, упакованная в пакеты RTP, номер порта RTP – 3456, алгоритм кодирования речи G.711, закон µ.

8.8 Установление, изменение и разрушение соединений

В данном параграфе будет показано, каким образом при помощи протокола MGCP устанавливаются, изменяются и завершаются речевые соединения в сетях с маршрутизацией пакетов IP. Пример охватывает взаимодействие протокола MGCP с протоколом OKC7 (рис. 8.6).

От телефонной станции ATC1 к шлюзу сигнализации SG1 по общему каналу сигнализации поступает запрос соединения – сообщение IAM. Шлюз SG1 передает сообщение IAM устройству управления шлюзами Call Agent, которое обрабатывает запрос и определяет, что вызов должен быть направлен к телефонной станции ATC2 посредством шлюза TGW2.

Далее Call Agent резервирует порт шлюза TGW1 (разговорный канал). С этой целью Call Agent передает шлюзу команду CreateConnection. Отметим, что порт шлюза TGW1 может только принимать информацию (режим «recvonly»), так как он еще не осведомлен о том, на какой адрес и каким образом ему следует передавать информацию.

```
CRCX 1204 trunk-group-1/17@tgw1.whatever.net MGCP 0.1
C: A3C47F21456789F0
L: p:10, a:G.711
M: recvonly
```

В ответе на принятую команду шлюз TGW1 возвращает описание сеанса связи.

```
200 1204 OK
I:FDE234C8

v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0
```

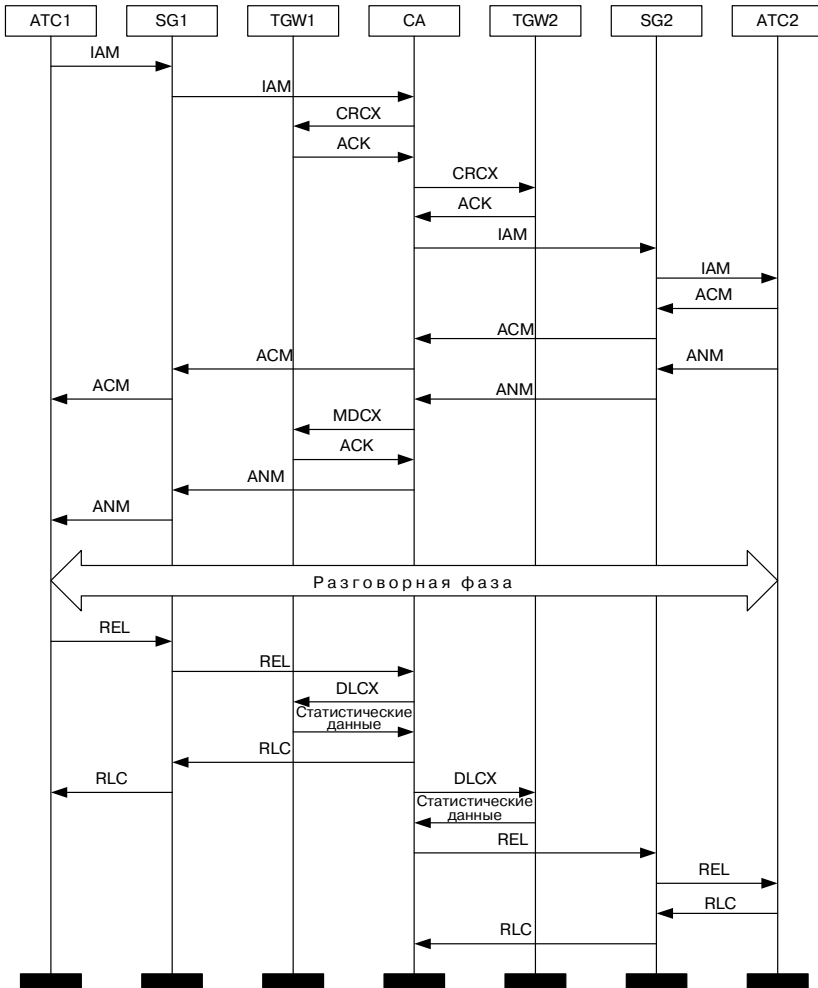


Рис. 8.6 Установление и разрушение соединения с использованием протокола MGCP

После приема от шлюза TGW1 подтверждения Call Agent передает команду CRCX второму шлюзу TGW2 с целью зарезервировать в нем порт:

```
CRCX 1205 trunk-group-2/$@tgw2.whatever.net MGCP 0.1
C: A3C47F21456789F0
M: sendrecv
```

```
v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0
```

Шлюз TGW 2 выбирает порт, который будет участвовать в связи, и подтверждает прием команды CRCX.

```
200 1205 OK
I:abc0
```

```
v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0
```

При помощи двух команд CRCX создается однонаправленный разговорный канал для передачи вызываемому абоненту акустических сигналов или речевых подсказок и извещений. В то же время, порт шлюза TGW 2 уже может не только принимать, но и передавать информацию, так как он получил описание сеанса связи от встречного шлюза. Далее Call Agent передает сообщение IAM к телефонной станции ATC2. На сообщение IAM станция ATC2 отвечает сообщением ACM, которое немедленно пересылается к станции ATC1.

После того как вызываемый абонент примет вызов, телефонная станция ATC2 передает к Call Agent сообщение ANM. Далее Call Agent меняет режим соединения «recvonly» в шлюзе TGW1 на полнодуплексный режим:

```
MDCX 1206 trunk-group-1/17@tgw1.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: FDE234C8
M: sendrecv
```

```
v=0
c=IN IP4 128.96.63.25
m=audio 1296 RTP/AVP 0
```

Шлюз TGW1 выполняет и подтверждает изменение режима соединения:

```
200 1206 OK
```

Call Agent передает сообщение ANM к телефонной станции ATC1, после чего наступает разговорная фаза соединения.

Завершение разговорной фазы происходит следующим образом. В нашем случае вызвавший абонент дает отбой первым, телефонная станция ATC1 через шлюз сигнализации передает к Call Agent сообщение REL. На основании этого сообщения Call Agent завершает соединение с вызвавшим абонентом:

```
DLCX 1207 trunk-group-1/17@tgw1.whatever.net MGCP 0.1
C: A3C47F21456789F0
I: FDE234C8
```

Шлюз подтверждает завершение соединения и передает к Call Agent собранные за время соединения статистические данные:

```
250 1217 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

Далее Call Agent передает к ATC1 сообщение RLC с целью подтвердить разрушение соединения.

Параллельно Call Agent завершает соединение с вызванной стороной:

```
DLCX 1208 trunk-group-2/13@tgw2.whatever.net MGCP 0.1  
C: A3C47F21456789F0  
I: abc0
```

Шлюз TGW2 подтверждает завершение соединения и передает к Call Agent собранные за время соединения статистические данные

```
250 1218 OK  
P: PS=790, OS=45700, PR=1230, OR=61875, PL=15, JI=27, LA=48
```

После приема ответа на команду DLCX Call Agent может начинать процедуру завершения соединения с ATC2, которая должна подтвердить разъединение, после чего соединение считается разрушенным.

8.9 Реализация оборудования с поддержкой протокола MGCP

Рассмотрим реализацию протокола MGCP на примере оборудования IPConnect производства компании Nortel Networks. IPConnect – это набор совместимых аппаратно-программных средств, объединенных единой идеологией и технологической базой. Он охватывает системы управления соединениями и обработки вызовов, серверы приложений, шлюзы, а также аппаратуру, устанавливаемую непосредственно у пользователей. Масштабы сетей, создаваемых на базе этого решения, практически не ограничены.

И еще одна важная особенность: концепция построения оборудования IPConnect дает оператору возможность выбрать то решение, которое отвечает его текущим потребностям, и постепенно наращивать мощность сети, инвестируя средства поэтапно и соотнося этот процесс с темпами развития бизнеса и финансовыми возможностями.

Как известно, до недавнего времени единственным протоколом, регулирующим процесс управления соединениями в сетях IP-телефонии, был протокол H.323. Этот протокол достаточно широко распространен и хорошо зарекомендовал себя в решениях IP-телефонии. Но, к сожалению, H.323 не может гарантировать прозрачность соединений с ТфОП, использующими сигнализацию ОКС7. Это обстоятельство предопределило выбор протокола MGCP в качестве основного протокола в оборудовании IPConnect (что, впрочем, не отвергает полностью протокол H.323). Протокол MGCP специально оптимизирован для нужд телефонии, и компания Nortel Networks принимает активное участие в его разработке и внедрении.

С точки зрения функционального построения в IPConnect можно выделить четыре основных элемента (рис. 8.7):

- шлюз между ТФОП и IP-сетью (CVX 1800);
- система обработки вызовов (IPConnect Call Engine – ICE);
- шлюз сигнализации (USP);
- различные приложения – например, IVR.

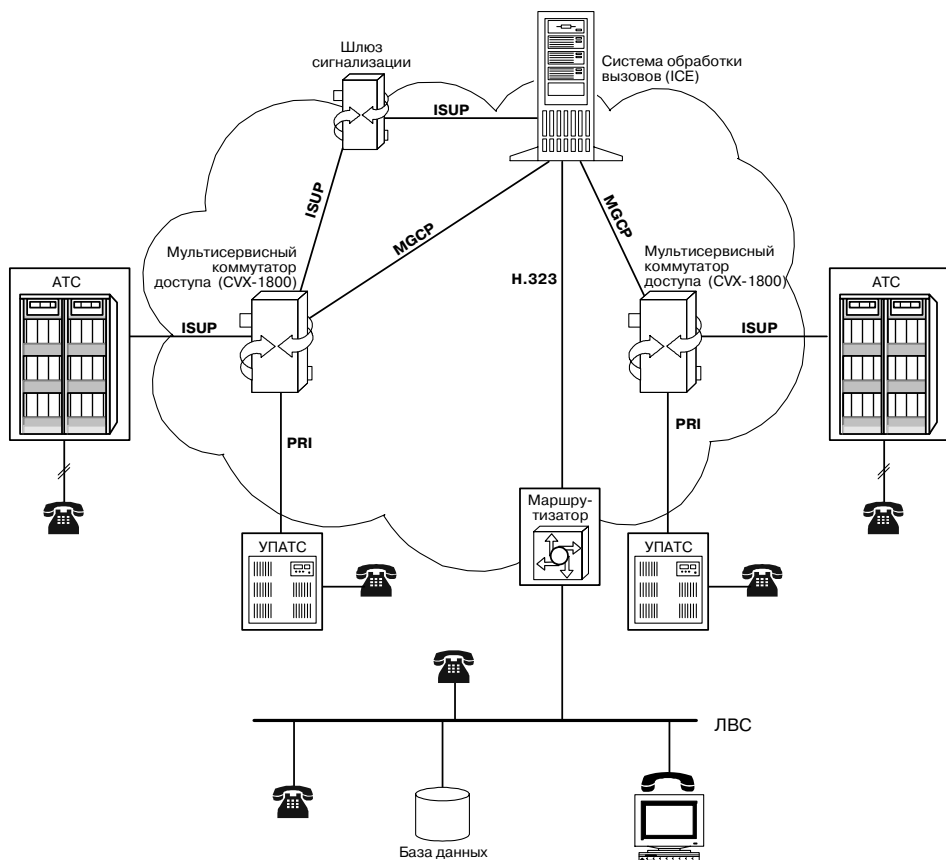


Рис. 8.7 Инфраструктура IPConnect

Результатом эволюции ОКС7 в направлении пакетной передачи информации стало появление семейства протоколов IPS7 (более 20), описывающих конкретные процедуры упаковки/распаковки сигналов ОКС7. В их числе – аналоги протоколов ISUP, INAP и других, используемых в системе сигнализации ОКС7. Эти протоколы имеют различные модификации, учитывающие особенности, присущие национальным системам сигнализации. В частности, имеются специальные разновидности и для России (например, аналог протокола ISUP-R).

8.10 Возможности и перспективы протокола MGCP

Для построения хорошо функционирующих и совместимых с ТфОП сетей IP-телефонии сегодня подходят протоколы H.323 и MGCP. Подход с использованием MGCP обладает весьма важным преимуществом перед подходом, предложенным ITU в рекомендации H.323: Call Agent поддерживает сигнализацию OKC7 и другие виды телефонной сигнализации; поддерживается также прозрачная трансляция сигнальной информации по сети IP-телефонии. В сети, построенной на базе рекомендации H.323, сигнализация OKC7, как и любая другая сигнализация, должна конвертироваться шлюзом в сигнальные сообщения H.225.0 (Q.931).

В целом же, анализируя функциональные возможности протокола MGCP, можно сделать следующий вывод: протокол, предлагаемый рабочей группой MEGACO организации IETF, лучше других подходит для развертывания глобальных сетей IP-телефонии, приходящих на смену традиционным телефонным сетям.

Но, в то же время, следует отметить, что в существующих сегодня приложениях IP-телефонии, таких как предоставление услуг международной и междугородной связи, использовать протокол MGCP (так же, как и протокол SIP) нецелесообразно в связи с тем, что подавляющее большинство сетей IP-телефонии сегодня построено на базе протокола H.323. Оператору придется строить на базе протокола MGCP (или SIP) отдельную сеть IP-телефонии, что потребует значительных капиталовложений, в то время как оператор связи, имеющий оборудование стандарта H.323, может легко присоединить свою сеть к существующим сетям.

Глава 9

Протокол MEGACO/Н.248

9.1 История создания и особенности протокола MEGACO/Н.248

Рабочая группа MEGACO комитета IETF, продолжая исследования, направленные на усовершенствование протокола управления шлюзами, создала более функциональный (по сравнению с рассмотренным в предыдущей главе протоколом MGCP) протокол MEGACO. Но разработкой протоколов управления транспортными шлюзами, кроме комитета IETF, занималась еще и исследовательская группа SG 16 Международного союза электросвязи. Так, в проекте 4-й версии рекомендации Н.323 ITU-T ввел принцип декомпозиции шлюзов, уже описанный с той или иной степенью детализации в главах 1, 2 и 8. Важной особенностью нововведения ITU-T явилось то, что управление транспортными шлюзами – Media Gateway (MG) – осуществляется контроллером транспортных шлюзов – Media Gateway Controller (MGC) – при помощи протокола MEGACO, адаптированного под сетевое окружение Н.323. Спецификации адаптированного протокола приведены в недавно утвержденной рекомендации ITU-T Н.248. В данной книге этот протокол называется MEGACO/Н.248, так как авторам не хотелось бы умалить чьи-либо заслуги в разработке и адаптации этого протокола. На рис. 9.1. представлено дерево эволюции протокола MEGACO/Н.248.

Рассмотрим кратко основные особенности протокола MEGACO/Н.248. Для переноса сигнальных сообщений MEGACO/Н.248 могут использоваться протоколы UDP, TCP, SCTP или транспортная технология ATM. Поддержка для этих целей протокола UDP – одно из обя-

зательных требований к контроллеру шлюзов. Протокол TCP должен поддерживаться и контроллером, и транспортным шлюзом, а поддержка протокола SCTP, так же, как и технологии ATM, является необязательной.

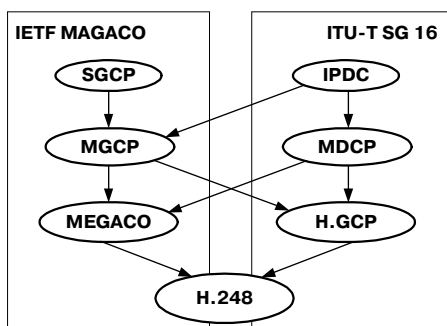


Рис. 9.1 Дерево эволюции протокола MEGACO/H.248

Еще одной особенностью протокола MEGACO/H.248 является то, что сообщения этого протокола могут кодироваться двумя способами. Комитет IETF предложил текстовый способ кодирования сигнальной информации, а для описания сеанса связи предложил использовать протокол SDP. ITU-T предусматривает бинарный способ представления сигнальной информации – ASN.1, а для описания сеансов связи рекомендует специальный инструмент – Tag-length-value (TLV). Контроллер шлюза должен поддерживать оба способа кодирования, в то время как шлюз – только один из этих способов.

9.2 Модель процесса обслуживания вызова

При описании алгоритма установления соединения с использованием протокола MEGACO комитет IETF опирается на специальную модель процесса обслуживания вызова, отличную от модели MGCP. Протокол MEGACO оперирует с двумя логическими объектами внутри транспортного шлюза: порт (termination) и контекст (context), которыми может управлять контроллер шлюза. Пример модели процесса обслуживания вызова приведен на рис. 9.2.

Порты являются источниками и приемниками речевой информации. Определено два вида портов: физические и виртуальные. Физические порты, существующие постоянно с момента конфигурации шлюза, это аналоговые телефонные интерфейсы оборудования, поддерживающие одно телефонное соединение, или цифровые каналы, также поддерживающие одно телефонное соединение и сгруппированные по принципу временного разделения каналов в тракт E1. Виртуальные порты, существующие только в течение разговорной сессии, являются портами со стороны IP сети (RTP-порты), через которые ведутся передача и прием пакетов RTP.

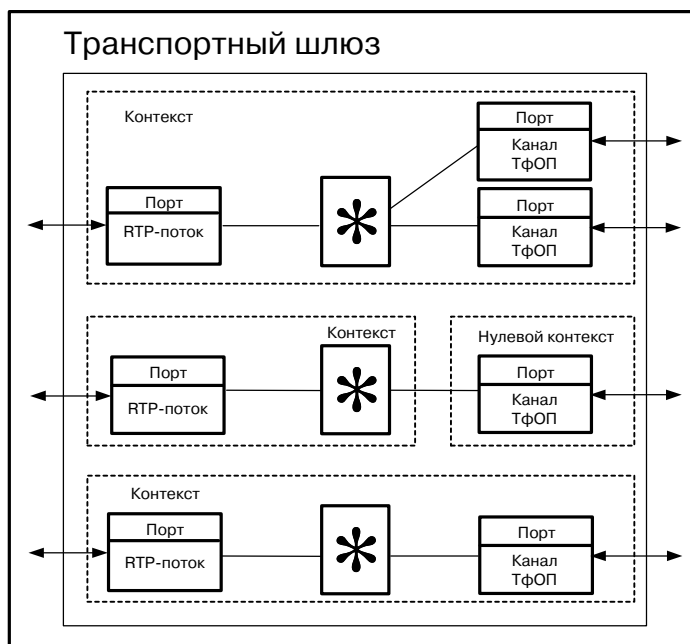


Рис. 9.2 Примеры модели процесса обслуживания вызова

Виртуальные порты создаются шлюзом при получении от контроллера команды Add и ликвидируются при получении команды Subtract, тогда как физические порты при получении команды Add или Subtract, соответственно, выводятся из нулевого контекста или возвращаются обратно в нулевой контекст.

Порт имеет уникальный идентификатор (TerminationID), который назначается шлюзом при конфигурации порта. Например, идентификатором порта может служить номер тракта E1 и номер временного канала внутри тракта. Иногда команды могут относиться ко всему шлюзу, тогда используется специальный идентификатор порта (TerminationID) – «Root».

Порты обладают рядом свойств (properties), каждое из которых имеет уникальный идентификатор (propertyID). Например, порты могут обладать свойствами генерировать речевые подсказки, акустические и вызывные сигналы, а также детектировать сигналы DTMF.

При создании портов некоторые свойства присваиваются им по умолчанию. При помощи протокола MEGACO контроллер может изменять свойства портов шлюза. Свойства портов группируются в дескрипторы, которые включаются в команды управления портами (таблица 9.1).

Таблица 9.1 Дескрипторы протокола MEGACO

Название дескриптора	Описание
Modem	Идентифицирует тип и параметры модема
Mux	Описывает тип мультимплексирования информации, используемый мультимедийными терминалами, например, H.221, H.223, H.225.0
Media	Специфицирует параметры информационного потока
TerminationState	Специфицирует свойства порта шлюза. Дескриптор содержит два параметра. Параметр ServiceStates описывает статус порта (работает в тестовом режиме - test, находится в нерабочем состоянии - out of service, по умолчанию указывается, что порт работает в нормальном режиме - in service). Параметр BufferedEventProcessingMode описывает реакцию шлюза на событие, о котором не надо немедленно оповещать контроллер. Определены две реакции на событие: игнорировать или обработать
Stream	Включает в себя ряд дескрипторов (Remote, Local, LocalControl, Signals, Events), специфицирующих параметры отдельного двунаправленного информационного потока
Local	Содержит параметры, описывающие информационный поток, передаваемый или принимаемый данным шлюзом. Информация, содержащаяся в этом дескрипторе, переносится от одного шлюза к другому
Remote	Содержит параметры, описывающие информационный поток, передаваемый или принимаемый удаленным шлюзом. Информация, содержащаяся в этом дескрипторе, переносится от одного шлюза к другому
LocalControl	Содержит параметр Mode - режим работы и ряд свойств порта. Параметр Mode может принимать значения send-only, receive-only, send/receive, inactive, loop-back и delete. Дескриптор передается на участке между шлюзом и контроллером
Events	Определяет события, которые шлюз должен отслеживать, и реакцию на эти события. Определены следующие реакции: NotifyAction (известить контроллер), Accumulate (сохранить информацию о событии в буфере), AccumulateByDigitMap (накопить цифры номера в соответствии с планом нумерации), KeepActive (известить контроллер, и продолжить передачу сигнала)
Signals	Описывает сигналы конечному пользователю, передачу которых порт шлюза должен начать или прекратить
Audit	Содержит информацию (в виде ряда дескрипторов), которую контроллер запрашивает у шлюза. Посылается в командах AuditValue и AuditCapabilities
Packages	Описывает совокупность свойств порта, передается в команде AuditValue
DigitMap	При помощи этого дескриптора контроллер информирует шлюз об используемом плане нумерации
ServiceChange	Содержит информацию, относящуюся к изменению состояния порта шлюза, такую как причина, метод изменения и др.
ObservedEvents	Содержит информацию о произошедших событиях. Передается в командах Notify и AuditValue
Statistics	Содержит статистическую информацию, собранную портом за время соединения
Extension	Позволяет передавать информацию, не специфицированную в протоколе

Контекст – это отображение связи между несколькими портами, то есть абстрактное представление соединения двух или более портов одного шлюза. В любой момент времени порт может относиться только к одному контексту, который имеет свой уникальный идентификатор. Существует особый вид контекста – нулевой. Все порты, входящие в нулевой контекст, не связаны ни между собой, ни с другими портами. Например, абстрактным представлением свободного (не занятого) канала в модели процесса обслуживания вызова является порт в нулевом контексте.

В общем случае для присоединения порта к контексту служит команда Add. При этом, если контроллер не специфицирует существующий контекст, к которому должен быть добавлен порт, то шлюз создает новый контекст.

Если шлюз поддерживает конференцию, то контекст определяет топологию связей между портами, участвующими в конференции, то есть возможные направления потоков информации для каждой пары портов. Примеры топологий, поддерживаемых протоколом MEGACO/H.323, приведены на рис. 9.3.

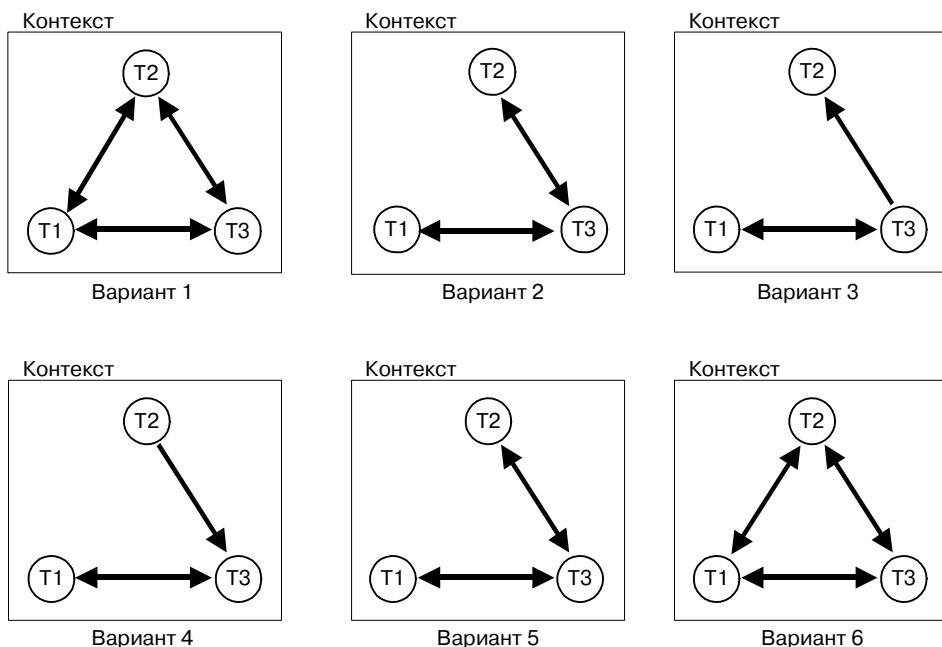


Рис. 9.3 Варианты топологии связей между портами внутри одного контекста

Краткое описание вариантов топологии связей в конференции, представленных на рис. 9.3, сведено в таблицу 9.2.

Таблица 9.2 Описание вариантов топологии

Вариант топологии	Описание
1	Топология связей не специфицирована, любой порт передает информацию другим портам и принимает информацию от других портов, участвующих в конференции
2	Порты 1 и 2 изолированы друг от друга, порт 3 передает информацию портам 1 и 2 и принимает информацию от них
3	Порты 1 и 2 изолированы друг от друга, порт 2 только принимает информацию от порта 3, обмен информацией между портами 1 и 3 - двусторонний
4	Порты 1 и 2 изолированы друг от друга, порт 3 только принимает информацию от порта 2 и обменивается информацией с портом 1
5	Двусторонняя связь между окончаниями 2 и 3 (как во втором случае)
6	Двусторонняя связь между всеми окончаниями (как в первом случае)

Следует отметить, что порты шлюза не знают о режиме, который поддерживают другие порты, участвующие в конференции.

9.3 Сравнительный анализ протоколов MGCP и MEGACO

Цель данного параграфа – определить, в чем сходны и чем различаются протоколы MGCP и MEGACO. Начнем с общих черт протоколов.

Оба протокола используются в сетях с одинаковой архитектурой, где транспортными шлюзами управляют высокоинтеллектуальные контроллеры. Оба протокола умеют работать со шлюзами одних и тех же видов, классификация шлюзов была дана в предыдущей главе. Порты шлюзов поддерживают детектирование одних и тех же событий и генерацию одних и тех же сигналов. Используются одинаковые транспортные механизмы для доставки сообщений систем сигнализации ОКС7, DSS1, ВСК. Процедуры установления и разрушения соединений, реализуемые обоими протоколами, идентичны. Кроме того, используются одинаковые механизмы поддержания защиты сети. На этом сходство протоколов MGCP и MEGACO/Н.248 заканчивается.

Самым важным отличием протокола MEGACO/Н.248 от протокола MGCP является использование иной модели организации связи. Протокол MEGACO/Н.248 работает не только с телефонными портами, но и UDP-портами. Кроме того, connection в модели MGCP – это, в общем случае, подключение к соединению между портами разного оборудования, в то время как context в модели MEGACO/Н.248 всегда отображает связь между портами одного шлюза (рис. 9.4).

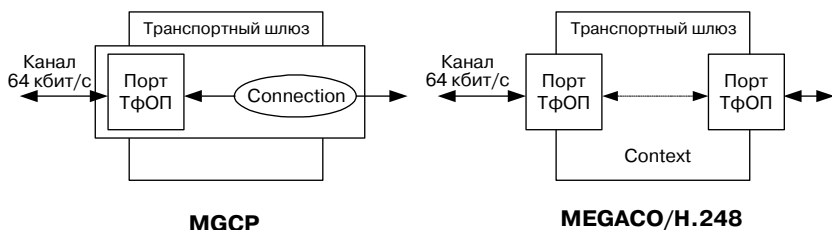


Рис. 9.4 Модели MGCP и MEGACO/Н.248

Меняя топологию связей портов, относящихся к одному контексту, при помощи протокола MEGACO контроллер может гибко управлять конференциями. Данной возможности в протоколе MGCP не предусмотрено.

Выше уже отмечалось, что для протокола MEGACO/Н.248 предусмотрено два способа кодирования, тогда как сообщения протокола MGCP представляются в текстовом формате, а бинарный способ кодирования не поддерживается. Кроме того, в протоколах используются разные параметры команд и коды ошибок.

Протокол MEGACO/Н.248, так же, как и протокол MGCP, предусматривает корреляцию команд и ответов. Но если в протоколе MGCP транзакция образуется из команды и ответа на нее, то в протоколе MEGACO/Н.248 транзакция состоит из запроса – совокупности акций и отклика на запрос. Общая структура запроса выглядит так:

