# 1. Введение. В чем состоит планирование сети

Корпоративная сеть - это сложная система, включающая тысячи самых разнообразных компонентов: компьютеры разных типов, начиная с настольных и кончая мейнфремами, системное и прикладное программное обеспечение, сетевые адаптеры, концентраторы, коммутаторы и маршрутизаторы, кабельную систему. Основная задача системных интеграторов и администраторов состоит в том, чтобы эта громоздкая и весьма дорогостоящая система как можно лучше справлялась с обработкой потоков информации, циркулирующих между сотрудниками предприятия и позволяла принимать им своевременные и рациональные решения, обеспечивающие выживание предприятия в жесткой конкурентной борьбе. А так как жизнь не стоит на месте, то и содержание корпоративной информации, интенсивность ее потоков и способы ее обработки постоянно меняются. Последний пример резкого изменения технологии автоматизированной обработки корпоративной информации у всех на виду - он связан с беспрецедентным ростом популярности Internet в последние 2 - 3 года.

Изменения, причиной которых стал Internet, многогранны. Гипертекстовая служба WWW изменила способ представления информации человеку, собрав на своих страницах все популярные ее виды - текст, графику и звук. Транспорт Internet - недорогой и доступный практически всем предприятиям (а через телефонные сети и одиночным пользователям) - существенно облегчил задачу построения территориальной корпоративной сети, одновременно выдвинув на первый план задачу защиты корпоративных данных при передаче их через в высшей степени общедоступную публичную сеть с многомиллионным "населением". Стек TCP/IP сразу же вышел на первое место, потеснив прежних лидеров локальных сетей IPX и NetBIOS, а в территориальных сетях - Х.25.

Популярность Internet оказывает на корпоративные сети не только техническое и технологическое влияние. Так как Internet постепенно становится общемировой сетью интерактивного взаимодействия людей, то Internet начинает все больше и больше использоваться не только для распространения информации, в том числе и рекламной, но и для осуществления самих деловых операций - покупки товаров и услуг, перемещения финансовых активов и т.п. Это в корне меняет для многих предприятий саму канву ведения бизнеса, так как появляются миллионы потенциальных покупателей, которых нужно снабжать рекламной информацией, тысячи интересующихся продукцией клиентов, которым нужно предоставлять дополнительную информацию и вступать в активный диалог через Internet, и, наконец, сотни покупателей, с которыми нужно совершать электронные сделки. Сюда нужно добавить и обмен информацией с предприятиями-соисполнителями или партнерами по бизнесу. Изменения схемы ведения бизнеса меняют и требования, предъявляемые к корпоративной сети. Например, использование технологии Intranet сломало привычные пропорции внутреннего и внешнего трафика предприятия в целом и его подразделений - старое правило, гласящее, что 80% трафика является внутренним и только 20% идет вовне, сейчас не отражает истинного положения дел. Интенсивное обращение к Web-сайтам внешних организаций и других подразделений предприятия резко повысило долю внешнего трафика и, соответственно, повысило нагрузку на пограничные маршрутизаторы и межсетевые экраны (firewalls) корпоративной сети. Другим примером влияния Internet на бизнес-процессы может служить необходимость аутентификации и авторизации огромного числа клиентов, обращающихся за информацией на серверы предприятия извне. Старые способы, основанные на заведении учетной информации на каждого пользователя в базе данных сети и выдаче ему индивидуального пароля, здесь уже не годятся - ни администраторы, ни серверы аутентификации сети с таким объемом работ не справятся. Поэтому появляются новые методы проверки легальности пользователей, заимствованные из практики организаций, имеющих дело с большими потоками клиентов - магазинов, выставок и т.п. Влияние Internet на корпоративную сеть - это только один, хотя и яркий, пример постоянных изменений, которые претерпевает технология автоматизированной обработки информации на современном предприятии, желающем не отстать от конкурентов. Постоянно появляются технические, технологические и организационные новинки, которые необходимо использовать в корпоративной сети для поддержания ее в состоянии, соответствующем требованиям времени. Без внесения изменений корпоративная сеть быстро морально устареет и не сможет работать так, чтобы предприятие смогло успешно выдерживать жесткую конкурентную борьбу на мировом рынке. Как правило, срок морального старения продуктов и решений в области информационных технологий находится в районе 3 - 5 лет.

Как же нужно поступать, чтобы предприятию не нужно было бы полностью перестраивать свою корпоративную сеть каждые 3 - 5 лет, что безусловно связано с огромными расходами? Ответ простой - нужно постоянно следить за основными тенденциями развития мира сетевых и информационных технологий и постоянно вносить в сеть (в программы, сервисы, аппаратуру) такие изменения , которые позволили бы сети плавно отрабатывать каждый резкий поворот. То есть нужно правильно видеть стратегическое направление развития вашей корпоративной сети, постоянно коррелировать его с направлением развития всего сетевого мира и тогда меньше шансов завести корпоративную сеть в такой тупик, откуда нет иного выхода, кроме полной перестройки сети. По крайней мере, нельзя вкладывать большие деньги и силы в решения, в будущности которых имеются большие сомнения. Например, весьма рискованно строить сегодня новую сеть исключительно на сетевой операционной системе NovellNetWare, которая переживает всеми признаваемый кризис. Если в вашей сети уже работает с десяток серверов NetWare, то добавление к ним нового сервера IntranetWare может быть и целесообразно, так как дает возможность старым серверам возможность работы с Internet и сетями TCP/IP. Но построение новой сети за счет покупки нескольких десятков копий IntranetWare трудно назвать стратегически верным решением, WindowsNT и Unix сейчас дают гораздо больше гарантий относительно своей жизнеспособности.

Стратегическое планирование сети состоит в нахождении компромисса между потребностями предприятия в автоматизированной обработке информации, его финансовыми возможностями и возможностями сетевых и информационных технологий сегодня и в ближайшем будущем.

При стратегическом планировании сети нужно принять решения по четырем группам вопросов:

1. Какие новые идеи, решения и продукты являются стратегически важными? Какие решения в стратегически важных областях являются перспективными? Какие из них могут оказаться полезными в вашей корпоративной сети?
2. Каким образом новые решения и продукты нужно внедрять в существующую сеть? На какие этапы нужно разбить процесс перехода на новые решения и продукты, как обеспечить максимально безболезненное взаимодействие новых и старых частей и компонентов сети?
3. Как рационально выбрать внешних соисполнителей для внедрения в сеть новых решений и продуктов? Как выбрать интеграторов, производителей и поставщиков программных и аппаратных продуктов, провайдеров услуг территориальных сетей?
4. Как организовать процесс обучения своих сотрудников новым технологиям и продуктам? Стоит ли набирать уже обученных специалистов со стороны?

Рассмотрим эти вопросы более подробно.

### 1.1. Многослойное представление корпоративной сети

Корпоративную сеть полезно рассматривать как сложную систему, состоящую из нескольких взаимодействующих слоев. В основании пирамиды, представляющей корпоративную сеть, лежит слой компьютеров - центров хранения и обработки информации, и транспортная подсистема (рис. 1.1), обеспечивающая надежную передачу информационных пакетов между компьютерами.

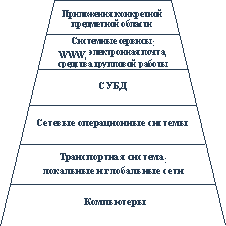


Рис. 1.1. Иерархия слоев корпоративной сети

Над транспортной системой работает слой сетевых операционных систем, который организует работу приложений в компьютерах и предоставляет через транспортную систему ресурсы своего компьютера в общее пользование.

Над операционной системой работают различные приложения, но из-за особой роли систем управления базами данных, хранящих в упорядоченном виде основную корпоративную информацию и производящих над ней базовые операции поиска, этот класс системных приложений обычно выделяют в отдельный слой корпоративной сети.

На следующем уровне работают системные сервисы, которые, пользуясь СУБД, как инструментом для поиска нужной информации среди миллионов и миллиардов байт, хранимых на дисках, предоставляют конечным пользователям эту информацию в удобной для принятия решения форме, а также выполняют некоторые общие для предприятий всех типов процедуры обработки информации. К этим сервисам относится служба WorldWideWeb, система электронной почты, системы коллективной работы и многие другие.

И, наконец, верхний уровень корпоративной сети представляют специальные программные системы, которые выполняют задачи, специфические для данного предприятия или предприятий данного типа. Примерами таких систем могут служить системы автоматизации банка, организации бухгалтерского учета, автоматизированного проектирования, управления технологическими процессами и т.п.

Конечная цель корпоративной сети воплощена в прикладных программах верхнего уровня, но для их успешной работы абсолютно необходимо, чтобы подсистемы других слоев четко выполняли свои функции.

Стратегические решения, как правило, влияют на облик сети в целом, затрагивая несколько слоев сетевой "пирамиды", хотя первоначально касаются только одного конкретного слоя или даже отдельной подсистемы этого слоя. Такое взаимное влияние продуктов и решений нужно обязательно учитывать при планировании технической политики развития сети, иначе можно столкнуться с необходимостью срочной и непредвиденной замены, например, сетевой технологии, из-за того, что новая прикладная программа испытывает острый дефицит пропускной способности для своего трафика.

### 1.2. Стратегические проблемы построения транспортной системы корпоративной сети

Из-за того, что транспортная система создает основу для взаимосвязанной работы отдельных компьютеров, ее часто отождествляют с самим понятием "корпоративная сеть", считая все остальные слои и компоненты сети просто надстройкой. В свою очередь, транспортная система корпоративной сети состоит из ряда подсистем и элементов. Наиболее крупными составляющими транспортной системы являются такие подсистемы как локальные и глобальные сети корпорации, опять же понимаемые как чисто транспортные средства. В свою очередь каждая локальная и глобальная сеть состоит из периферийных подсетей и магистрали, которая эти подсети связывает воедино. Например, крупная локальная сеть, приведенная на рисунке 1.2, состоит из подсетей, объединенных магистралью, включающих два кольца FDDI и четыре маршрутизатора. Каждая подсеть также может иметь иерархическую структуру, образованную своими маршуртизаторами, коммутаторами, концентраторами и сетевыми адаптерами. Все эти коммуникационные устройства связаны разветвленной кабельной системой.

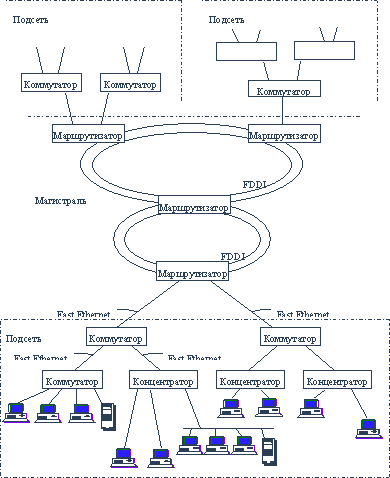


Рис. 1.2.Структура локальной сети

Глобальная сеть, объединяющая отдельные локальные сети, разбросанные по большой территории, также имеет, как правило, иерархическую структуру с высокоскоростной магистралью (например, АТМ), более медленными периферийными сетями (например, framerelay) и каналами доступа локальных сетей к глобальным. Эти составляющие глобальной сети представлены на рисунке 1.3.

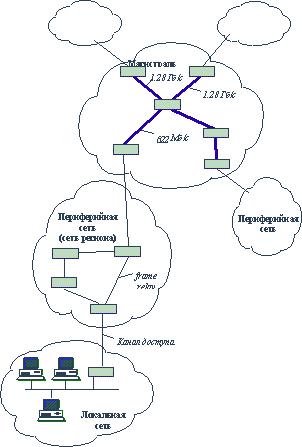


Рис. 1.3. Структура глобальной сети

При создании и модернизации транспортной системы стратегически значимыми сегодня являются в первую очередь следующие проблемы.

#### 1.2.1. Создание транспортной инфраструктуры с масштабируемой производительностью для сложных локальных сетей

Сегодня все чаще и чаще возникают повышенные требование к пропускной способности каналов между клиентами сети и серверами. Это происходит по разным причинам: из-за повышения производительности клиентских компьютеров, увеличения числа пользователей в сети, появления приложений, работающих с мультимедийной информацией, которая хранится в файлах очень больших размеров, увеличением числа сервисов, работающих в реальном масштабе времени. Особенно резко возросла нагрузка на серверы, которые публикуют корпоративные данные в Internet. Хотя такой трафик большую часть пути между сервером и клиентом проходит по глобальным каналам Internet, последний отрезок пути приходится на сегменты локальной сети предприятия, которые должны справляться с такой повышенной нагрузкой.

Требования к пропускной способности каналов связи к тому же очень неоднородны для различных сегментов и подсетей крупной локальной сети. Так как очень маловероятно, что все клиенты с одинаковой интенсивностью обмениваются данными со всеми серверами предприятия и внешними серверами, то часть сегментов загружена больше, а часть - меньше (рис. 1.4).

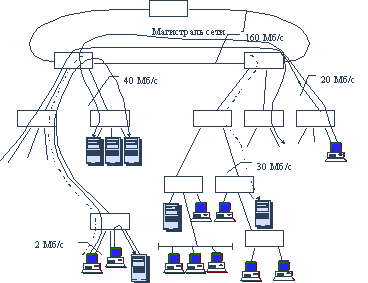


Рис. 1.4. Интенсивности потоков данных в разных сегментах локальной сети

Следовательно, имеется потребность в экономичном решении, предоставляющем сегментам и подсетям ту пропускную способность, которая им требуется.

Тем не менее, 10-Мегабитный Ethernet устраивал большинство пользователей на протяжении около 15 лет. Это объясняется тем, что пропускная способность в 10 Мб/с с большим запасом перекрывала потребности клиентских и серверных компьютеров сетей тех лет. До появления персональных компьютеров в локальную сеть объединялись миникомпьютеры, которых на предприятиях было не так уж и много, поэтому подсети включали по 5 - 20 компьютеров. Трафик состоял в основном из алфавитно-цифровых данных, интенсивность которых обычно не превышала нескольких десятков Кбайт в секунду для одного компьютера. Персональные компьютеры, массово появившиеся в середине 80-х, были весьма маломощными, с медленными дисками, и также не создавали проблем для 10-Мегабитных каналов.

Большая избыточность 10-Мегабитных каналов также не очень беспокоила специалистов, так как технология Ethernet была достаточно дешевой, коммуникационное оборудование сети состояло из одного-двух маршрутизаторов, коаксиального кабеля и сетевых адаптеров, стоимость которых была весьма небольшой по сравнению со стоимостью компьютеров, в которые они устанавливались.

Однако в начале 90-х годов начала ощущаться недостаточная пропускная способность каналов Ethernet. Для компьютеров на процессорах Intel 80286 или 80386 с шинами ISA (8 Мбайт/с) или EISA (32 Мбайт/с) пропускная способность сегмента Ethernet составляла 1/8 или 1/32 канала "память - диск", и это хорошо согласовывалось с соотношением объемов локальных данных и внешних данных для компьютера. Теперь же у мощных клиентских станций с процессами Pentium или PentiumPRO и шиной PCI (133 Мбайт/с) эта доля упала до 1/133, что явно недостаточно. Еще больший недостаток в пропускной способности стали ощущать серверы, как на основе RISC-, так и на основе Intel-процессоров. Основным решением в этой области стало использование нескольких сетевых адаптеров, работающих на разные подсети.

В начале 90-х годов наметились сдвиги и в характере передаваемой по сети информации. Наряду с алфавитно-цифровыми данными появились графические, звуковые и видеоданные, хранящиеся в многомегабайтных файлах. Это еще больше усугубило ситуацию, так как теперь даже несколько персональных компьютеров, работающих с мультимедийной информацией, могли перегрузить 10-Мегабитный сегмент сети.

Поэтому многие сегменты 10 Мегабитного Ethernet'а стали перегруженными, реакция серверов в них значительно упала, а частота возникновения коллизий существенно возросла, еще более снижая номинальную пропускную способность.

Самое простое решение - повышение битовой скорости единственного протокола, работающего во всех сегментах сети, как происходило ранее с сетями на основе Ethernet - не является уже рациональным для скоростей больших чем 30 - 40 Мб/с. Это стало ясно после разработки и применения первого высокоскоростного протокола локальных сетей - протокола FDDI, работающего на битовой скорости 100 Мб/с. Стоимость сегментов FDDI оказалась для этого слишком высокой, поэтому протокол FDDI стал применяться в основном только для построения магистралей крупных локальных сетей и подключения централизованных серверов предприятия. Для связи сегментов Ethernet с сегментами FDDI потребовалось применение маршрутизаторов или транслирующих коммутаторов.

Такая схема построения локальной сети, когда в ней существует несколько сегментов (в случае применения коммутаторов) или подсетей (в случае применения маршрутизаторов или умеющих маршрутизировать коммутаторов), в каждом из которых применяется один из двух протоколов в зависимости от той пропускной способности, которая нужна компьютерам, работающих в этой части сети, является прообразом схемы, к которой сегодня стремятся производители сетевого оборудования и сетевые интеграторы.

Более совершенная схема построения локальной сети должна опираться не на две доступные скорости, а на более дробную иерархическую линейку скоростей для компьютеров сети. Тогда можно будет более точно и с меньшими затратами учесть потребности каждой группы компьютеров, объединенных в сегмент, или даже каждого отдельного компьютера. Для согласования скоростей работы каналов между сегментами сети необходимо применять устройства, обрабатывающие трафик с буферизацией пакетов - коммутаторы или маршрутизаторы, но не концентраторы, которые организуют побитную передачу данных из сегмента в сегмент.

#### 1.2.2.Предоставление индивидуального качества обслуживания для различных типов трафика и различных приложений в локальных сетях

Можно пойти и дальше в детализации требований к пропускной способности. В конце концов пропускная способность каналов связи нужна не компьютеру в целом, а отдельным приложениям, которые выполняются на этом компьютере. У файлового сервиса одни требования к пропускной способности, у электронной почты - другие, а у сервиса интерактивных видеоконференций - третьи. Особенно остро эти различия стали ощущаться с начала 90-х годов, когда наряду с традиционным файловым сервисом и сервисом печати в локальных сетях стали использоваться новые виды сервисов, порождающих трафик реального времени, очень чувствительного к задержкам. Типичным представителем такого сервиса является компьютерная телефония. Каждый телефонный разговор двух абонентов порождает в сети трафик, имеющий постоянную битовую скорость, чаще всего 64 Кб/с (рис.1.5), когда источник голосовой информации порождает поток байт с частотой 8 КГц.

Более сложные методы кодирования могут уменьшить интенсивность голосового трафика до 9.6 Кб/с, и даже до 4 - 5 Кб/с.

Независимо от способа кодирования и интенсивности трафика, качество воспроизводимого на приемном конце голоса очень зависит от задержек поступления байт, несущих замеры амплитуды голоса. Вся техника передачи голоса в цифровой форме основана на том, что замеры должны поступать на воспроизводящее устройство через те же интервалы, через которые они производились на приемном устройстве, которое преобразовывало голос в последовательность чисел. Задержка поступления очередного байта более чем на 10 мс может привести к появлениям эффекта эха, большие задержки могут исказить тембр голоса до неузнаваемости или привести к затруднениям в распознавании слов. Компьютерные сети - как локальные, так и глобальные - это сети с коммутацией пакетов, в которых задержки передачи пакетов трудно предсказать. В силу самого способа буферизации пакетов в промежуточных коммутаторах и маршрутизаторах задержки в компьютерных сетях имеют переменный характер, так как пульсирующий характер файлового сервиса, Web-сервиса и многих других популярных компьютерных сервисов создают постоянно меняющуюся загрузку коммутаторов и маршрутизаторов.

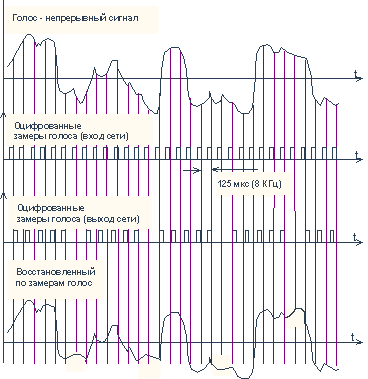


Рис. 1.5. Трафик, порождаемый в сети при передаче телефонного разговора

Особенно большие проблемы создает интерактивный обмен голосовыми сообщениями, проще говоря - обычный разговор. При передаче голоса только в одну сторону, например, при воспроизведении заранее записанной музыки, на приемном конце можно поставить буфер, в котором будут накапливаться неравномерно поступающие замеры звука, которые с некоторой задержкой затем будут извлекаться из буфера строго с частотой 8 КГц (рис.1.6).