```
TransactionRequest(TransactionID {
    ContextID {Command ... Command},
    ContextID {Command ... Command } })
```

Общая структура отклика на запрос приведена ниже:

```
TransactionReply(TransactionID {
    ContextID { Response ... Response },
    ContextID { Response ... Response } })
```

Каждая акция, в свою очередь, состоит из одной или нескольких команд, относящихся к одному контексту, и ответов на них (Рис. 9.5). Использование такого инструмента позволяет значительно уменьшить объем передаваемой сигнальной информации и увеличить скорость установления соединений за счет того, что контроллер может параллельно вести обработку сигнальной информации, относящейся к разным соединениям.

Аналоги двух избыточных команд EndpointConfiguration и NotificationRequest протокола MGCP в протоколе MEGACO/Н.248 отсутствуют, но, в тоже время, добавлена команда Move, позволяющая в одно действие перевести порт из одного контекста в другой. В качестве примера использования команды Move приведем сценарий допол-

нительных услуг «Уведомление о входящем вызове и перевод существующего соединения в режим удержания», англоязычное название услуг – Call Waiting и Call Hold.

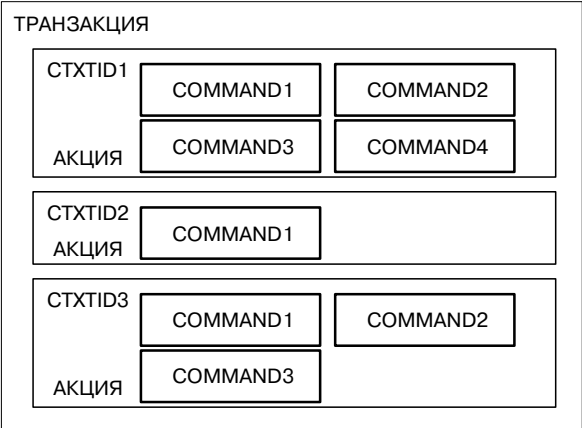


Рис. 9.5 Транзакция протокола MEGACO/H.248

Абонент А разговаривает с абонентом В, а абонент С вызывает абонента А, при этом вызываемому абоненту передается акустическое уведомление о входящем вызове (рис. 9.6). Далее абонент А переводит соединение с абонентом В в режим удержания и соединяется с абонентом С (рис. 9.7). Реализация комбинации дополнительных услуг Call Waiting и Call Hold, т.е. передача порта из одного контекста в другой, стала возможной благодаря команде Move.

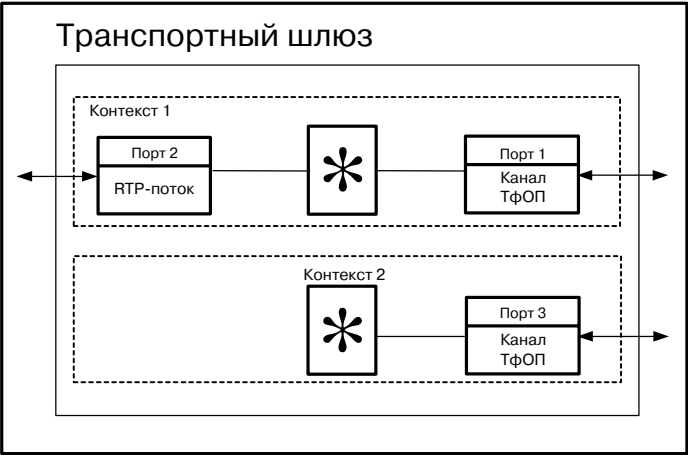


Рис. 9.6 Сценарий реализации услуги Call Waiting

В заключение данного параграфа хотелось бы отметить, что неизвестно, как скоро понадобится расширенная, по сравнению с MGCP, функциональность протокола MEGACO/H.248. Кроме того, на

базе протокола MGCP построен ряд сетей IP-телефонии. Все это означает, что оба протокола MGCP и MEGACO/H.248 вполне могут совместно использоваться в одной сети.

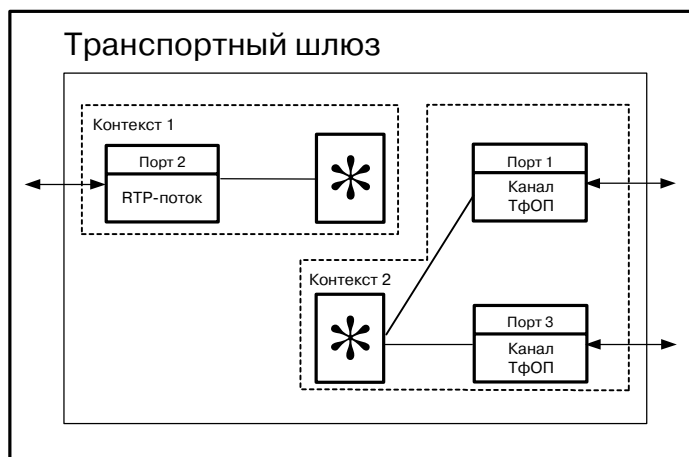


Рис. 9.7 Сценарий реализации услуги Call Hold

9.4 Структура команд и ответов

Далее идет описание команд, которые используются для манипулирования двумя основными объектами протокола MEGACO/H.248: портами и контекстами. В большинстве случаев команды передает контроллер, но существуют два исключения: команда Notify, передается шлюзом, а команда ServiceChange может передаваться и шлюзом, и контроллером. В квадратных скобках указаны необязательные дескрипторы команд. Те дескрипторы, которые расположены над командами, передаются в ответах на команды.

Команда Add добавляет порт к контексту. Если команда относится к первому порту, который должен быть добавлен к контексту, то создается новый контекст.

```
[TerminationID]
[,MediaDescriptor]
[,ModemDescriptor]
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
Add( TerminationID
    [, MediaDescriptor]
    [, ModemDescriptor]
    [, MuxDescriptor]
    [, EventsDescriptor]
    [, SignalsDescriptor]
    [, DigitMapDescriptor]
    [, AuditDescriptor]
),
```

где TerminationID – это идентификатор порта, который должен быть добавлен к контексту. Для уже существующего порта должен быть указан его идентификатор, для несуществующего порта должен быть указан идентификатор «\$». В ответе на команду должен передаваться TerminationID, назначенный шлюзом.

MediaDescriptor – необязательный дескриптор, описывающий информационные потоки.

ModemDescriptor – необязательный дескриптор, описывающий тип модема, который должен быть подключен к контексту.

MuxDescriptor – необязательный дескриптор, содержащий список портов, которые должны быть подключены к контексту.

EventsDescriptor – необязательный дескриптор, определяющий список событий, при детектировании которых порт должен оповестить контроллер.

SignalsDescriptor – необязательный дескриптор, определяющий сигналы, которые порт должен передавать в канал.

DigitMapDescriptor – необязательный дескриптор, определяющий план нумерации, который должен быть использован для соединения.

AuditDescriptor – необязательный дескриптор, специфицирующий параметры порта, которые должны быть переданы шлюзом контроллеру.

PackagesDescriptor – необязательный дескриптор, описывающий пакет поддерживаемых сигналов и событий.

Команда Modify изменяет свойства, события или сигналы для существующего порта.

```
[TerminationID]
[,MediaDescriptor]
[,ModemDescriptor]
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
Modify( TerminationID
        [, MediaDescriptor]
        [, ModemDescriptor]
        [, MuxDescriptor]
        [, EventsDescriptor]
        [, SignalsDescriptor]
        [, DigitMapDescriptor]
        [, AuditDescriptor]
      )
```

Если команда относится к конкретному порту шлюза, участвующего в контексте, то должен быть указан идентификатор порта.

В команде Modify используются такие же дескрипторы, как и в команде Add.

Команда Subtract отключает порт от существующего контекста.

```
[TerminationID]
[,MediaDescriptor]
[,ModemDescriptor]
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
    Subtract(TerminationID
              [, AuditDescriptor]
    )
```

где TerminationID – идентификатор порта, который должен быть отсоединен от контекста. В случае отключения всех портов от контекста используется TerminationID «*».

В ответ на команду Subtract в дескрипторе StatisticsDescriptor шлюз посылает статистику, собранную за время соединения.

Команда Move переводит порт из текущего контекста в другой контекст в одно действие.

```
[TerminationID]
[,MediaDescriptor]
[,ModemDescriptor]
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
    Move( TerminationID
          [, MediaDescriptor]
          [, ModemDescriptor]
          [, MuxDescriptor]
          [, EventsDescriptor]
          [, SignalsDescriptor]
          [, DigitMapDescriptor]
          [, AuditDescriptor]
    )
```

где TerminationID – идентификатор порта, который должен быть переведен из одного контекста в другой. Дескрипторы здесь используются такие же, как в команде Modify.

При помощи команды AuditValue контроллер запрашивает сведения о свойствах порта, произошедших событиях или сигналах, передаваемых в канал, а также статистику, собранную на текущий момент.

```
[TerminationID]
[,MediaDescriptor]
[,ModemDescriptor]
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
    AuditValue(TerminationID,
               AuditDescriptor
    )
```

В ответ на команду передаются запрашиваемые параметры порта или портов шлюза.

При помощи команды `AuditCapabilities` контроллер запрашивает возможные значения свойств порта, список событий, которые могут быть обнаружены портом, список сигналов, которые порт может передавать в канал, статические данные.

```
[TerminationID]
[,MediaDescriptor]
[,ModemDescriptor]
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
    AuditCapabilities(TerminationID,
                      ,AuditDescriptor
                      )
```

В ответ на команду передаются запрашиваемые параметры порта.

Команда `Notify` служит для того, чтобы известить контроллер о событиях, которые произошли в шлюзе.

```
Notify(TerminationID,
       ObservedEventsDescriptor
),
```

где `TerminationID` идентифицирует порт, передавший команду `Notify`.

`ObservedEventsDescriptor` – дескриптор, содержащий список произошедших событий (в том порядке, в каком они происходили).

Команда `ServiceChange` позволяет шлюзу известить контроллер о том, что порт или группа портов вышли из обслуживания или вернулись в обслуживание. `Media Gateway Controller` может предписать порту выйти из обслуживания или вернуться в обслуживание. При помощи данной команды контроллер может передать управление шлюзом другому контроллеру.

```
[ServiceChangeDescriptor]
    ServiceChange(TerminationID
                  ,ServiceDescriptor
                  ),
```

где `TerminationID` идентифицирует порт или порты, вышедшие из обслуживания или вернувшиеся в обслуживание. Значение «Root» дескриптора `TerminationID` показывает, что весь шлюз вышел из обслуживания или вернулся в обслуживание.

ServiceDescriptor – дескриптор, содержащий поля со сведениями: о методе изменения состояния; причине изменения; задержке; адресе, куда должны передаваться сообщения; профиле поддерживаемого протокола и другие поля.

По аналогии с предыдущими главами, в таблицу 9.3 сведены все команды протокола MEGACO/H.248.

В заключение данного параграфа в таблице 9.4 приведены коды ошибок, используемые в протоколе MEGACO/H.248.

Таблица 9.3 Команды протокола MEGACO/H.248

Команда	Направление передачи	Назначение
Add (Добавить)	MGC → MG	Контроллер дает указание шлюзу добавить порт к контексту
Modify (Изменить)	MGC → MG	Контроллер дает указание шлюзу изменить свойства порта
Subtract (Отключить)	MGC → MG	Контроллер изымает порт из контекста
Move (Перевести)	MGC → MG	Контроллер переводит порт из одного контекста в другой в одно действие
AuditValue (Проверить порт)	MGC → MG	Контроллер запрашивает свойства порта, произошедшие события или сигналы, передаваемые в канал, а также статистику, собранную на текущий момент времени
AuditCapabilities (Проверить возможности порта)	MGC → MG	Контроллер запрашивает возможные значения свойств порта, список событий, которые могут быть выявлены портом, список сигналов, которые порт может посылать в канал, статические данные
Notify (Уведомить)	MG → MGC	Шлюз информирует контроллер о произошедших событиях
ServiceChange (Рестарт)	MG → MGC, MGC → MG	Шлюз информирует контроллер о том, что один или несколько портов выходят из рабочего состояния или возвращаются в рабочее состояние. Контроллер может предписать порту или группе портов выйти из обслуживания или вернуться в обслуживание

Таблица 9.4 Коды ошибок

Код ошибка	Описание
400	Некорректный запрос
401	Ошибка в протоколе
402	Авторизация не подтверждена
403	Синтаксическая ошибка в транзакции
410	Некорректный идентификатор
411	В транзакции указан идентификатор несуществующего контекста
412	Отсутствуют свободные идентификаторы контекста
420	Нет такого события или сигнала в пакете (package)
421	Неизвестная акция или некорректная комбинация акций
422	Синтаксическая ошибка в акции
430	Неизвестный идентификатор порта
431	Несуществующий идентификатор порта
432	Отсутствуют свободные идентификаторы портов
433	Порт, с указанным идентификатором, уже добавлен к контексту
440	Не поддерживаемый или неизвестный пакет
441	Отсутствует дескриптор Remote
442	Синтаксическая ошибка в команде
443	Не поддерживаемая или неизвестная команда
444	Не поддерживаемый или неизвестный дескриптор
445	Не поддерживаемое или неизвестное свойство
446	Не поддерживаемый или неизвестный параметр
447	Дескриптор не совместим с командой
448	Два одинаковых дескриптора в команде
450	Отсутствующее в пакете свойство
451	Отсутствующее в пакете событие
452	Отсутствующий в пакете сигнал
453	Отсутствующая в пакете статистическая информация
454	Отсутствующее значение параметра в пакете
455	Параметр не совместим с дескриптором
456	Два одинаковых параметра или свойства в дескрипторе
500	Внутренняя ошибка в шлюзе
501	Не поддерживается
502	Оборудование не готово
503	Услуга не реализована
510	Недостаточно ресурсов
512	Шлюз не оборудован для детектирования требуемого события
513	Шлюз не оборудован для генерирования требуемого сигнала
514	Шлюз не может воспроизвести уведомление или подсказку
515	Не поддерживаемый вид информации
517	Не поддерживаемый или некорректный режим
518	Переполнение буфера, в котором хранится информация о произошедших событиях
519	Не хватает памяти для хранения плана нумерации
520	Шлюз не имеет информации об используемом плане нумерации
521	Порт рестартовал
526	Недостаточная полоса пропускания
529	Внутренняя неисправность в аппаратном обеспечении
530	Временная неисправность сети
531	Постоянная неисправность сети
581	Не существует

9.5 Пример установления и разрушения соединения

На рисунке 9.8 приведен пример установления соединения с использованием протокола MEGACO между двумя шлюзами (Residential Gateway), управляемыми одним контроллером.

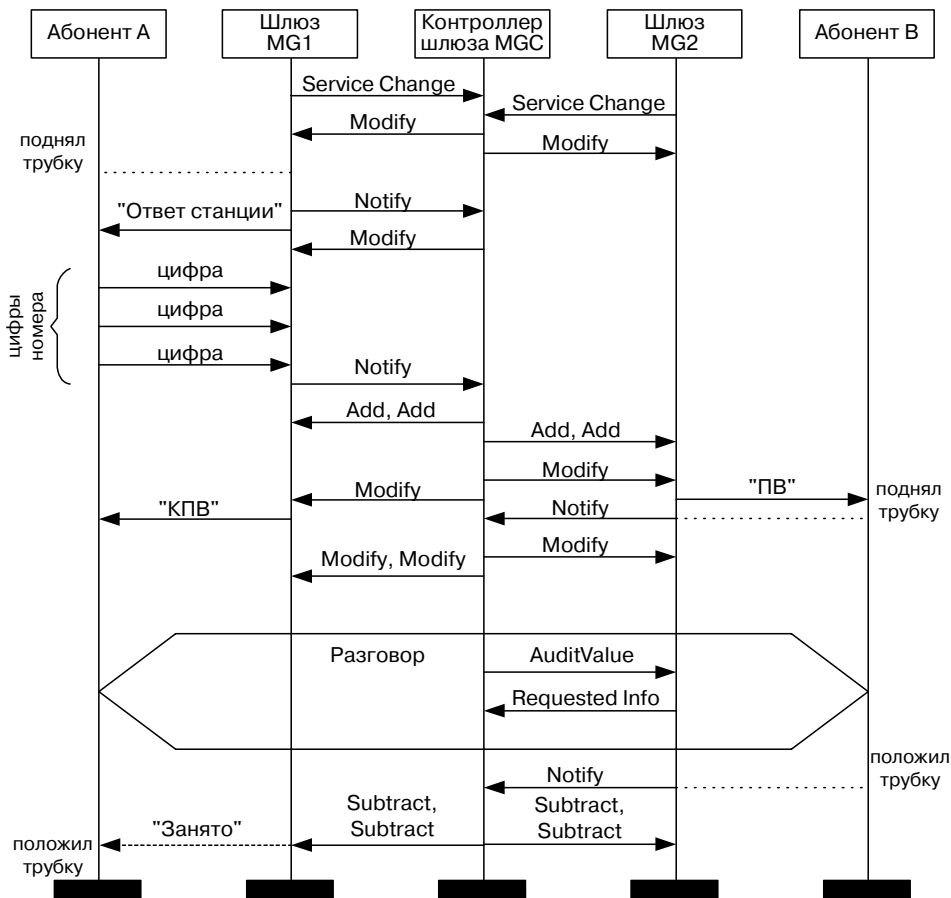


Рис. 9.8 Алгоритм установления и разрушения соединения с помощью протокола MEGACO

В данном примере вызывающий шлюз MG1 – имеет IP-адрес 124.124.124.222, адрес вызываемого шлюза MG2 – 125.125.125.111, адрес контроллера шлюзов MGC – 123.123.123.4. Порт для связи по протоколу MEGACO для всех трех устройств по умолчанию имеет значение 55555.

1. Шлюз MG1 регистрируется у контроллера MGC при помощи команды ServiceChange. Использование нулевого контекста означает, что порт в настоящий момент не участвует ни в каком соедине-

нии, а использование идентификатора порта ROOT означает, что команда относится ко всему шлюзу, а не к какому-нибудь определенному порту.

```
MG1 to MGC:
MEGACO/1.0 [124.124.124.222]
Transaction = 9998 {
    Context = - {
        ServiceChange = ROOT {Services {
            Method=Restart, Port=55555, Profile=ResGW/1.0}
        }
    }
}
```

2. Контроллер подтверждает регистрацию шлюза:

```
MGC to MG1:
MEGACO/1.0 [123.123.123.4]:55555
Reply = 9998 {
    Context = - {ServiceChange = ROOT { Services {
        Port=55555, Profile=ResGW/1.0} }
    }
}
```

3. Шлюз имеет свободные аналоговые порты, которые должны быть запрограммированы для отслеживания изменения сопротивления абонентского шлейфа, означающего поднятие абонентом трубки, после чего шлюз должен передать абоненту акустический сигнал «Ответ станции». Программирование производится при помощи команды Modify с соответствующими параметрами, причем программируется порт, находящийся в нулевом контексте. В команде указывается идентификатор порта (terminationId) – A4444, идентификатор информационного потока (streamId) – 1, транспортный адрес оборудования, передавшего команду – [123.123.123.4]:55555, специфицируется режим функционирования – дуплексный (SendReceive).

```
MGC to MG1:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 9999 {
    Context = - {
        Modify = A4444 {
            Media {
                TerminationState {
                    BufferedEventHandling{Step,Process}
                },
                Stream = 1 {
                    LocalControl {
                        Mode = SendReceive,
                        g/GainControl=2, ; in dB,
                        g/Encryption=xxx,
                        g/EchoCancellation=G165,
                        g/VoiceActDet=yes
                    }
                }
            }
        },
        Events = 2222 {glinesup/offhook}
    },
    Signals {g/PlayTone{tone=dialtone}
}
}
```

На этом же этапе в шлюз может быть загружен план нумерации (в дескрипторе digit map). В этом случае, после того как абонент поднимет трубку, шлюз должен передать ему акустический сигнал «Ответ станции» и начинать прием сигналов DTMF в соответствии с планом нумерации. Однако в нашем примере план нумерации будет загружен только после того, как абонент поднимет трубку, на 8 шаге.

Кроме того, следует отметить, что шаги 3 и 4 данного алгоритма могут быть совмещены с шагами 8 и 9, соответственно, при помощи дескриптора EventsDescriptor. При этом шаги 6 и 7 опускаются.

4. Шлюз MG1 подтверждает выполнение команды Modify:

```
MG1 to MGC:
MEGACO/1.0 [124.124.124.222]:55555
Reply = 9999 {
    Context = - {Modify}
}
```

5. Подобным же образом (шаги 1 – 4) программируется аналоговый порт шлюза MG2, в нашем примере имеющий идентификатор A5555.

6. Далее шлюз MG1 обнаруживает, что абонент А поднял трубку, и извещает об этом событии Media Gateway Controller при помощи команды Notify.

```
MG1 to MGC:
MEGACO/1.0 [124.124.124.222]:55555
Transaction = 10000 {
    Context = - {
        Notify = A4444 {ObservedEvents =2222 {
            19990729T22000000:glinesup/offhook}
        }
    }
}
```

7. Контроллер подтверждает получение команды Notify:

```
MGC to MG1:
MEGACO/1.0 [123.123.123.4]:55555
Reply = 10000 {
    Context = - {Notify}
}
```

8. На следующем шаге MGC дает шлюзу инструкцию накапливать цифры номера вызываемого абонента в соответствии с выбранным планом нумерации. Кроме того, после получения первой цифры номера необходимо остановить передачу акустического сигнала «Ответ станции».

```

MGC to MG1:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 10001 {
    Context = - {
        Modify = A4444 {
            Events = 2223 {
                glinesup/onhook {
                    Action { DigitMap=Dialplan0 }
                }
            },
            Signals { g/StopTone
            },
            DigitMap= Dialplan0{
(0T|00T|[17]xxx|8xxxxxxxx|#xxxxxxxx|*xx|91xxxxxxxxxxx|9011x.T)}
        }
    }
}

```

9. Шлюз подтверждает получение команды Modify:

```

MG1 to MGC:
MEGACO/1.0 [124.124.124.222]:55555
Reply = 10001 {
    Context = - {Modify}
}

```

10. Цифры номера вызываемого абонента собираются шлюзом MG1, после чего они сравниваются с планом нумерации и передаются к контроллеру.

```

MG1 to MGC:
MEGACO/1.0 [124.124.124.222]:55555
Transaction = 10002 {
    Context = - {
        Notify = A4444 {ObservedEvents =2223 {
            19990729T22010001:g/tonesdt{tones=«16135551212»}}
        }
    }
}

```

11. Полученное сообщение подтверждается контроллером:

```

MGC to MG1:
MEGACO/1.0 [123.123.123.4]:55555
Reply = 10002 {
    Context = - {Notify}
}

```

12. Далее контроллер MGC анализирует цифры номера вызываемого абонента, полученные от шлюза MG1, и определяет, что соединение должно проходить через шлюз MG2, к которому подключен вызываемый абонент. К вновь образованному контексту в шлюзе MG1 добавляются при помощи команды Add физический порт (аналоговый абонентский интерфейс) – A4444 и виртуальный порт (RTP-порт). Так как в этот момент шлюз, к которому прикреплен вызывающий абонент, не имеет информации о шлюзе MG2 (такой как IP-адрес, RTP-порт и поддерживаемые алгоритмы декодирования принимаемой речевой информации), то контроллер MGC предписывает шлюзу MG1 только принимать информацию (режим ReceiveOnly). Кроме того, контроллер MGC специфицирует в команде Add предпочтительные для использования алгоритмы кодирования.

```

MGC to MG1:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 10003 {
    Context = $ {
        Add = A4444 {
            Media {
                Stream = 1 {
                    Events = 2224 {glinesup/onhook}
                }
            }
        },
        Add = $ {
            Media {
                Stream = 1 {
                    LocalControl {
                        Mode = ReceiveOnly,
                        g/NetworkType = RTP/IP4, ;
                        g/MaxJitterBuffer=40, ; in ms
                        g/PreferredPacketization=30, ; in ms
                        g/PreferredPacketization=30, ; in ms
                        g/PreferredEncoders =[G723, PCMU],
                        g/PreferredDecoders=[G723, PCMU],
                        g/Gain=0 ; in dB
                    }
                }
            }
        }
    }
}

```

Примечательно то, что MGC указывает предпочтительные параметры в дескрипторе LocalControl, как это показано выше. В требовании, получаемом шлюзом от контроллера, отсутствуют дескрипторы Local и Remote, в которых от вызывающей стороны к вызываемой переносится вся адресная информация и сведения об используемых алгоритмах кодирования. Дескрипторы Local и Remote передаются шлюзом в ответе на это требование.

13. Шлюз MG1 создает новый виртуальный порт (RTP-порт), указывает его транспортный адрес: IP-адрес (124.124.124.222) и UDP/RTP-порт (22220). Кроме того, шлюз выбирает алгоритмы кодирования информации, которые будут использоваться в соединении, основываясь на предпочтениях контроллера.

```

MG1 to MGC:
MEGACO/1.0 [124.124.124.222]:55555
Reply = 10003 {
    Context = 2000 {
        Add,
        Add=A4444{
            Media {
                Stream = 1 {
                    Local = SDP {
                        v=0
                        c=IN IP4 124.124.124.222
                        m=audio 2222 RTP/AVP 4 0
                        a=ptime:20
                        a=recvonly
                    } ; RTP profile for G.723 is 4
                }
            }
        }
    }
}

```

14. Контроллер MGC создает в шлюзе MG2 контекст для установления дуплексного соединения (режим SendReceive) с вызывающим пользователем.

```
MGC to MG2:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 50003 {
  Context = $ {
    Add = A5555 {
      Media {
        Stream = 1 {
        }
      }
    },
    Add = $ {
      Media {
        Stream = 1 {
          LocalControl {
            Mode = SendReceive,
            g/NetworkType = RTP/IP4,
            g/MaxJitterBuffer=40, ; in ms
            g/PreferredPacketization=30, ; in ms
            g/PreferredEncoders =[G723, PCMU],
            g/PreferredDecoders=[G723, PCMU],
            g/Gain=0 ; in dB
          },
          Remote=SDP{
            v=0
            c=IN IP4 124.124.124.222
            m=audio 2222 RTP/AVP 4 0
            a=sendrecv
          } ; RTP profile for G.723 is 4
        }
      }
    }
  }
}
```

15. Создание контекста подтверждается, физический порт шлюза MG2 A5555 соединяется с UDP/RTP портом, имеющим идентификатор A5556. Отметим, что RTP-порт имеет номер 1111, т.е. отличный от номера порта Megaco/H.248 – 55555.

```
MG2 to MGC:
MEGACO/1.0 [124.124.124.222]:55555
Reply = 50003 {
  Context = 5000 {
    Add,
    Add = A5556{
      Media {
        Stream = 1 {
          Local = SDP {
            v=0
            c=IN IP4 125.125.125.1111
            m=audio 1111 RTP/AVP 4 0
            a=sendreceive
          }
        } ; RTP profile for G723 is 4
      }
    }
  }
}
```

16. Контроллер MGC предписывает порту A5555 шлюза MG2 начать передачу вызывного сигнала.

```
MGC to MG2:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 50004 {
  Context = 5000 {
    Modify = A5555 {
      Signals {glinesup/PlayTone{tone=ring}}
    }
  }
}
```

17. Шлюз MG2 подтверждает передачу сигнала «Посылка вызова» вызываемому абоненту.

```
MG2 to MGC:
MEGACO/1.0 [125.125.125.111]:55555
Reply = 50004 {
  Context = 5000 {Modify}
}
```

18. Контроллер предписывает шлюзу MG1 начать передачу вызываемому абоненту акустического сигнала «Контроль посылки вызова (КПВ)».

```
MGC to MG1:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 10005 {
  Context = 2000 {
    Modify = A4444 {
      Signals {g/PlayTone{tone=ringback}}
    }
  }
}
```

19. Шлюз MG1 подтверждает передачу указанного акустического сигнала в порт A4444.

```
MG1 to MGC:
MEGACO/1.0 [124.124.124.222]:55555
Reply = 10005 {
  Context = 2000 {Modify}
}
```

На этом этапе обоим абонентам, участвующим в соединении, посылаются соответствующие сигналы, и шлюз MG2 ждет, пока вызываемый абонент примет входящий вызов, после чего между двумя шлюзами будут организованы двунаправленные разговорные каналы.

20. Шлюз MG2 обнаружил, что вызываемый абонент поднял трубку, и извещает об этом контроллер MGC.

```
MG2 to MGC:
MEGACO/1.0 [125.125.125.111]:55555
Transaction = 50005 {
  Context = 5000 {
    Notify = A5555 {ObservedEvents =1234 {
      19990729T22020002:glinesup/offhook}
    }
  }
}
```

21. Контроллер подтверждает получение команды Notify.

```
MGC to MG2:
MEGACO/1.0 [123.123.123.4]:55555
Reply = 50005 {
    Context = - {Notify}
}
```

22. Далее контроллер MGC предписывает шлюзу MG2 прекратить передачу вызывного сигнала.

```
MGC to MG2:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 50006 {
    Context = 5000 {
        Modify = A5555 {
            Events = 1235 {glinesup/onhook},
            Signals {g/StopTone} ; to turn off ringing
        }
    }
}
```

23. Шлюз MG2 подтверждает выполнение команды.

```
MG2 to MGC:
MEGACO/1.0 [125.125.125.111]:55555
Reply = 50006 {
    Context = 5000 {Modify}
}
```

24. Далее, контроллер разрешает шлюзу MG1 не только принимать, но и передавать информацию (режим SendReceive), и останавливает передачу вызывающему абоненту акустического сигнала «КПВ».

```
MGC to MG1:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 10006 {
    Context = 2000 {
        Modify = A4445 {
            Media {
                Stream = 1 {
                    LocalControl {
                        Mode=SendReceive
                    }
                }
            }
        },
        Modify = A4444 {
            Signals { g/StopTone}
        }
    }
}
```

25. Шлюз MG1 подтверждает выполнение команды.

```
MG1 to MGC:
MEGACO/1.0 [124.124.124.222]:55555
Reply = 10006 {
    Context = 2000 {Modify, Modify}}
```

26. После этого начинается разговорная фаза соединения, в течение которой участники обмениваются речевой информацией. Следующим шагом контроллер MGC принимает решение проверить RTP-порт в шлюзе MG2.

```

MGC to MG2:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 50007 {
    Context = - {AuditValue = A5556{
        Audit{Media, DigitMap, Events, Signals, Packages, Statistics }}
    }
}

```

27. Шлюз MG2 выполняет команду. В ответе на команду AuditValue передается вся запрашиваемая информация, в том числе статистика, собранная за время соединения. Кроме того, из ответа видно, что не произошло никаких событий и не передавалось никаких сигналов.

```

MEGACO/1.0 [125.125.125.111]:55555
Reply = 50007 {
    Context = - {
        AuditValue {
            Media {
                TerminationState {
                    BufferedEventHandling{Process}
                },
                Stream = 1 {
                    LocalControl {
                        Mode = SendReceive,
                        g/MaxJitterBuffer=40, ; in ms
                        g/PreferredPacketization=30, ; in ms
                        g/PreferredEncoders =[G723, PCMU],
                        g/PreferredDecoders=[G723, PCMU],
                        g/Gain=0 ; in dB
                    },
                    Local = SDP {
                        v=0
                        c=IN IP4 125.125.125.111
                        m=audio 1111 RTP/AVP 4 0
                        a=sendrecv
                    },
                    Remote = SDP{
                        v=0
                        c=IN IP4 124.124.124.222
                        m=audio 2222 RTP/AVP 4 0
                        a=sendrecv
                    } ; RTP profile for G.723 is 4
                }
            },
            Packages {g, glinesup, RTPPkg},
            Statistics { RTPPkg/PacketsSent=1200,
                RTPPkg/OctetsSent=62300,
                RTPPkg/PacketsReceived=700,
                RTPPkg/OctetsReceived=45100,
                RTPPkg/PacketsLost=6,
                RTPPkg/Jitter=20,
                RTPPkg/AverageLatency=40 }
        }
    }
}

```


28. Вызываемый абонент первым завершает соединение, и шлюз MG2 извещает об этом контроллер MGC.

```
MG2 to MGC:
MEGACO/1.0 [125.125.125.111]:55555
Transaction = 50008 {
    Context = 5000 {
        Notify = A5555 {ObservedEvents =1235 {
            19990729T24020002:glinesup/onhook}
        }
    }
}
```

29. Контроллер MGC подтверждает получение сообщения Notify.

```
MGC to MG2:
MEGACO/1.0 [123.123.123.4]:55555
Reply = 50008 {
    Context = - {Notify}
}
```

30. Получив информацию от любого из шлюзов о том, что один из абонентов положил трубку, контроллер MGC завершает соединение. К обоим шлюзам передается команда Subtract. Алгоритм завершения соединения предусматривает одинаковый обмен сигнальными сообщениями между контроллером и обоими шлюзами, поэтому здесь этот алгоритм рассматривается на примере шлюза MG2.

```
From MGC to MG2:
MEGACO/1.0 [123.123.123.4]:55555
Transaction = 50009 {
    Context = 5000 {
        Subtract = A5555 {Audit{Statistics}},
        Subtract = A5556 {Audit{Statistics}}
    }
}
```

31. Каждый из портов шлюза MG2, участвующих в соединении (физический порт – A5555 и RTP-порт – A5556), возвращает статистику, собранную за время соединения. В общем случае, контроллер может запрашивать статистическую информацию только у одного из портов.