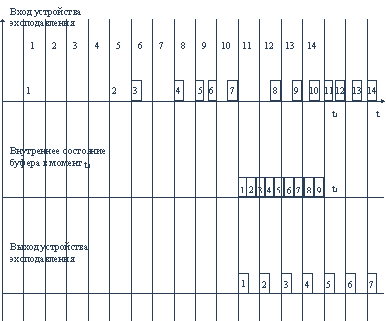


Рис. 1.6. Сглаживание неравномерности задержек, вносимых сетью

Такой буфер обычно называется устройством эхоподавления и используется в протяженных цифровых телефонных сетях. При интерактивном обмене постоянные значительные задержки, вносимые буфером в разговор, становятся очень неудобными для собеседников - приходится долго ждать ответа, как при разговоре с космонавтами.

Аналогичные требования к передаче своих данных предъявляет и трафик, переносящий видеоизображение. Трафик, требующий, чтобы его данные поступали к приемному узлу через строго определенные промежутки времени, называется *синхронным* в отличие от *асинхронного* трафика, мало чувствительного к задержкам его данных. Почти весь трафик традиционных сервисов компьютерных сетей является асинхронным - задержка передачи части файла даже на 200 мс будет мало заметна для пользователя.

Как правило, асинхронный и синхронный виды трафика существенно отличаются и еще в одном важном отношении - чувствительности к потере пакетов. Асинхронный трафик очень чувствителен к таким потерям, так как потеря даже небольшой части файла делает всю операцию передачи файла по сети бессмысленной - файл или его потерянную часть обязательно нужно передавать заново. Потеря же одного замера голоса или одного кадра изображения не очень заметно сказывается на качестве воспроизводимого сигнала, так как инерционный характер физических процессов приводит к тому, что два последовательных замера не очень отличаются друг от друга, поэтому воспроизводящему устройству не составляет труда восстановить, хотя и приблизительно, потерянную информацию.

Использование мультимедийной информации и интерактивных сервисов в компьютерных сетях создало сложную проблему совмещения двух очень разных по требованиям к характеру передачи пакетов через сеть типов данных. Сложности совмещения синхронного и асинхронного трафика в одной сети коммутации пакетов иллюстрирует рисунок 1.7.

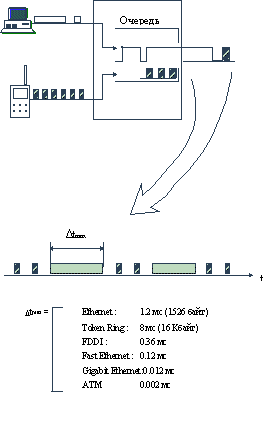


Рис. 1.7. Проблема совмещения синхронного и асинхронного трафика в одной сети с коммутацией пакетов

При передаче синхронных данных в обычных пакетах и кадрах локальной сети, такие пакеты будут встречаться во внутренних очередях коммутаторов и маршрутизатров с пакетами обычного асинхронного компьютерного трафика. Если коммутатор или маршрутизатор не делает различий между пакетами синхронного и асинхронного трафика, то задержки могут быть очень большими и очень неравномерными, особенно при загрузке коммуникационного устройства, близкой к его максимальным возможностям обработки пакетов. Естественным выходом из этой ситуации является приоритетная обработка пакетов синхронного трафика - и это очень распространенный прием, применяемый многими производителями коммутаторов, маршрутизаторов, а также разработчиками новых протоколов локальных сетей, например, протокола 100VG-AnyLAN, в котором существует два уровня приоритетов.

Однако, даже при приоритетной обработке пакеты синхронного трафика могут задерживаться в коммуникационных устройствах, так как в них можно реализовать только алгоритмы обработки с относительными приоритетами, а не с абсолютными. Это значит, что если низкоприоритетный пакет уже начал передаваться в сеть, то устройство не может прервать его передачу при приходе в это время высокоприоритетного пакета. Поэтому максимальное время ожидания синхронного пакета всегда равно времени передачи пакета максимальной длины, которую допускает тот или иной протокол или действующая конфигурация сети.

Так, для классического 10-Мегабитного Ethernet'а максимальный размер пакета равен 1526 байт (со всеми служебными полями и преамбулой). Значит, максимальное время его передачи составит 1.2 мс. Это не так много для большинства видов синхронного мультимедийного трафика. Хуже обстоят дела в сетях TokenRing, где кадры могут достигать размера в 16 Кбайт. При скорости в 16 Мб/с это может привести к задержке в 8 мс, уже оказывающей заметное влияние на качество голоса или изображения. Для сетей FDDI с битовой скоростью 100 Мб/с и максимальным размером кадра 4500 байт задержка составит всего 0.36 мс, для сетей Fast Ethernet - 0.12 мс, GigabitEthernet - 0.012 мс, а АТМ при скорости 155 Мб/c и размере ячейки в 53 байта - всего 2.7 мкс.

Однако, этот идеальный случай соответствует полностью свободной в любой момент времени среде передачи данных на выходном порту коммутатора или маршрутизатора. Такая ситуация встречается в локальных сетях не часто, так как микросегментация, когда каждый компьютер связан с сетью своей индивидуальной линией связи, пока еще слишком дорогое удовольствие для применения его во всех сегментах сети. При использовании в сегменте разделяемой среды высокоприоритетный пакет должен ждать не только завершения передачи низкоприоритетного пакета, но и освобождения разделяемой среды, а это составляющая является чисто случайной и с ней бороться можно только уменьшая загрузку сегмента.

Разделяемые среды передачи данных традиционно использовались в локальных сетях для уменьшения стоимости сетевого оборудования. Практически все протоколы локальных сетей - от Ethernet до 100VG-AnyLAN и GigabitEthernet (АТМ не относится к протоколам, разработанным для локальных сетей, эта технология в гораздо большей степени близка к технологиям передачи данных в глобальных сетях) могут работать на разделяемых средах передачи данных.

В разных протоколах локальных сетей реализованы разные методы доступа к разделяемой среде. В некоторых новых протоколах предусмотрен механизм приоритетного предоставления доступа к среде. Обычно, разработчики протокола ограничиваются двумя уровнями приоритетов - один, низкий, для асинхронного трафика, и второй, высокий, для синхронного. Так поступили разработчики протоколов FDDI и 100VG-AnyLAN. В протоколе TokenRing существует 8 уровней приоритетов, а во всех протоколах семейства Ethernet - FastEthernet - GigabitEthernet понятие приоритета кадра отсутствует. Безусловно, приоритетное предоставление доступа к разделяемой среде намного уменьшает задержки доставки пакетов к узлу назначения.

Однако, какой бы метод доступа к разделяемой среде передачи данных не использовался, возможна ситуация, когда несколько узлов с высоким приоритетом будут требовать одновременный доступ к разделяемой среде. Протоколы локальных сетей, даже самые последние, не решают задачу предоставления каких-то гарантий в распределении полосы пропускания общего канала передачи данных между равноприоритетными узлами. И, если все узлы сети будут иметь равные приоритеты, то разделяемая среда по-прежнему будет предоставлять каждому узлу заранее неизвестную часть своей пропускной способности.

Обеспечение для абонентов сети требуемого уровня задержек - это частный случай обеспечения нужного *качества обслуживания - QualityofService, QoS.* Анализ типов трафика, создаваемого современными приложениями, позволил выделить несколько основных типов, для которых понятие QoS имеет различный смысл и характеризуется различными параметрами.

Трафик реального времени с постоянной битовой скоростью обычно требует предоставления ему постоянной полосы пропускания, причем в понятие качества обслуживания входит не только величина предоставляемой ему пропускной способности, но и величина задержек передачи каждого пакета - обычно среднее время задержки и величина ее вариации. Типичный представитель этого типа трафика - голосовой трафик.

Существует также тип трафика реального времени, создающий поток данных с пульсирующей битовой скоростью, но так же чувствительный к задержкам передачи пакетов. Такой трафик создают источники, выполняющие компрессию голоса или видеоизображения, когда, например, при неизменной картинке интенсивность потока данных резко уменьшается. Для такого трафика в понятие качества обслуживание по прежнему входят средняя величина и вариация задержек, а вместо одного параметра пропускной способности обычно от сети требуется обеспечить два - среднюю скорость передачи данных и передачу всплеска трафика до определенной величины в течение оговоренного периода времени.

Для пульсирующего компьютерного трафика, который не является трафиком реального времени, так как нечувствителен к задержкам, обычно достаточно обеспечить аналогичные предыдущему случаю параметры пропускной способности, а о величинах задержек не заботиться.

Для случая, когда трудно точно оценить среднюю скорость передачи данных приложением и максимальный всплеск интенсивности, применяют упрощенное толкование понятия качества обслуживания - как верхний и нижний пределы пропускной способности, предоставляемой сетью абоненту в течение достаточно длительного промежутка времени.

В предыдущих примерах качество обслуживания трактовалось только относительно временных характеристик работы сети. Однако, вероятность успешной доставки данных абоненту также, естественно, может учитываться в качестве обслуживания. Для многих видов компьютерного сервиса, где потери пакетов ведут к существенному снижению полезной пропускной способности сети, надежность доставки пакета - существенная составляющая качественного обслуживания абонента сетью.

До сих пор мы больше говорили о предоставлении определенного уровня качества обслуживания узлам сети, то есть компьютерам в целом. Однако, на самом деле источником трафика с определенными требованиями к качеству его обслуживания является не компьютер, а отдельное приложение, работающее на этом компьютере. Вполне реальной является ситуации когда на одном компьютере одновременно в режиме разделения времени выполняются несколько приложений и у каждого имеются свои требования к передаче его данных через сеть. Большинство современные ОС поддерживают режим мультипрограммирования, так что сосуществование фонового приложения рассылки электронной почты или факсов с сессией видеоконференции вполне возможно. Поэтому современная сеть должна допускать обслуживание с разными классами качества и с разными параметрами качества приложений одного и того же компьютера.

**Совмещенная передача голоса и данных и гарантированное качество обслуживания в глобальных сетях**

В глобальных сетях проблема совмещения голоса и данных, или, в более широкой постановке задачи, обеспечение гарантированного качества обслуживания для различных классов трафика стоит еще более остро. Это объясняется тем, что глобальные каналы связи существенно дороже локальных, поэтому гораздо сильнее стимулы для использования одной и той же транспортной инфраструктуры для передачи компьютерного трафика и трафика, который обычно передается через телефонные сети.

Попытки обеспечить приемлемое качество обслуживания для голосового и компьютерного трафика делались в территориальных сетях уже давно - практически с самого начала внедрения глобальных компьютерных сетей. Компьютерный трафик при отсутствие специальных каналов связи передавали по телефонным каналам с помощью модемов. Телефонные сети всегда работали по технологии коммутации каналов, поэтому проблема задержек голосовых данных длинными компьютерными пакетами здесь не возникала - после коммутации составной канал оказывался полностью в распоряжении либо компьютерного, либо голосового трафика.

Однако, при этом определенные неудобства испытывали компьютерные абоненты сети - канал с постоянной пропускной способностью не может хорошо передавать пульсации трафика. Если нужно передать трафик со средней интенсивностью 10 Кб/с и пульсацией до 500 Кб/с на протяжении одной секунды, то, очевидно, что канал с пропускной способностью 28.8 Кб/с не сможет хорошо справиться с этой задачей. Пакеты, принадлежащие периоду всплеска трафика, будут ждать в очереди, которая образуется на входе такого канала. В то же время в периоды трафика низкой интенсивности (а они, безусловно, будут иметь место, так как средняя интенсивность трафика всего 10 Кб/c) канал будет использоваться всего на небольшую долю своей пропускной способности, а так как в сетях с коммутацией каналов оплата всегда осуществляется на повременной основе, то и платить компьютерные абоненты всегда будут не только за полезную пропускную способность канала, но и за неиспользуемую часть времени его работы.

Такое положение дел всегда сохраняется при использовании сетей с коммутацией каналов, в том числе и сетей ISDN. Сети ISDN изначально проектировались как сети с интегральными услугами, в которых компьютерный трафик должен передаваться наравне с телефонным, трафиком факсов, службы телетекста и трафиками других служб. Однако первая попытка построения интегральной территориальной сети удалась далеко не в полной мере. Сервис коммутации пакетов, так нужный для качественной и экономной передачи пульсаций трафика, оказался в этих сетях пасынком. Только немногие провайдеры сетей ISDN предоставляют такой вид услуг своим абонентам, да и то на медленных каналах типа D в 16 Кб/с или 64 Кб/с, а такие скорости вряд ли удовлетворят пользователей современных корпоративных сетей. Поэтому для передачи компьютерного трафика через сети ISDN используется сервис коммутации каналов со скоростью до 2 Мб/с, а значит все проблемы с передачей пульсаций остаются.

При использовании же для передачи голосового и других видов трафика реального времени сетей, разработанных как чисто компьютерные, пользователи сталкиваются с той же проблемой неравномерных и значительных задержек пакетов с мультимедийными данными, которая присуща и локальным сетям. При более низких скоростях передачи данных задержки могут быть достаточно чувствительными. Даже в ненагруженной сети framerelay при скорости передачи данных по каналу в 1.5 Мб/с передача пакета компьютерных данных длиной 4096 байт может задержать пакет голосовых данных на 22 мс, что скорее всего очень сильно снизит качество передачи голоса.

Большие размеры пакетов, которые выгодны для передачи компьютерных данных, так как увеличивают полезную пропускную способность канала за счет снижения доли служебных данных в заголовках, вредно влияют на качество передачи трафика реального времени не только за счет задержек в промежуточных узлах, то есть коммутаторах и маршрутизаторах. Большое влияние на качество обслуживания может оказывать так называемая задержка пакетизации, которая тем больше, чем больше размер пакета, используемого для передачи мультимедийных данных. Механизм возникновения задержки пакетизации иллюстрирует рисунок 1.8.

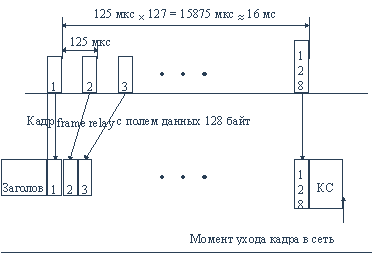


Рис. 1.8. Задержка пакетизации голосовых данных при передаче через сети коммутации пакетов

Пусть мы хотим использовать для передачи голоса сеть framerelay с максимальным размером пакета 4096 байт. Оцифрованные замеры голоса поступают на вход устройства доступа к глобальной сети - FrameRelayAccessDevice, FRAD, с частотой 8 КГц. FRAD упаковывает каждый байт в пакет, при этом первый байт, попавший в какой-либо пакет, должен ждать отправки в сеть 4095 интервалов по 125 мкс (период следования байт при частоте 8 КГц), пока пакет на заполнится полностью. Эта задержка и называется задержкой пакетизации, в данном случае она составит 511 мс, то есть полсекунды, что совершенно недопустимо. Поэтому обычно FRAD настраивается на отправку в сеть голосовых данных в пакетах гораздо меньшей длины, например, 128 байт, но и при этом задержка составит порядка 16 мс и для ее компенсации нужно устройство эхоподавления на приемном конце.

Из-за задержек пакетизации в сетях с коммутацией пакетов при необходимости совмещать передачу голоса и данных применяют пакеты небольших размеров, которые также уменьшают и задержки ожидания в коммутаторах сети. Однако, при этом уменьшается полезная пропускная способность сети для компьютерных данных.

В глобальных сетях обычно не ставится задача поддержки качества обслуживания для всех возможных типов трафика, которые были рассмотрены выше. Чаще всего, глобальная сеть считается очень хорошей, если она может дифференцированно обслуживать по крайней мере два вида трафика, голосовой и компьютерный, прием с очень упрощенной поддержкой качества обслуживания для каждого типа.

Более тонкая поддержка качества обслуживания для всех основных типов трафика обеспечивается сегодня только в сетях АТМ, и то часто потенциально, так как не все провайдеры АТМ предлагают абонентам своих сетей все возможные способы поддержки качества обслуживания, определенные в стандарте технологии АТМ.

#### 1.2.3. Выбор технологии магистрали для крупных локальных сетей предприятия

Технология определяется используемыми протоколами нижнего уровня, такими как Ethernet, TokenRing, FDDI, FastEthernet и т.п. и существенно влияет на типы используемого в сети коммуникационного оборудования. Магистраль - это одна из наиболее дорогостоящих частей любой сети. Кроме того, так как через нее проходит значительная часть трафика сети, то ее свойства сказываются практически на всех сервисах корпоративной сети, которыми пользуются конечные пользователи. Поэтому решение о технологии работы магистрали явно относится к разряду стратегических решений.

Кроме протокола, который будет работать на магистрали, необходимо также выбрать рациональную структуру магистрали. Эта структура будет затем положена в основу структуры кабельной системы, стоимость которой может составлять 15% и более процентов всей стоимости сети. Рациональная структура магистрали должна обеспечить компромисс между качеством передачи трафика (пропускная способность, задержки, приоритеты для ответственных приложений) и стоимостью. На структуру магистрали сильнейшее влияние оказывает выбранная технология, так как она определяет максимальные длины кабелей, возможность использования резервных связей, типы кабелей и т.п. Так как магистраль крупной сети строится практически всегда на основе активного коммуникационного оборудования - коммутаторов и маршрутизаторов - фильтрующего и перераспределяющего трафик между подсетями, то в понятие рациональной структуры входит и выбор активного оборудования. При этом вопрос состоит не столько в выборе определенной модели оборудования от определенного производителя, а в основном в выборе типа оборудования - маршрутизатор, коммутатор - и режима работы этого оборудования по объединению подсетей и установлению барьеров от нежелательного межсетевого трафика.

Сегодня существует несколько режимов работы маршрутизаторов и коммутаторов, отличающихся от стандартных: образование виртуальных сетей коммутаторами, ускоренная маршрутизация для долговременных потоков данных (IPswitching, tagswitching и т.п.), спуффинг широковещательного трафика и некоторые другие. Пока что эти режимы, часто весьма полезные для работы на магистралях современных сетей, каждый производитель реализует по-своему, хотя работы по стандартизации идут, и некоторые приемы и алгоритмы уже близки к тому, чтобы обрести свое стандартное выражение.

**1.2.4. Организация высокоскоростного и экономичного доступа удаленных пользователей и сетей филиалов к центральной сети предприятия**

Организация удаленного доступа сотрудников предприятия к информационным ресурсам, сосредоточенным в центральных базах данных компьютеров корпоративной сети, перешла в последнее время в разряд вопросов стратегически важных для большинства предприятий. Быстрый доступ к корпоративной информации из любой географической точки определяет для многих видов деятельности качество принятия решений его сотрудниками. Важность этого фактора растет с увеличением числа сотрудников, работающих на дому (telecommuters - телекоммьютеров), сотрудников, часто находящихся в командировках, и с ростом количества небольших филиалов предприятий, находящихся в различных городах и, может быть, разных странах.

Количество сотрудников предприятий, которым нужен регулярный компьютерный доступ к корпоративной сети, с каждым годом увеличивается. Так, по данным нью-йоркской исследовательской компании FIND/SVP, количество телекоммьютеров в 1995 году во всем мире составило 9.1 миллионов человек.

В качестве отдельных удаленных узлов могут также выступать банкоматы или кассовые аппараты, требующие доступ к центральной базе данных о легальных клиентах банка, пластиковые карточки которых необходимо авторизовать на месте. Банкоматы или кассовые аппараты обычно рассчитаны на взаимодействие с центральным компьютером по сети Х.25, которая в свое время специально разрабатывалась как сеть для удаленного доступа неинтеллектуального терминального оборудования к центральному компьютеру.