```
From MG2 to MGC:
MEGACO/1.0 [125.125.125.111]:55555
Reply = 50009 {
    Context = 5000 {
        Subtract {
            Statistics { ; what are the stats for a TDM connection?
                TDMPkg/OctetsSent=45123,
                TDMPkg/Duration=40 ; in seconds
            }
        },
        Subtract {
            Statistics {
                RTPPkg/PacketsSent=1245,
                RTPPkg/OctetsSent=62345,
                RTPPkg/PacketsReceived=780,
                RTPPkg/OctetsReceived=45123,
                RTPPkg/PacketsLost=10,
                RTPPkg/Jitter=27,
                RTPPkg/AverageLatency=48
            }
        }
    }
}
```

32. После завершения соединения контроллер MGC предписывает шлюзам MG1 и MG2 быть готовыми к тому, что кто-то из обслуживаемых ими абонентов поднимет трубку. Примечательно, что портам шлюза, отображаемым окончаниями в нулевом контексте, по умолчанию может быть предписано обнаруживать, что абонент поднял трубку, при этом контроллер не передает шлюзам специальные команды, как это было показано ранее (шаг 3).

Глава 10

Качество обслуживания в сетях IP-телефонии

10.1 Что понимается под QoS?

Характер информации, передаваемой по сетям с маршрутизацией пакетов IP, сегодня драматически меняется. Кроме передачи данных, IP-сети используются для прослушивания музыкальных программ, просмотра видеоклипов, обмена речевой информацией, проведения мультимедийных конференций, оперативного контроля/управления, сетевых игр и других приложений реального времени.

Протокол IP, подробно рассмотренный в Главе 4, первоначально не предназначался для обмена информацией в реальном времени. Ведь пакеты одного и того же потока данных маршрутизируются по сети независимо друг от друга, а время обработки пакетов в узлах может меняться в широких пределах, в силу чего такие параметры передачи как задержка и вариация задержки пакетов также могут меняться. А параметры качества сетевых услуг, обеспечивающих передачу информации в реальном времени, как известно, сильно зависят от характеристик задержек пакетов, в которых эта информация переносится.

Транспортные протоколы стека TCP/IP, реализуемые в оборудовании пользователей и функционирующие поверх протокола IP, также не обеспечивают высокого качества обслуживания трафика, чувствительного к задержкам. Протокол TCP, хоть и гарантирует достоверную доставку информации, но переносит ее с непредсказуемы-

ми задержками. Протокол UDP, который, как правило, используется для переноса информации в реальном времени, обеспечивает меньшее, по сравнению с протоколом TCP, время задержки, но, как и протокол IP, не содержит никаких механизмов обеспечения качества обслуживания.

Кроме того, в самой сети Интернет нет никаких механизмов, поддерживающих на должном уровне качество передачи информации в реальном времени. Иными словами, ни в узлах IP-сетей, ни в оборудовании пользователей в настоящее время нет средств, обеспечивающих гарантированное качество обслуживания.

Вместе с тем, налицо необходимость получения от сети гарантий, что в периоды перегрузки пакеты с информацией, чувствительной к задержкам, не будут простаивать в очередях или, по крайней мере, получат более высокий приоритет, чем пакеты с информацией, не чувствительной к задержкам. Иначе говоря, необходимо гарантировать доставку такой информации, как речь, видео и мультимедиа, в реальном времени с минимально возможной задержкой. Для этой цели в сети должны быть реализованы механизмы, гарантирующие нужное качество обслуживания (Quality of Service – QoS). Анализу таких механизмов и посвящена эта глава.

Идеальной была бы следующая ситуация. Приложение «договаривается» с сетью о том, что пакеты такого-то потока данных со средней скоростью передачи X Кбит/с будут доставляться от одного конца соединения к другому с задержкой не более Y мс, и что сеть в течение всего соединения будет следить за выполнением этого договора. Кроме указанной характеристики, сеть должна поддерживать согласованные значения таких параметров передачи как минимально доступная полоса частот, максимальное изменение задержки (джиттер), максимальные потери пакетов.

В конечном счете, качество обслуживания зависит не только от сети, но и от оборудования пользователя. Слабые системные ресурсы оборудования пользователя – малый объем оперативной памяти, невысокая производительность центрального процессора и др. – могут сделать показатели качества обслуживания неприемлемыми для пользователя вне зависимости от того, как соблюдает «договоренность» сеть. Хорошее качество обслуживания достигается лишь тогда, когда пользователь удовлетворительно оценивает работу системы в целом.

Следует отметить, что высокое качество обслуживания представляет интерес не только для конечного пользователя, но и для самого поставщика услуг. Например, исследования, проведенные в сетях мобильной связи, показали, что с улучшением качества передачи речи абоненты чаще и дольше пользуются услугами таких сетей, что означает увеличение годовых доходов операторов.

Чтобы добиться гарантий качества обслуживания от сетей, изначально на это не ориентированных, необходимо «наложить» на сеть так называемую QoS-архитектуру, которая включает в себя поддержку качества на всех уровнях стека протоколов TCP/IP и во всех сетевых элементах. Но и при этом обеспечение гарантированного качества обслуживания все равно остается самым слабым местом процесса передачи информации от источника к приемнику.

Поскольку все больше приложений становятся распределенными, все больше возрастает потребность в поддержке качества обслуживания на нижних сетевых уровнях. Это может вызвать определенные трудности, так как даже стандартные операционные системы рабочих станций не поддерживают доставку информации в реальном времени.

Кроме того, качество обслуживания – это относительное понятие; его смысл зависит от приложения, с которым работает пользователь. Как уже отмечалось раньше, разные приложения требуют разных уровней или типов качества. Например, скорее всего, пользователя не огорчит тот факт, что его текстовый файл будет передаваться на секунду дольше, или что за первую половину времени передачи будет передано 80% файла, а за вторую – 20%. В то же время, при передаче речевой информации такого рода явления весьма нежелательны или даже недопустимы.

10.2 Качество обслуживания в сетях пакетной коммутации

Идея обеспечить гарантированное качество обслуживания в сетях передачи данных впервые возникла в 1970 году, когда некий пользователь получил вместо одних данных другие. Идея была воплощена в сети X.25. Однако пакетные системы X.25, производя проверку ошибок на каждом сетевом узле, вносили задержку порядка нескольких сотен миллисекунд в каждом узле на пути информации от отправителя до получателя.

В сетевых узлах (коммутаторах пакетов) высокоскоростных транспортных сетей Frame Relay проверка и коррекция ошибок не производятся. Эти функции возложены на оборудование пользователя, вследствие чего задержка при передаче информации по таким сетям намного ниже, чем в сетях X.25.

Одной из широко известных технологий пакетной передачи с гарантированным качеством обслуживания является транспортная технология ATM. И хотя не так давно на ATM возлагались огромные надежды (предполагалось, например, использование ATM в качестве базы для широкополосных сетей ISDN), эта технология не по-

лучила широкого распространения из-за своей сложности и высокой стоимости оборудования. Скорее всего, технология АТМ будет использоваться на магистральных участках сетей связи до тех пор, пока ее не вытеснят более простые и скоростные транспортные технологии.

Но самой популярной сегодня технологией пакетной передачи информации является технология маршрутизации пакетов IP. Объем потоков данных, передаваемых по глобальной информационной сети Интернет, удваивается каждые три месяца. Частично это происходит из-за постоянного увеличения количества новых пользователей сети Интернет, а также из-за того, что мультимедийная передача и видеоконференции через Интернет стали, наконец-то, доступны и популярны среди обеспеченных пользователей. Качество обслуживания в этой сети привлекает все более пристальное внимание специалистов и пользователей, так как в Интернет заключается все больше сделок и контрактов, а рост ее пропускной способности несколько отстает от роста спроса.

10.3 Трафик реального времени в IP-сетях

Все большую часть трафика в IP-сетях составляют потоки информации, чувствительной к задержкам. Максимальная задержка не должна превышать нескольких десятых долей секунды, причем сюда входит и время обработки информации на конечной станции. Вариацию задержки также необходимо свести к минимуму. Кроме того, необходимо учитывать, что при сжатии информации, обмен которой должен происходить в реальном времени, она становится более чувствительной к ошибкам, возникающим при передаче, и их нельзя исправлять путем переспроса именно из-за необходимости передачи в реальном времени.

Телефонный разговор – это интерактивный процесс, не допускающий больших задержек. В соответствии с рекомендацией ITU-T G.114 для большинства абонентов задержка речевого сигнала на 150 мс приемлема, а на 400 мс – недопустима.

Общая задержка речевой информации делится на две основные части – задержка при кодировании и декодировании речи в шлюзах или терминальном оборудовании пользователей (об этом рассказывалось в главе 3) и задержка, вносимая самой сетью. Уменьшить общую задержку можно двумя путями: во-первых, спроектировать инфраструктуру сети таким образом, чтобы задержка в ней была минимальной, и, во-вторых, уменьшить время обработки речевой информации шлюзом.

Для уменьшения задержки в сети нужно сокращать количество транзитных маршрутизаторов и соединять их между собой высокоскоростными каналами. А для сглаживания вариации задержки можно использовать такие эффективные методы как, например, механизмы резервирования сетевых ресурсов.

IP-телефонию часто считают частью пакета услуг Интернет-провайдеров, что, вообще говоря, неверно. Известно множество примеров, когда технология передачи речевой информации внедрялась в корпоративные IP-сети и когда строились (в том числе, в России) выделенные сети IP-телефонии.

Как правило, с корпоративными сетями все обстоит сравнительно просто. Они имеют ограниченные размеры и контролируемую топологию, а характер трафика обычно бывает известен заранее. Возьмем простой пример: речь передается по существующей ЛВС, которая слишком загружена, чтобы обеспечить приемлемое качество обслуживания. Решением этой проблемы будет изоляция серверов и клиентов, работающих с трафиком данного типа, и сегментация сети. Разбить сеть на сегменты можно, либо установив коммутатор Ethernet, либо добавив порты в маршрутизатор.

Выделенные сети IP-телефонии обычно используются для междугородной и международной связи. Такие сети лучше строить по иерархическому принципу, возлагая на каждый уровень иерархии свои функции. На входе в сеть главное – обеспечить подключение речевых шлюзов, а внутри сети – высокоскоростную пересылку пакетов. В такой сети очень просто производить расширение и внедрять новые услуги. Проблема проектирования также не доставляет особых проблем: характер трафика определен, полоса пропускания также легко рассчитывается. Трафик одностипный, а значит, не требуется вводить приоритетность пакетов.

В сетях традиционных операторов обслуживается трафик разных видов, поэтому в таких сетях, чтобы обеспечить приемлемое качество, целесообразно применять дифференцированное обслуживание разнотипного трафика (Diff-Serv).

10.4 Дифференцированное обслуживание разнотипного трафика – Diff-Serv

Для обеспечения гарантированного качества обслуживания комитет IETF разработал модель дифференцированного обслуживания разнотипного трафика – Diff-Serv. В соответствии с этой моделью байт ToS (Type of Service) в заголовке IP-пакета получил другое название DS (Differentiated Services), а шесть его битов отведены под код Diff-Serv. Каждому значению этого кода соответствует свой класс

PHB (Per-Hop Behavior Forwarding Class), определяющий уровень обслуживания в каждом из сетевых узлов. Пакеты каждого класса должны обрабатываться в соответствии с определенными для этого класса требованиями к качеству обслуживания.

Модель Diff-Serv описывает архитектуру сети как совокупность пограничных участков и ядра. Пример сети согласно модели Diff-Serv приведен на рисунке 10.1.

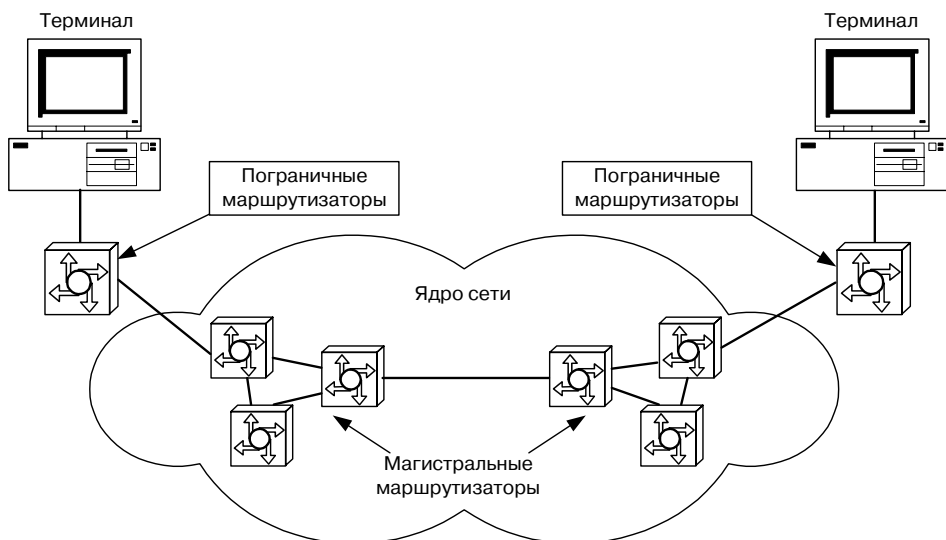


Рис. 10.1 Модель Diff-Serv

Поступающий в сеть трафик классифицируется и нормализуется пограничными маршрутизаторами. Нормализация трафика предусматривает измерение его параметров, проверку соответствия заданным правилам предоставления услуг, профилирование (при этом пакеты, не укладывающиеся в рамки установленных правил, могут быть отсеяны) и другие операции. В ядре магистральные маршрутизаторы обрабатывают трафик в соответствии с классом PHB, код которого указан в поле DS.

Достоинства модели Diff-Serv состоят в том, что она, во-первых, обеспечивает единое понимание того, как должен обрабатываться трафик определенного класса, а во-вторых, позволяет разделить весь трафик на относительно небольшое число классов и не анализировать каждый информационный поток отдельно. К настоящему времени для Diff-Serv определено два класса трафика:

- класс срочной пересылки пакетов (Expedited Forwarding PHB Group);
- класс гарантированной пересылки пакетов (Assured Forwarding PHB Group).

Механизм обеспечения QoS на уровне сетевого устройства, применяемый в Diff-Serv, включает в себя четыре операции. Сначала пакеты классифицируются на основании их заголовков. Затем они маркируются в соответствии с произведенной классификацией (в поле Diff-Serv). В зависимости от маркировки выбирается алгоритм передачи (при необходимости – с выборочным удалением пакетов), позволяющий избежать заторов в сети. Заключительная операция, чаще всего, состоит в организации очередей с учетом приоритетов [15].

Хотя эта модель и не гарантирует качество обслуживания на 100%, у нее есть серьезные преимущества. Например, нет необходимости в организации предварительного соединения и в резервировании ресурсов. А так как в модели Diff-Serv используется небольшое, фиксированное количество классов и трафик абонентов распределяется по общим очередям, не требуется высокая производительность сетевого оборудования.

10.5 Интегрированное обслуживание IntServ

Этот подход явился одной из первых попыток комитета IETF разработать действенный механизм обеспечения качества обслуживания в IP-сетях. Для трафика реального времени вводятся два класса обслуживания: контролируемой загрузки сети и гарантированного обслуживания.

Классу гарантированного обслуживания предоставляется определенная полоса пропускания, а также гарантируются задержка в определенных пределах и отсутствие потерь при переполнении очередей.

Класс контролируемой загрузки сети идентичен традиционному подходу «best effort», но уровень QoS для уже обслуживаемого потока данных остается неизменным при увеличении нагрузки в сети.

Основными компонентами модели IntServ являются система резервирования ресурсов, система контроля доступа, классификатор и диспетчер очередей. Архитектура модели изображена на рис. 10.2.

Спецификация потока (flow specification) нужна для определения необходимого уровня качества обслуживания потока.

Система контроля доступа, получив запрос сеанса связи, в зависимости от наличия требуемых ресурсов, либо допускает этот запрос к дальнейшей обработке, либо дает отказ. Классификатор определяет класс обслуживания на основе содержания поля приоритета в заголовке. Диспетчер определяет способ организации и механизм обслуживания очереди. Система резервирования ресурсов использует специальный протокол сигнализации, который служит для за-

проса приложением нужного ему уровня качества обслуживания и для координации обработки этого запроса всеми устройствами сети. Этому протоколу посвящен следующий параграф.

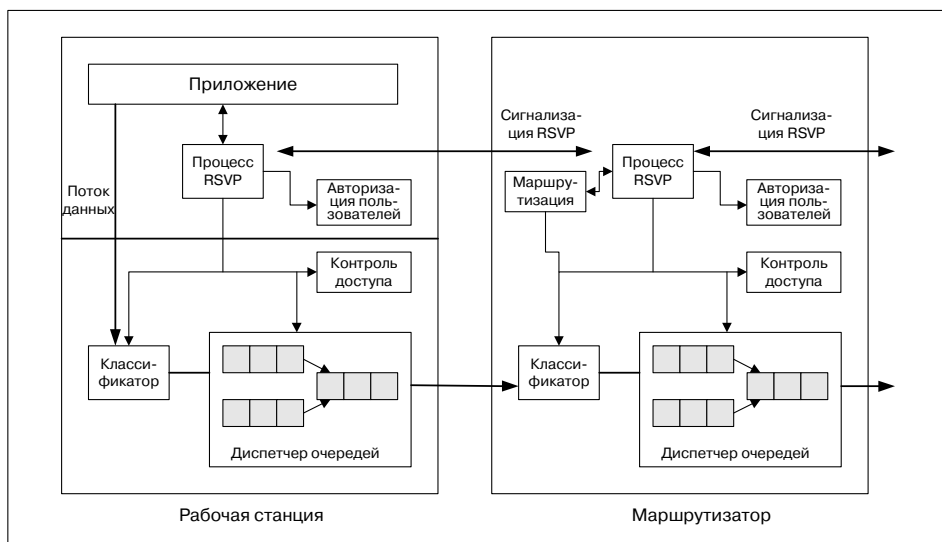


Рис. 10.2 Модель IntServ

10.6 Протокол резервирования ресурсов – RSVP

10.6.1 Общие принципы протокола

Чтобы обеспечить должное качество обслуживания трафика речевых и видеоприложений, необходим механизм, позволяющий приложениям информировать сеть о своих требованиях. На основе этой информации сеть может резервировать ресурсы для того, чтобы гарантировать выполнение требований к качеству, или отказать приложению, вынуждая его либо пересмотреть требования, либо отложить сеанс связи. В роли такого механизма выступает протокол резервирования ресурсов RSVP (Resource Reservation Protocol).

При одноадресной передаче процесс резервирования выглядит довольно просто. Многоадресная же рассылка делает задачу резервирования ресурсов гораздо более сложной. Приложения, использующие многоадресную рассылку, могут генерировать огромные объемы трафика, например, в случае организации видеоконференции с большой рассредоточенной группой участников. Однако в этом случае существуют возможности значительного снижения трафика.

Во-первых, некоторым участникам конференции в какие-то периоды времени может оказаться ненужной доставка к ним всех данных от всех источников – например, участник может быть заинтересован

в получении речевой информации и не заинтересован в получении видеоинформации.

Во-вторых, оконечное оборудование некоторых участников конференции может оказаться способным обрабатывать только часть передаваемой информации. Например, поток видеоданных может состоять из двух компонентов – базового и дополнительного, нужного для получения более высокого качества изображения. Оборудование части пользователей может не иметь достаточной вычислительной мощности для обработки компонентов, обеспечивающих высокое разрешение, или может быть подключено к сети через канал, не обладающий необходимой пропускной способностью.

Процедура резервирования ресурсов позволяет приложениям заранее определить, есть ли в сети возможность доставить многоадресный трафик всем адресатам в полном объеме, и, если нужно, принять решение о доставке отдельным получателям усеченных версий потоков.

RSVP – это протокол сигнализации, который обеспечивает резервирование ресурсов для предоставления в IP-сетях услуг эмуляции выделенных каналов. Протокол позволяет системам запрашивать, например: гарантированную пропускную способность такого канала, предсказуемую задержку, максимальный уровень потерь. Но резервирование выполняется лишь в том случае, если имеются требуемые ресурсы.

В основе протокола RSVP лежат три компонента:

- *сеанс связи*, который идентифицируется адресом получателя данных;
- *спецификация потока*, которая определяет требуемое качество обслуживания и используется узлом сети, чтобы установить соответствующий режим работы диспетчера очередей;
- *спецификация фильтра*, определяющая тип трафика, для обслуживания которого запрашивается ресурс.

10.6.2 Процедура резервирования ресурсов

Отправитель данных передает на индивидуальный или групповой адрес получателя сообщение *Path*, в котором указывает желательные характеристики качества обслуживания трафика – верхнюю и нижнюю границу полосы пропускания, величину задержки и вариации задержки. Сообщение *Path* пересылается маршрутизаторами сети по направлению к получателю данных с использованием таблиц маршрутизации в узлах сети. Каждый маршрутизатор, поддерживающий протокол RSVP, получив сообщение *Path*, фиксирует определенный элемент «структуры пути» – адрес предыдущего маршрутизатора. Таким образом, в сети образуется фиксированный маршрут. Поскольку сообщения *Path* содержат те же адреса отправителя и полу-

чателя, что и данные, пакеты будут маршрутизироваться корректно даже через сетевые области, не поддерживающие протокол RSVP.

Сообщение *Path* должно нести в себе шаблон данных отправителя (Sender Template), описывающий тип этих данных. Шаблон специфицирует фильтр, который отделяет пакеты данного отправителя от других пакетов в пределах сессии (по IP-адресу отправителя и, возможно, по номеру порта). Кроме того, сообщение *Path* должно содержать спецификацию потока данных отправителя *Tspec*, которая определяет характеристики этого потока. Спецификация *Tspec* используется, чтобы предотвратить избыточное резервирование.

Шаблон данных отправителя имеет тот же формат, что и спецификация фильтра в сообщениях *Resv* (см. ниже).

Приняв сообщение *Path*, его получатель передает к маршрутизатору, от которого пришло это сообщение (т.е. по направлению к отправителю), запрос резервирования ресурсов – сообщение *Resv*. В дополнение к информации *Tspec*, сообщение *Resv* содержит спецификацию запроса (*Rspec*), в которой указываются нужные получателю параметры качества обслуживания, и спецификацию фильтра (*filterspec*), определяющую, к каким пакетам сессии относится данная процедура (по IP-адресу отправителя и, возможно, по номеру порта).

Когда получатель данных передает запрос резервирования, он может запросить передачу ему ответного сообщения, подтверждающего резервирование.

При получении сообщения *Resv* каждый маршрутизатор резервируемого пути, поддерживающий протокол RSVP, определяет, приемлем ли этот запрос, для чего выполняются две процедуры. С помощью механизмов управления доступом проверяется, имеются ли у маршрутизатора ресурсы, необходимые для поддержки запрашиваемого качества обслуживания, а с помощью процедуры авторизации пользователей (policy control) – правомерен ли запрос резервирования ресурсов. Если запрос не может быть удовлетворен, маршрутизатор отвечает на него сообщением об ошибке.

Если же запрос приемлем, данные о требуемом качестве обслуживания поступают для обработки в соответствующие функциональные блоки (способ обработки параметров QoS маршрутизатором в протоколе RSVP не определен), и маршрутизатор передает сообщение *Resv* следующему (находящемуся ближе к отправителю данных) маршрутизатору. Это сообщение несет в себе спецификацию *flowspec*, содержащую два набора параметров:

- «*Rspec*», который определяет желательное QoS,