Качественный скачок в расширении возможностей удаленного доступа произошел в связи со стремительным ростом популярности и распространенности сети Internet. Транспортные услуги Internet дешевле, чем услуги междугородных и международных телефонных сетей, а их качество быстро улучшается. Кроме транспортных услуг, Internet предоставляет средствам удаленного доступа единую технологию доступа к корпоративной информации, основанную на технологии Web-серверов и Web-броузеров и названную технологией Intranet. Технология Intranet как единый для всех типов сетей и операционных систем стандарт, удешевляет развертывание систем удаленного доступа, что в свою очередь, дает дополнительный стимул для широкого их использования на предприятиях.

Подтверждением роста популярности средств удаленного доступа является быстрый рост доходов, получаемых в этом секторе рынка.

Анализ мирового рынка коммуникационного оборудования и услуг, проведенный редакцией журнала "DataCommunicationsInternational" совместно с ведущими исследовательскими компаниями (IDC, Dell'OroGroup, IDG, YankeyGroup и др.) показал, что в 1996 году рост доходов от продаж оборудования удаленного доступа составил 151% (в абсолютном исчислении доходы этого сектора составили 2.157 миллиарда долларов), что уступает темпам роста только коммутаторов локальных сетей (216%) и высокоскоростных сетевых адаптеров (160%). Прогноз на 1997 год также очень благоприятен - ожидается дальнейший рост продаж аппаратных средств удаленного доступа на 100% при достижении абсолютной цифры доходов в 4.308 миллиардов долларов.

Ввиду массовости клиентов, пользующихся сервисом удаленного доступа, основным видом телекоммуникационного транспорта, подходящего для этих целей остаются телефонные сети - как аналоговые, так и ISDN. Для быстрой передачи данных сети ISDN подходят в гораздо большей степени, чем узкополосные и зашумленные каналы аналоговых сетей. В секторе услуг ISDN также имеется устойчивый рост. В 1996 году он составил 102% (доходы от продажи сервисов ISDNBRI составили 1.230 миллиардов долларов), и прогнозируется рост на 99% в 1997 году.

Чуть меньшие темпы роста зафиксированы в 1996 году в секторе коммерческих услуг Internet - 78% (при абсолютной цифре доходов в 2.395 миллиардов долларов), но в 1997 году ожидается их повышение до 98%.

Сегодня разработчики средств и систем удаленного доступа преследуют следующие стратегические цели:

* Повышение скорости доступа для домашних и мобильных пользователей. Скорости модемов, работающих по коммутируемым телефонным каналам, сейчас для многих видов приложений уже оказывается недостаточным. Максимальная скорость модема последнего стандарта V.34+ составляет 33.6 Кб/c и то только в случае очень хорошего качества телефонного канала. В то же время считается, что такой популярный для удаленного доступа сервис, как WWW, требует в среднем скорости 64 Кб/c или даже 128 Кб/с. Для телекоммьютеров могут потребоваться и более высокие скорости доступа, если они используют корпоративные приложения, перекачивающие к клиенту значительные объемы данных. Поэтому остро стоит проблема доведения высокоскоростных каналов 125 Кб/с - 10 Мб/с до каждого здания и каждой квартиры по крайней мере в крупных городах. Телефонные сети ISDN - хорошее решение этой проблемы, но не долговременное, так как их скорость доступа для массовых абонентов ограничена порогом в 128 Кб/с. Большие надежды специалисты по удаленному доступу возлагают на новые технологии "последней мили", использующие существующую инфраструктуру каналов связи квартир с АТС или центрами кабельного телевидения для несимметричной высокоскоростной передачи компьютерных данных.
* Создание интегрированных серверов удаленного доступа, способных принимать данные от большого числа пользователей по нескольким высокоскоростным каналам. При большом числе пользователей сервер удаленного доступа, построенный по обычной схеме пула аналоговых модемов становится слишком сложным в обслуживании - слишком большим становится необходимое число модемов, кабелей, кроссовых средств, телефонных номеров и т.п. Интегрированный сервер должен одновременно обслуживать несколько сотен соединений по таким высокоскоростным каналам как T1/E1, ISDNPRI или SONET/SDH.
* Создание централизованной системы аутентификации удаленных корпоративных пользователей, взаимодействующей с системой аутентификации провайдеров территориальных сетей (POTS, ISDN) и с системами аутентификации и авторизации сетевых ОС - NDS, Kerberos и т.п.

Конечно, кроме перечисленных, существуют и другие задачи, которые могут быть отнесены к стратегическим для транспортной системы корпоративной сети того или иного предприятия.

### 1.3. Стратегические проблемы выбора сетевой операционной системы и СУБД

При принятии стратегического решения относительно используемых в корпоративной сети сетевых операционных систем, необходимо учитывать, что все сетевые ОС делятся по своим функциональным возможностям на два четко различимых класса: сетевые ОС масштаба отдела и корпоративные сетевые ОС.

При выборе корпоративной сетевой ОС в первую очередь нужно учитывать следующие критерии:

* *Масштабируемая в широких пределах производительность*, основанная на хорошей поддержке многопроцессорных и кластерных платформ (здесь сегодня лидерами являются фирменные версии Unix, показывающие рост производительности близкий к линейному при росте числа процессоров до 64).
* *Возможность использования данной ОС в качестве сервера приложений*. Для этого ОС должна поддерживать несколько популярных универсальных API, таких, которые позволяли бы, например, выполняться в среде этой ОС приложениям Unix, Windows, MSDOS, OS/2. Эти приложения должны выполняться эффективно, а это означает, что данная ОС должна поддерживать многонитевую обработку, вытесняющую многозадачность, мультипроцессирование и виртуальную память.
* *Наличие мощной централизованной справочной службы* (такой, например, как NDS компании Novell или StreetTalk компании Banyan). Справочная служба должна обладать масштабируемостью, то есть хорошо работать при очень большом числе пользователей и разделяемых ресурсов, а для этого необходимо, чтобы база справочных данных была распределенной. Нужно учитывать, что справочные службы, также как и многие другие сетевые сервисы, сейчас часто поставляются не встроенными в конкретную ОС, а в виде отдельного продукта, например, StreetTalkforWindowsNT (компания Novell планирует выпуск NDS для WindowsNT).

И, хотя существует еще ряд не менее важных характеристик, которые надо учитывать при выборе сетевой ОС, таких, например, как степень стабильности и безопасности ОС, наличие программных средств удаленного доступа, способность работать в гетерогенной среде и т.д., реальная жизнь упрощает задачу выбора. Сегодня рынок корпоративных ОС поделен между несколькими операционными системами: примерно по одной трети имеют NetWare и WindowsNT, 10% приходится на разные версии Unix и 20% представлены остальными типами ОС.

Похожая ситуация складывается и на рынке СУБД. Число явных лидеров не так велико, если рассматривать наиболее распространенные классы компьютерных платформ - RISC-серверы и RISC-рабочие станции, а также многочисленную армию серверов и рабочих станций на платформе процессоров Intel. Однако, более тонкий подбор подходящей СУБД и ее версии для используемых на предприятии прикладных задач и технологий хранения и обработки данных требует знания основных сегодняшних свойств каждой СУБД и представления о том, какие новые свойства, желательные для вашей сети, можно ожидать от данной СУБД в ближайшем будущем.

### 1.4. Стратегические проблемы создания корпоративных приложений

Для слоя приложений чаще всего важен выбор не самого приложения, а той технологии, в соответствии с которой приложение создается. Это связано с тем, что большая часть приложений создается силами сотрудников предприятия или же силами сторонней организации, но по конкретному техническому заданию для этого предприятия. Случаи использования готовых крупных приложений, настраиваемых на потребности данного предприятия, например SAPR/3, более редки по сравнению с созданием специальных приложений. Специальные приложения часто модифицируются, добавляются, снимаются с работы, поэтому важно, чтобы технология их создания допускала быструю разработку (например, на основе объектного подхода) и быстрое внесение изменений при возникновении такой необходимости. Кроме того, важно, чтобы технология позволяла строить распределенные системы обработки информации, использующие все возможности транспортной подсистемы современной корпоративной сети.

Технология Intranet удовлетворяет этим требованиям, являясь одновременно и самой перспективной технологией создания приложений на ближайшие несколько лет. Однако, и при выборе Intranet для создания корпоративных приложений, остается немало проблем, которые можно отнести к стратегическим, так как существует несколько вариантов реализации этой технологии - вариант Microsoft, варианты Sun, IBM, Netscape и другие.

В конечном итоге свойства приложений определяют требования, предъявляемые к остальным слоям и подсистемам корпоративной сети. Объемы хранимой информации, их распределение по сети, тип и интенсивность трафика - все эти параметры, влияющие на выбор СУБД, операционной системы и коммуникационного оборудования и т.п. являются следствием того, какие приложения работают в сети. Поэтому знание свойств приложений и их сознательное формирование разработчиком корпоративной сети позволяют более рационально планировать развитие остальных ее слоев.

### 1.5. Защита корпоративной информации при использовании публичных глобальных сетей (в том числе и Internet)

Корпоративные сети более подвержены вторжениям, чем локальные сети меньшего масштаба или централизованные информационные системы предприятия, построенные на базе мэйнфреймов. Все главные особенности корпоративной сети обуславливают повышенную опасность этого типа сетей.

И первой такой особенностью является наличие глобальных связей. По своей природе глобальные связи, простирающиеся на много десятков и тысяч километров, не позволяют воспрепятствовать злонамеренному доступу к передаваемым по этим линиям связи данным. Нельзя дать никаких гарантий, что в некоторой, недоступной для контроля точке пространства, некто, используя, например, анализатор протокола, не подключится к передающей среде для захвата и последующего декодирования пакетов данных. Такая опасность одинакова присуща всем видам территориальных каналов связи и не связана с тем, используются ли собственные, арендуемые каналы связи или услуги общедоступных территориальных сетей, подобные Internet.

Однако использование общественных сетей (речь в основном идет об Internet) еще более усугубляет ситуацию, хотя бы потому, что в такой сети для доступа к корпоративным данным в распоряжении злоумышленника имеются более разнообразные и удобные средства, чем выход в чистое поле с анализатором протоколов. Кроме того, огромное число пользователей увеличивают вероятность попыток несанкционированного доступа.

Другая особенность корпоративной сети - масштабность, она заключается в том, что имеется очень большое количество рабочих станций, серверов, пользователей, мест хранения данных и т.п. В таких условиях администратору оказывается гораздо труднее построить надежную защиту сети, предусматривающую адекватную реакцию на все возможные попытки взлома системы.

Гетерогенность - это еще одна особенность корпоративной сети, которая усложняет работу администратора по обеспечению ее безопасности. Действительно, в программно и аппаратно неоднородной среде гораздо сложнее проверить согласованность конфигурации разных компонентов и осуществлять централизованное управление. К тому же надо учесть, что в большой гетерогенной сети резко возрастает вероятность ошибок как пользователей, так и администраторов.

По сравнению с сетями масштаба отдела или небольшого предприятия обеспечение безопасности в корпоративной сети является задачей не только более сложной, но и более важной: огромные материальные потери, к которым может привести доступность некоторых данных для заинтересованных в этом людей, переводит безопасность из разряда чисто технических вопросов в разряд самых приоритетных бизнес-проблем.

Поэтому защита данных в корпоративной сети в любом случае является стратегически важной задачей, а при использовании публичных территориальных сетей ее важность увеличивается многократно.

Проблема защиты данных при передачи их через публичные сети осложняется и тем обстоятельством, что во многих странах правительства вводят ограничения на использование основных средств защиты данных, а именно, средств их шифрации. Практически любой метод защиты данных основан на том или ином виде шифрации. Правительственные ограничения на использование средств шифрации преследуют несколько целей:

* предотвращение утечек государственных секретов, которые могут иметь место при использовании в государственных учреждениях непроверенных средств шифрации данных при отправке их в публичные сети (телефонные или компьютерные);
* возможность расшифровки данных, пересылаемых лицами или организациями, подозреваемыми в преступных действиях;
* защита отечественных производителей средств шифрации;
* контроль за рынком средств шифрации.

Правительственные ограничения особенно осложняют решение задачи защиты корпоративных данных при создании интернациональных корпоративных сетей, так как на одном конце сети могут действовать одни ограничения, диктующие использование определенных средств шифрации, а на другом - другие.

Повсеместное распространение сетевых продуктов массового потребления, имеющих встроенные средства защиты данных, например, сетевых операционных систем WindowsNT и Windows 95 c протоколом защиты данных PPTP , с одной стороны упрощает защиту данных, а с другой стороны часто создает только видимость надежной защиты. Эта видимость - следствие того, что в соответствии с ограничениями правительства США допускается экспорт продуктов шифрации только с ключами длиной до 40 бит. Поэтому, внутри США используются те же операционные системы или другие продукты с ключами 56и бит и выше, а за пределы страны американские продукты поставляют версии с усеченными возможностями. В то же время мощности компьютеров, в том числе и персональных, выросли настолько, что расшифровать сообщение, зашифрованное с помощью 40-битного ключа, можно за один день, даже не имея в распоряжении мощных суперкомпьютеров (и такие случаи зафиксированы).

Поэтому, использование для защиты данных тех средств, которые имеются под рукой, защищает данные только от просто любопытствующих людей. Заинтересованный в прочтении этих данных человек может достаточно легко их расшифровать, потратив на это не так уж много сил и средств.

Подобная ситуация складывается не только с недорогими встроенными средствами защиты, так как большая часть профессиональных и достаточно дорогих систем также производится в США и также подвержена правительственным ограничениям. Выход - использование средств защиты, произведенных или адаптированных в России или же в европейских странах, в которых нет аналогичных ограничений на экспорт.

Надежная шифрация - не единственная проблема, возникающая при защите корпоративных данных. Достаточно сложно решить и проблему надежной аутентификации пользователей.

*Аутентификация* - это получение уверенности в том, что данный пользователь является тем индивидуумом, за кого себя выдает. Использование средств удаленного доступа к корпоративной сети существенно усложняет эту задачу. При аутентификации пользователей локальной сети успешно решить эту задачу помогают организационные меры - отсечение посторонних пользователей от клиентских компьютеров и терминалов, контроль за подключениями к кабельной системе здания и т.п. При удаленном доступе эти средства не работают, а пароли, передаваемые легальными пользователями по сети в открытом виде по публичной сети, могут быть перехвачены и использованы впоследствии нелегальными пользователями. Даже при использовании более сложных схем аутентификации, не передающих пароли по сети, в схеме аутентификации имеется уязвимое звено - процедура передачи удаленному пользователю его пароля. Хотя эта процедура, в отличие от процедуры входа в сеть, выполняется редко, использование для нее электронных средств коммуникаций или обычной почты не дает хороших гарантий от перехвата пароля.

Новые проблемы создает проблема аутентификации пользователей при ведении бизнеса через Internet. Число пользователей вырастает настолько, что количество переходит в качество, и старые методы аутентификации на основе индивидуальных паролей начинают работать плохо - слишком большой объем работы падает на администратора, раздающего пароли, и средства аутентификации, эти пароли проверяющие. Нужны новые механизмы идентификации категорий пользователей, например, при издании журнала через Internet, нужно различать: пользователей, вообще не оформивших подписку, которым нужно предоставлять доступ только к небольшому списку избранных статей, рекламирующих журнал; пользователей, оформивших подписку только на определенную рубрику журнала; пользователей, оформивших полную подписку. Кроме этого, нужно проверять срок окончания действия подписки.

Ведение бизнеса через Internet выдвигает на первый план и такую проблему, как обеспечение целостности переданных через сеть данных. Пользователь, купивший через Internet новую программу, должен быть уверен, что он получил оригинальную копию этой программу, а не подделку от нечестного продавца.

### 1.6. Создание интегрированной системы управления

Корпоративная сеть - это очень сложная система, включающая большое количество систем разного типа и назначения: от кабельной системы до программной системы коллективной работы. Кроме того, в сети обычно сосуществуют элементы одного типа, но разных производителей, например, маршрутизаторы Cisco и BayNetworks, СУБД Oracle и DB2 и т.п.

Из-за этого очень трудно создать единую интегрированную систему управления корпоративной сетью, которая бы контролировала все процессы и все системы сети (сеть здесь понимается в широком смысле, как объединение всех слоев - от компьютеров до приложений). Тем не менее потребность в такой системе управления имеется, так как все подсистемы сети тесно взаимосвязаны и при управлении коммуникационным оборудованием могут потребоваться данные об установленных в операционных системах приложениях или о конфигурационных параметрах коммуникационных протоколов в стеках сетевых операционных систем.

Идеальная система управления корпоративной сетью должна состоять из нескольких подсистем, каждая из которых контролирует и управляет своим типом оборудования или программных систем: подсистема управления локальной сетью, подсистема управления глобальной сетью, подсистема управления компьютерами и сетевыми операционными системами, подсистема управления СУБД, подсистема управления приложениями. Все эти подсистемы должны быть интегрированы общей платформой управления, поддерживающей удобный графический интерфейс с администраторами сети и обеспечивающей общую базу данных управляющей информации, а также обмен данными между подсистемами.

К сожалению, реальная ситуация в этой области далека от идеала. Даже для более узкой области, а именно для транспортной системы, сейчас отсутствует система управления, которая бы хорошо справлялась с контролем и управлением разнородным коммуникационным оборудованием локальных и глобальных сетей. Работы по созданию интегрированных систем управления ведутся, но даже лучшие образцы пока справляются с поддержкой вполне определенного вида оборудования, например, только одного или максимум нескольких производителей, а при управлении оборудованием другого типа могут выполнять только самые примитивные функции.

Сами по себе существуют и так называемые системы управления системами, область действия которых ограничена операционными системами. Еще один обособленный вид систем управления - системы администрирования СУБД, не говоря уже о специальных приложениях.

Задача интеграции разрозненных систем - это задача номер один в такой важной для больших корпоративных сетей области как централизованное управление. Второй по важности задачей является создание масштабируемой системы управления, которая может управлять сетью любого масштаба.

### 1.7. Планирование этапов и способов внедрения новых технологий в существующие сети

Важно не только принять стратегически верное решение, но и правильно внедрить его в существующую сеть. Так как это решение долговременное, то оно совсем не обязательно одномоментно должно найти свое воплощение в новых программных или аппаратных средствах сети. Например, внедрение технологии Intranet не означает быстрый отказ от всех приложений другого типа. Возможность поэтапного и как можно менее болезненного способа постепенного перехода на новый продукт или новую технологию - это тоже обязательное свойство хорошего стратегического решения. Если же новое решение технически очень привлекательно, но путей его постепенного внедрения в существующую сеть нет, то от него лучше отказаться. Примером может служить технология АТМ до разработки таких стандартов как LANEmulation или ClassicalIP. Красивое с технической точки зрения решение требовало полной замены всего коммуникационного оборудования локальной сети и поэтому не находило применения до тех пор, пока на появились коммутаторы АТМ, которые за счет реализации в них клиентов и серверов LANEmulation могут теперь без проблем взаимодействовать с традиционными сетями Ethernet или TokenRing.

Обычно процесс внедрения нового продукта или новой технологии в сеть разбивается на несколько этапов. На первом этапе в сети появляется небольшая подсеть или даже несколько компьютеров, работающих по-новому. На этом этапе специалисты, обслуживающие сеть, и пользователи знакомятся с принципиальными свойствами нового подхода и оценивают возможность сосуществования его с остальной частью сети. При положительной оценке новинки ее область применения постепенно расширяется, захватывая новые подсети, серверы или другие компоненты сети. Постепенность внедрения позволяет вовремя отказаться от новинки, не затратив при этом вхолостую большие средства.

### 1.8. Выбор интеграторов, производителей и поставщиков программных и аппаратных продуктов, провайдеров услуг территориальных сетей

При внедрении в сеть принципиально новых технологий или продуктов желательно привлечение внешних организаций, уже имеющих опыт работы с этими технологиями и продуктами. В такой ситуации слишком рискованно уповать только на свои собственные силы и осваивать все с нуля. Правильный выбор соисполнителей работ по модернизации корпоративной сети также является необходимой компонентой стратегического планирования сети.

Вариантов привлечения сил сторонних организаций может быть несколько. Новая технология или новый продукт - это плод трудов какой-нибудь компании-производителя. Наиболее верный путь получения хорошего конечного результата состоит в получении консультаций или даже выполнения внедренческих работ специалистами компании-производителя. Однако такие услуги обычно предоставляются только достаточно крупным и интересным заказчикам, так как специалисты таких компаний как, например, Oracle или BayNetworks, заняты в основном разработкой и не могут тратить слишком много сил на внедрение своих продуктов. Тем не менее практика привлечения такого сорта специалистов существует и, если это возможно, то ей желательно пользоваться.

Гораздо более распространенным является привлечение специалистов фирм, основной специализацией которых является системная или сетевая (транспортная) интеграция. В этом случае нужно быть уверенным, что специалисты этой фирмы действительно хорошо знают продукты, которые внедряют.

Выбор производителя нового продукта определяется многими факторами. Обязательными требованиями при выборе производителя стратегически важного продукта или технологии являются стабильность его технической репутации и устойчивость финансового положения. Почти беспроигрышным является приобретение продуктов у признанных лидеров определенного сектора рынка, например, Oracle, Cisco, Netscape, Sun и т.п. Часто хорошие новинки появляются у малоизвестных компаний, но через некоторое время лидеры обязательно применяют эти новинки в своих продуктах, так что ставка на лидера и в этих случаях оказывается правильной, так как небольшой инкубационный период позволяет определить качество и перспективность нового решения. Примером может служить новая технология IPswitching, которую компания Ipsilon применила для ускоренной передачи IP-пакетов через магистрали АТМ. Через полгода компания Cisco разработала аналогичную технологию tagswitching, внеся в исходную идею некоторые усовершенствования. Единственным недостатком ставки на лидеров является более высокая стоимость их продуктов по сравнению с компаниями второго эшелона.

### 1.9. Обучение и набор персонала

Для того, чтобы новое решение прижилось в корпоративной сети и заработало так, как планировали его разработчики, необходимо, чтобы обслуживающий сеть персонал хорошо понимал технические особенности нового продукта или технологии. Для решения этой задачи существуют два способа - обучения своих сотрудников и набор уже обученного персонала со стороны.

И тот и другой подходы имеют свои плюсы и минусы. Обучение своих сотрудников - более длительный процесс и часто он связан со значительными затратами на оплату обучения. Зато вы имеете дело с проверенными людьми. Наем новых уже обученных сотрудников дает более быстрый результат, но при этом человеческие качества нового сотрудника могут заставить в скором времени отказаться от его услуг. Да и уровень профессиональных знаний может оказаться не тем, на который вы рассчитывали.

Результаты опроса, проведенного журналом DataCommunications среди посетителей Web-узла журнала, дали следующие результаты.

На вопрос "Испытываете ли вы сложности при подборе специалистов для построения и обслуживания вашей сети" 26% опрошенных ответило "Нет" и 74% ответило "Да". На вопрос "Что вы делаете для решения проблемы со специалистами" были получены следующие ответы:

22% - нанимаем новых сотрудников;  
50% - переобучаем своих сотрудников;  
34% - обращаемся к внешним консультантам;  
25% - нанимаем внешних специалистов на временную работу.

Переобучение своих сотрудников имеет обратную сторону медали - именно эти люди часто и являются теми специалистами, которых нанимает другое предприятие для решения аналогичных проблем. Поэтому некоторые предприятия очень неохотно платят деньги за переобучение своих сотрудников, особенно в новых модных областях, например, создания собственных Web-узлов. Слишком велика, по их мнению, вероятность перехода таких сотрудников на новое место работы, где им предлагают более высокий уровень заработка.

После принятия решения об обучении своих сотрудников руководителю соответствующего подразделения предприятия нужно принять решения по ряду вопросов: Какой учебный центр выбрать? Должен ли этот учебный центр быть авторизован производителем продукта или это может быть независимый учебный центр ? Насколько необходима дальнейшая сертификация специалиста после прохождения обучения? Повышать ли зарплату сотрудникам, успешно прошедшим обучение и, возможно, сертификацию, для их закрепления на предприятии ?

# 2. Главные тенденции развития локальных сетей

Ранее были описаны основные проблемы, которые нужно решить сетевым специалистам для того, чтобы локальные сети успешно выполняли возлагаемые на них в корпоративных сетях задачи. В данном разделе излагаются способы решения этих проблем, появившиеся в последнее время в продуктах и технологиях ведущих производителей коммуникационного оборудования для локальных сетей.

### 2.1. Обеспечение иерархии скоростей и качества обслуживания

К началу 90-х стало ясно, каким требованиям должны удовлетворять новые транспортные технологии для локальных сетей:

* Поддержка иерархии скоростей от 10 Мб/с до нескольких сотен Мб/с.
* Обеспечение гарантий требуемого качества транспортного обслуживания для каждого приложения, работающего в сети. В понятие качества обслуживания должны включаться такие параметры как средняя пропускная способность, пиковая пропускная способность, допускаемая в течение небольшого периода времени, максимальная величина задержек пакетов, вариация задержек пакетов.
* Наличие дешевых решений для нетребовательных сегментов сети и более дорогих, но и более высокопроизводительных решений для сегментов, поддерживающих трафик реального времени. Обычно удешевление локальной сети всегда достигалось за счет использования разделяемых сред передачи данных между всеми компьютерами сегмента. По такому пути шли авторы стандарта Ethernet в 70-е годы, по такому же пути пошли и авторы практически всех новых технологий для локальной сети.
* Возможность поэтапного внедрения новой технологии в существующие сети, не требующего чрезмерных разовых вложений.

Все работы по созданию технологий, удовлетворяющих этим требованиям, можно разделить на три большие группы:

1. Создание масштабируемой по скорости технологии на основе технологии Ethernet: линия Ethernet - FastEthernet - GigabitEthernet. Качество обслуживания не обеспечивается ни одной из входящих в триаду технологий, поэтому для его поддержки необходима реализация дополнительных механизмов в коммутаторах и маршрутизаторах.
2. Создание технологии с масштабируемой скоростью, частично совместимой с Ethernet, и имеющей встроенные возможности для обеспечения начального уровня качества обслуживания для трафика реального времени: линия 100VG-AnyLAN - 1000VG.
3. Использование в локальных сетях технологии АТМ, изначально разработанной для поддержки тонкой градации качества обслуживания для соединений "приложение - приложение" и обеспечения иерархии скоростей в рамках одной и той же технологии. Так как технология АТМ существенно отличается от остальных технологий локальных сетей и не имеет дешевого варианта работы на разделяемой среде, то основные усилия разработчиков сосредоточены на реализации механизмов наименее болезненного внедрения этой технологии в существующие локальные сети и удешевлении АТМ-оборудования.

Необходимо подчеркнуть, что появление в начале 90-х годов быстродействующих многопортовых мостов, которыми в сущности являются современные коммутаторы локальных сетей, резко расширило функциональные возможности протоколов локальных сетей. Использование микросегментации, когда в сети отсутствуют разделяемая среда между конечными узлами и портами коммутаторов, снимает многие ограничения, свойственные тому или иному протоколу. Крайней формой отхода от классического использования разделяемой во времени среды нужно считать полнодуплексные версии протоколов локальных сетей, которые работают исключительно в микросегментах.

В полнодуплексной версии из протокола локальной сети удаляются все алгоритмы, связанные с предоставлением доступа к разделяемой среде, а они обычно влияют на значительную часть микросхем сетевых адаптеров и их драйверов, а также соответствующих схем портов коммутаторов. От протокола остается только метод физического кодирования сигналов (ман- честерский код или избыточные коды типа 4B/5B), формат пакета и, возможно, способ тестирования работоспособности связей и узлов, а также организация обхода отказавших элементов сегмента (эти процедуры развиты только в протоколе FDDI, а также присутствуют в рудиментарной форме в протоколе TokenRing).

Зато взамен ненужных процедур доступа в полнодуплексной версии соответствующего протокола должны присутствовать новые процедуры - процедуры управления потоком кадров, так как теперь ничто не мешает станциям сети посылать кадры в коммутаторы с максимальной скоростью и создавать на определенных портах перегрузки, которые могут привести к потерям кадров и к существенному замедлению работы локальной сети. Таким образом, локальная сеть, использующая полнодуплекесную версию протокола, по принципам работы становится очень близкой к глобальной сети (например, Х.25 или framerelay), в которой узлы всегда использовали канал связи с коммутатором в полнодуплексном режиме.

Ввиду большой популярности коммутаторов и, соответственно, полнодуплексных режимов работы протоколов в локальных сетях при сравнении протоколов и выборе наиболее перспективного для вашей сети необходимо всегда учитывать существование двух режимов работы каждого протокола - полудуплексного (в сети с концентраторами-повторителями) и полнодуплексного (в сети на основе коммутаторов). Сравнение возможностей и стоимости только полудуплексных версий не даст правильной картины, так как эти показатели могут отличаться значительно. Так, например, максимальный диаметр сегмента FastEthernet даже при использовании оптоволокна составляет менее 400 метров в полудуплексном режиме, а при использовании полнодуплексного режима увеличивается до 2-х километров, как и у других технологий, таких как FDDI, ATM и100VG-AnyLAN.

#### 2.1.1. От Ethernet к Fast и Gigabit Ethernet

**2.1.1.1. Причины создания стандарта Fast Ethernet и его основные характеристики**

Первая высокоскоростная технология - FDDI - была создана в середине 80-х годов для работы на магистралях крупных сетей и для подключения серверов и мощных компьютеров, которым не хватало пропускной способности в 10 Мб/с, обеспечиваемой самой популярной технологией локальных сетей - Ethernet. Поэтому разработчики заботились в первую очередь о повышении пропускной способности и отказоустойчивости, так как эти свойства наиболее важны для магистрали сети. Разработанная технология действительно удовлетворяет поставленным требованиям - двойное оптоволоконное кольцо гарантирует работоспособность сети при одиночных обрывах кабеля и одиночных отказах оборудования конечных узлов, обеспечивает высокую (для середины 80-х - очень высокую) скорость передачи данных в 100 Мб/c. Уже тогда стала понятной необходимость обеспечения в локальных сетях поддержки трафика реального времени, и разработчики стандарта включили в него механизм предоставления приоритетного доступа к разделяемому между всеми узлами кольцу для синхронного трафика реального времени. Кроме того, использование оптоволокна позволило даже для такой высокой скорости обеспечить расстояние между узлами сети до 2-х километров, а общий диаметр - до 100 километров, что вывело сети FDDI из класса чисто локальных сетей в класс сетей масштаба крупного города (MetropolitanAreaNetwork).

В дальнейшем технология FDDI развивалась незначительно и, несмотря на попытки ее удешевления за счет использования высококачественной витой пары, осталась технологией магистралей и серверных парков. Механизм приоритетов для поддержки синхронного трафика также остался в зачаточном состоянии, без предоставления узлам сети возможности автоматически и гибко распределять полосу пропускания кольца между приложениями с синхронным трафиком.

Поэтому со временем все больше ощущалась потребность в технологии, которая бы предоставляла большие, чем Ethernet скорости передачи данных для массовых недорогих компьютеров, таких как персональные компьютеры конца 80-х годов. Особенно остро эта проблема встала перед сетевым сообществом в начале 90-х, когда пропускная способность канала диск-память многих моделей персональных компьютеров превзошла рубеж в 1 Мбайт/c, уже недоступный для сетевых адаптеров Ethernet.

Так как поддержку всех необходимых механизмов качества обслуживания тогда обещала быстро приобретающая черты реального стандарта технология АТМ, то решено было вдохнуть новую жизнь в такую знакомую и проверенную технологию как Ethernet, не занимающуюся вопросами обслуживания трафика разного типа ни в коей степени, но хорошо и эффективно обслуживающую многие виды приложений. Несмотря на постоянные упоминания о мультимедийных приложениях, доля их в общей смеси приложений многих сетей не так уж велика и до сих пор, поэтому отсутствие средств поддержки качества обслуживания не казалось разработчикам нового Ethernet'а чем-то трагическим. Многие специалисты оправдывали перенос недостатков технологии Ethernet в новый стандарт его временным сроком жизни - 5 - 8 лет, сроком, который по их мнению был нужен технологии АТМ для завоевания рынка локальных сетей.

Пользователи с большим энтузиазмом восприняли сообщения, появившиеся в 1992 году о начале работ по разработке высокоскоростного Ethernet'а, обещавшие им продление жизни привычной и недорогой технологии. Однако вскоре сетевой мир разделился на два соперничающих лагеря, что и привело в конце концов к появлению двух различных технологий - FastEthernet и100G-AnyLAN.

Сторонники первого подхода считали, что новая технология должна в максимальной степени быть похожа во всем на Ethernet - за исключением только битовой скорости передачи данных.

Сторонники второго подхода призывали воспользоваться удобным случаем для устранения недостатков, связанных со слишком "случайным" механизмом предоставления доступа к разделяемой среде CSMA/CD, используемым в Ethernet.

И у них были достаточно веские причины для критики алгоритма CSMA/CD. Его основное преимущество - простота реализации, за счет чего Ethernet и является самой дешевой технологией. Но простота метода доступа к разделяемой среде имеет и несколько отрицательных последствий. Наиболее важны следующие два:

1. Столкновения кадров нескольких станций - коллизии - являются допустимыми событиями в сегменте Ethernet, и при достаточно низком коэффициенте загрузки сегмента мало сказываются на пропускной способности канала. Однако, при повышении загруженности сегмента, которое обычно со временем наблюдается во всех сетях, коллизии начинают "отбирать" все больше и больше полезной пропускной способности сети (рис. 2.1), так как каждая коллизия связана с непроизводительным использованием сегмента. Метод CSMA/CD не гарантирует для узла получения доступа к среде даже за весьма большой интервал времени, и такие ситуации иногда случаются в сетях Ethernet в реальной жизни, когда постоянно генерирующая ошибочные кадры станция не дает возможности системе управления передать на нее управляющие кадры.

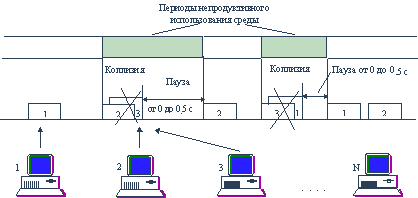


Рис. 2.1. Уменьшении полезной пропускной способности сегмента Ethernet при повышении коэффициента загрузки

1. Длина сегмента Ethernet всегда ограничена очень жестким соотношением, которое вряд ли можно преодолеть за счет технического прогресса. Для того, чтобы конечные узлы сети всегда четко распознавали коллизии и автоматически организовывали повторную передачу искаженного в результате коллизии кадра, нужно, чтобы время передачи кадра всегда было больше времени двойного оборота сигнала по сегменту Ethernet (рис.2.2). Так как время распространения сигнала ограничено скоростью света, то максимальная длина сегмента Ethernet для битовой скорости 10 Мб/c составляет примерно 2500 м. При увеличении битовой скорости в 10 раз и сохранении минимального размера кадра в 64 байта максимальный размер сегмента сокращается соответственно в 10 раз, то есть становится равным 250 метрам, а при увеличении битовой скорости еще в 10 раз - 25 метрам.

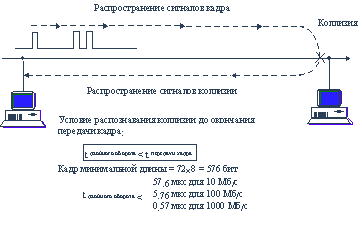


Рис.2.2. Ограничение накладываемое методом доступа CSMA/CD на длину сегмента Ethernet

Правда, ситуация улучшается при использовании коммутаторов и полнодуплексного режима работы, но при этом теряется низкая стоимость сегмента.

Тем не менее, недостатки, связанные с методом доступа CSMA/CD, не испугали сторонников "чистого" Ethernet'а и они в 1992 году образовали неформальное объединение *FastEthernetAlliance,* куда первоначально вошли такие лидеры технологии Ethernet как SynOptics, 3Com и ряд других.

Одновременно были начаты работы в институте IEEE по стандартизации новой технологии - там была сформирована исследовательская группа для изучения технического потенциала высокоскоростных технологий. За период с конца 1992 года и по конец 1993 года группа IEEE изучила 100-Мегабитные решения, предложенные различными производителями. Наряду с предложениями FastEthernetAlliance группа рассмотрела также и другой подход к созданию недорогого высокоскоростного стандарта, предложенный компаниями*Hewlett-Packard и AT&T*.

В 1995 году комитет IEEE принял спецификацию FastEthernet в качестве стандарта, и сетевой мир получил технологию, с одной стороны решающую самую болезненную проблему - нехватку пропускной способности на нижнем уровне сети, а с другой стороны очень легко внедряющуюся в существующие сети Ethernet, которые и сегодня дают миру **около 80% всех сетевых соединений**.

Легкость внедрения FastEthernet объясняется следующими факторами:

* общий метод доступа позволяет использовать в сетевых адаптерах и портах FastEthernet до 80% микросхем адаптеров Ethernet;
* драйверы также содержат большую часть кода для адаптеров Ethernet, а отличия вызваны новым методом кодирования (4B/5B или 8B/6T) и наличием полнодуплексной версии протокола;
* формат кадра остался прежним, что дает возможность анализаторам протоколов применять к сегментам FastEthernet те же методы анализа, что и для сегментов Ethernet, лишь механически повысив скорость работы.

Отличия FastEthernet от Ethernet сосредоточены в основном на физическом уровне. Разработчики стандарта FastEthernet учли тенденции развития структурированных кабельных систем и реализовали физический уровень для всех популярных типов кабелей, входящих в стандарты на структурированные кабельные системы (такие как EIA/TIA 568A) и реально выпускаемые кабельные системы.

Существует три варианта физического уровня FastEthernet:

1. *100Base-TX* для двухпарного кабеля на неэкранированной витой паре UTPCategory 5 (или экранированной витой паре STPType 1);
2. *100Base-T4* для четырехпарного кабеля на неэкранированной витой паре UTPCategory 3,4 или 5;
3. *100Base-FX*для многомодового оптоволоконного кабеля.

При создании сегментов FastEthernet с разделяемой средой нужно использовать концентраторы, при этом максимальный диаметр сети колеблется от 136 до 205 метров, а количество концентраторов в сегменте ограничено одним или двумя, в зависимости от типа концентратора. При использовании двух концентраторов расстояние между ними не может превышать 5 - 10 метров, так что существование 2-х устройств мало что дает, кроме увеличения количества портов - расстояние между компьютерами сегмента от добавления второго концентратора практически не изменяется.

В разделяемом сегменте FastEthernet нет возможности обеспечить какие-либо преимущества при обслуживании трафика приложений реального времени, так как любой кадр получает равные шансы захватить среду передачи данных в соответствии с логикой алгоритма CSMA/CD.

**2.1.1.2. В каких случаях рекомендуется использовать Fast Ethernet**

У технологии FastEthernet есть несколько ключевых свойств, которые определяют области и ситуации ее эффективного применения. К этим свойствам относятся:

* большая степень преемственности по отношению к классическому 10 мегабитному Ethernet'у;
* высокая скорость передачи данных - 100 Mб/c;
* возможность работать на всех основных типах современной кабельной проводки - UTPCategory 5, UTPCategory 3, STPType 1, многомодовом оптоволокне.

Наличие многих общих черт у технологий FastEthernet и Ethernet дает простую общую рекомендацию - FastEthernet следует применять в тех организациях и в тех частях сетей, где до этого широко применялся 10 Мегабитный Ethernet, но сегодняшние условия или же ближайшие перспективы требуют в этих частях сетей более высокой пропускной способности. При этом сохраняется весь опыт обслуживающего персонала, привыкшего к особенностям и типичным неисправностям сетей Ethernet. Кроме того, можно по-прежнему использовать средства анализа протоколов, работающие с агентами MIB-II, RMONMIB и привычными форматами кадров.

**FastEthernet в сетях рабочих групп**

Область применения разделяемых сегментов FastEthernet достаточно ясна - это объединение близко расположенных друг от друга компьютеров, трафик которых имеет ярко выраженный пульсирующий характер с большими, но редкими всплесками. Большие всплески хорошо передаются незагруженным каналом 100 Мб/c, а редкое их возникновение приводит к возможности совместного использования канала без частого возникновения коллизий. Типичным примером такого трафика является трафик файлового сервиса, электронной почты, сервиса печати.