- «*Tspec*», который описывает информационный поток.

Вместе *flowspec* и *filterspec* представляют собой дескриптор потока, используемый маршрутизатором для идентификации каждой процедуры резервирования ресурсов.

Когда маршрутизатор, ближайший к инициатору процедуры резервирования, получает сообщение *Resv* и выясняет, что запрос приемлем, он передает подтверждающее сообщение получателю данных. При групповом резервировании учитывается тот факт, что в точках слияния дерева доставки несколько потоков, для которых производится резервирование, сливаются в один, так что подтверждающее сообщение передает маршрутизатор, находящийся в точке их слияния.

После окончания вышеописанной процедуры ее инициатор начинает передавать данные и на их пути к получателю будет обеспечено заданное QoS.

Для простой выделенной линии требуемое QoS будет получено с помощью диспетчера пакетов в драйвере уровня звена данных. Если технология уровня звена данных предусматривает свои средства управления QoS, протокол RSVP должен согласовать получение нужного QoS с этим уровнем.

Отменить резервирование можно двумя путями – прямо или косвенно. В первом случае отмена производится по инициативе отправителя или получателя с помощью специальных сообщений RSVP. Во втором случае резервирование отменяется по таймеру, ограничивающему срок существования резервирования.

Протокол RSVP поддерживает улучшенную версию однопроходного варианта алгоритма, известного под названием OPWA (One Path With Advertising) (OPWA95). С помощью OPWA управляющие пакеты RSVP, передаваемые вдоль маршрута, могут быть использованы для переноса данных, которые позволяют предсказывать QoS маршрута в целом.

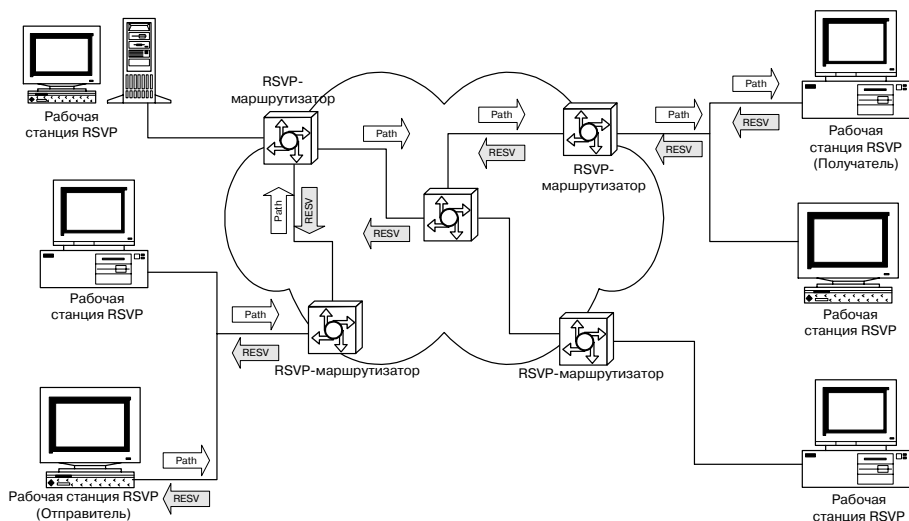


Рис. 10.3 Резервирование ресурсов при помощи протокола RSVP

С точки зрения узла сети работа протокола RSVP выглядит так:

1. Получатель вступает в группу многоадресной рассылки, отправляя соответствующее сообщение протокола IGMP ближайшему маршрутизатору;
2. Отправитель передает сообщение по адресу группы;
3. Получатель принимает сообщение Path, идентифицирующее отправителя;
4. Теперь получатель имеет информацию об обратном пути и может отправлять сообщение Resv с дескрипторами потока;
5. Сообщения Resv передаются по сети отправителю;
6. Отправитель начинает передачу данных;
7. Получатель начинает передачу данных.

Несмотря на то, что протокол RSVP является важным инструментом в арсенале средств, обеспечивающих гарантированное качество обслуживания, этот протокол не может решить все проблемы, связанные с QoS. Основные недостатки протокола RSVP – большой объем служебной информации и большие затраты времени на организацию резервирования.

Протокол RSVP не размещается в крупномасштабных средах. В лучшем случае, магистральный маршрутизатор имеет возможность резервировать ресурсы для нескольких тысяч потоков и управлять очередями для каждого из них.

Протокол RSVP работает с пакетами IP и не затрагивает схем сжатия, циклического контроля (CRCs) или работы с кадрами уровня звена данных (Frame Relay, PPP, HDLC).

Например, при использовании для IP-телефонии алгоритма кодирования речи G.729, обеспечивающего, с учетом сжатия заголовков RTP-пакетов, передачу речи со скоростью около 11 Кбит/с, в оборудовании Cisco по протоколу RSVP резервируется ресурс с пропускной способностью 24 Кбит/с. Другими словами, в канале с пропускной способностью 56 Кбит/с разрешено резервировать ресурс только для двух потоков со скоростью 24 Кбит/с каждый, даже если полоса пропускания располагает ресурсом для трех потоков со скоростью 11 Кбит/с каждый. Чтобы обойти это ограничение, можно применить следующий прием. Средствами эксплуатационного управления функциональному блоку RSVP маршрутизатора сообщается, например, что канал с фактической полосой пропускания 56 Кбит/с имеет, якобы, пропускную способность 100 Кбит/с и что допускается использовать для резервирования 75% его полосы пропускания. Такой «обман» разрешит протоколу RSVP резервировать полосу пропускания, которая необходима для трех речевых потоков, закодированных по алгоритму G.729. Очевидно, что при этом есть опасность

перегрузки канала с реальной полосой 56 Кбит/с, если сжатие заголовков RTP-пакетов не применяется.

10.7 Технология MPLS

Технология многопротокольной коммутации по меткам (MultiProtocol Label Switching – MPLS), разработанная комитетом IETF, явилась результатом слияния нескольких разных механизмов, таких как IP Switching (Ipsilon), Tag Switching (Cisco Systems), Aris (IBM) и Cell Switch Router (Toshiba). В архитектуре MPLS собраны наиболее удачные элементы всех упомянутых механизмов, и благодаря усилиям IETF и компаний, заинтересованных в скорейшем продвижении этой технологии на рынке, она превратилась в стандарт Internet.

В обычных IP-сетях любой маршрутизатор, находящийся на пути следования пакетов, анализирует заголовок каждого пакета, чтобы определить, к какому потоку этот пакет относится, и выбрать направление для его пересылки к следующему маршрутизатору. При использовании технологии MPLS соответствие между пакетом и потоком устанавливается один раз, на входе в сеть MPLS. Более точно, соответствие устанавливается между пакетом и так называемым «классом эквивалентности пересылки» FEC (Forwarding Equivalence Class); к одному FEC относятся пакеты всех потоков, пути следования которых через сеть (или часть сети) совпадают. С точки зрения выбора ближайшего маршрутизатора, к которому их надо пересылать, все пакеты одного FEC неразличимы. Пакеты снабжаются метками – идентификаторами небольшой и фиксированной длины, которые определяют принадлежность каждого пакета тому или иному классу FEC. Метка имеет локальное значение – она действительна на участке между двумя соседними маршрутизаторами, являясь исходящей меткой определенного FEC для одного из них и входящей – для второго. Второй маршрутизатор, пересылая пакет этого FEC к следующему маршрутизатору, снабжает его другой меткой, которая идентифицирует тот же FEC на следующем участке маршрута, и т.д. Таким образом, каждый FEC имеет свою систему меток.

Использование меток значительно упрощает процедуру пересылки пакетов, так как маршрутизаторы обрабатывают не весь заголовок IP-пакета, а только метку, что занимает значительно меньше времени.

На рис. 10.4 показана, в качестве примера, простейшая MPLS-сеть, содержащая маршрутизаторы двух типов:

- пограничные маршрутизаторы MPLS (Label Edge Routers – LER),
- транзитные маршрутизаторы MPLS (Label Switching Routers – LSR).

По отношению к любому потоку пакетов, проходящему через MPLS-сеть, один LER является входным, а другой LER – выходным.

Входной LER анализирует заголовок пришедшего извне пакета, устанавливает, какому FEC он принадлежит, снабжает этот пакет меткой, которая присвоена данному FEC, и пересылает пакет к соответствующему LSR. Далее, пройдя, в общем случае, через несколько LSR, пакет попадает к выходному LER, который удаляет из пакета метку, анализирует заголовок пакета и направляет его к адресату, находящемуся вне MPLS-сети.

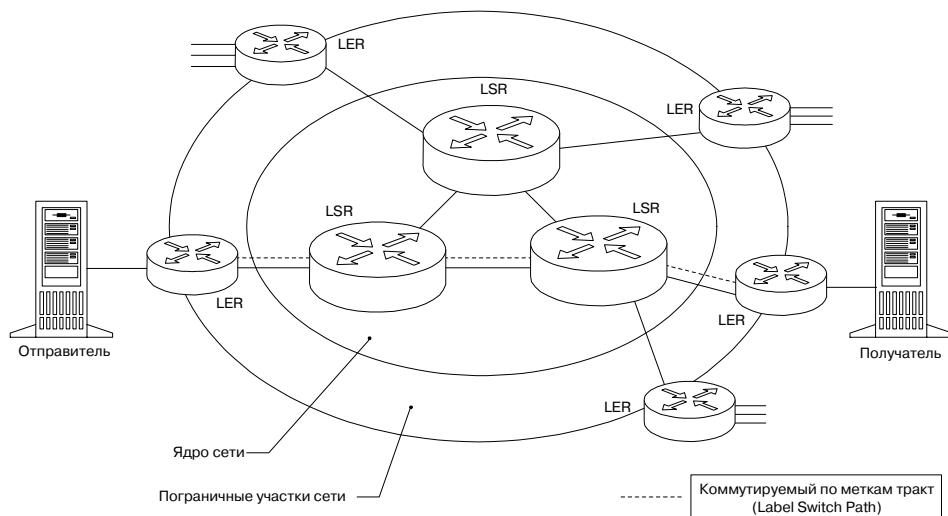


Рис. 10.4 Сеть, построенная на базе технологии MPLS

Последовательность $(LER_{вх}, LSR_1, \dots, LSR_n, LER_{вых})$ маршрутизаторов, через которые проходят пакеты, принадлежащие одному FEC, образует виртуальный коммутируемый по меткам тракт LSP (Label Switched Path). Так как один и тот же LER для одних потоков является входным, а для других – выходным, в сети, содержащей N LER, в простейшем случае может существовать $N(N-1)$ FEC и, соответственно, $N(N-1)$ LSP.

Заметим, однако, что потоки пакетов из разных FEC, приходящие к одному выходному LER от разных входных LER, могут в каких-то LSR сливаться в более мощные потоки, каждый из которых образует новый FEC со своей системой меток. Возможно и обратное, то есть группа потоков может идти до некоторого LSR по общему маршруту и, следовательно, принадлежать одному и тому же FEC, а затем разветвиться, и тогда каждая ветвь будет иметь свой FEC (со своей системой меток). Кроме того, существует возможность образования внутри некоторого LSP одного или нескольких вложенных в него LSP (так называемых LSP-туннелей).

То обстоятельство, что система меток, присваиваемых пакету, может изменяться, приводит к образованию так называемого «стека ме-

ток». При переходе потока пакетов в другой FEC, метка нового FEC помещается поверх метки прежнего FEC и используется для коммутации, а прежняя метка сохраняется под ней, но не используется до тех пор, пока не восстановится прежний FEC. Ясно, что если FEC пакета меняется несколько раз, в стеке накапливается несколько меток.

Все это, с одной стороны, демонстрирует, насколько широки возможности MPLS в части распределения ресурсов сети при ее проектировании и в части их оперативного перераспределения при эксплуатации, но, с другой стороны, предъявляет непростые требования к средствам, с помощью которых устанавливается соответствие «FEC-метка» в каждом LER и LSR сети.

Итак, метка, помещаемая в некоторый пакет, представляет FEC, к которому этот пакет относится. Как правило, отнесение пакета к определенному классу производится на основе сетевого адреса получателя. Метка может быть помещена в пакет разными способами – вписываться в специальный заголовок, «вставляемый» либо между заголовками уровня звена данных и сетевого уровня, либо в свободное и доступное поле заголовка какого-то одного из этих двух уровней, если таковое имеется. В любом случае этот специальный заголовок содержит поле, куда записывается значение метки, и несколько служебных полей, среди которых имеется и то, которое представляет особый интерес с точки зрения данной главы – поле QoS (три бита, т.е. до восьми классов качества обслуживания).

Метки для каждого FEC всегда назначаются «снизу», то есть либо выходным LER, либо тем LSR, который является для этого FEC «нижним» (расположенным ближе к адресату), и распределяются им по тем маршрутизаторам, которые расположены «выше» (ближе к отправителю).

Распределение меток может быть независимым или упорядоченным. В первом случае LSR может уведомить вышестоящий LSR о привязке метки к FEC еще до того, как получит информацию о привязке «метка-FEC» от нижестоящего маршрутизатора. Во втором случае высылать подобное уведомление разрешается только после получения таких сведений «снизу». Метки могут выдаваться нижним маршрутизатором как по собственной инициативе, так и по запросу верхнего. Наконец, возможен «либеральный» или «консервативный» режим распределения меток. В либеральном режиме нижний LSR раздает метки вышестоящим LSR, как имеющим с ним прямую связь, так и доступным лишь через промежуточные LSR. В консервативном режиме вышестоящий LSR обязан принять метку, если ее выдает смежный LSR, но может отказаться от метки, пришедшей к нему транзитом.

Как уже отмечалось, метка должна быть уникальной лишь для каждой пары смежных LSR. Поэтому одна и та же метка в любом LSR может быть связана с несколькими FEC, если разным FEC принадлежат пакеты, идущие от разных маршрутизаторов, и имеется возможность определить, от которого из них пришел пакет с данной меткой. В связи с этим обстоятельством вероятность того, что пространство меток будет исчерпано, очень мала.

Для распределения меток может использоваться либо специальный протокол LDP (Label Distribution Protocol), либо модифицированная версия одного из существующих протоколов сигнализации (например, протокола RSVP).

Каждый LSR содержит таблицу, которая ставит в соответствие паре величин «входной интерфейс, входящая метка» пару величин «выходной интерфейс, исходящая метка». Получив пакет, LSR определяет для него выходной интерфейс (по входящей метке и по номеру интерфейса, куда пакет поступил). Входящая метка заменяется исходящей (записанной в соответствующем поле таблицы), и пакет пересылается к следующему LSR. Вся операция требует лишь одно-разовой идентификации значений в полях одной строки таблицы и занимает гораздо меньше времени, чем сравнение IP-адреса отправителя с адресным префиксом в таблице маршрутов при традиционной маршрутизации.

MPLS предусматривает два способа пересылки пакетов. При одном способе каждый маршрутизатор выбирает следующий участок маршрута самостоятельно, а при другом заранее задается цепочка маршрутизаторов, через которые должен пройти пакет. Второй способ основан на том, что маршрутизаторы на пути следования пакета действуют в соответствии с инструкциями, полученными от одного из LSR данного LSP (обычно – от нижнего, что позволяет совместить процедуру «раздачи» этих инструкций с процедурой распределения меток).

Поскольку принадлежность пакетов тому или иному FEC определяется не только IP-адресом, но и другими параметрами, нетрудно организовать разные LSP для потоков пакетов, предъявляющих разные требования к QoS. Каждый FEC обрабатывается отдельно от остальных – не только в том смысле, что для него образуется свой LSP, но и в смысле доступа к общим ресурсам (полосе пропускания канала, буферному пространству). Поэтому технология MPLS позволяет очень эффективно поддерживать требуемое QoS, соблюдая предоставленные пользователю гарантии.

Конечно, подобный результат удастся получить и в обычных IP-сетях, но решение на базе MPLS проще и гораздо лучше масштабируется.

10.8 Обслуживание очередей

Алгоритмы обслуживания очередей позволяют предоставлять разный уровень QoS трафику разных классов. Обычно используется несколько очередей, каждая из которых занимается пакетами с определенным приоритетом. Требуется, чтобы высокоприоритетный трафик обрабатывался с минимальной задержкой, но при этом не занимал всю полосу пропускания, и чтобы трафик каждого из остальных типов обрабатывался в соответствии с его приоритетом.

Обслуживание очередей включает в себя алгоритмы:

- организации очереди;
- обработки очередей.

10.8.1 Алгоритмы организации очереди

Существует два основных алгоритма организации очереди: Tail Drop и Random Early Detection.

10.8.1.1 Алгоритм Tail Drop

Принцип алгоритма прост – задается максимальный размер очереди (в пакетах или в байтах), вновь прибывающий пакет помещается в конец очереди, а если очередь уже полна – отбрасывается.

Однако при использовании протокола TCP, когда пакеты начинают теряться, модули TCP в рабочих станциях решают, что в сети перегрузка и замедляют передачу пакетов. При заполненной очереди возможны случаи, когда несколько сообщений отбрасываются друг за другом – в результате целый ряд приложений решит замедлить передачу. Затем приложения начнут зондировать сеть, чтобы определить, насколько она загружена, и буквально через несколько секунд возобновят передачу в прежнем темпе, что опять приведет к потерям сообщений.

В некоторых ситуациях данный алгоритм может вызвать так называемый эффект «локаута» (lock-out). Это происходит в тех случаях, когда очередь монополизирована либо одним потоком пакетов, либо несколькими потоками, случайно или по необходимости синхронизированных (например, потоков, несущих изображение и его звуковое сопровождение), что препятствует попаданию в очередь пакетов остальных потоков.

Алгоритм Tail Drop приводит к тому, что очереди оказываются заполненными (или почти заполненными) в течение длительного периода времени. Так происходит, поскольку алгоритм сигнализирует только о том, что очередь полна. Большой размер очередей сильно увеличивает время доставки пакета от одной рабочей станции к другой. Поэтому желательно, чтобы средний размер очередей в мар-

шрутизаторах был невелик. С другой стороны, известно, что трафик в сети, как правило, неравномерен, и поэтому маршрутизатор должен иметь буфер, размер которого достаточен для того, чтобы «амортизировать» неравномерность трафика.

Имеются альтернативные алгоритмы отбрасывания пакетов: Random Drop (отбрасывание случайно выбранного) и Drop from Front (отбрасывание первого). Принцип работы алгоритма Random Drop понятен из его названия. При заполнении очереди отбрасывается не пакет, пришедший последним, а пакет, выбираемый из очереди случайно. Такой алгоритм предъявляет более высокие требования к вычислительным ресурсам маршрутизатора, поскольку он производит с очередью более сложные операции. Алгоритм Drop from Front отбрасывает пакет, стоящий в очереди первым. Помимо значительно-го снижения вероятности «локаута», этот алгоритм выгодно отличается от алгоритма Tail Drop тем, что при использовании протокола TCP рабочая станция раньше узнает о перегрузке в сети и, соответственно, раньше снижает скорость передачи пакетов.

10.8.1.2 Алгоритм Random Early Detection (RED)

Суть алгоритма в том, что когда размер очереди превышает некоторое пороговое значение, прибывающий пакет отбрасывается с вероятностью, зависящей от того, насколько превышен установленный порог. Обычно предусматривается два пороговых значения. Когда длина очереди переходит за первый порог, вероятность отбрасывания пакета линейно возрастает от 0 (у первого порога) до 1 (у второго порога). Положение второго порога выбирается с таким расчетом, чтобы в оставшемся после него «хвосте» очереди мог поместиться пакет, длина которого несколько превышает среднюю. По мере приближения длины очереди ко второму порогу, растет вероятность того, что прибывший пакет будет отброшен в связи с тем, что он не умещается в очереди (несмотря на наличие «хвоста»), при этом пакеты большего размера имеют больше шансов быть отброшенными, чем пакеты меньшего размера.

При малых размерах очередей метод *RED* более эффективен, чем другие методы. Он также более устойчив к трафику, имеющему «взрывной» характер.

10.8.2 Алгоритмы обработки очередей

Алгоритмы обработки очередей составляют одну из основ механизмов обеспечения гарантированного качества обслуживания в сетевых элементах.

Разные алгоритмы нацелены на решение разных задач и по-разному воздействует на качество обслуживания сетью трафика разных типов. Тем не менее, возможно и комбинированное применение нескольких алгоритмов.

10.8.2.1 Стратегия FIFO

Алгоритм обслуживания очередей FirstIn, FirstOut (FIFO), также называемый First Come First Served является самым простым. Пакеты обслуживаются в порядке поступления без какой-либо специальной обработки.



Рис. 10.5 Очередь FIFO

Такая схема приемлема, если исходящий канал имеет достаточно большую свободную полосу пропускания. Алгоритм FIFO относится к так называемым неравноправным схемам обслуживания очередей, так как при его использовании одни потоки могут доминировать над другими и захватывать несправедливо большую часть полосы пропускания. В связи с этим применяются равноправные схемы обслуживания, предусматривающие выделение каждому потоку отдельного буфера и равномерное разделение полосы пропускания между разными очередями.

10.8.2.2 Очередь с приоритетами

Очередь с приоритетами (Priority Queuing) – это алгоритм, при котором несколько очередей FIFO (могут использоваться алгоритмы Tail Drop, RED и т.д.) образуют одну систему очередей. В случае простейшего приоритетного обслуживания трафик определенных классов имеет безусловное преимущество перед трафиком других классов. Например, если все IPX-пакеты имеют более высокий приоритет, чем IP-пакеты, то какова бы ни была ценность IP данных, IPX данные будут иметь первоочередной приоритет при разделе доступной полосы. Такой алгоритм гарантирует своевременную доставку лишь наиболее привилегированного трафика.

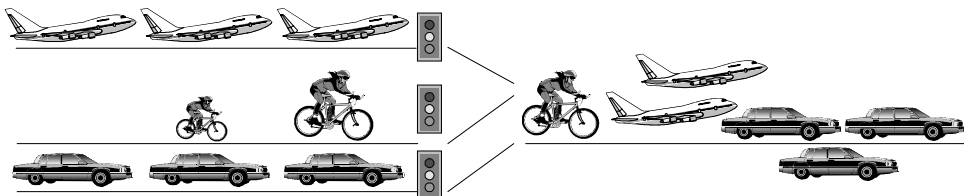


Рис. 10.6 Очереди с приоритетами

Назначение разным потокам нескольких разных приоритетов производится по ряду признаков, таких как источник и адресат пакета, транспортный протокол, номер порта. Пакет каждого потока помещается в очередь, имеющую соответствующий приоритет.

Хотя трафик более высокого приоритета получает лучшее обслуживание, чем он мог бы получить при использовании FIFO, некоторые фундаментальные проблемы остаются нерешенными. Например, если используются очереди Tail Drop, то остаются проблемы больших задержек, «локаута» и т.д. Некоторые прикладные программы пытаются использовать весь доступный ресурс. Если им предоставлена наиболее приоритетная очередь, то очереди с низким приоритетом будут блокированы в течение длительного времени, или же низкоприоритетный трафик встретит настолько большую задержку в результате следования по окружному пути, что станет бесполезным. Это может привести к прекращению менее приоритетных сеансов связи или, по крайней мере, сделает их практически непригодными.

10.8.2.3 Class-Based Queuing (CBQ)

При использовании этого механизма трафику определенных классов гарантируется требуемая скорость передачи, а оставшийся ресурс распределяется между остальными классами. Например, администратор может зарезервировать 50% полосы пропускания для SNA-трафика, а другие 50% разделить между остальными протоколами, такими как IP и IPX.

Обработка очередей по алгоритму Class-Based Queuing, CBQ предполагает, что трафик делится на классы. Определение класса трафика в значительной степени произвольно. Класс может представлять весь трафик, проходящий через данный интерфейс, трафик определенных приложений, трафик, направленный к заданному подмножеству получателей, трафик с качеством услуг, гарантированным протоколом RSVP. Каждый класс имеет собственную очередь, и ему гарантируется, по крайней мере, некоторая доля пропускной способности канала. Драйвер интерфейса обходит все очереди по кругу и передает некоторое количество пакетов из каждой очереди. Если какой-либо класс не исчерпывает предоставленный ему лимит пропускной способности, то доля полосы пропускания, выделяемая каждому из остальных классов, пропорционально увеличивается.

Алгоритм CBQ использует иерархическую организацию классов. Например, в корпоративной сети иерархия может выглядеть так, как показано на рис. 10.7. При разделении недоиспользованного кем-то ресурса классы, принадлежащие той же ветви дерева, имеют первоочередной приоритет.