Поэтому основная область использования FastEthernet с общей средой - это настольные компьютеры, сети рабочих групп и отделов, где компьютерам требуется пиковая пропускная способность выше 100 Мб/c. Такими компьютерами чаще всего являются файловые серверы, но и клиентским компьютерам может понадобиться скорость выше 10 Мб/c.

При этом целесообразно совершать переход к FastEthernet постепенно, оставляя Ethernet там, где он хорошо справляется со своей работой. Одним из очевидных случаев, когда Ethernet не следует заменять FastEthernet'ом, является подключение к сети старых персональных компьютеров с шиной ISA - их пропускная способность канала "сеть - диск" не позволит пользователю ощутить выгоды от повышения в 10 раз скорости сетевой технологии. Для устранения узких мест для сетей, состоящих из таких компьютеров, больше подходит использование коммутаторов с портами 10 Мб/с, так как в этом случае узлам гарантированно предоставляется по 10 Мб/с - как раз столько, сколько им нужно при их архитектуре и параметрах производительности.

Новые клиентские компьютеры с процессором PentiumPro и шиной PCI - очевидные претенденты на использование скорости 100 Мб/c. Поэтому даже при весьма неопределенных требованиях их пользователей к пропускной способности сети имеет смысл покупать для них сетевые адаптеры FastEthernet, которые могут работать на скорости 10 Мб/c, пока у организации не появятся концентраторы или коммутаторы с портами FastEthernet. Переход к скорости 100 Мб/c будет для пользователей практически безболезненным, так как большинство сетевых адаптеров не нужно конфигурировать для перехода на FastEthernet.

**FastEthernet в магистралях зданий и кампусов**

Создание достаточно крупных сетей, к которым относятся сети зданий и кампусов с количеством узлов в несколько сотен, также возможно с использованием технологии FastEthernet. Эта технология может использоваться в таких сетях как в "чистом" виде, так и в сочетании с другими технологиями, например, FDDI или ATM.

Сети зданий и даже крупных этажей сейчас практически не строятся без использования коммутаторов, поэтому ограничения на максимальный диаметр сети в 250 - 272 метра легко преодолеваются, так как соединение коммутатор-коммутатор позволяет удлинить сеть до 412 м при полудуплексной связи на оптоволокне, и до 2 км при аналогичной полнодуплексной связи.

Отсутствие стандартного резервирования на уровне повторителей также мало ограничивает построения отказоустойчивых магистралей - поддержка коммутаторами алгоритма SpanningTree позволяет автоматически переходить с основной отказавшей связи на резервную.

Основными двумя факторами, сдерживающими применение технологии FastEthernet на магистралях, являются:

* широкое использование в настоящее время для этой цели технологии FDDI;
* отсутствие у технологии FastEthernet средств поддержки трафика реального времени.

Поэтому, если эти факторы не относятся к вашей сети, то ее магистраль можно успешно строить и на коммутируемой технологии FastEthernet, особенно на ее полнодуплексной версии. Правда при этом все равно остаются нерешенными некоторые проблемы, присущие сети, построенной на коммутаторах.

Большая часть производителей коммуникационного оборудования для локальных сетей поддерживают технологию FastEthernet во всем спектре своих изделий - сетевых адаптерах, повторителях, коммутаторах и маршрутизаторах.

Наиболее распространенный тип физического интерфейса - 100Base-TX, а интерфейсы 100Base-T4 распространены в меньшей степени (по прогнозам компании SMC, внесшей большой вклад в разработку этой версии физического уровня, доля оборудования T4, которое может работать на обычной витой паре категории 3, составит в недалеком будущем 25% от всего рынка оборудования FastEthernet). Интерфейсы 100Base-FX часто поддерживаются не непосредственно, а через интерфейс MII и соответствующий оптоволоконный трансивер.

Стоимость технологии FastEthernet при использовании разделяемой среды передачи данных составляет около $100 - $160 на узел (стоимость сетевого адаптера и порта концентратора), что приближает эту технологию к классическому 10 Мегабитному Ethernet'у по стоимости.

**2.1.1.3. Переход Ethernet на гигабитные скорости**

Достаточно быстро после появления на рынке продуктов FastEthernet сетевые интеграторы и администраторы почувствовали определенные ограничения при построении корпоративных сетей на базе этих двух технологий. Во многих случаях серверы, подключенные по 100-Мегабитному каналу, перегружали магистрали сетей, работающие также на скорости 100 Мб/c - магистрали FDDI и FastEthernet. Ощущалась потребность в следующем уровне иерархии скоростей. В 1995 году более высокий уровень скорости могли предоставить только коммутаторы АТМ, а при отсутствии в то время удобных средств миграции этой технологии в локальные сети (хотя спецификация LANEmulation - LANE, была принята в начале 1995 года, практическая ее реализация была впереди) внедрять их в локальную сеть почти никто не решался.

Поэтому логичным выглядел следующий шаг, сделанный IEEE - через 5 месяцев после окончательного принятия стандарта FastEthernet в июне 1995 исследовательской группе по изучению высокоскоростных технологий IEEE было предписано заняться изучением возможности выработки стандарта Ethernet с еще более высокой битовой скоростью.

Летом 1996 было объявлено о создании группы 802.3z для разработки протокола, максимально подобного Ethernet, но с битовой скоростью 1000 Мб/c. Как и в случае FastEthernet, сообщение было воспринято сторонниками Ethernet с большим энтузиазмом, а лагерь приверженцев технологии АТМ это сообщение насторожило.

Основной причиной энтузиазма была перспектива такого же плавного перевода магистралей сетей на GigabitEthernet, подобно тому, как были переведены на FastEthernet перегруженные сегменты Ethernet, расположенные на нижних уровнях иерархии сети.

Образованный для согласования усилий в этой области GigabitEthernetAlliance сразу же включал таких флагманов отрасли как BayNtworks, CiscoSystems и 3Com. За год своего существования GigabitEthernetAlliance существенно вырос и насчитывает сейчас более 100 членов.

Основная идея разработчиков стандарта GigabitEthernet состоит в максимальном сохранении идей классической технологии Ethernet при достижении битовой скорости в 1000 Мб/с. GigabitEthernet также как и его менее скоростные собратья не будет на уровне протокола поддерживать:

* качество обслуживания;
* избыточные связи;
* тестирование работоспособности узлов и оборудования (в последнем случае - за исключением тестирования связи порти - порт, как это делается для Ethernet и FastEthernet).

По-прежнему будут существовать полудуплексная версия протокола, поддерживающая метод доступа CSMA/CD, и полнодуплексная версия, работающая с коммутаторами.

Основные проблемы, которые решают разработчики стандарта GigabitEthernet, сосредоточены в следующих областях:

* В связи с ограничениями, накладываемыми методом CSMA/CD на длину кабеля, версия GigabitEthernet для разделяемой среды допускала бы длину сегмента всего в 25 метров. Так как существует большое количество применений, когда нужно повысить диаметр сегмента хотя бы до 100 метров, то сейчас предпринимаются усилия по использованию всего потенциала современных технологий для преодоления "врожденного" ограничения метода CSMA/CD.

Одно из предложений состоит в расширении минимального размера кадра с 64 до 512 байт. Если поле данных состоит из меньшего количества байт, то кадр дополняется до 512 байт служебными символами. Такой подход позволяет увеличить диаметр сети до 100 м, но снижает полезную пропускную способность сети.

Второе предложение основано на применении для соединения узлов в сегмент буферизующего полнодуплексного повторителя. Такой повторитель разрешает станциям работать со своими портами по полнодуплексной схеме, снимая реализацию метода доступа CSMA/CD с сетевых адаптеров компьютеров. Однако, повторитель по прежнему реализует разделяемую среду в 2 Гб/c за счет применения алгоритма CSMA/CD к кадрам, поступившим в буфер порта. Такой подход позволяет строить связи между узлом и повторителем той же длины, что и в случае использования коммутатора. Сам же полнодуплексный повторитель оказывается дешевле коммутатора, так как его внутренняя производительность должна составлять всего 2 Гб/c вместо N/2x2 Гб/c при построении коммутатора с N портами GigabitEthernet. Полнодуплексный повторитель позволяет соединять сегменты GigabitEthernet с сегментами Ethernet и FastEthernet.

* Достижение битовой скорости 1000 Мб/c на основных типах кабелей. Даже для оптоволокна достижение такой скорости представляет некоторые проблемы, так как технология FibreChannel, физический уровень которой был взят за основу для оптоволоконной версии GigabitEthernet, обеспечивает скорость передачи данных всего в 800 Мб/c (битовая скорость на линии равна в этом случае примерно 1000 Мб/c, но при методе кодирования 8B/10B полезная битовая скорость на 20% меньше скорости на линии). Разработчики стандарта считают, что на многомодовом оптоволокне им удастся обеспечить расстояние между узлами для полнодуплексного режима работы в 550 м, а для одномодового волокна - до 3000 м.
* Поддержку кабеля на витой паре. Отдельный комитет 802.3ab был создан для разработки стандарта GigabitEthernet на витой паре 5 категории. У председателя группы 802.3z Говарда Фрэйзера, специалиста из компании Cisco, есть оптимизм относительно возможностей техники кодирования при использовании всех четырех пар кабеля категории 5. При использовании всех четырех пар (это возможно при отказе от применения алгоритма CSMA/CD на отрезке сети между конечным узлом и повторителем или же при работе с коммутатором) задача все равно остается непростой, так как по каждой паре нужно передать данные со скоростью 250 Мб/c - максимальная скорость работы на витой паре категории 5 в 155 Мб/c достигается в настоящее время технологией АТМ. Так что задача чрезвычайно непростая. Использование же двух пар витой пары категории 6 вызывает сомнения - экранирование при скорости передачи по одной 1000 Мб/c необходимо как средство защиты людей от вредного излучения, а экранирование связано с решением проблем заземления, часто весьма сложных, кроме того, стоимость кабельной системы на экранированной витой паре категории 6 будет сравнима со стоимостью оптоволоконной кабельной системы.

Первый проект стандарта GigabitEthernet был представлен на рассмотрение группы 802.3z в январе 1997 года, а окончательное принятие ожидается в начале 1998 года.

GigabitEthernetAlliance предполагает, что стоимость одного порта концентратора GigabitEthernet в 1998 году составит от $920 до $1400, а стоимость одного порта коммутатора GigabitEthernet составит от $1850 до $2800.

Как и в случае с FastEthernet, оборудование GigabitEthernet появилось на рынке задолго до окончательного принятия стандарта. 13 компаний производили летом 1997 года коммутаторы GigabitEthernet, 3 компании - концентраторы GigabitEthernet, 6 - сетевые адаптеры GigabitEthernet. Небольшое количество моделей концентраторов GigabitEthernet связано с трудностями реализации этой технологии на разделяемой среде. Коммутаторы на оптоволокне реализовать проще - для этого нужно взять микросхемы FibreChannel, разогнать их до тактовой скорости 1.2 Ггц и снабдить устройство коммутирующим ядром достаточной производительности.

Для технологии GigabitVG предлагается реализовать скорость 500 Мб/с для витой пары и 1 Гб/с для оптоволокна. Предельные расстояния между узлами ожидаются следующие: для витой пары - 100 м, для многомодового оптоволокна - 500 м и для одномодового оптоволокна - 2 км.

**2.1.2. Технология 100VG-AnyLAN - улучшенное качество обслуживания за ту же стоимость**

В качестве альтернативы технологии FastEthernet фирмы AT&T и HP выдвинули проект новой недорогой технологии со скоростью передачи данных 100 Мб/с - 100Base-VG (VoiceGrade - технология, способная работать на кабеле категории 3, предназначенном первоначально для передачи голоса). В этом проекте было предложено усовершенствовать метод доступа с учетом потребности мультимедийных приложений, а для формата пакета сохранить совместимость с форматом пакета сетей 802.3. В сентябре 1993 года по инициативе фирм IBM и HP был образован комитет IEEE 802.12, который занялся стандартизацией новой технологии. Проект был расширен за счет поддержки в одной сети кадров не только формата Ethernet, но и формата TokenRing. В результате новая технология получила название *100VG-AnyLAN,* то есть технология для любых сетей, где под любыми сетями понимаются сети Ethernet и TokenRing.

Летом 1995 года технология 100VG-AnyLAN получила статус стандарта IEEE 802.12.

В технологии 100VG-AnyLAN определен новый метод доступа DemandPriority с двумя уровнями приоритетов - для обычных приложений и для мультимедийных, а также новая схема квартетного кодирования QuartetCoding, использующая избыточный код 5В/6В, и позволяющая передавать по каждой из 4-х пар категории 3 данные с полезной скоростью 25 Мб/c.

**Пропускная способность и качество обслуживания**

Метод доступа DemandPriority основан на передаче концентратору функций арбитра, решающего проблему доступа к разделяемой среде. Концентратор отличается от обычных повторителей за счет того, что он изучает адреса присоединенных к нему узлов (в момент физического подключения) и поэтому не передает принятый от узла кадр на все порты, а только на тот, на который нужно. Среда по-прежнему разделяемая, так как концентратор за один цикл опроса портов принимает в свой буфер только один кадр - от приоритетного порта или же при равных приоритетах от первого по порядку. Однако, некоторые этапы работы с разными узлами совмещаются во времени, и за счет этого ускоряется передача кадров.

Работа сети 100VG-AnyLAN не дает гарантий приложениям по поддержанию для них определенного качества обслуживания, как это делает технология АТМ. Приоритеты только уменьшают задержки трафика реального времени, но это сервис по принципу besteffort, то есть обслуживание "по возможности" лучшее, но без каких-либо количественных гарантий.

Метод DemandPriority повышает коэффициент использования пропускной способности сети - до 95% по утверждению компании Hewlett-Packard.

**Используемые кабельные системы и максимальный диаметр сети**

Отсутствие требования распознавания коллизий позволяет без проблем строить протяженные сегменты сети без коммутаторов, только на концентраторах - до 2-х километров между узлами на оптоволокне и до 100 метров на витой паре.

Общий диаметр сети, построенной на концентраторах, может составлять при использовании многомодового оптоволокна до 5000 м.

Связь, соединяющая концентратор и узел, может быть образована:

* 4 парами неэкранированной витой пары категорий 3,4 или 5 (4-UTPCat 3,4,5);
* 2 парами неэкранированной витой пары категории 5 (2-UTPCat 5);
* 2 парами экранированной витой пары типа 1 (2-STPType 1);
* 2 парами многомодового оптоволоконного кабеля.

Хотя могут использоваться любые варианты кабельной системы, наиболее распространен вариант 4-UTP, который был разработан первым, кроме того, его популярность объясняется тем, что он работает на витой паре категории 3, установленной во многих существующих локальных сетях.

**Совместимость с существующими локальными сетями**

Сегменты 100VG-AnyLAN достаточно просто могут быть внедрены в существующие сети. Каждый концентратор может быть сконфигурирован на поддержку либо кадров 802.3 Ethernet либо кадров 802.5 TokenRing. Все концентраторы, расположенные в одном и том же логическом сегменте (не разделенном мостами, коммутаторами или маршрутизаторами), должны быть сконфигурированы на поддержку кадров одного типа. Для связи сегмента 100G-AnyLAN сегментами Ethernet или TokenRing нужно использовать коммутатор или маршрутизатор, так как частота передачи бит и способ их кодирования отличаются, и концентратор не может справиться с такими проблемами. Коммутатор может достаточно быстро передавать кадры из сегмента 100VG-AnyLAN в сегмент традиционной технологии и обратно, так как трансляция формата кадра и пересчет контрольной суммы не требуется.

**Перспективы и области применения 100VG-AnyLAN**

Технология 100VG-AnyLAN имеет меньшую популярность среди производителей коммуникационного оборудования, чем конкурирующее предложение - технология FastEthernet. Компании, которые не поддерживают технологию 100VG-AnyLAN, объясняют это тем, что для большинства сегодняшних приложений и сетей достаточно возможностей технологии FastEthernet, которая не так заметно отличается от привычной большинству пользователей технологии Ethernet. В более далекой перспективе эти производители предлагают использовать для мультимедийных приложений технологию АТМ, или же GigabitEthernet, а не 100VG-AnyLAN.

Тем не менее, число сторонников технологии 100VG-AnyLAN растет и насчитывает около 30 компаний. Среди них находятся не только копании Hewlett-Packard и IBM, но и такие лидеры как CiscoSystems, Cabletron, D-Link и другие. Все эти компании поддерживают обе конкурирующие технологии в своих продуктах, выпуская модули с портами как FastEthernet, так и 100VG-AnyLAN.

Наиболее очевидным случаем применения технологии 100VG-AnyLAN является модернизируемые сети TokenRing. Технология TokenRing широко используется на протяжении многих компанией IBM и некоторыми другими (Madge, Thomas-Conrad) для построения сегментов локальных сетей, решающих ответственные бизнес-задачи. Эта технология популярна в западных банках, использующих мейнфреймы и миникомпьютеры производства IBM, и многих других отраслях бизнеса. Кроме мощной поддержки компанией IBM, популярность TokenRing объясняется наличием у нее встроенных в протокол (и, соответственно, в оборудование и сетевые адаптеры) процедур самотестирования сети, не таких развитых как у FDDI, но тем не менее позволяющих обнаружить источник неисправности кольца.

Однако, перспектив дальнейшего развития у технологии TokenRing практически нет - это признают многие ведущие специалисты (смотрите заметки Ника Липписа "TheTokenRingTrap" в февральском номере DataCommunications и Робина Лейланда "TimetoMoveOnThePriceofEthernetSwitching" в июльском номере того же журнала). Предел скорости в 16 Мб/c не дает возможности масштабирования производительности сетей TokenRing в широких масштабах, а сеть, построенная на коммутаторах TokenRing может оказаться дороже сети, построенной на концентраторах и коммутаторах FastEthernet. В примере, рассмотренном в статье Робина Лейланда, сравнивается стоимость коммутируемой сети TokenRing и коммутируемой сети Ethernet/FastEthernet/GigabitEthernet для сети 7-этажного здания. Этот пример более подробно рассмотрен в разделе 2.2.4, а здесь приведем только окончательные результаты - общая стоимость полностью коммутируемой сети TokenRing составила $415 625, в то время как иерархически построенная сеть Ethernet/FastEthernet/GigabitEthernet "потянула" всего лишь на $270 000.

Но на полную одномоментную замену оборудования TokenRing решится мало предприятий, поэтому в условиях необходимости сосуществования со старыми сетями TokenRing технология 100VG-AnyLAN может найти свое место.

**Гигабитные сети 1000VG**

Комитет 802.12, ведомый специалистами компании Hewlett-Packard, также ведет работы по разработке варианта этой технологии для скорости передачи данных в 1 Гигабит в секунду. Вариант этой технологии также ориентируется на физический уровень стандарта FibreChannel, а в качестве метода доступа предполагается использовать метод DemandPriority.

К энтузиастам перевода технологии VG на гигабитные скорости относятся также компании CompaqComputer, TexasInstrument и Motorola.

### 2.5. Выбор технологии для построения магистрали крупной локальной сети

Магистраль крупной локальной сети - это очень ответственный участок, во многом определяющий все свойства сети в целом. Причина - через магистраль проходят все основные пути взаимодействия между сетями рабочих групп, отделов и подразделений в том случае, если требуемые клиенту ресурсы находятся за пределами сети его рабочей группы, отдела и т.п. Применение технологии Intranet, поиск нужной информации в Internet приводят к тому, что все чаще и чаще нужные пользователю ресурсы находятся за пределами его сегмента сети, а это в свою очередь резко повышает интенсивность трафика, проходящего через магистраль сети. У разных клиентских сессий могут быть существенно разные требования к качеству обслуживания - интерактивное телевидение предъявляет самые жесткие требования к задержкам и вариации задержек предаваемых пакетов при постоянной скорости обмена, передача больших графических файлов менее чувствительна к задержкам передачи отдельных пакетов, но требует большой пропускной способности и быстрой реакции магистрали на значительные пульсации трафика.

Резко возросшие объемы передаваемых данных и разнородность требований клиентов к качеству обслуживания делают реализацию магистрали современной корпоративной локальной сети очень непростой задачей. Сегодня у администратора такой сети имеется несколько вариантов ее решения:

* улучшение традиционной схемы построения магистрали на основе колец FDDI с подсетями, подключенными через маршрутизаторы, за счет применения высокопроизводительных маршрутизаторов нового поколения, отличающихся внутренним параллелизмом операций и ускоренной передачей долговременных потоков данных;
* использование на магистрали стандартной технологии АТМ;
* использовании на магистрали технологий ускоренной передачи IP-трафика через нестандартные коммутаторы АТМ, например, технологии IPswitching компании Ipsilon;
* применение на магистрали технологии GigabitEthernet.

**2.5.1. Построение магистрали с использованием технологии FDDI и высокопроизводительных маршрутизаторов**

Этот вариант связан с сохранением существующей магистрали, построенной как правило на технологии FDDI, и подключением подсетей с помощью традиционных маршрутизаторов. У этого решения имеется два узких места - сама скорость технологии FDDI в 100 Мб/c и значительные задержки, создаваемые маршрутизаторами при обработке пакетов. Если скорость в 100 Мб/c сама по себе достаточна для передачи в среднем всего магистрального трафика, то сейчас существуют решения, позволяющие ускорить работу маршрутизаторов, а значит и оставить общую структуру магистрали в неизменном виде, не применяя каких-либо революционных решений. Ускорение же работы маршрутизаторов многие производители обеспечивают двумя способами. Во-первых, за счет распараллеливания обработки пакетов мультипроцессорными маршрутизаторами, подобными Cisco 7500 или BayNetworksBCN, или переноса процедур маршрутизации с процессорного уровня на уровень заказных БИС (ASIC), как это сделано в маршрутизаторе GRF 400 компании AscendCommunications, способном обрабатывать около 2.8 миллионов IP-пакетов в секунду.

Во-вторых, за счет сокращения числа операций при маршрутизации пакетов, образующих в сети стабильные потоки данных - dataflow. Многие производители разработали и внедрили в свои маршрутизаторы частные схемы, у которых есть общая черта - они выявляют в сети долговременные соединения между двумя определенными узлами и приложениями, кэшируют маршрутную информацию, необходимую для обработки пакетов каждого потока, а затем не тратят много времени на обработку каждого пакета потока, так как такой пакет помечается на входе магистрали специальной меткой потока, а все маршрутизаторы магистрали оказываются уже настроенными на обработку помеченных пакетов без необходимости анализа IP-заголовка пакета и просмотра всей таблицы маршрутизации. Примером такого частного решения является техника NetFlow компании Cisco, которая позволяет ускорить обработку потоков данных не только для протокола IP, но и других протоколов, например, IPX.

**2.5.2. Построение магистрали на основе технологии АТМ**

Данный вариант построения магистрали основан на переходе в магистрали сети к такой новой технологии как АТМ. В этом случае магистраль будет обладать не только более высокой скоростью по сравнению с вариантом, основанным на FDDI, но и очень хорошей масштабируемостью, так как большинство коммутаторов АТМ для локальных сетей имеют интерфейсы не только 155 Мб/с, но и 622 Мб/c, а в недалеком будущем возможно появление интерфейсов и в 1.28 Гб/c - стандарт АТМ поддерживает всю иерархическую лестницу скоростей технологии SONET/SDH. Кроме иерархии скоростей, позволяющей модернизировать магистраль без замены оборудования, АТМ обладает наиболее развитым на сегодняшний день механизмом поддержания требуемого качества обслуживания для трафика разного типа, от голосового трафика и трафика видеоконференций с постоянной битовой скоростью до пульсирующего трафика Web-узлов с гарантированной средней пропускной способностью.

Однако, переход на магистрали к АТМ требует замены традиционных коммутаторов и маршрутизаторов с интерфейсами Ethernet, FastEthernet или FDDI на коммутаторы АТМ со сложной системой сигнализации и существенно более высокой стоимостью за порт. Такой переход требует как значительных инвестиционных вложений, так и обучения обслуживающего сеть персонала. Кроме того, сегодня до конца не решена проблема взаимодействия магистрали АТМ с подсетями, работающими на основе традиционных протоколов локальных сетей. В этой области определенно ощущается недостаток стандартов, на основе которых могло бы работать в одной сети оборудование разных производителей. Два стандарта из этой области - LANE и ClassicalIP - только частично решают проблему. Спецификация LANE (LANEmulation) превращает магистраль, построенную из АТМ коммутаторов в распределенный коммутатор, поддерживающий традиционные протоколы локальных сетей, например, Ethernet или FDDI. Но это не решает проблемы работы через магистраль подсетей, подключенных через маршрутизаторы, так как механизмы LANE работают на канальном уровне. Отказ от маршрутизаторов при создании магистрали крупной сети невозможен, так как надежная защита от широковещательного шторма, а также решение некоторых других проблем только средствами канального уровня невозможно - это известный факт, подтверждающийся опытом эксплуатации локальных сетей, построенных только на коммутаторах.

Стандарт ClassicalIP описывает работу маршрутизаторов через магистраль АТМ только для протокола IP. Кроме того, он был разработан достаточно давно как первый вариант механизма сосуществования маршрутизируемых сетей и сетей АТМ, и потому не решает всех задач автоматизации процесса работы маршрутизатора через сеть, не поддерживающую широковещательность.

Спецификация MPOA (MultiprotocolOverATM), которая должна определить, как должны работать многопротокольные маршрутизаторы через магистраль АТМ и быть одновременно совместимой с LANE, пока все еще не нашла окончательного одобрения в ATMForum из-за несовместимости позиций некоторых ключевых участников согласительного процесса, таких, например, как Cisco и BayNetworks.

Даже при наличии стандарта на работу многопротокольных маршрутизаторов через АТМ и достаточной производительности самих маршрутизаторов, особенности работы системы сигнализации коммутаторов АТМ затрудняют сегодня использование АТМ в качестве магистрали крупной локальной сети. Эти особенности, описанные выше, приводят к слишком большому времени установления новых соединений на магистрали, измеряемому единицами милисекунд.

**2.5.3. Применение на магистрали методов ускоренной передачи IP-трафика типа IP switching и tag switching**

Для устранения замедления работы магистрали, вносимого коммутаторами АТМ при установлении и разрыве динамических виртуальных соединений (SVC), компания Ipsilon предложила свой собственный подход использования технологии АТМ, названный ею IP-switching, который затем подхватили и развили многие компании, породив большое количество частных и несовместимых решений (Tag-switching компании Cisco, ARIS компании IBM и т.п.). Все вместе эти решения позволяют говорить о третьем вариантепостроения магистрали.

Этот вариант связан с нестандартным способом работы коммутаторов АТМ. Формат ячеек АТМ, способы разбиения пакетов на ячейки и последующей сборки ячеек в пакеты локальных сетей, а также использование небольшого поля номера виртуального канала VPI/VCI для принятия коммутатором решения о передаче ячейки с порта на порт остаются теми же. Изменяется способ прокладки нового виртуального соединения через сеть АТМ. Для этого не требуется прохождение запроса через все коммутаторы на пути нового виртуального канала. Каждый коммутатор строит таблицу номеров VPI/VCI для продвижения ячеек через свои порты относительно автономно от других коммутаторов, так что для образования виртуального пути не требуется согласия всех коммутаторов, через которые он проходит. Процесс создания таблиц VPI/VCI может быть растянут во времени, так что часть коммутаторов уже готова к быстрой коммутации ячеек нового виртуального пути, а часть - еще нет.

Виртуальные пути прокладываются через магистраль только для долговременных потоков пакетов данных. Для одиночных пакетов, которые не образуют потока (например, для пакетов сервиса DNS), коммутаторы АТМ работают как обычные IP-маршрутизаторы, то есть обрабатывают каждый пакет в соответствии с его IP-заголовком и просматривают обычную таблицу маршрутизации для принятия решения о продвижении пакета через сеть АТМ. Как только коммутатор выявляет устойчивый поток, проходящий через его порты, он устанавливает для него новое виртуальное соединение и присваивает ему новый адрес VPI/VCI. Затем все пакеты, приходящие в пограничный АТМ-коммутатор магистрали, при разбиении на ячейки АТМ помечаются этим адресом VPI/VCI и коммутируются без задержек на выполнение маршрутизации всеми коммутаторами магистрали АТМ, встречающимися на пути маршрута.

Ускорение передачи потоков данных достигается за счет сокращения времени образования виртуальных каналов в коммутаторах АТМ по сравнению со стандартной процедурой. Однако при этом коммутаторы, используемые на магистрали сети, уже трудно назвать стандартными АТМ-коммутаторами. Они выполняют и функции обычных маршрутизаторов, так как для обработки IP-пакетов строят таблицы маршрутизации, для чего поддерживают стандартные протоколы обмена маршрутной информацией, такие как RIP или OSPF. Они также поддерживают частный протокол распространения маршрутной информации по ATM-сети для построения таблиц VPI/VCI номеров потоков (например, протокол IFMP для коммутаторов Ipsilon). Для поддержки стандартных АТМ-сетей такие коммутаторы могут создавать виртуальные пути и стандартным способом, с помощью системы сигнализации, предложенной АТМ Forum.

К сожалению, техника ускоренной передачи IP-трафика через АТМ-магистрали остается пока нестандартной, хотя в комитет IETF поступило ряд предложений по выработке общего стандарта (в том числе и от пионера этого подхода Ipsilon, а также от лидера в области маршрутизации - компании Cisco, чья схема Tag-switching позволяет передавать через магистраль не только IP-пакеты, но и пакеты других популярных протоколов, например, IPX).

**2.5.4. Магистраль на базе технологии GigabitEthernet**

Четвертый вариантпостроения магистрали связан с использованием на ней новой технологии GigabitEthernet. При этом скорость отдельных частей магистрали может гибко подстраиваться под нужды трафика, так как наряду с GigabitEthernet может использоваться и FastEthernet.

GigabitEthernet, очевидно, будет сильным конкурентом для технологии АТМ при построении магистралей больших локальных сетей. Он превосходит существующие коммутаторы АТМ по битовой скорости (1000 Мб/c против 622 Мб/c), является более дешевым решением, и, кроме того, не требует существенного переобучения персонала. Основной недостаток технологии GigabitEthernet по сравнению с АТМ - как и ее предшественники Ethernet и FastEthernet, она не поддерживает такое понятие как качество обслуживания пользовательского трафика. Этот недостаток может быть существенным, если в сети действительно передаются чувствительные к задержкам данные. Если же основной поток данных составляют данные файлового сервиса (или аналогичного по требованиям к задержкам сервиса), то отсутствие гарантий качества обслуживания практически не будет сказываться на работе пользователей сети. Кроме того, высокая скорость передачи данных в какой-то степени компенсирует отсутствие механизмов гарантии пропускной способности и задержек, так как пакет 1500 байт передается через незагруженную магистраль GigabitEthernet всего за 12 мкс. При коэффициенте загрузки в 30% - 50%, характерном для многих сетей Ethernet, задержка будет составлять в среднем 30 мкс, что намного меньше уровня в 20 мс, при котором участники видеоконференции начинают замечать ухудшение качества изображения.

Тем не менее, приверженцы технологии GigabitEthernet заботятся и о поддержке качества обслуживания. Они рассчитывают использовать для этой цели такие внешние по отношению к этой технологии протоколы ,как протокол резервирования пропускной способности IP-маршрутизаторов для потоков данных RSVP, а также протоколы 802.1q и 802.1p, обеспечивающие приоритезацию трафика в локальных сетях на основе коммутаторов. Очевидно, что такая смесь различных протоколов хотя и улучшит обслуживание трафика разных классов, но не сможет конкурировать со стройной системой поддержки качества обслуживания в сетях АТМ.

Недостатком технологии GigabitEthernet является на сегодняшний день и тот факт, что принятие для нее окончательного стандарта планируется в комитете 802.3 института IEEE только в середине 1998 года, так что нет никаких гарантий, что выпускаемое сейчас различными производителями гигабитное оборудование будет полностью совместимо со стандартом.

**2.5.5. Сравнение различных вариантов построения магистрали крупной локальной сети**

В целом необходимо отметить, что выбор определенного способа построения магистрали корпоративной локальной сети сейчас представляет достаточно сложное дело. Очевиден только тот факт, что традиционно используемая на магистралях крупных локальных сетей технология FDDI начинает постепенно сдавать свои позиции. Свидетельством этому является опрос, проведенный SageResearchInc. среди 200 американских компаний, применяющих FDDI в своих корпоративных сетях. Вопрос звучал так: "Собирается ли ваша компания в ближайшие 3 года перейти на другую высокоскоростную технологию?". Результаты опроса, представленные на диаграмме (рис. 2.22.), показывают, что популярность FDDI падает, так как больше половины компаний хотят заменить оборудование FDDI на оборудование, поддерживающее другую высокоскоростную технологию. Возможно, популярность FDDI падала бы еще более быстрыми темпами, но присущая ей внутренняя отказоустойчивость, которой нет ни в сетях АТМ, ни в сетях FastEthernet или GigabitEthernet, продолжает привлекать сетевых администраторов и интеграторов.

Выбор же технологии для замены FDDI на магистрали пока не очевиден, особенно ввиду отсутствия стандарта на большинство перспективных вариантов. Во многом выбор определяется требованиями приложений к качеству обслуживания их трафика. Во многих случаях для не очень чувствительных к задержкам приложений хорошим выбором будет GigabitEthernet, а для трафика реального времени - АТМ, возможно с модификациями стандартного варианта технологии для ускоренной передачи стандартных потоков данных типа IP-switching или Tag-switching. При использовании GigabitEthernet достигается хорошая совместимость с подсетями Ethernet и FastEthernet, то есть с внутренними подсетями корпорации, а при использовании АТМ нет проблем при подключении к территориальной магистрали провайдера, также все чаще использующей технологию АТМ как основной вид транспорта.

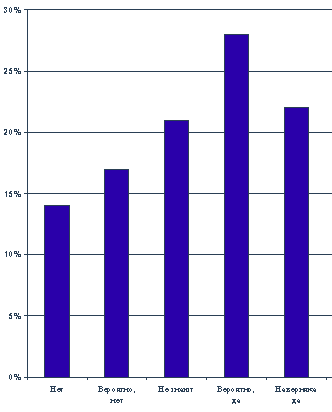


Рис. 2.22. Результаты опроса о замене технологии FDDI в ближайшие 3 года

Приведем основные доводы в пользу построения магистрали корпоративной сети на GigabitEthernet или АТМ двух авторитетных и заинтересованных собеседников. По просьбе редакции журнала DataCommunications о перспективах двух технологий спорят Джо Скорупа (JoeSko- rupa) - представитель одного из лидеров технологии ATM компании ForeSystems, и Джордж Продан (GeorgeProdan), сотрудник компании ExtremeNetworks, пионера технологии GigabitEthernet (DataCommunications, April 97).

Скорупа обосновывает хорошие шансы технологии АТМ ее способностью дать высокую и гарантированную пропускную способность для приложений различных типов, и критикует GigabitEthernet за отсутствие механизмов для предоставления потребителям определенных параметров пропускной способности, а также за то, что до появления стандартов еще нужно ждать еще как минимум год, в то время как продукты АТМ давно имеются на рынке. Продан в свою очередь приводит в пользу GigabitEthernet такие доводы, как плавность и легкость перехода, трехступенчатую иерархию скоростей 10 - 100 - 1000 (в сочетании с Ethernet и FastEthernet), обеспечение совместимости продуктов разных производителей за счет усилий GigabitEthernetAlliance, насчитывающем более чем 100 членов. Замечание Скорупы о том, что у продуктов FastEthernet есть проблемы с совместимостью даже через год после принятия стандарта, Продан парирует фактами о несовместимости АТМ-продуктов. Значительная часть дебатов посвящена проблеме обеспечения качества сервиса обеими технологиями. Скорупа не соглашается с утверждением Продана о том, что протокол RSVP сможет обеспечить в сетях GigabitEthernet требуемое качество обслуживания, так как он разработан совсем для других целей. Продан же в свою очередь считает, что нужное качество обслуживания для конечных пользователей дает в сетях АТМ только сервис ABR, а так как многие коммутаторы пока не поддерживают ABR, то о хорошем качестве обслуживания в сетях АТМ пока говорить рано. Дискуссия заканчивается выражением общего мнения о том, что технология GigabitEthernet будет играть заметную роль в ближайшем будущем, однако ее место в сетях собеседники видят по разному - Скорупа в качестве сети доступа к магистрали на основе АТМ, а Продан - в качестве самой магистрали.

### 2.6. Перспективы развития кабельных систем

Вопросы построения физической инфраструктуры сети, а именно кабельной системы, без всякого сомнения также могут быть отнесены к стратегическим моментам создания сети. Являясь фундаментом сети, кабельная система в конечном счете определяет предельно возможную пропускную способность, предоставляемую в распоряжение приложений. Не менее важной характеристикой сети является отказоустойчивость. Согласно зарубежным исследованиям (журнал LANTechnologies), 70% времени простоев обусловлено проблемами, возникшими вследствие низкого качества применяемых кабельных систем. Правильное проектирование кабельной системы является необходимым условием не только для достижения необходимой производительности и надежности сети, но и для обеспечения ее гибкости, способности к развитию. На самых первых этапах внедрения какого-либо новшества в сети руководитель проекта должен убедиться, допускает ли кабельная система подобные изменения.

Вот почему так важно правильно построить фундамент сети - кабельную систему. Задача создания эффективной кабельной системы все чаще решается путем использования структурированной кабельной системы. *Структурированная кабельная система* (StructuredCablingSystem, SCS) - это набор коммутационных элементов (кабелей, разъемов, коннекторов, кроссовых панелей и шкафов), а также методика их совместного использования, которая позволяет создавать регулярные, легко расширяемые структуры связей в вычислительных сетях. Если внутри здания или в пределах комплекса зданий установлена структурированная кабельная система, то путем перекоммутации кабелей в специальных кроссовых секциях и шкафах можно гибко и без больших дополнительных затрат приспосабливаться в течение 5 - 10 лет к изменяющейся структуре сети и появляющимся новым протоколам.

Кабельная система такого типа должна обладать некоторой долей избыточности. Так, в каждой комнате здания должно быть разведено достаточное количество оконечных розеток, к которым подключаются сетевые адаптеры компьютеров, даже если в настоящее время в таком количестве розеток и нет необходимости. Эти ненужные розетки могут никуда не подключаться, но быть подведенными к ближайшему кроссовому шкафу, чтобы подключиться к новому концентратору, когда это станет необходимо. Начальная избыточность структурированной кабельной системы окупится достаточно быстро, так как стоимость наращивания кабелей и розеток в действующей кабельной системе всегда выше, чем их установка в период установки всей кабельной системы. К избыточности приводит также желание получить ясную структуру кабельной системы, так как здесь иногда приходится жертвовать элегантным, но отклоняющимся от общей схемы решением, в пользу избыточного, но единообразного решения.

Преимущества структурированной кабельной системы:

* Универсальность. Структурированная кабельная система при продуманной организации может стать единой средой для передачи компьютерных данных в локальной вычислительной сети, организации локальной телефонной сети, передачи видеоинформации и даже передачи сигналов от датчиков пожарной безопасности или охранных систем. Это позволяет автоматизировать многие процессы по контролю, мониторингу и управлению хозяйственными службами и системами жизнеобеспечения.
* Увеличение срока службы. Срок морального старения хорошо структурированной кабельной системы может составлять 8-10 лет.
* Уменьшение стоимости добавления новых пользователей и изменения их мест размещения. Стоимость кабельной системы в основном определяется не стоимостью кабеля, а стоимостью работ по его прокладке (затраты на выполнение работ по инсталляции кабельной системы зачастую в 2-3 раза превосходят стоимость материалов и оборудования). Поэтому более выгодно провести однократную работу по прокладке кабеля, возможно с большим запасом по длине, чем несколько раз выполнять прокладку, наращивая длину кабеля. Это помогает быстро и дешево изменять структуру кабельной системы при перемещениях персонала или смене приложений.
* Возможность легкого расширения сети. Структурированная кабельная система является модульной, поэтому ее легко наращивать, что позволяет легко и ценой малых затрат переходить на более совершенное оборудование, удовлетворяющее растущим требованиям к системам коммуникаций.
* Обеспечение более эффективного обслуживания. Структурированная кабельная система облегчает обслуживание и поиск неисправностей по сравнению с шинной кабельной си- стемой.
* Надежность. Структурированная кабельная система имеет повышенную надежность поскольку обычно производство всех ее компонентов и техническое сопровождение осуществляется одной фирмой-производителем.