Как и в предыдущем случае, используются очереди FIFO, следовательно, для потоков, разделяющих одну очередь, остаются проблемы, присущие FIFO, но гарантируется некоторая справедливость распределения ресурсов в сети между разными очередями. Кроме того, в отличие от приоритетного обслуживания, CBQ не допускает блокировки очереди и дает возможность учитывать использование сети разными классами.

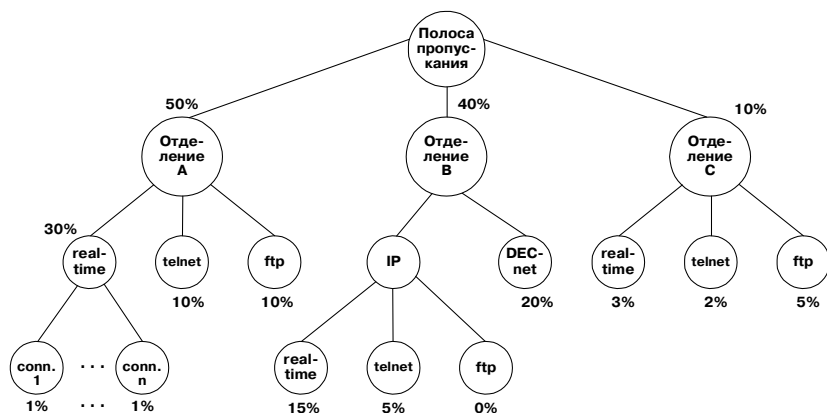


Рис. 10.7 Иерархия классов при использовании алгоритма CDQ

10.8.2.4 Взвешенные очереди

Если необходимо обеспечить для всех потоков постоянное время задержки, и не требуется резервирование полосы пропускания, то можно воспользоваться алгоритмом **Weighted Fair Queuing**.

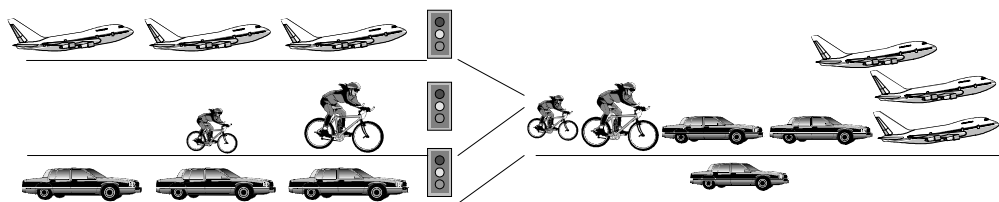


Рис. 10.8 Взвешенные очереди

Взвешенная справедливая очередь (Weighted Fair Queuing, WFQ) является частным случаем CBQ, когда каждому классу соответствуют свой поток (TCP-сеанс и т.д.). Как и в случае CBQ, каждому классу WFQ отводится одна очередь FIFO и гарантируется некоторая часть пропускной способности канала, в соответствии с весовым коэффициентом потока. Если некоторые потоки не используют предоставленную им полосу пропускания полностью, то другие потоки соответственно увеличивают свою долю. Так как в данном случае каждый класс – это отдельный поток, то гарантия пропускной способности эквивалентна гарантии максимальной задержки. Зная параметры сообщения, можно по известным формулам вычислить его максимальную задержку при передаче по сети. Выделение дополнительной пропускной способности позволяет уменьшить максимальную задержку.

Алгоритм WFQ гарантирует, что очереди не будут лишены своей доли полосы пропускания, и что трафик получит предсказуемое QoS. Трафик, не использующий целиком свою долю полосы, будет обслуживаться в первую очередь, а оставшаяся полоса будет разделена между остальными потоками.

Одна из версий этого алгоритма, называемая Generic Cell Rate Algorithm (GCRA), применяется в сетях ATM для контроля некоторых параметров.

10.8.3.2 Алгоритм «Token Bucket»

Алгоритм выполняет «калибровку» трафика, т.е. уменьшает до заданного предела пульсацию скорости потока данных и гарантирует, что не будет превышена заданная средняя скорость этого потока.

Имеется некое «ведро», в которое через равные промежутки времени поодиночке падают одинаковые жетоны; каждый жетон равноценен определенному числу байтов. Имеется буферный накопитель, в котором образуется очередь пакетов, требующих дальнейшей обработки (или передачи). Система работает так, что если количество жетонов в ведре равноценно числу байтов, не меньшему чем содержится в пакете, который стоит в очереди первым, этот пакет выводится из очереди для дальнейшей обработки, и одновременно соответствующее количество жетонов изымается из ведра. Если же жетонов в ведре недостаточно, пакет ожидает, пока их наберется столько, сколько нужно. Таким образом, генератор, определяющий частоту, с которой жетоны падают в ведро, контролирует скорость продвижения пакетов, а буферный накопитель сглаживает ее пульсацию.

Если трафик снизился настолько, что в буферном накопителе не осталось ни одного пакета, подача жетонов в ведро прекращается, когда в нем наберется такое их количество, которое примерно равноценно числу байтов в пакете средней длины. Как только в накопитель снова начнут поступать пакеты, подача жетонов в ведро должна быть возобновлена.

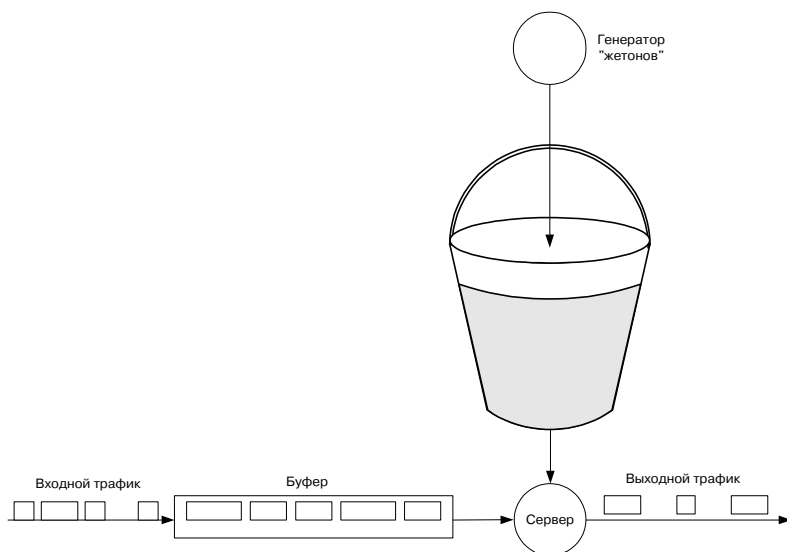


Рис. 10.10 Алгоритм "Token Bucket"

Глава 11

Принципы реализации

11.1 Оборудование IP-телефонии

Как оказалось, мудрым советом знаменитого бизнесмена Аристотеля Онассиса «Не гонись за деньгами – иди им навстречу» воспользовался целый ряд компаний, преуспевших в разработке программных средств и оборудования IP-телефонии, среди которых – VocalTec, Dialogic, Cisco, Ascend, 3Com, Nortel, Lucent, IBM, Motorola, RAD, Rockwell, Digitcom и др.

Некоторые из продуктов названных компаний уже упоминались на страницах этой книги. Так, в главе 5 обсуждались принципы построения оборудования, реализующего самый распространенный на сегодняшний день протокол H.323; в главе 7 были отчасти затронуты технические решения, поддерживающие протокол SIP, а в главе 8 рассмотрены аспекты реализации протокола MGCP. Здесь же рассматриваются некоторые интересные решения в построении оборудования IP-телефонии, не вошедшие в предыдущие главы.

Начнем анализ с продукции компании Nortel Networks, выдвинувшей новый подход, который назван философией вечной молодости (Evergreen). Применительно к тематике данной книги этот принцип нашел свое воплощение в концепции Inca (Internet Communications Architecture), объявленной 8 июня 1999 г. и предусматривающей постепенное оснащение уже существующих IP-сетей функциями IP-телефонии.

Примером практической реализации концепции Nortel Networks является платформа MMCS (MultiMedia Carrier Switch), прошедшая сертификацию для ВСС России и известная по публикациям в жур-

налах. Другими примерами являются семейство Magellan – пакетные коммутаторы серии DPN (протоколы X.25, FR) и модельный ряд Passport – устройства доступа с компрессией речи по протоколу FR – Passport 4400, мультипротокольные маршрутизаторы серии Passport 7000/6000, пограничные устройства Passport Voice Gateway, сопрягающие телефонные сети и сети ATM, а также высокоскоростные ATM-коммутаторы Passport 15000. Все это оборудование позволяет полностью интегрировать речь, факсимильные сообщения, видеoinформацию, данные по протоколам IP, FR, SNA, X.25, HDLC, и обеспечивать мультимедийные услуги, оптимизируя использование имеющихся ресурсов (например, при передаче речи применяется технология передачи пакетов с переменной скоростью).

Еще одним примером оборудования IP-телефонии может служить универсальный маршрутизатор IP45/951 с функциями передачи речи и мультимедийной информации по IP-сетям, входящий в гамму продуктов корпорации NEC, Япония. Маршрутизатор IP45/951 реализует функции шлюза и привратника. Маршрутизатор поддерживает большое количество алгоритмов кодирования речи, в том числе, ITU-T G.729, G.729a, G.729b, G.729ab, G.723.1, G.729.1a, G.711, G.711VAD, G.728 и G.728VAD. Это позволяет маршрутизатору IP45/951 соединяться практически со всеми шлюзами, поддерживающими протокол H.323, в то время как возможности многих шлюзов ограничены небольшим количеством поддерживаемых алгоритмов кодирования. Маршрутизатор IP45/951 обеспечивает хорошее качество передачи речи благодаря следующим особенностям:

- применение современных алгоритмов кодирования;
- подавление эха (64 мс);
- сглаживание джиттера;
- подавление пауз в разговоре;
- генерация комфортного шума;
- поддержка протокола RSVP;
- сжатие заголовков IP/UDP/RTP;
- поддержка приоритетов для различных видов трафика.

Даже при кратком анализе характеристик оборудования IP-телефонии ведущих фирм-производителей обращает на себя внимание модульность и масштабируемость продуктов и их ценовой диапазон, не превышающий нескольких десятков тысяч долларов.

Это же справедливо и в отношении программного обеспечения IP-телефонии, оно доступно и недорого. Популярные продукты Microsoft NetMeeting, IDT Net2Phone и DotDialer реализуют разные схемы телефонной связи через IP-сети: NetMeeting используется для связи по схеме «компьютер-компьютер», а Net2Phone и DotDialer

реализуют схему «компьютер-телефон». Оба эти сценария IP-телефонии уже обсуждались в главе 2. Там же отмечались сложность и актуальность программно-аппаратных средств, реализующих сценарий «телефон-телефон».

За последнее время появились следующие виды оборудования IP-телефонии для всех этих сценариев:

1. Автономные шлюзы IP-телефонии, подключаемые к АТС через цифровые и аналоговые интерфейсы и осуществляющие предварительную обработку речевых сигналов, компрессию, упаковку в IP-пакеты и передачу их по сети.

2. Магистральные речевые платы с интерфейсом 10/100BaseT (ЛВС Ethernet) для подключения учрежденческих АТС существующих моделей к корпоративной IP-сети. После установки в АТС такой платы речевой трафик в виде IP-пакетов может быть направлен по локальной или глобальной пакетной сети подобно тому, как он сейчас передается от АТС по телефонной сети.

3. Телефонные аппараты, упаковывающие речевую информацию в IP-пакеты (IP-телефоны) и подключаемые не к телефонной сети, а непосредственно к ЛВС Ethernet. Как правило, такие аппараты требуют от сетевого администратора минимальных настроек, используя протокол динамической конфигурации - Dynamic Host Configuration Protocol (DHCP).

4. Специализированные коммутаторы речевых пакетов, предназначенные для выполнения функций традиционной АТС на базе протокола IP. В литературе такие устройства часто называют IP-АТС, но это название представляется нам не совсем корректным, поскольку в данном случае осуществляется не автоматическая коммутация каналов, а коммутация пакетов.

Аппаратура IP-телефонии выпускается в совмещенной или автономной конструкции. Совмещенный сервер выполняет функции шлюза, привратника и администратора (manager), т.е. маршрутизацию, сбор биллинговой информации (IP-адрес, время начала и конца разговора и т.п.), подавление эхосигналов, детектирование пауз в разговоре, заполнение пауз на приеме комфортным шумом (comfort noise), буферизацию принятых пакетов для уменьшения джиттера, интерполяцию потерянных речевых пакетов, а также контроль состояния разговорного канала (среднее время задержки, джиттер, процент потерь пакетов). В автономной конструкции эти функции выполняются отдельными устройствами.

В ранних моделях цифровая обработка сигнала производилась программными средствами. Позднее программную обработку сменила аппаратная, основную роль стали выполнять уже упоминавшиеся в главе 3 платы DSP (Digital Signal Processing), что разгрузило ос-

новой процессор и оперативную память, увеличило число портов оборудования и уменьшило время задержки речевой информации. Наиболее известны платы DSP фирм Texas Instrument, Dialogic (DM3 IP Link) и Natural MicroSystems (Quad E1).

Рассмотренная в начале книги конвергенция сетей электросвязи, в рамках которых передается информация всех видов (речь, видео и данные), обусловила появление упомянутых выше продуктов в качестве отклика ведущих телекоммуникационных компаний на интерес потенциальных потребителей этой части рынка. Пока большинство разработок используется в корпоративных сетях или в небольших офисах, а не в глобальных сетях операторов связи. Создание более мощного и емкого оборудования IP-телефонии в самое ближайшее время потребует значительных усилий, среди полезных результатов которых, возможно, найдется место и данной книге. Обсуждаемые в ней требования к аппаратуре IP-телефонии можно сформулировать, в общих чертах, так:

- полная поддержка рекомендаций H.323;
- поддержка всех основных алгоритмов кодирования речи;
- поддержка основных систем телефонной сигнализации (ОКС7, DSS1, R1.5 и т.д.);
- удобство и функциональность средств управления и контроля.

Важная категория удовлетворяющего этим требованиям оборудования IP-телефонии предназначена для построения *сетевой инфраструктуры*. Сегодня это главное препятствие для развития IP-телефонии, с которым столкнулись региональные операторы. Проблема состоит в построении структуры IP-сети, которая могла бы дать им шанс скоординировать свои действия и собрать свои ресурсы для того, чтобы составить конкуренцию операторам и поставщикам традиционного оборудования междугородной и международной связи. Для решения этой проблемы и создания магистральных транзитных узлов сегодня имеются сверхскоростные маршрутизаторы IP-пакетов производства Avici Systems Inc. (Челмсфорд, Массачусетс), Berkeley Networks Inc. (Сан Хоце, Калифорния), Gigapacket Networks Inc. (Литтлтон, Массачусетс), Juniper Networks (Санта Клара, Калифорния), Neonet LLC (Уэстборо, Массачусетс) и Torrent Networking Technologies Inc. (Лендовер, Мэриленд). Эти маршрутизаторы могут объединяться в IP-сети коммутаторами ATM, SDH/Sonet производства Ascend, Cisco и др.

Другим, не менее важным аспектом внедрения IP-телефонии являются *шлюзы*, обеспечивающие взаимодействие сетей с коммутацией каналов и с коммутацией пакетов.

В настоящее время несколько десятков компаний выпускают подобные изделия, среди них Cisco Systems, VocalTec, Lucent Technologies и др. Более того, на базе этих шлюзов почти каждая крупная

телекоммуникационная компания имеет или заявленное, или уже поставляемое изделие IP-телефонии. Предлагаются АТС, реализованные на основе технологии маршрутизации IP-пакетов. Компания Cisco выпустила интегрированный сервер доступа AS5300 с коммутатором Catalyst 5500. Компания Ascend Communications Inc. объединила модем для коммутируемых каналов ТНТ с гигабитным маршрутизатором GRF. Компания 3Com добавила передачу речи по IP и факс в свой концентратор Total Control Hub.

Согласно некоторым оценкам, объем рынка сетевых изделий IP-телефонии в 2001 году составит 1.8 миллиарда долларов, а рынка устройств доступа к сетям IP-телефонии – 7.5 миллиарда долларов. Как известно из предыдущего опыта, эксплуатация оборудования создает рынок в десять раз выше, чем общая стоимость установленного оборудования. Поэтому общий рынок оборудования передачи речи по IP-сетям уже сегодня можно оценить в 90 миллиардов долларов.

11.2 Особенности оборудования IP-телефонии для России

При внедрении технологии передачи речевой информации по сетям с маршрутизацией IP-пакетов во Взаимоувязанной сети связи Российской Федерации (ВСС РФ), помимо рассмотренных выше, возникают следующие специфические трудности:

- при подключении оборудования IP-телефонии к АТС телефонной сети общего пользования по двухпроводным аналоговым абонентским линиям препятствием часто становится большое затухание сигналов в этих линиях;
- при подключении оборудования IP-телефонии к коммутационному оборудованию ТфОП по межстанционным соединительным линиям затруднения связаны с тем, что декадно-шаговые и координатные АТС имеют специфические системы сигнализации [6], основная из которых определяется неформальным, но весьма точным термином «R полтора»;
- присутствующие в ТфОП декадно-шаговые АТС создают большие помехи и поддерживают только импульсный набор номера.

Специфические требования к оборудованию IP-телефонии, подключаемому к ВСС России, изложены в руководящем документе отрасли РД 45.046-99 «Аппаратура связи, реализующая функции передачи речевой информации по сетям передачи данных с протоколом IP. Технические требования», утвержденном Министерством связи 12.11.99 г., изложение которого выходит за рамки данной книги. Сэкономленное таким образом место отдано краткому описанию отечественной платформы IP-телефонии ПРОТЕЙ-IP, реализующей все

требования данного РД и учитывающей упомянутую выше специфику ВСС России.

11.3 Шлюз IP-телефонии Протей-ITG

Шлюз IP-телефонии Протей-ITG реализует передачу речевого трафика и факсимильной информации по сетям с маршрутизацией пакетов IP по протоколу H.323, версия 2. Основным функциональным назначением шлюза является преобразование речевой информации, поступающей от ТфОП с постоянной скоростью передачи, в вид, пригодный для передачи по сетям с маршрутизацией пакетов IP: кодирование и упаковка речевой информации в пакеты RTP/UDP/IP, а также обратное преобразование. Кроме того, шлюз конвертирует сигнальные сообщения систем сигнализации E-DSS1 и ОКС7 (ISUP-R, российская версия) в сигнальные сообщения H.323 и производит обратное преобразование по рекомендации ITU H.246.

Шлюз Протей-ITG подключается к ТфОП по цифровым линиям со скоростью передачи 2048 Кбит/с (Е1) с использованием сигнализации ISUP-R системы общеканальной сигнализации ОКС7, абонентской сигнализации E-DSS1, а также сигнализации по двум выделенным сигнальным каналам «R1.5», а к сетям с маршрутизацией пакетов IP – при помощи интерфейса 10/100BaseT.

В таблицу 11.1 сведены основные технические характеристики шлюза.

Таблица 11.1 Технические спецификации шлюза IP-телефонии «Протей-ITG»

Емкость системы	2 тракта Е1. 60 одновременных соединений
Интерфейсы оборудования	Для подключения к ТфОП:Е1 симметричный, 120 Ом по рекомендации ITU G.703; Для подключения к сети с маршрутизацией пакетов IP: 10/100BaseT
Протоколы и системы сигнализации	TCP/IP, RTP/RTCP, H.323 Версия 2: H.245, H.225 (Q.931 и RAS); DSS1 (Q.931, Q.921), QSIG (ETS 300172), ОКС№ 7 - Российские национальные спецификации MTP, ISUP-R, RADIUS - для подключения к биллинговой системе, T.37 - для передачи факсимильной информации в реальном времени
Алгоритмы кодирования речи	G.711, G.722, G.723.1, G.728, G.729;
Техническое обслуживание	Интуитивно понятный Web-интерфейс

На рис.11.1. изображена обобщенная структура шлюза IP-телефонии Протей-ITG. Следует отметить, что кодирование и пакетирование речевых сигналов, поступающих из ТфОП для последующей их

передачи по IP-сети, реализованы в Протей-ITG на базе специализированных процессоров обработки цифровых сигналов – Digital Signal Processors (DSP). Остальные функции выполняются программным обеспечением, использующим универсальный процессор.

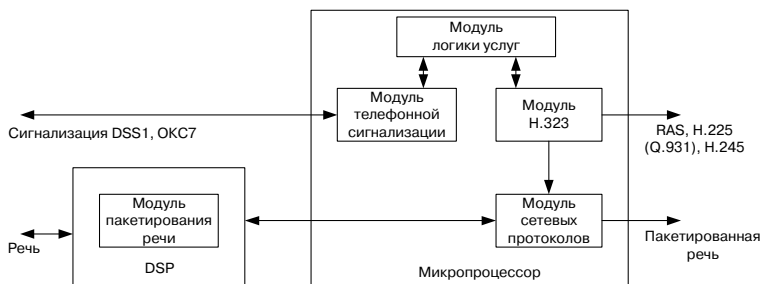


Рис. 11.1 Структурная схема шлюза Протей-ITG

Модуль обработки телефонной сигнализации взаимодействует с телефонным оборудованием, преобразуя сигналы систем DSS1 и OKC7 во внутрисистемные примитивы, которые отражают состояния процесса обслуживания вызова (установление соединения, отбой и т.п.) и используются модулем логики услуг шлюза для установления соединений между ТФОП и IP-сетью.

Модуль сигнализации H.323 обрабатывает сигнальную информацию протоколов RAS, H.225.0 (Q.931) и H.245. Информация о состояниях процесса обслуживания вызова в IP-сети передается в модуль логики услуг шлюза.

Модуль логики услуг шлюза IP-телефонии отвечает за маршрутизацию вызова, поступившего из ТФОП в IP-сеть. Производятся такие операции, как контроль доступа и анализ телефонного номера вызываемого абонента с последующим определением и предоставлением требуемой услуги. При наличии в сети IP-телефонии привратника многие функции могут быть возложены на него.

Модуль пакетирования речи выполняет функции подготовки речевого сигнала, поступающего из ТФОП с постоянной скоростью, для дальнейшей его передачи по сети с маршрутизацией пакетов IP. Основными функциями модуля являются: преобразование речевого сигнала методом импульсно-кодовой модуляции, эхокомпенсация, кодирование речевого сигнала, обнаружение активных периодов и пауз в речи и адаптация воспроизведения. Кроме того, модуль отвечает за детектирование и генерацию сигналов DTMF и за обработку факсимильных и модемных сигналов. Структура модуля пакетирования речи представлена на рис. 11.2.

Механизм обнаружения активных периодов речи проверяет получаемый из ТФОП сигнал на наличие в нем речевой информации. Если в течение определенного времени речевая информация не обнаружена, передача речевых пакетов в IP-сеть прекращается. Ис-

пользование этого механизма существенно повышает эффективность использования полосы пропускания. Экономия полосы может достигать до 60%.

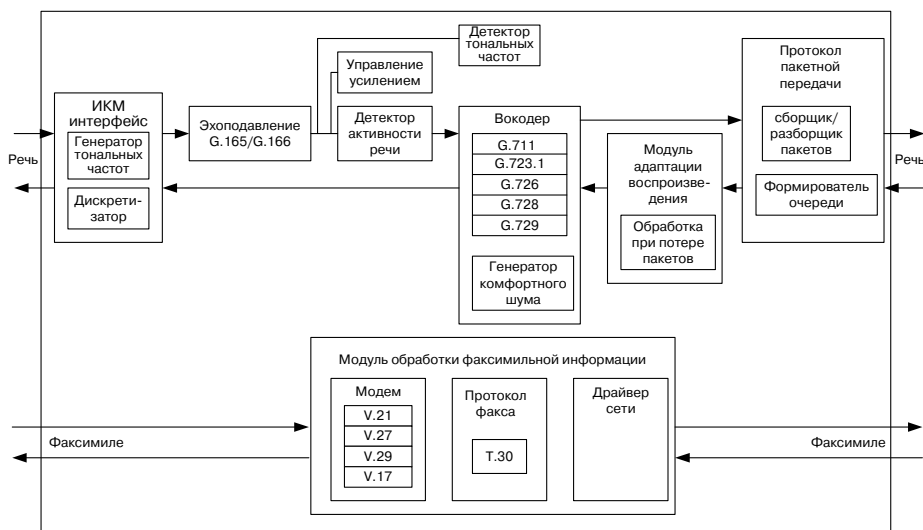


Рис. 11.2 Модуль пакетирования речи

Суть механизма адаптации воспроизведения заключается в буферизации речевых пакетов для сглаживания вариации их задержки. Механизм использует буфер FIFO, хранящий речевые элементы перед их воспроизведением. Далее измеряется джиттер и производится адаптивное управление задержкой пакетов в буфере.

В архитектуру платформы включен упрощенный вариант шлюза IP-телефонии – *устройство уплотнения соединительных линий* – для уплотнения межстанционных соединительных линий связи (сигнальных и разговорных каналов) и передачи мультимплексированной информации через сеть IP с последующим ее демультимплексированием на удаленной стороне. Данное устройство может использоваться, например, операторами сотовой связи для уменьшения числа соединительных линий. В таком варианте шлюза IP-телефонии отсутствует стандартная обработка сигнальных сообщений (сигнальная информация ОКС7 передается прозрачно), так как разговорные каналы постоянно открыты и по ним передается сжатая речь с подавленными паузами. Кроме того, система автоматически обнаруживает сигналы факса и начинает вести обработку информации по протоколу T.38; после окончания факсимильной сессии система возвращается на прежний режим работы.

11.4 Привратник Протей-GK и варианты организации связи

В *привратнике* сосредоточен весь интеллект сети IP-телефонии. Он выполняет функции управления зоной сети IP-телефонии, в которую входят терминалы, шлюзы и устройства управления конференциями, зарегистрированные в этом привратнике.

В число наиболее важных функций, выполняемых привратником с целью обеспечения нормального функционирования управляемой зоны сети, входят:

- регистрация окончного оборудования;
- контроль доступа пользователей системы к услугам IP-телефонии при помощи сигнализации RAS (Рекомендация ITU H.225.0);
- преобразование *alias*-адреса (имени абонента, телефонного номера, адреса электронной почты и др.) в транспортный адрес сети с маршрутизацией пакетов IP (IP адрес + номер порта TCP/UDP);
- контроль, управление и резервирование пропускной способности сети;
- ретрансляция сигнальных сообщений H.225.0 и H.245 между терминалами.

В последнем случае привратник в любое время знает состояние конечных пользователей и может предоставлять дополнительные услуги, такие как переключение связи, переадресация, постановка на ожидание, перехват вызова и т.д.

Возможны следующие два *варианта организации связи* с использованием оборудования IP-телефонии платформы Протей.

В первом варианте шлюз IP-телефонии Протей-ITG и привратник Протей-GK подключаются к существующей сети IP-телефонии.

Подключение может осуществляться на правах корпоративного клиента, т.е. биллинговым центром поставщика услуг IP-телефонии выставляется групповой счет местному оператору у которого установлено оборудование Протей-IP, а оператор в свою очередь выставляет групповой счет за терминированный трафик поставщику услуг, т.е. производится взаиморасчет. Что касается расчетов со «своими» абонентами (пользователями услуги), то для этого местному оператору предпочтительно использовать предоплатные механизмы, т.е. систему сервисных телефонных карт, предоставляя доступ к услугам только по собственным картам оператора. Такой вариант требует от оператора минимальных вложений (особенно, если он уже предоставляет услуги обычной телефонной связи с использованием систе-