Большинство стандартных сетевых технологий, как старых (Ethernet, TokenRing, FDDI), так и новых (FastEthernet, 100VG-AnyLAN, ATM), использует три основных типа кабелей - неэкранированную витую пару (UTP), экранированную витую пару (STP) и многомодовый оптоволоконный кабель. В состав любой кабельной системы входят кабели различных типов, каждый из которых имеет свою область или области назначения.

Для определения областей назначения того или иного типа кабеля полезно выделять в кабельной системе отдельные подсистемы. В типичную иерархическую структурированную кабельную систему входят горизонтальные и вертикальные подсистемы, а также подсистема кампуса. *Горизонтальные подсистемы* работают в пределах отдела и соединяют кроссовый шкаф этажа с розетками пользователей. Подсистемы этого типа соответствуют этажам здания. *Вертикальные подсистемы* работают внутри здания, соединяют кроссовые шкафы каждого этажа с центральной аппаратной здания. *Кампусовская система*, работающая в пределах территории между зданиями, соединяет несколько зданий с главной аппаратной всего кампуса. Эта часть кабельной системы обычно называется backbone (или магистралью).

Помимо технических характеристик при выборе кабеля нужно учитывать, какая кабельная система уже установлена на вашем предприятии, и какие тенденции и перспективы существуют на рынке в данный момент.

В России СКС нашли коммерческое применение сравнительно недавно - в начале 90-х годов в результате рыночных реформ и появления частного бизнеса. До этого локальные сети и учрежденческие АТС, как правило, были не интегрированы друг с другом.

Хотя в настоящее время наибольшее число инсталляций СКС осуществляются в мелких локальных сетях (с числом ПК от 10 до 20), многие крупные предприятия, учреждения и банки в последние годы также проявляют заинтересованность в установке у себя структурированных кабельных систем. Благодаря отсутствию унаследованного оборудования, в России, как правило, используются самые современные системы от ведущих производителей. Наибольшая доля рынка принадлежит таким известным западным маркам, как LucentTechnologies, BICCBrand-Rex, MOD-TAP, Alcatel, Siemens, AMP, IBM, а также российским маркам, как "АйТи".

Среди российских компаний-системных интеграторов, способных реализовать крупные проекты кабельных систем, специалисты отмечают IBS, "АйТи", "Черус", "Демос", R-Style, "Ланит", LVC, "Крок", "Руслан". Они располагают отделами, специализирующимися по локальным сетям, структурированным кабельным системам и голосовой связи.

Основная доля рынка СКС принадлежит LucentTechnologies. Около 73% установленных систем используют проводку UTP.

Решения на основе экранированного кабеля связываются главным образом с системами TokenRing, но все большее число заказчиков в России предпочитает использовать эти кабели и в других сетях передачи данных. Коаксиальный кабель для горизонтальной проводки пользовался популярностью 2 года назад, но в последнее время он был вытеснен кабелями с витыми парами.

В последние два года существенные изменения произошли и в распределении рынка между кабелями разных категорий. Доля кабелей Категории 3 уменьшилась с двух третьих в 1994 до 37% в 1996 году.

Согласно прогнозам, весь рынок кабелей для передачи данных будет расти на 14,7% ежегодно, и в 2001 году его объем составит 29,9 млн. долларов. Наиболее быстрые темпы роста ожидаются на рынке STP - 33,6% ежегодно в денежном исчислении, далее следуют оптические кабели и UTP - 22,1% и 16,2% соответственно. Доля коаксиального кабеля будет неуклонно снижаться.

Сегодня в мире существуют три весьма сходных между собой стандарта на кабельные системы для ЛВС:

1. американский стандарт TIA/EIA 568A;
2. международный стандарт ISO/IEC 11801;
3. европейский стандарт EN50173.

Все эти стандарты кабельных систем определяет основные параметры неэкранированной витой пары UTP, экранированной витой пары STP и волоконно-оптического кабеля, то есть тех видов кабелей, которые покрывают все разнообразие физических уровней современных стандартов для локальных сетей.

Необходимо отметить, что стандарт EIA/TIA 568A относится только к сетевому кабелю. Но реальные системы, помимо кабеля, содержат также коннекторы, розетки, распределительные панели и др., т.е. все, что в совокупности составляет понятие кабельной системы**.** Использование только кабеля типа 5 не гарантирует создание кабельной системы этой категории. Все составные части кабельной системы также должны удовлетворять требованиям соответствующей категории, то есть работать без ухудшения электрических параметров передаваемых сигналов в заданных частотах. Важное значение имеет также технология инсталляции всех компонентов системы, нарушение которой приведет к снижению категории.

К числу наиболее распространенных стандартизованных структурированных кабельных систем относятся кабельная система CablingSystem, разработанная компанией IBM, кабельные системы PremisesDistributedSystem и SYSTIMAX, разработанные AT&T, а также кабельная система OPENDECсonnect корпорации DigitalEquipment. Различные производители дают своим кабельным системам различные названия, но общие принципы их организации остаются одинаковыми.

Наряду с кабельными системами, соответствующими этим стандартам, в настоящее время имеется ряд проектов новых широкополосных кабелей, например, медных кабелей категории 6. Появились сообщения о гарантированных скоростях передачи данных 350 и 622 Мб/с обеспечиваемых новыми продуктами. Но стандартов на новые широкополосные кабели пока не существует. Специалисты высказывают большие сомнения в рыночном успехе этого проекта: во-первых, проблема совместимости с огромным числом уже установленных кабельных систем категорий 3 и 5, во-вторых, высокая стоимость, сравнимая со стоимостью сетей на основе волоконно-оптического кабеля, в-третьих, необходимость надежного заземления, которую не всегда просто реализовать.

Еще далеко не исчерпаны возможности кабельных систем категории 5. Вполне вероятно, они смогут поддерживать и сети GigabitEthernet. Если же говорить о высокоскоростных сетях (ATM и GigabitEthernet на скоростях 622 и 1000 Мбит/с соответственно), то для их реализации хорошо подходит волоконно-оптический кабель.

Поэтому, если вы думаете над тем, как улучшить работу кабельной системы, то естественным решением является ее модернизация путем прокладки горизонтального кабеля (проводки на этажах) категории 5, а в качестве магистрали оптического кабеля. В любом случае необходимо получить гарантию на кабельную систему и предусмотреть возможность последующей ее модернизации в будущем.

# 4. Стратегии защиты данных

### 4.1. Комплексный подход - необходимое условие надежной защиты корпоративной сети

Построение и поддержка безопасной системы требует системного подхода. В соответствии с этим подходом прежде всего необходимо осознать весь спектр возможных угроз для конкретной сети и для каждой из этих угроз продумать тактику ее отражения. В этой борьбе можно и нужно использовать самые разноплановые средства и приемы - организационные и законодательные, административные и психологические, защитные возможности программных и аппаратных средств сети.

Законодательные средства защиты - это законы, постановления правительства и указы президента, нормативные акты и стандарты, которыми регламентируются правила использования и обработки информации ограниченного доступа, а также вводятся меры ответственности за нарушения этих правил.

Административные меры - это действия общего характера, предпринимаемые руководством предприятия или организации. Администрация предприятия должна определить *политику информационной безопасности*, которая включает ответы на следующие вопросы:

какую информацию и от кого следует защищать;

кому и какая информация требуется для выполнения служебных обязанностей;

какая степень защиты требуется для каждого вида информации;

чем грозит потеря того или иного вида информации;

как организовать работу по защите информации.

К организационным (или процедурным) мерам обеспечения безопасности относятся конкретные правила работы сотрудников предприятия, например, строго определенный порядок работы с конфиденциальной информацией на компьютере.

К морально-этическим средствам защиты можно отнести всевозможные нормы, которые сложились по мере распространения вычислительных средств в той или иной стране (например, Кодекс профессионального поведения членов Ассоциации пользователей компьютеров США).

К физическим средствам защиты относятся экранирование помещений для защиты от излучения, проверка поставляемой аппаратуры на соответствие ее спецификациям и отсутствие аппаратных "жучков" и т.д.

К техническим средствам обеспечения информационной безопасности могут быть отнесены:

системы контроля доступа, включающие средства аутентификации и авторизации пользователей;

средства аудита;

системы шифрования информации;

системы цифровой подписи, используемые для аутентификации документов;

средства доказательства целостности документов (использующие, например, дайджест-функции);

системы антивирусной защиты;

межсетевые экраны.

Все указанные выше средства обеспечения безопасности могут быть реализованы как в виде специально разработанных для этого продуктов (например, межсетевые экраны), так и в виде встроенных функций операционных систем, системных приложений, компьютеров и сетевых коммуникационных устройств.

### 4.2. Усугубление проблем безопасности при удаленном доступе. Защитные экраны - firewall'ы и proxy-серверы

Обеспечение безопасности данных при удаленном доступе - проблема если и не номер один, то, по крайней мере, номер два, после проблемы обеспечения приемлемой для пользователей пропускной способности. А при активном использовании транспорта Internet она становится проблемой номер один.

Неотъемлемым свойством систем удаленного доступа является наличие глобальных связей. По своей природе глобальные связи, простирающиеся на много десятков и тысяч километров, не позволяют воспрепятствовать злонамеренному доступу к передаваемым по этим линиям данным. Нельзя дать никаких гарантий, что в некоторой, недоступной для контроля точке пространства, некто, используя, например, анализатор протокола, не подключится к передающей среде для захвата и последующего декодирования пакетов данных. Такая опасность одинаково присуща всем видам территориальных каналов связи и не связана с тем, используются ли собственные, арендуемые каналы связи или услуги общедоступных территориальных сетей, подобные Internet.

Однако использование общественных сетей (речь в основном идет об Internet) еще более усугубляет ситуацию, хотя бы потому, что в такой сети для доступа к корпоративным данным в распоряжении злоумышленника имеются более разнообразные и удобные средства, чем выход в чистое поле с анализатором протоколов. Кроме того, огромное число пользователей увеличивает вероятность попыток несанкционированного доступа.

Безопасная система - это система, которая, во-первых, надежно хранит информацию и всегда готова предоставить ее своим пользователям, а во-вторых, система, которая защищает эти данные от несанкционированного доступа.

Межсетевой экран (firewall, брандмауэр) - это устройство, как правило, представляющее собой универсальный компьютер с установленным на нем специальным программным обеспечением, который размещается между защищаемой (внутренней) сетью и внешними сетями, потенциальными источниками опасности. Межсетевой экран контролирует все информационные потоки между внутренней и внешними сетями, пропуская данные, в соответствии с заранее установленными правилами. Эти правила являются формализованным выражением политики безопасности, принятой на данном предприятии.

Межсетевые экраны базируются на двух основных приемах защиты:

1. пакетной фильтрации;
2. сервисах-посредниках (proxyservices).

Эти две функции можно использовать как по отдельности, так и в комбинации.

**Пакетная фильтрация. Использование маршрутизаторов в качестве firewall**

Фильтрация осуществляется на транспортном уровне: все проходящие через межсетевой экран пакеты или кадры данных анализируются, и те из них, которые имеют в определенных полях заданные ("неразрешенные") значения, отбрасываются.

Пропуск во внутреннюю сеть пакетов сетевого уровня или кадров канального уровня по адресам (MAC-адреса, IP-адреса, IPX-адреса) или номерам портов TCP, соответствующих приложениям. Например, для того, чтобы трафик telnet не пересекал границу внутренней сети, межсетевой экран должен отфильтровывать все пакеты, в заголовке TCP которых указан адрес порта процесса-получателя, равный 23 (этот номер зарезервирован за сервисом telnet). Сложнее отслеживать трафик FTP, который работает с большим диапазоном возможных номеров портов, что требует задания более сложных правил фильтрации.

Конечно, для фильтрации пакетов может быть использован и обычный маршрутизатор, и действительно, в Internet 80% пакетных фильтров работают на базе маршрутизаторов. Однако маршрутизаторы не могут обеспечить ту степень защиты данных, которую гарантируют межсетевые экраны.

Главные преимущества фильтрации межсетевым экраном по сравнению с фильтрацией маршрутизатором состоят в следующем:

* межсетевой экран обладает гораздо более развитыми логическими способностями, поэтому он в отличие от маршрутизатора легко может, например, обнаружить обман по IP-адресу;
* у межсетевого экрана большие возможности аудита всех событий, связанных с безопасностью.

**Сервисы - посредники (Proxy-services)**

Сервисы-посредники не допускают возможности непосредственной передачи трафика между внутренней и внешней сетями. Для того, чтобы обратиться к удаленному сервису, клиент-пользователь внутренней сети устанавливает логическое соединение с сервисом-посредником, работающим на межсетевом экране. Сервис-посредник устанавливает отдельное соединение с "настоящим" сервисом, работающим на сервере внешней сети, получает от него ответ и передает по назначению клиенту - пользователю защищенной сети.

Для каждого сервиса необходима специальная программа: сервис-посредник. Обычно, защитный экран включает сервисы-посредники для FTP, HTTP, telnet. Многие защитные экраны имеют средства для создания программ-посредников для других сервисов. Некоторые реализации сервисов-посредников требуют наличия на клиенте специального программного обеспечения. Пример: Sock - широко применяемый набор инструментальных средств для создания программ-посредников.

Сервисы-посредники не только пересылают запросы на услуги, например, разработанный CERN сервис-посредник, работающий по протоколу HTTP, может накапливать данные в кэше межсетевого экрана, так что пользователи внутренней сети могут получать данные с гораздо меньшим временем доступа.

Журналы событий, поддерживаемые сервисами-посредниками, могут помочь предупредить вторжение на основании записей о регулярных неудачных попытках. Еще одним важным свойством сервисов-посредников, положительно сказывающимся на безопасности системы, является то, что при отказе межсетевого экрана защищаемый посредником сервис-оригинал остается недоступным.

Трансляция сетевых адресов - новый вид сервиса-посредника. Трансляторы адресов заменяют "внешние" IP-адреса серверов своих сетей на "внутренние". При таком подходе топология внутренней сети скрыта от внешних пользователей, вся сеть может быть представлена для них одним-единственным IP-адресом. Такая непрозрачность сети усложняет задачу несанкционированного доступа. Кроме этого, трансляция адресов дает еще одно преимущество - позволяет иметь внутри сети собственную систему адресации, не согласованную с Internet, что снимает проблему дефицита IP-адресов.

Сервисы-посредники намного надежнее фильтров, однако они снижают производительность обмена данными между внутренней и внешней сетями, они также не обладают той степенью прозрачности для приложений и конечных пользователей, которая характерна для фильтров.

### 4.3. Использование сертификатов для аутентификации массовых пользователей при ведении бизнеса через Internetи другие публичные сети

Обеспечение безопасности при работе в Internet стало особенно важной проблемой в условиях массового интереса к построению частных виртуальных сетей с использованием транспортных средств Internet, а также использования методов Internet для хранения, представления и поиска информации в локальных сетях предприятий. Все это можно назвать одним словом - intranet. Специфика Internet сказывается и на используемых средствах обеспечения безопасности. Остановимся на некоторых из них.

При организации доступа к некоторым ресурсам Internet все чаще возникает необходимость во введении некоторых ограничений. Это означает, что среди множества пользователей Internet, владелец ресурса должен определить некоторые правила определения тех, кому доступ разрешен, и предоставить им способ, с помощью которого они могли бы доказывать свою принадлежность к легальным пользователям. Следовательно, необходима процедура аутентификация, пригодная для использования в Internet.

Аутентификация с применением сертификатов является альтернативой использованию паролей и представляется естественным решением в условиях, когда число пользователей сети измеряется миллионами, что мы имеем в Internet. В таких обстоятельствах процедура предварительной регистрации пользователей, связанная с назначением и хранением их паролей становится крайне обременительной, опасной, а иногда и просто нереализуемой. При использовании сертификатов сеть, которая дает пользователю доступ к своим ресурсам, не хранит никакой информации о своих пользователях - они ее предоставляют сами в своих запросах в виде сертификатов, удостоверяющих личность пользователей. Сертификаты выдаются специальными уполномоченными организациями - центрами сертификации. Поэтому задача хранения секретной информации (закрытых ключей) возлагается теперь на самих пользователей, что делает это решение гораздо более масштабируемым, чем вариант с использованием паролей.

Аутентификация личности на основе сертификатов происходит примерно так же, как на проходной большого предприятия. Вахтер пропускает людей на территорию на основании пропуска, который содержит фотографию и подпись сотрудника, удостоверенных печатью предприятия и подписью лица, выдавшего пропуск. Сертификат является аналогом пропуска и выдается по запросам специальными сертифицирующими центрами при выполнении определенных условий. Он представляет собой электронную форму, в которой имеются такие поля, как имя владельца, наименование организации, выдавшей сертификат, открытый ключ владельца. Кроме того, сертификат содержит электронную подпись выдавшей организации - зашифрованные закрытым ключом этой организации все остальные поля сертификата.

Использование сертификатов основано на предположении, что сертифицирующих организаций немного, и их открытые ключи могут быть всем известны каким-либо способом, например, с помощью тех же публикаций в журналах.

Когда пользователь хочет подтвердить свою личность, он предъявляет свой сертификат в двух формах - открытой, то есть такой, в которой он получил его в сертифицирующей организации, и в зашифрованной с применением своего закрытого ключа. Сторона, проводящая аутентификацию, берет из открытого сертификата открытый ключ пользователя и расшифровывает с помощью него зашифрованный сертификат. Совпадение результата с открытым сертификатом подтверждает факт, что предъявитель действительно является владельцем закрытого ключа, парного с указанным открытым.

Затем с помощью известного открытого ключа указанной в сертификате организации проводится расшифровка подписи этой организации в сертификате. Если в результате получается тот же сертификат с тем же именем пользователя и его открытым ключом - значит он действительно прошел регистрацию в сертификационном центре, является тем, за кого себя выдает, и указанный в сертификате открытый ключ действительно принадлежит ему.

Сертификаты можно использовать не только для аутентификации, но и для предоставления избирательных прав доступа. Для этого в сертификат могут вводиться дополнительные поля, в которых указывается принадлежность его владельца к той или иной категории пользователей. Эта категория указывается сертифицирующей организацией в зависимости от условий, на которых выдается сертификат. Например, организация, поставляющая через Internet на коммерческой основе информацию, может выдавать сертификаты определенной категории пользователям, оплатившим годовую подписку на некоторый бюллетень, а Web-сервер будет предоставлять доступ к страницам бюллетеня только пользователям, предъявившим сертификат данной категории.

При использовании сертификатов отпадает необходимость хранить на серверах корпораций списки пользователей с их паролями, вместо этого достаточно иметь на сервере список имен и открытых ключей сертифицирующих организаций. Может также понадобится некоторый механизм отображений категорий владельцев сертификатов на традиционные группы пользователей для того, чтобы можно было использовать в неизменном виде механизмы управления избирательным доступом большинства операционных систем или приложений.

Механизм получения пользователем сертификата хорошо автоматизируется в сети в модели клиент-сервер, когда браузер выполняет роль клиента, а в сертифицирующей организации установлен специальный сервер выдачи сертификатов. Браузер вырабатывает для пользователя пару ключей, оставляет закрытый ключ у себя и передает частично заполненную форму сертификата серверу. Для того, чтобы неподписанный еще сертификат нельзя было подменить при передаче по сети, браузер зашифровывает сертификат выработанным закрытым ключом. Сервер сертификатов подписывает полученный сертификат, фиксирует его в своей базе данных и возвращает его каким-либо способом владельцу. Очевидно, что при этом может выполняться еще и неформальная процедура подтверждения пользователем своей личности и права на получение сертификата, требующая участия оператора сервера сертификатов. Это могут быть доказательства оплаты услуги, доказательства принадлежности к той или иной организации - все случаи жизни предусмотреть и автоматизировать нельзя.