мы обработки телефонных карт с Протей-ТК) и, по мнению авторов, является на сегодня оптимальным.

Если же в существующей сети IP-телефонии выставление счетов производится централизованно, то проблема взаимодействия решается следующим образом.

Если сеть построена на базе оборудования VocalТес или, по крайней мере, с наличием привратника, произведенного фирмой VocalТес, который, как правило, занимается начислением платы за разговоры абонентов, то шлюз Протей-ITG общается с привратником по протоколу RAS, входящему в семейство протоколов H.323.

Если сеть построена на базе оборудования Cisco, то шлюз взаимодействует с биллинговым центром сети по протоколу RADIUS.

Второй вариант организации связи заключается в создании на базе оборудования Протей собственной сети IP-телефонии, имеющей выход на другие сети. Между операторами производятся взаиморасчеты.

11.5 Экономические аспекты применения оборудования IP-телефонии

Шлюз IP-телефонии обеспечивает возможность использования сети с маршрутизацией пакетов IP для предоставления услуг междугородной и международной телефонной связи, что приводит к удешевлению услуг при приемлемых показателях качества обслуживания, сравнимых с теми же показателями в ТФОП. Удешевление достигается за счет использования современных алгоритмов кодирования речи, подавления пауз в разговоре и статистического мультиплексирования пользовательской информации.

В главе 1 отмечалось, что подобное удешевление не является единственным (или даже основным) преимуществом внедрения IP-телефонии, но для того, чтобы наглядно продемонстрировать (именно с этой точки зрения) привлекательность предоставления услуг междугородной связи с помощью технологии IP-телефонии, в таблице 11.2 представлены оценки возможных затрат и доходов, ожидаемых при развертывании и эксплуатации узла IP-телефонии для организации связи между Санкт-Петербургом и Москвой. Следует подчеркнуть, что оценки являются упрощенными и приведены исключительно в качестве примера.

Указанная в таблице комплектация оборудования может изменяться в зависимости от того, какое оборудование уже имеется у местного оператора. В полный комплект оборудования, кроме шлюза, входят также привратник, биллинговая система, коммутатор Ethernet и маршрутизатор.

Таблица 11.2 Расчет окупаемости узла IP-телефонии

Исходные данные			
Статья	Единицы	Кол-во	Примечания
Выделенный цифровой канал между С.-Петербургом и Москвой со скоростью	Кбит/с	64	
Количество телефонных линий	шт.	30 (Е1)	
Расчетная загрузка оборудования узла связи	%	20	В дальнейшем – увеличение нагрузки
Допустимая загрузка оборудования узла связи	%	до 100	При увеличении скорости передачи по выделенному каналу
Усредненный добавочный тариф (неочищенная прибыль)	\$/мин	0,06	
Лицензия на предоставление услуг IP-телефонии	МРОТ	40	
Рабочий персонал	чел.	2	
Налог на прибыль	%	34	
Налог на добавленную стоимость (НДС)	%	20	Учен в дальнейшем
Расчетный период	мес.	12	
Разовые затраты			
Получение лицензии	МРОТ	40	
Стоимость оборудования Протей, пуско-наладочных работ, включения в сеть, сетевой мониторинг	\$	11760	Комплектация может быть изменена
Инсталляция выделенного канала	\$	428	через Ростелеком
Инсталляция канала Е1	\$	1972	
Инсталляция серийного тел. номера	\$	185	
Итого	\$	14464	
Ежемесячные затраты на содержание узла			
Аренда выделенного канала	\$	660	
Аренда канала Е1	\$	1825	
Зарплата персонала	\$	1000	
Аренда помещения	\$	150	
Амортизация	\$	93	10%/год от затрат на строительство
Коммерческие расходы (реклама)	\$	910	5% от производственной себестоимости
Отчисления на социальные нужды	\$	390	39% от фонда оплаты труда
Итого	\$	5028	
Расчет прибыли			
Месячный доход = $0,06\$/\text{мин} \times 60\text{мин.} \times 24\text{час} \times 30\text{дней} \times 30\text{каналов} \times 0,20$	\$	15552	
Месячная прибыль = доход - затраты/мес. налоги	\$	6946	
Расчетный период			
Валовый доход = месячный доход * 12	\$	186624	
Затраты на строительство узла (разовые)	\$	14464	
Расходы на содержание узла за 12 месяцев	\$	60336	
Прибыль	\$	73804	После уплаты налога
Резюме: экономическая эффективность узла связи			
Общие затраты за расчетный период	\$	74800	
Валовый доход за расчетный период	\$	186624	
Рентабельность вложений	%	99	

Приведенные данные не претендуют на глубину экономического анализа внедрения IP-телефонии. Более того, по мнению авторов, время для такого анализа еще не настало, слишком молода сама индустрия IP-телефонии. И, все же, содержание таблицы 11.2 может подбодрить читателя, добравшегося почти до конца книги, и показать ему возможность получения реальных доходов от IP-телефонии уже сегодня.

А если приведенные в таблице 11.2 суммы не показались достаточной компенсацией потраченного на чтение данной книги времени, то в следующем параграфе излагается техническая идея, экономическая эффективность которой эквивалентна разве что известной карте боцмана Билли Бонса из стивенсоновского «Острова сокровищ».

11.6 Виртуальная телефонная линия

Сколько раз мы пытались безуспешно дозвониться до абонента, работающего в сети Интернет? Неудобство, бесполезно потраченное время, моральные издержки, возможные убытки и отсутствие возможности сообщить вовремя срочную информацию – далеко неполный перечень неприятностей при использовании телефонной связи, связанных с многочасовым доступом к Интернет. Известные пути решения этой проблемы при помощи линий ADSL, базового доступа ISDN типа 2B+D, установкой второго телефонного номера имеют весьма ограниченное применение в нашей российской действительности.

В основу излагаемого здесь решения положено использование технологии IP-телефонии. Программно-аппаратный комплекс, в состав которого входят шлюз Протей-ITG и привратник Протей-GK, позволяет Оператору связи предоставить абонентам новую услугу: *работать в сети Интернет и разговаривать по телефону одновременно, занимая всего лишь одну обычную аналоговую линию*. Ниже представлено описание механизма организации виртуальной телефонной линии.

Перед началом работы в сети Интернет абонент активизирует на своей телефонной станции дополнительную услугу «Переадресация при занятости абонента». Предусмотрены два способа активизации услуги: самим абонентом при помощи сигналов DTMF или персоналом АТС при помощи средств эксплуатационного управления.

Далее абонент стандартным образом устанавливает соединение со своим поставщиком услуг сети Интернет и получает IP-адрес, назначаемый, как правило, динамически.

Абонент запускает любое клиентское приложение IP-телефонии, например, популярное программное обеспечение NetMeeting. При запуске клиентского приложения автоматически, по протоколу H.323,

инициируется процедура регистрации абонента у привратника сети Протей-GK, в ходе которой указывается телефонный номер и IP-адрес абонента. Кроме того, вводится PIN-код для идентификации абонента. Получив от абонента запрос регистрации – Registration Request, привратник обращается к базе данных для проверки прав абонента на пользование данной услугой. Если абонент подписан на эту услугу, то привратник подтверждает регистрацию сообщением Registration Confirm, после чего абонент может быть доступен во время работы в сети Интернет.

Дополнительно, по IP-адресу пользователя, может быть проведена идентификация Интернет-провайдера, организовавшего доступ абонента к услугам глобальной сети. Это может понадобиться в тех случаях, когда оператор телефонной связи заключает соглашения на организацию второй виртуальной телефонной линии только с некоторыми поставщиками услуг сети Интернет.

Вместо клиентского программного обеспечения может использоваться специальная плата (например, PhoneJack или LineJack производства фирмы Quicknet), вставляемая в персональный компьютер, к которой по двухпроводной линии подключается аналоговый телефонный аппарат.

На рис. 11.3 представлен алгоритм вызова пользователя, работающего в сети Интернет.

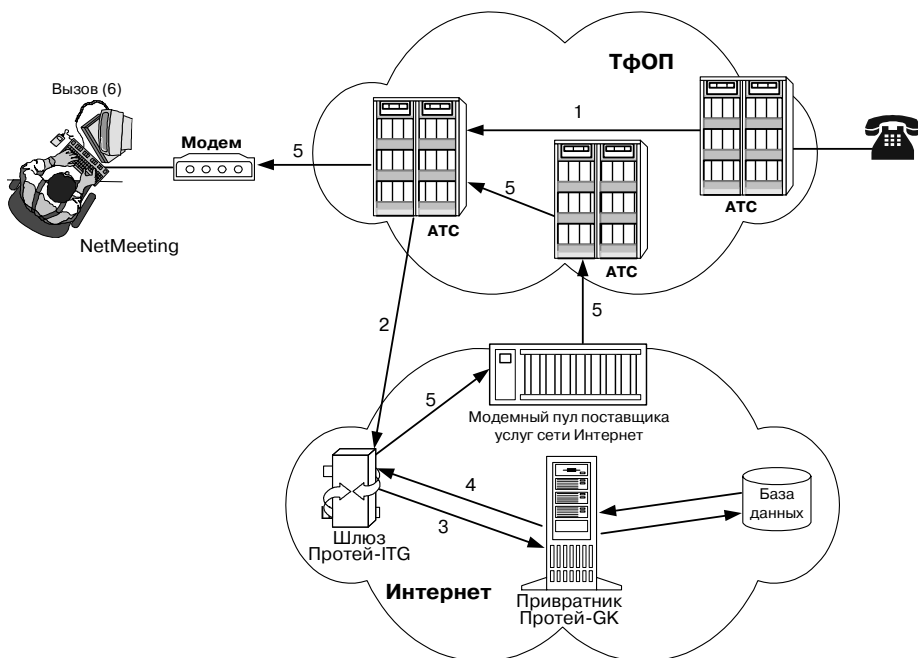


Рис. 11.3 Вызов абонента, работающего в сети Интернет

Вызывающий абонент набирает номер вызываемого абонента (1), и, если этот номер занят (вызываемый пользователь работает в Интернете), вызов переадресуется телефонной станцией к шлюзу Протей-ITG (2). Для того, чтобы вызов мог быть автоматически переадресован к шлюзу, на станции для шлюза должно быть выделено отдельное внутристанционное направление (не включенное в городской план нумерации, чтобы не расходовать номерную емкость). Кроме того, каждому абоненту, подписавшемуся на услугу «Виртуальная телефонная линия», должен быть присвоен внутристанционный (внутрисетевой) номер, на который переадресуется вызов при работе абонента в сети Интернет.

Шлюз, в свою очередь, передает привратнику запрос допуска к использованию сетевых ресурсов Admission Request по протоколу H.323 (3). В запросе указывается телефонный номер вызываемого абонента. Привратник дает шлюзу разрешение использовать сетевые ресурсы (Admission Confirm), в котором указывается IP-адрес вызываемого абонента (4). Далее шлюз маршрутизирует вызов к вызываемому абоненту через IP-сеть (5). У вызываемого абонента на экране появляется сообщение о входящем вызове с указанием телефонного номера вызывающего абонента и акустическое извещение. Он может либо принять этот входящий вызов, либо отказаться от приема.

Чтобы обеспечить хорошее качество воспроизведения речевой информации, оператору необходимо задействовать механизмы предоставления гарантированного качества обслуживания, например, настроить маршрутизатор таким образом, чтобы речевому трафику, передаваемому в пакетах UDP, присваивался более высокий приоритет, чем трафику данных, передаваемому в пакетах TCP. Кроме того, желательно, чтобы шлюз с привратником и модемным пулом поставщика услуг сети Интернет располагались в одной локальной сети.

Если автоматически реализовать услугу «Виртуальная телефонная линия» не представляется возможным по техническим причинам, например, в сети не поддерживается дополнительная услуга «Переадресация при занятости абонента», то ее можно реализовать при помощи системы обработки телефонных карт (СТК) Протей-ТК. Алгоритм такой реализации услуги представлен на рис.11.4. В этом случае услуга может предоставляться любым абонентам ТФОП, но для обеспечения хорошего качества речи, опять-таки, следует задействовать механизмы предоставления гарантированного качества обслуживания.

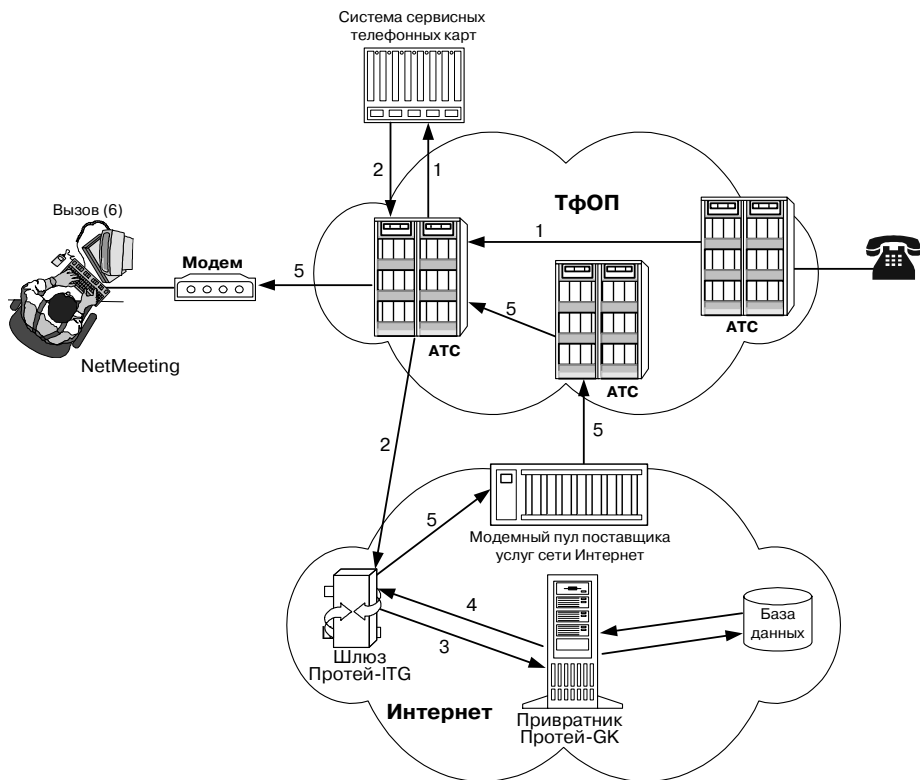


Рис. 11.4 Вызов абонента, работающего в сети Интернет, через СТК

Вызываемый абонент (абонент, подписавшийся на услугу) выполняет действия, аналогичные предыдущему алгоритму.

Вызывающий абонент соединяется с СТК (1). Отметим, что на станции для услуги СТК выделяется серийный номер. Далее абоненту передается приглашение к набору номера. После того, как абонент введет при помощи сигналов DTMF номер вызываемого абонента, СТК устанавливает через АТС соединение со шлюзом Протей-ITG (2), который подключается к станции таким же образом, как и в предыдущем случае. Дальнейший алгоритм установления соединения идентичен алгоритму, описанному выше.

Следует особо отметить, что с помощью вышеизложенного пользователь, работающий в сети Интернет, имеет возможность не только принимать входящие вызовы, но также может сам инициировать вызовы, не прерывая своей Интернет-сессии.

11.7 Центр обработки вызовов

В центре обработки вызовов (Call-center) будут интегрированы как операторские, так и автоматизированные информационные службы, что позволит реализовать в одном комплексе максимально широкий спектр взаимосвязанных услуг. Центр реализован на базе технологии IP-телефонии с поддержкой протокола H.323.

Для организации информационно-справочных служб в центре обработки вызовов предусмотрены ступень распределения вызовов (CPB) и интерактивная речевая система (IVR).

Для офисного рынка в центре реализованы функции учрежденческой телефонной станции, так называемой IP-PBX. Благодаря этому отпадает необходимость разворачивать в офисе две сетевые инфраструктуры: телефонную и компьютерную. IP-PBX не только сохранит функциональные возможности традиционных УАТС (предоставление базовых соединений и стандартных дополнительных услуг), но и предложит сотрудникам офиса новые возможности, например, пользование телефонными услугами своей корпоративной УАТС независимо от того, где они находятся, например, при работе дома.

При помощи центра обработки вызовов абонент может воспользоваться услугами Web-телефонии, т.е. он может инициировать телефонный вызов, выбрав ссылку на имя вызываемого абонента прямо на страницах Internet. Для этого нужно подвести курсор к имени вызываемого абонента в специальной базе данных на Web-странице и щелкнуть клавишей мыши. Другими словами, нужное телефонное соединение устанавливается при щелчке мышью, когда курсор установлен на имени, номере или другом обозначении вызываемого абонента, через ссылку на Web-странице.

Кроме того, центр сможет сопровождать в реальном времени каждого клиента с момента его появления на домашней странице компании в сети Internet до оформления заказа на покупку необходимого продукта, проводя этого клиента через такие этапы, как демонстрация каталога предлагаемых изделий и выяснение возникающих вопросов путем телефонного общения с представителем компании.

Техническое обслуживание оборудования центра обработки вызовов осуществляется через протокол http, т.е. при помощи обычного Web Browser. Так же, через Web-интерфейс, пользователь сможет управлять обслуживанием вызова, заказывать и отменять дополнительные услуги, регистрироваться и получать доступ к услугам и т.д.

11.8 Модуль IPU как средство интеграции цифровых АТС с IP-сетями

Все чаще телефонные сети общего пользования (ТфОП) используются для организации доступа к глобальной сети Internet. Операторы телефонной связи, как правило, не получают от этого существенного дохода, в то время как поставщики услуг сети Интернет получают значительную прибыль. Кроме того, коммутационное оборудование ТфОП проектировалось из расчета интенсивности нагрузки 0,1 – 0,15 Эрланга в час наибольшей нагрузки (ЧНН) на одну абонентскую линию, и именно при таких параметрах обеспечивалось удовлетворительное качество обслуживания телефонных вызовов. Анализ нагрузки с учетом модемных соединений между абонентами ТфОП и поставщиками услуг сети Интернет показывает, что в некоторых случаях интенсивность нагрузки на одну абонентскую линию может достигать 0,8 Эрланга в ЧНН. При этом занимаются не только абонентские, но и межстанционные соединительные линии, что приводит к ощутимым перегрузкам в ТфОП.

Предлагаются различные решения проблемы отвода трафика сети Интернет от соединительных линий ТфОП. Одним из наиболее действенных решений является организация в коммутационном оборудовании точки присутствия Интернет – Internet Point of Presence (IPoP). При этом не только отводится трафик Интернет, но и операторы телефонной связи сами становятся поставщиками услуг глобальной сети Интернет, что позволяет существенно повысить их доходы.

На рис.11.5 представлен входящий в комплекс оборудования АТСЦ-90 модуль IPU (Internet Point of Presence Unit), позволяющий организовать интеграцию коммутационного оборудования АТС в сети с маршрутизацией пакетов IP. Модуль IPU является интегрированным сервером доступа: в качестве шлюза IP-телефонии он обеспечивает передачу речевого трафика и факсимильных сообщений по сетям с маршрутизацией пакетов IP, а в качестве сервера удаленного доступа к IP-сетям предоставляет абонентам ТфОП доступ к Интернет или к удаленным ЛВС по коммутируемым линиям.

Модуль IPU подключается к АТС по цифровым трактам E1 с использованием систем сигнализации OKC7 (ISUP) или E-DSS1, а к сетям с маршрутизацией пакетов IP – при помощи интерфейса 10/100BaseT. Сервер автоматически (по набранному номеру) распознает, какое из приложений должно быть использовано для обслуживания поступившего вызова.

11.9 Тестирование протоколов IP-телефонии

Для тестирования семейства протоколов сигнализации H.323, рассмотренных в главах 5 и 6, а также протоколов стека TCP/IP, в том числе, протоколов прикладного уровня RTP/RTCP, рассмотренных в главе 4, используется протокол-тестер SNT, представленный на рис.11.6.



Рис. 11.6 Протокол-тестер SNT

Протокол-тестер SNT-7531, наряду с тестированием протоколов OKC7, V5, DSS1, проверяет протокол H.323 (как и другие рассмотренные в книге протоколы IP-телефонии) на соответствие международным рекомендациям и российским национальным требованиям. Протокол-тестер может использоваться операторами связи для проведения пуско-наладочных работ и сопряжения оборудования разных фирм-производителей, разработчиками оборудования для отладки своего оборудования, сертификационными и испытательными центрами – для проведения испытаний по «Типовой программе и методике».

Определены два основных режима функционирования тестера: мониторинг и симуляция.

Мониторинг предусматривает пассивное чтение данных в сигнальных каналах. При работе тестера SNT в режиме мониторинга передаваемые и принимаемые сигнальные сообщения выводятся на экран в порядке их передачи и приема. Различные фильтры и настройки монитора позволяют выводить на экран в удобном для пользователя формате только необходимые данные (например, только сигнализацию RAS и т.д.). Существует возможность сохранения данных в файле в ASCII формате.

Симулятор является эталонной моделью оборудования, в котором реализован тестируемый протокол. При совместной работе с терминальным оборудованием Н.323 (шлюзом) или привратником протокол-тестер симулирует, соответственно, режимы работы привратника, терминала или шлюза и позволяет определить, насколько функционирование тестируемого оборудования соответствует требованиям международных и национальных стандартов. Реализованный в протокол-тестере симулятор протокола Н.323 позволяет создать исходящий вызов, ответить на вызов, имитировать занятость абонента и т.д., а также содержит тестовые сценарии для всех основных этапов установления и завершения соединения при помощи протокола Н.323, включая:

- обнаружение привратника и регистрацию в нем (сигнализация RAS);
- установление сигнального соединения (сигнализация RAS и Н.225.0/Q.931);
- определение ведущего и ведомого оборудования и обмен информацией о его функциональных возможностях (сигнализация Н.245);
- открытие логических каналов (сигнализация Н.245);
- передачу речевой информации (протокол RTP/RTCP);
- завершение соединения.

Остановимся немного более подробно на каждом из этих этапов. В тестовые сценарии регистрации шлюза включены типичные ошибочные ситуации, такие, например, как:

- отсутствие ответа от привратника;
- отказ в регистрации.

Для установления соединения между двумя терминалами (шлюзами) с участием привратника используется два способа передачи сигнальной информации: непосредственно от одного оборудования к другому или с маршрутизацией сигнальных сообщений привратником.

В протокол-тестере реализованы следующие ошибочные ситуации, возникающие при установлении сигнального соединения:

- запрет доступа привратником (из-за того, что не зарегистрирована вызываемая сторона, не зарегистрирован вызывающий терминал (шлюз), или отсутствует запрошенная полоса пропускания);
- не получен ответ на сообщение Setup: отсутствие сообщений Call Proceeding/Alerting/Connect.

После успешного установления сигнального соединения терминалы (шлюзы) Н.323 открывают управляющий канал Н.245. На этом этапе выполняются две процедуры: определение ведущего и ведомого оборудования и обмен информацией о функциональных возможностях.

Во время выполнения любой из этих процедур могут возникнуть сбои в работе протокола H.245, отработка которых предусмотрена в тестовых сценариях протокол-тестера:

- сбой в определении ведущего и ведомого оборудования (из-за того, что совпали числовые значения в поле типа оборудования, или иницилирующая сторона не принимает ответа от встречной стороны в течение назначенного промежутка времени);
- сбой в процедуре обмена информацией о функциональных возможностях (из-за того, что шлюз, иницировавший процедуру, не принимает сообщение подтверждения в течение назначенного промежутка времени).

После выполнения вышеуказанных процедур начинается процедура открытия логических каналов. В тестовые сценарии для проверки этого этапа включены следующие возможные случаи:

- приемный шлюз запрещает открытие канала (из-за того, что вид информации неизвестен или не поддерживается, ширина полосы недостаточна, идентификатор сеанса связи недействителен и т.д.);
- иницилирующий шлюз своевременно не получает подтверждения и завершает соединение.

Специальные опции протокол-тестера SNT-7531 позволяют измерять время установления соединения – один из важнейших параметров качества обслуживания. При их помощи определяются также и другие параметры качества обслуживания: количество потерянных RTP-пакетов, средняя задержка и вариация задержки RTP-пакетов.

Завершая эту главу и всю книгу, хочется пожелать читателю, дочитавшему ее до конца, не пожалеть о потраченном времени. Ко всем приведенным в книге аргументам и длинному списку упомянутых в данной главе продуктов IP-телефонии можно добавить и знаменательное событие, произошедшее 1 июня 1999 года, когда Министерство связи (в ту пору – Госкомитет по связи и информатизации) официально признало IP-телефонию подлежащим лицензированию видом услуг связи, хотя и под псевдонимом «Телематическая служба речевой информации».

Вспоминая замечание о попытках сдерживания IP-телефонии административными методами, с которого начиналась эта книга, можно с известной долей оптимизма заключить, что здесь также победил здравый смысл, позволивший применить вполне подходящий к развитию IP-телефонии принцип Авраама Линкольна: «Если вы держите слона за заднюю ногу, а он вырывается, то самое лучшее – отпустить его».

Глоссарий

ACELP	Algebraic Code-Excited Linear Prediction. Популярный алгоритм компрессии речевого сигнала (скорость передачи 8 Кбит/с)
ADPCM	Adaptive Differential PCM. Адаптивная дифференциальная ИКМ
API	Application Programming Interface. Интерфейс прикладного программирования. Это набор функций, с помощью которых приложения взаимодействуют с операционной системой для выполнения задач
ARP	Address Resolution Protocol. Протокол, обеспечивающий динамическое преобразование адресов Интернет в физические (аппаратные) адреса оборудования локальной сети
ARPA	Advanced Research Projects Agency. Специальное агентство при Министерстве обороны США (Department of Defense – DoD), создавшее сеть ARPANET
ASCII	American Standard Code for Information Interchange. Американский стандартный код для обмена информацией
ASN.1	Abstract Syntax Notation One. Спецификация абстрактной синтаксической нотации, версия 1. Служит для описания информационных объектов прикладного уровня
BGP	Border Gateway Protocol. Одноуровневый многопунктовый протокол динамической междоменной маршрутизации. При выборе маршрута учитывает число пересылок, пропускную способность каналов и др. Выявляет загромождение маршрутов.
CBQ	Class Based Queuing. Организация очередей с учетом класса трафика. Каждый класс имеет свою очередь, и ему гарантируется некоторая доля пропускной способности канала. Если какой-то класс не использует весь отведенный ему ресурс, доля каждого из остальных классов пропорционально увеличивается.
CNG	Comfort Noise Generator. Генератор комфортного шума. Используется в системах с компрессией речевого сигнала для устранения эффекта «гробовой тишины» в паузах, когда передача сигнала прекращается, и у слушающего создается иллюзия, что связь прервана.
CS-ACELP	Conjugate Structure – Algebraic Code – Excited Linear Prediction. Технология CS-ASELP применяется в кодеках G.729, используемых для передачи речи по сетям Frame Relay. Обеспечивает скорость передачи 8 Кбит/с при длительности кадра 10 мс.
CSRC	Contributing Source Identifier. Список полей с идентификаторами источников, речевая информация которых смешивается при создании RTP-пакета. Во время речевой конференции каждый RTP-пакет содержит свой SSRC.
DiffServ	Differentiated Services. Система дифференцированного обслуживания трафика разных классов, разработанная комитетом IETF
DNS	Domain Name System. Сервер имен доменов в Интернет, преобразующий эти имена в IP-адреса.
DSP	Digital Signal Processor. Процессор цифровой обработки сигналов.
DTMF	Dual Tone Multi-Frequency. Многочастотная система кодирования цифр номера. Имеется две группы частот, и цифра кодируется двумя частотами из разных групп.
DTX	Discontinuous Transmission. Прерываемая передача. Кодек прекращает передачу пакетов, когда детектор речевой активности обнаруживает паузу в речи.
DVMRP	Distance Vector Multicast Routing Protocol. Один из основных протоколов, на базе которых реализуется многоадресная рассылка в IP-сетях.
E1	Цифровая система передачи со скоростью 2.048 Мбит/с, используемая в России и других европейских странах.
ECMA	European Computer Manufacturer's Association. Ассоциация европейских производителей компьютеров.

EIA	Electronic Industries Association. Ассоциация электронной промышленности. Группа, разрабатывающая стандарты передачи данных. Разработчик ряда коммуникационных стандартов, в том числе, стандарты EIA/TIA-232 и EIA/TIA-449.
ETSI	European Telecommunication Standards Institute. Европейский институт стандартов в области связи.
FIFO	First In, First Out. Дисциплина обслуживания, при которой первой обслуживается та заявка, которая первой поступила в очередь.
FTP	File Transfer Protocol. Протокол переноса файлов.
GCP	Gateway Control Protocol. Протокол управления шлюзом.
GK	Gatekeeper. Привратник. Выполняет функции управления зоной сети H.323.
GW	Gateway. Шлюз. Аппаратно-программный комплекс, обеспечивающий обмен данными между сетями разных типов.
H.323	Рекомендация ITU-T, которая определяет системы мультимедийной связи в сетях с пакетной коммутацией, не обеспечивающие гарантированное качество обслуживания
ICMP	Internet Control Message Protocol Протокол управляющих сообщений в Интернет. Предоставляет программному обеспечению рабочей станции или маршрутизатора возможность обмениваться с другими устройствами информацией, относящейся к маршрутизации пакетов. Является необходимой частью стека протоколов TCP/IP.
IDCP	IP Device Control Protocol. Протокол управления оборудованием, реализующим технологию маршрутизации пакетов IP. Предложен фирмой Level 3.
IEEE	Institute of Electrical and Electronics Engineers. Институт инженеров электротехнической и электронной отраслей. Ведет исследования в области связи и разрабатывает коммуникационные и сетевые стандарты.
IETF	Internet Engineering Task Force. Группа инженерных проблем Интернет. Включает в себя более 80 самостоятельных подгрупп, отвечает за разработку стандартов для Интернет. Устанавливает приоритеты и вырабатывает решения по краткосрочным вопросам и проблемам, включая протоколы, архитектуру и эксплуатацию. Работает под эгидой ISO.
IP	Internet Protocol. Протокол межсетевого взаимодействия.
IPOP	Internet Point of Presence. Точка присутствия поставщика услуг Интернет в коммутационном оборудовании.
ISO	International Organization for Standardization. ISO – это не просто аббревиатура: греческое слово <i>iso</i> означает «равный». ISO Основана в 1946 г. и представляет собой международную организацию, объединяющую более 75 национальных бюро разных стран. Разработала важнейшие стандарты, в том числе и компьютерные.
ISP	Internet Service Provider. Поставщик услуг Интернет. Компания, предоставляющая доступ в Интернет другим компаниям или частным лицам.
ITU-T	International Telecommunications Union Telecommunication Standardization Sector. Сектор Международного Союза Электросвязи, разрабатывающий рекомендации в области телекоммуникаций. Выполняет функции бывшего CCITT (МККТТ).
LDP	Label Distribution Protocol. Протокол распределения меток. Поддерживает процедуры «раздачи» и согласования меток между маршрутизаторами сети MPLS.
LER	Label Edge Router. Пограничный маршрутизатор сети MPLS.
LPC	Linear Prediction Coding. Кодирование с линейным предсказанием.
LSP	Label Switched Path. Коммутируемый по меткам тракт.
LSR	Label Switching Router. Маршрутизатор коммутации по меткам.

MCU	Multipoint Control Unit. Устройство управления конференцией. Обеспечивает согласование функциональных возможностей участников конференции, то есть алгоритмов, используемых каждым из них для кодирования речи, видеoinформации и данных; может транскодировать информацию любого вида, если терминалы, участвующие в конференции, используют разные алгоритмы кодирования; в процессе согласования выбирает режим конференции. Производит смешивание или коммутацию информации, принимаемой от участников, и ее распределение.
MEGACO/ H.248	Протокол управления транспортным шлюзом. Создан рабочей группой MEGACO комитета IETF.
MG	Media Gateway. Транспортный шлюз.
MGCP	Media Gateway Control Protocol. Протокол управления шлюзами.
MOS	Mean Opinion Score. Характеристика качества передачи речи с применением кодека того или иного типа, получаемая как среднее значение результатов оценки качества большой группой экспертов по пятибалльной шкале.
MP	Multipoint processor. Процессор для обработки информации пользователей при централизованных конференциях.
MPLS	Multi-Protocol Label Switching. Многопротокольная коммутация по меткам. Основана на том, что пакеты, поступающие в пограничный маршрутизатор сети MPLS извне, разделяются на «классы эквивалентности пересылки». Каждому классу назначается система меток – коротких заголовков, используемых для маршрутизации пакетов внутри сети MPLS вместо длинных заголовков сетевого уровня.
Multicasting	Многоадресная рассылка информации (аудио, видео или данных).
OPWA	One Path With Advertising. Вариант алгоритма, который поддерживается протоколом RSVP. С помощью OPWA информация управляющих пакетов RSVP может быть использована для предсказания «сквозного» QoS всего маршрута.
OSI	Open System Interconnection. Взаимодействие открытых систем. Созданная ISO и ITU-T международная программа разработки стандартов сетевой передачи данных.
POTS	Plain Old Telephone Service. Услуги традиционной телефонии.
PPP	Point-to-Point Protocol. Протокол двусторонней связи. Используется при связи через Интернет. Реализует обмен пакетами между компьютерами или иными устройствами
PSTN	Public Switched Telephone Network. Телефонная сеть общего пользования (ТфОП). Общий термин, используемый для обозначения телефонных сетей во всем мире.
QCIF	Quarter-Common Intermediate Format. Обязательный формат изображений, который должны обеспечивать кодеки.
QoS	Quality of Service. Качество обслуживания (услуги).
Router	Маршрутизатор. Программно-аппаратное устройство, которое направляет информацию пользователя по маршруту, ведущему к адресату. Ведет статистику использования ресурсов сети, обеспечивает определенный уровень защиты информации.
RADIUS	Remote Authentication Dial-In User Service. Протокол аутентификации и авторизации абонентов, а также учета объема предоставленных им услуг.
RAS	Registration Admission and Status. Протокол взаимодействия терминального оборудования с привратником. Входит в семейство протоколов H.323.
RSVP	Resource Reservation Protocol. Протокол резервирования ресурсов.
RTCP	Real-time Transport Control Protocol. Протокол контроля транспортировки информации в реальном времени. Организует обратную связь получателя информации с ее отправителем, используемую для обмена сведениями о числе переданных и потерянных пакетов, о задержке, ее джиттере и т.д. Эти сведения отправитель может использовать для изменения параметров передачи с целью улучшить QoS (например, уменьшить коэффициент сжатия информации).

RTP	Real Time Transport Protocol. Протокол транспортировки в реальном времени. Служит базисом практически для всех приложений, связанных с интерактивным обменом речевой и видеоинформацией в IP-сети. Позволяет компенсировать отрицательное влияние джиттера на качество передачи такой информации, но не гарантирует своевременную доставку пакетов (за это отвечают нижележащие протоколы) и не выполняет функций исправления ошибок и управления потоком. Обычно базируется на протоколе UDP и использует его возможности, но может работать и поверх других транспортных протоколов.
SGSP	Simple Gateway Control Protocol. Простой протокол управления шлюзами. Разработан компанией Telecordia (бывшая компания Bellcore).
SIP	Session Initiation Protocol. Протокол инициирования сеансов связи. Является протоколом прикладного уровня и предназначается для организации, изменения и завершения сеансов связи – мультимедийных конференций, телефонных соединений и соединений пользователей с разнообразными приложениями.
SNMP	Simple Network Management Protocol. Простой протокол эксплуатационного управления сетью.
TAPI	Telephony Applications Programming Interface. Интерфейс для программирования телефонных приложений. Это – стандартный программный интерфейс для создания приложений компьютерной телефонии в среде Windows, определяющий набор функций, которые необходимы программе для взаимодействия с телефонами, линиями и коммутаторами. Стандарт TAPI используется при разработке приложений для рабочих станций, выполняющих все интеллектуальные действия.
TCP	Transmission Control Protocol. Протокол управления передачей (данных). Основной транспортный протокол в стеке протоколов TCP/IP. Устанавливает связь между двумя компьютерами и организует надежный обмен данными в дуплексном режиме.
TCP/IP	Transmission Control Protocol/Internet Protocol. Стек протоколов, обеспечивающих организацию связи между компьютерами в сети Интернет. Встроен в операционную систему UNIX и широко используется при операциях в сети Интернет, являясь de facto сетевым стандартом для передачи данных. Даже те сетевые операционные системы, которые имеют собственные протоколы, поддерживают также и TCP/IP.
TIPHON	Telecommunications and Internet Protocol Harmonisation Over Networks. Проект под эгидой ETSI (начат в 1997 г), в котором участвует более 40 крупных компаний. Цель проекта – поддержка рынка, предусматривающего сочетание телекоммуникационных услуг сетей с коммутацией каналов и сетей с коммутацией пакетов.
TSAPI	Telephony Services Application Programming Interface. API для создания телефонных услуг. Разработан Novell и AT&T. Позволяет программистам конструировать телефонные и СТИ-приложения. Аналогичен TAPI, но, в отличие от него, функционирует на платформах Netware. Другое отличие – TAPI используется как для клиентских, так и для серверных приложений, а TSAPI предназначен исключительно для сервера.
UAC	User Agent Client. Агент пользователя – клиент. Прикладная программа, которая инициирует SIP-запрос.
UAS	User Agent Server. Агент пользователя – сервер. Прикладная программа, которая принимает запрос и от имени пользователя возвращает ответ с информацией о том, что запрос принимается, не принимается или переадресуется.
UDP	User Datagram Protocol. Протокол передачи дейтаграмм пользователя. Подобно TCP, использует для доставки данных протокол IP. В отличие от TCP/IP, предусматривает обмен дейтаграммами без подтверждения (то есть доставка не гарантируется).
VAD	Voice Activity Detector. Детектор речевой активности. Используется в кодеках со сжатием речевой информации для определения момента окончания паузы в речи.
VoIP	Voice over Internet Protocol. Технология, позволяющая использовать IP-сеть для передачи речевой информации.
WFQ	Weighted Fair Queuing. Взвешенная справедливая очередь. Один из алгоритмов управления обслуживанием очередей.

Литература

1. Бакланов И.Г. ISDN и IP-телефония / Вестник связи, 1999, №4.
2. Брау Д. Грядет год стандарта H.323? / Сети и системы связи, 1999, №14.
3. Будников В.Ю., Пономарев Б.А. Технологии обеспечения качества обслуживания в мультисервисных сетях / Вестник связи, 2000, №9.
4. Варакин Л. Телекоммуникационный феномен России / Вестник связи International, 1999, №4.
5. Варламова Е. IP-телефония в России / Connect! Мир связи, 1999, №9.
6. Гольдштейн Б.С. Сигнализация в сетях связи. Том 1. М.: Радио и связь, 1998.
7. Гольдштейн Б.С. Протоколы сети доступа. Том 2. М.: Радио и связь, 1999.
8. Гольдштейн Б.С., Ехриель И.М., Перле Р.Д. Интеллектуальные сети. М.: Радио и связь, 2000.
9. Кузнецов А.Е., Пинчук А. В., Суховицкий А.Л. Построение сетей IP-телефонии / Компьютерная телефония, 2000, №6.
10. Кульгин М. Технологии корпоративных сетей. Изд. «Питер», 1999.
11. Ломакин Д. Технические решения IP-телефонии / Мобильные системы, 1999 № 8.
12. Мюнх Б., Скворцова С. Сигнализация в сетях IP-телефонии. – Часть I, II / Сети и системы связи, 1999. – №13(47), 14(48).
13. Уиллис Д. Интеграция речи и данных. В начале долгого пути. / Сети и системы связи, 1999. – №16.
14. Шнепс-Шнеппе М.А. Интеллектуальные услуги – это ДВО / Информ-курьер-связь, 2000 – №9.
15. Armitage Grenville. Quality of Service in IP Networks. – Macmillan Technical Publishing, 2000.
16. Anquetil L-P., Bouwen J., Conte A., Van Doorselaer. B. Media Gateway Control Protocol and Voice over IP Gateway. – Alcatel Telecommunications Review, 2nd Quarter 1999.
17. Black Uyless. Voice Over IP. – Prentice Hall, 08 / 99. 0130224634
18. Caputo R. Cisco Packetized Voice and Data Integration. – McGraw-Hill Cisco Technical Expert, 2000
19. Curtin P., Whyte B. Tigris – A gateway between circuit-switched and IP networks / Ericson Rewiew, 1999, №2.
20. Davidson J., Peters J. Voice Over IP Fundamentals. – Cisco Press, 2000.
21. DeMartino K. ISDN and the Internet. – Computer Networks, 1999.

22. Douskalis B. IP Telephony. The Integration of Robust VoIP Services. – Prentice Hall, 1999.
23. Durham D., Yavatkar R.. Inside the Internet's Resource Reservation Protocol: Foundations for Quality of Service, 2000
24. Faynberg I., Gabuzda L., Lu Hui-Lan. Converged Networks and Services: Internetworking IP and the PSTN. – John Wiley & Sons, 2000.
25. Goncalves M. Voice Over IP Networks. – McGraw Hill Publishing, 1998.
26. Goralski W., Kolon M. IP Telephony. – McGraw Hill Publishing, 1999.
27. Harte . Voice Over Data Network Internet, Frame Relay, and ATM.– APDG Inc. 2000
28. Hersent O, Gurle D., Petit Jean-Pierre. IP Telephony: Packet-Based Multimedia Communications Systems.- Addison-Wesley Pub Co, 2000.
29. Horak R. Communications systems & networks / Second Edition, M Et T Books and IDG Books Worldwide, Inc., 2000
30. Houghton T. F., E. C. Schloemer, E. S. Szurkowski, W. P. Weber. A packet telephony gateway for public network operators. – Bell Laboratories, Lucent Technologies – U.S.A., XVI World Telecom Congress Proceeding, 1997.
31. ITU-T Recommendation E.164. Numbering Plan for the ISDN Era. – 1991.
32. ITU-T Recommendation G.711. Pulse Code Modulation of 3kHz Audio Channel. – 1988.
33. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kbit / sec. – 1996.
34. ITU-T Recommendation G.728. Coding of Speech at 16 kbit / s Using Low-delay Code Excited Linear Prediction (LD-CELP). – 1992.
35. ITU-T Recommendation G.729. Speech codec for multimedia telecommunications transmitting at 8 / 13 kbit / s. – 1996.
36. ITU-T Recommendation H.225.0. Call signaling protocols and media stream packetization for packet-based multimedia communication systems. – Geneva, 1998.
37. ITU-T Recommendation H.245. Control protocol for multimedia communication. – Geneva, 1998
38. ITU-T Recommendation H.248. Gateway control protocol. – Geneva, 2000.
39. ITU-T Recommendation H.320. Narrow-band Visual Telephone Systems and Terminal Equipment. – 1996.
40. ITU-T Recommendation H.321. Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments. – 1996.
41. ITU-T Recommendation H.322. Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Guaranteed Quality of Service. – 1996.

42. ITU-T Recommendation H.323. Packet based multimedia communication systems. – Geneva, 1998.
43. ITU-T Recommendation H.324. Terminal for Low Bit Rate Multimedia Communications. – 1996.
44. ITU-T Recommendation Q.931. ISDN User-Network Interface Layer 3 Specification for Basic Call Control. – 1993.
45. Lee J. Implementing Voice Over IP. McGraw Hill Text, 2000.
46. Luczywek M. Cisco Voice over IP Handbook. IDG Books Worldwide, 2000.
47. McDysan D. Phd. Qos & Traffic Management in Ip & Atm Networks
48. Miller M. Voice over IP: Strategies for the Converged Network. IDG Books Worldwide, 2000.
49. Minoli D., Minoli E. Delivering Voice over IP Networks / John Wiley & Sons, Inc., 1998.
50. Regis B., Donald G.. Voice & Data Communications Handbook Third Edition. – McGraw Hill Publishing, 2000.
51. Reid M. Multimedia conferencing over ISDN and IP networks using ITU-T H-series recommendations: architecture, control and coordination / Computer Networks, 1999 – №31.
52. RFC 2205. Resource Reservation Protocol (RSVP). Ver.1. Functional Specification. – September 1997.
53. RFC 2327. Session Description Protocol. M. Handley, V. Jacobson. April, 1998.
54. RFC 2543. SIP: Session Initiation Protocol. M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg. March 1999.
55. RFC 2616. Hypertext Transfer Protocol — HTTP / 1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June 1999.
56. RFC 2705. Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999.
57. RFC 2865. Remote Authentication Dial In User Service (RADIUS). C. Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.
58. RFC 2885. Megaco Protocol 0.8. F. Cuervo, N. Greene, C. Huitema, A. Rayhan, B. Rosen, J. Segers. August 2000.
59. Toga J., Ott J. ITU-T standardization activities for interactive multimedia communications on packet-based networks: H.323 and related recommendations / Computer Networks, 1999.
60. Uyless Black. Voice over IP, Prentice Hall PTR, 2000.
61. Walter J. Goralski, Matthew C. Kolon. IP telephony / The McGraw-Hill Co., Inc., 2000.



Б.С. Гольдштейн – автор более 130 печатных работ в области коммуникаций. Его перу принадлежат, в частности, «Сигнализация в сетях связи», «Протоколы сетей доступа» и «Интеллектуальные сети» (последняя в соавторстве с И.М. Ехриелем и Р.Д. Рерле). Будучи заместителем директора института и начальником научно-исследовательского отделения института, руководит разработками перспективных систем и технологий инфокоммуникаций. 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, профессор кафедры сетей связи Государственного университета телекоммуникаций им.М.А.Бонч-Бруевича, член IEEE.



А.В. Пинчук родился в 1973 году, а в 1996 году окончил Санкт-Петербургский государственный технический университет. Во время обучения на дневном отделении выполнил ряд разработок в области абонентской сети доступа, а непосредственно после защиты дипломного проекта возглавил ведущую лабораторию научно-исследовательского отделения института. Под его руководством и при непосредственном участии выполнен ряд ключевых НИОКР в областях компьютерной и IP-телефонии, организации телекоммуникационных услуг, преобразования протоколов и оборудования сети доступа. В 1999 году окончил аспирантуру Санкт-Петербургского государственного университета телекоммуникаций им.М.А.Бонч-Бруевича по кафедре сетей связи. В ведущих телекоммуникационных журналах отрасли опубликовал более 10 печатных работ.



А.Л. Суховицкий родился в 1973 году. В 1995 году окончил Санкт-Петербургский университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, а в 1999 году очную аспирантуру по специальности «Сети, системы связи и распределение информации». Защитил кандидатскую диссертацию по исследованию сетей АТМ. С 1996 года по настоящее время работает в ЛОНИИС. Занимался тестированием и адаптацией телекоммуникационного оборудования ISDN, АТМ, IP. В настоящее время в качестве научного сотрудника ведет системные исследования в области IP-телефонии и участвует в проекте создания отечественного оборудования IP-телефонии. С 1999 года также читает курс лекций по современным телекоммуникационным сетям и IP-телефонии на кафедре «Сетей связи» СПбГУТ. Автор 10 печатных работ.