После получения сертификата браузер хранит его вместе с закрытым ключом и использует при аутентификации на тех серверах, которые поддерживают такой процесс.

В настоящее время существует уже большое количество протоколов и продуктов, использующих сертификаты. Например, компания NetscapeCommunications поддерживает сертификаты стандарта X.509 в браузерах NetscapeNavigator и своих информационных серверах, а также выпустила сервер сертификатов, который организации могут у себя устанавливать для выпуска своих собственных сертификатов. Microsoft реализовала поддержку сертификатов в версии InternetExplorer 3.0 и в сервере InternetInformationServer.

### 4.4. Технологии защищенного канала

*Технология защищенного канала* призвана обеспечивать безопасность передачи данных по открытой транспортной сети, например, по сети Internet. Защищенный канал включает в себя выполнение трех основных функций:

1. взаимная аутентификация абонентов;
2. защита передаваемых по каналу сообщений от несанкционированного доступа;
3. подтверждение целостности поступающих по каналу сообщений.

Взаимная аутентификация обеих сторон при установлении соединения может быть выполнена, например, путем обмена сертификатами.

Секретность может быть обеспечена каким-либо методом шифрации, например, передаваемые сообщения шифруются с использованием симметричных сессионных ключей, которыми стороны обмениваются при установлении соединения. Сессионные ключи передаются также в зашифрованном виде, при этом они шифруются с помощью открытых ключей. Использование для защиты сообщений симметричных ключей связано с тем, что скорость процессов шифрации и дешифрации на основе симметричного ключа существенно выше, чем при использовании несимметричных ключей.

Целостность передаваемых сообщений достигается за счет того, что к сообщению (еще до его шифрации сессионным ключом) добавляется дайджест, полученный в результате применения односторонней функции к тексту сообщения.

Защищенный канал в публичной сети часто называют также виртуальной частной сетью - VirtualPrivateNetwork, VPN. Существует два способа образования VPN (рисунок 4.1):

1. с помощью специального программного обеспечения конечных узлов;
2. с помощью специального программного обеспечения шлюзов, стоящих на границе между частной и публичной сетями.

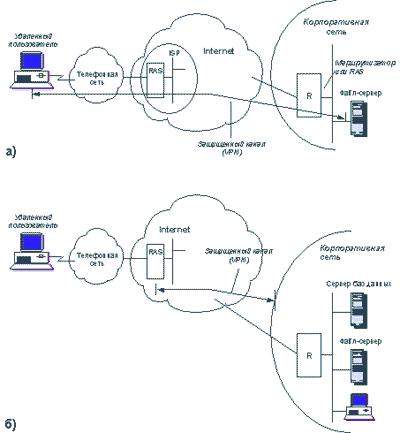


Рис. 4.1. VPN - частные виртуальные сети

В первом случае (рисунок 4.1, a) программное обеспечение, установленное на компьютере удаленного клиента, устанавливает защищенный канал с сервером корпоративной сети, к ресурсам которого клиент обращается. Преимуществом этого подхода является полная защищенность канала вдоль всего пути следования, а также возможность использования любых протоколов создания защищенных каналов, лишь бы на конечных точках канала поддерживался один и тот же протокол. Недостатки заключаются в избыточности и децентрализованности решения. Избыточность состоит в том, что уязвимыми для злоумышленников обычно являются сети с коммутацией пакетов, а не каналы телефонной сети или выделенные каналы. Поэтому установка специального программного обеспечения на каждый клиентский компьютер и каждый сервер корпоративной сети не является необходимой. Децентрализация процедур создания защищенных каналов не позволяет вести централизованное управление доступом к ресурсам сети. В большой сети необходимость отдельного администрирования каждого сервера и каждого клиентского компьютера с целью конфигурирования в них средств защиты данных - это трудоемкая процедура.

Во втором случае клиенты и серверы не участвуют в создании защищенного канала - он прокладывается только внутри публичной сети с коммутацией пакетов, например внутри Internet. Канал создается между сервером удаленного доступа провайдера услуг публичной сети и пограничным маршрутизатором корпоративной сети. Это хорошо масштабируемое решение, управляемого централизованно как администратором корпоративной сети, так и администратором сети провайдера. Для клиентских компьютеров и серверов корпоративной сети канал прозрачен - программное обеспечение этих конечных узлов остается без изменений. Реализация этого подхода сложнее - нужен стандартный протокол образования защищенного канала, требуется установка программного обеспечения, поддерживающего такой протокол, у всех провайдеров, необходима поддержка протокола производителями серверов удаленного доступа и маршрутизаторов.

### 4.5. Правовая регламентация деятельности в области защиты информации

Правовая регламентация деятельности в области защиты информации имеет целью защиту информации, составляющей государственную тайну, обеспечение прав потребителей на получение качественных продуктов, защиту конституционных прав граждан на сохранение личной тайны, борьбу с организованной преступностью.

Те, кто несут ответственность за корпоративную сеть, при приобретении тех или иных продуктов обеспечения безопасности, особенно связанных с шифрацией, должны выяснить некоторые правовые вопросы, например, может ли данный продукт быть экспортирован в другие страны или импортирован из других стран.

Регламентация может выражаться в следующей форме:

* обязательное лицензирование некоторых видов деятельности;
* необходимость иметь разрешение на некоторые виды деятельности;
* требование сертификации некоторых видов продуктов.

*Лицензия* является официальным документом, который разрешает осуществление на определенных условиях конкретного вида деятельности в течение установленного срока.

*Разрешение* выдается на некоторые виды разовых работ, независимо имеется ли у данной организации лицензия. Например, организация которая имеет лицензию на разработку шифровальных средств, должна получить разрешение на их экспорт.

*Сертификат* - официальный документ, удостоверяющий, что продукт прошел тестирование и соответствует требованиям нормативных документов.

В настоящее время правовая база, регулирующая отношения субъектов в области защиты информации, включает следующие основные документы:

* *Конституция РФ*, согласно которой "каждый гражданин имеет право заниматься любыми видами деятельности свободно и по своему выбору, кроме тех, которые запрещены законом, или для занятия которыми требуется наличие специального разрешения (лицензии).
* Гражданский кодекс РФ, в Части первой которого в ст.49 говорится: "Отдельными видами деятельности, перечень которых определяется законом, юридическое лицо может заниматься только на основании специального разрешения (лицензии)".
* Федеральные законы РФ ("Об информации, информатизации и защите информации" от 25 января 1995 года, "О государственной тайне" и др.).
* Постановление правительства РФ (например, постановление N 758 "О мерах по совершенствованию государственного регулирования экспорта товаров и услуг" от 1 июля 1994 г., постановление N1418 от 24.12.94 и др.).
* Указы президента (например, указ N1268 "О контроле за экспортом из Российской Федерации товаров и технологий двойного назначения" от 26 августа 1996 года, указ N334 "О мерах по соблюдению законности в области разработки, производства, реализации и эксплуатации шифровальных средств, а также услуг в области шифрования информации" от 3 апреля 1995 года.).
* Уголовный кодекс, в котором в разделе IX главы 28 "Преступления в сфере компьютерной информации" предусматривается наказания за "уничтожение, блокирование, модификацию и копирование информации, нарушение работы ЭВМ, системы ЭВМ или их сети".

С последними законодательными актами в области безопасности можно ознакомиться на сервере www.alpha.ru.

Работу по регламентации деятельности в области защиты информации проводит также Государственная техническая комиссия при президенте Российской федерации, издающая так называемые руководящие документы (РД). Один из этих РД, например, оценивает степень защищенности межсетевых экранов (firewall).

**Виды деятельности, лицензии на которые выдаются ФАПСИ:**

* В области шифровальных средств:
* разработка;
* производство;
* монтаж, наладка и установка;
* ремонт и сервисное обслуживание;
* реализация;
* предоставление услуг по шифрованию;
* предоставление консультационных услуг;
* эксплуатация.
* Те же виды деятельности, относящиеся к **системам, использующим шифровальные средства и предназначенным для телекоммуникаций.** Лицензии должны получать все предприятия и организации, независимо от их ведомственной принадлежности и прав собственности.
* Проведение сертификационных испытаний.
* Проведение работ по выявлению электронных устройств перехвата информации в технических средствах предприятий, банков и других учреждений, расположенных в Российской Федерации, независимо от их ведомственной принадлежности и форм собственности, на которых осуществляется обработка информации, составляющей государственную тайну.

**Виды деятельности, на которые выдаются разрешения ФАПСИ:**

* экспорт и импорт шифровальных средств, предназначенных для использования при обработке, хранении и передаче информации по каналам связи;
* экспорт и импорт закрытых (с помощью шифровальных средств) систем и комплексов телекоммуникаций;
* экспорт услуг в области шифрования;
* открытие учебных специальностей, курсов для организаций, имеющих лицензию на работу по подготовке кадров;

**Виды деятельности, на которые не нужны лицензии и разрешения ФАПСИ:**

* эксплуатация шифровальных средств физическими лицами и негосударственными организациями для защиты информации, не составляющей государственную тайну, во внутренних сетях без выхода в сети общего пользования или для связи с зарубежными партнерами;
* выявление электронных устройств перехвата информации в помещениях и устройствах негосударственных предприятий, если это не связано с обработкой информации, составляющей государственную тайну;
* издательская, рекламная и выставочная деятельность.

### 4.6. Продукты, сертифицированные для использования в России

Пока существует не так уж много продуктов, имеющих сертификат Гостехкомиссии, удостоверяющий пригодность этого продукта для применения в России в целях защиты данных. Ниже приводится описание некоторых из них.

**4.6.1. BlackHole компании MilkywayNetworks**

**BlackHole** (продукт компании MilkywayNetworks) - это firewall, работающий на proxy-серверах протоколов прикладного уровня (TELNET, FTP и т.д.). Он служит для разграничения доступа между локальной и глобальной сетью (INTERNET) или между двумя подразделениями локальной сети (ИНТРАНЕТ). BlackHole построен на принципе "Все что не разрешено, запрещено", т.е. любой вид доступа должен быть описан явно.

BlackHole поддерживает TELNET, FTP, Web-сервис, почту SMTP и некоторые другие виды сервисов.

Кроме этого, в состав BlackHole входят proxy-сервер уровня TCP и proxy сервер для UDP протокола. BlackHole осуществляет мониторинг всех 65 535 TCP/UDP портов и сбор статистики по попыткам доступа к этим портам. Правила доступа могут использовать в качестве параметров адрес источника, адрес назначения, вид сервиса (FTP, TELNET, и т.д.), дату и время доступа, идентификатор и пароль пользователя.

BlackHole поддерживает строгую аутентификацию как с использованием обычных паролей, так и различных типов одноразовых паролей S/Key, EnigmaLogicSafeword, SecurityDynamicsSecureID, при этом аутентификация может быть включена для любого вида сервиса.

Система выдачи предупреждений о попытках НСД в реальном времени в BlackHole позволяет администратору системы описывать опасные события и реакцию на них системы (вывод на консоль, звонок на пейджер и т.д.).

BlackHole предоставляет сервис для создания групп пользователей, сервисов, хостов и сетей и позволяет создавать правила для этих групп, что дает возможность легко описывать и администрировать большое количество пользователей.

BlackHole имеет удобную и дружественную графическую оболочку под X-Windows, так что настройкой системы может заниматься неискушенный в UNIX человек. Все административные функции могут быть выполнены из этой оболочки. Ядро ОС модифицировано для защиты графического интерфейса от внешнего доступа.

BlackHole предоставляет следующие возможности по конвертации адреса источника пакета при прохождении через firewall:

* адрес может быть заменен на адрес firewall;
* адрес может быть оставлен без изменения;
* адрес может быть заменен на выбранный администратором.

Последняя опция позволяет для пользователей INTERNET представлять внутреннюю сеть как состоящую из набора подсетей.

Для хранения и обработки статистической информации BlackHole использует реляционную базу данных с языком запросов SQL. BlackHole функционирует на PC и SunSparc платформах под управлением модифицированных версий операционных систем BSDI и SunOS.

BlackHole имеет сертификат Национального Агентства Компьютерной Безопасности США (NCSA), гарантирующий его высокую надежность и уровень защиты, но, что еще более важно этот продукт сертифицирован Государственной Технической Комиссией при Президенте России. Ниже приводится выдержка из сертификата N79:

"Выдан 30 января 1997г.

Действителен до 30 января 2000г."

Автоматизированная система разграничения доступа BlackHole версии BSDI-OS, функционирующая под управлением операционной системы BSDIBSD/OSv2.1, является программным средством защиты информации от НСД в сетях передачи данных по протоколу TCP/IP, обеспечивает защиту информационных ресурсов защищаемого участка локальной сети от доступа извне, не снижая уровня защищенности участка локальной сети, соответствует "Техническим условиям на Автоматизированную систему разграничения доступа BlackHole версии BSDI-OS" N 5-97 и требованиям Руководящего документа Гостехкомиссии России "Автомати- зированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации" в части администрирования для класса 3Б."

**4.6.2. Пандора (Gaumtlet) компании TIS**

Система **Пандора**, имеющая оригинальное название **Gauntlet,** это firewall (межсетевой экран), разработанный фирмой TIS. Этот продукт предоставляет возможность безопасного доступа и сетевых операций между закрытой, защищенной сетью и публичной сетью типа Internet.

Gauntlet представляет собой наиболее эффективный с точки зрения защиты вариант firewall - фильтр на уровне приложений, при этом обеспечивает максимальную прозрачность при использовании, возможность создания VPN и простое управление всем этим. Этот firewall позволяет пользоваться только теми протоколами, которые описал оператор и только в случае их безопасного использования.

Безопасность обеспечивается при работе через firewall в обоих направлениях в соответствии с политикой безопасности, определенной для данной организации. Продукт доступен в предустановленном виде на Pentium машинах, а также как отдельный пакет для BSD/OS, SunOS4\*, HP-UX, IRIX и других UNIX-систем. Открытый продукт фирмы - FWTK - на сегодняшний день является стандартом де-факто для построения firewall на уровне приложений.

Система Gauntlet получила сертификат Национального Агентства Компьютерной Безопасности США и была проверена Агентством Национальной безопасности США (NCSA). Продукт имеет сертификат Гостехкомиссии при президенте РФ номер 73, выдан 16 января 1997 года, действителен 3 года. На основании сертификационных испытаний системе присвоен класс 3Б.

**4.6.3. Застава (Sunscreen) компании "Элвис -Плюс"**

24 сентября 1997 года компания "Элвис-Плюс" анонсировала завершение бета-тестирования и начало промышленных поставок первого отечественного программного firewall'а "**Застава**". Одновременно объявлено о начале проведения сертификационных испытаний продукта Гостехкомиссией при Президенте РФ. К концу года будет предложена версия "Заставы" для платформы WindowsNT.

"Застава" относится к категории фильтрующих firewall'ов и функционирует на платформах Solaris/SPARC и Solaris/Intel и предназначен для использования в качестве первого эшелона защиты ресурсов корпоративных сетей от вторжения из Internet.

В настоящее время лабораторией средств сетевой безопасности "Элвис-Плюс" ведется интенсивная подготовка к производству аппаратно-программного варианта "Заставы" на базе системной платы SPARCengineUltraAXMotherboard производства компании SunMicroelectronics, по отношению к которой "Элвис-Плюс" выступает в качестве OEM-производителя. Гораздо более производительная по сравнению с программным аналогом новая версия "Заставы" будет включать модуль создания корпоративных VPN на базе протокола SKIP, а также поддерживать защищенное конфигурирование и управление с удаленного рабочего места администратора безопасности, реализованного по технологии Java.

По мнению сотрудников компании "Элвис-Плюс", семейство "Застава" может с успехом использоваться при создании комплексных систем защиты распределенных сетей федеральных агентств и ведомств при условии интеграции в продукты компании криптографических модулей легальных российских производителей.

Разработчики межсетевого экрана "Застава" ставили перед собой задачу не просто создать первый российский firewall, но, прежде всего, учесть быстрорастущие требования и современные приоритеты рынка защиты информации в IP сетях. Главными задачами при разработке межсетевого экрана были обеспечение "жесткой" и быстрой фильтрации, а также реализация эффективного proxy-модуля для обработки трафика электронной почты, представляющей в современных условиях серьезную потенциальную угрозу безопасности сети - по сравнению с прочими "внешними" сервисами.

На сервере компании "Элвис-Плюс" по адресу http://www.elvis.ru можно найти демонстрационные копии firewall'а "Застава".

**4.6.4. Продукты компании АНКАД**

Фирма "АНКАД" предлагает семейство программно-аппаратных средств криптографической защиты информации **КРИПТОН**, использующее ключ длиной 256 бит и алгоритмы шифрования и электронной подписи, которые соответствуют российским стандартами. В частности, шифр оплаты серии КРИПТОН для IBM-совместимых компьютеров широко применяются для защиты данных при передаче по открытым каналам связи.

Московская фирма "АНКАД" недавно получила от Федерального агентства правительственной связи и информации (ФАПСИ) лицензию на деятельность в сфере защиты информации. Лицензия предоставляет право заниматься разработкой, производством и реализацией средств криптографической защиты информации в коммерческом и государственном секторах экономики, включая работы в интересах зарубежных заказчиков. Эта деятельность осуществляется в рамках научно-технического сотрудничества и по согласованию с ФАПСИ.

# 6. Борьба сетевых операционных систем за корпоративный рынок

### 6.1. Что такое сетевая операционная система

Прежде, чем приступить к обсуждению вопросов о положении на рынке сетевых операционных систем и о проблеме их выбора, надо договориться о том, что собственно понимается под термином "сетевая операционная система". Одни считают, что это операционная система со встроенными сетевыми функциями, позволяющим пользователям совместно использовать ресурсы сети. Другие полагают, что сетевая операционная система - это просто набор сетевых служб, способных согласованно работать в общей операционной среде. При этом не имеет особого значения, входит ли эта служба в состав дистрибутива операционной системы или приобретена отдельно, поставляется ли она в виде дополнительного динамически загружаемого модуля ОС или в виде обычного приложения, разработана ли данная служба компанией-производителем ОС или какой-либо третьей фирмой. В многозначности термина "сетевая ОС" нет ничего страшного, важно только каждый раз, употребляя его отдавать себе отчет, что под этим понимается.

Каждая сетевая служба предоставляет пользователям сети некоторый вид сервиса, как правило, связанный с доступом к ресурсам сети. Например, файловый сервис - обеспечивает доступ пользователей сети к разделяемым файлам сети, факс- и принт-сервис - доступ к принтеру и факсу соответственно, сервис удаленного доступа - позволяет пользователям, связанным с основной сетью коммутируемыми каналами, получать доступ ко всем ресурсам сети, сервис электронной почты - предоставляет пользователям сети возможность обмениваться сообщениями.

Среди сетевых служб можно выделить такие, которые в основном ориентированы не на простого пользователя, а на администратора. Такие службы необходимы для организации правильной работы сети в целом, например, служба администрирования учетных записей о пользователях DomainUserManager в WindowsNT, которая позволяет администратору вести общую базу данных о пользователях сети. Более прогрессивным является подход с созданием централизованной справочной службы, или по-другому службы каталогов, которая предназначена для ведения базы данных не только обо всех пользователях сети, но и обо всех ее программных и аппаратных компонентах. В качестве примеров службы каталогов часто приводятся NDS компании Novell и StreetTalk компании Banyan. Другими примерами сетевых служб, предоставляющих сервис администратору, являются служба мониторинга сети, позволяющая захватывать и анализировать сетевой трафик, служба безопасности, в функции которой может входить в частности выполнение процедуры логического входа с проверкой пароля, служба резервного копирования и архивирования.

Если сетевые службы встроены в операционную систему, то есть рассматриваются как неотъемлемые ее части, то такая операционная система называется сетевой. Например, сетевая ОС WindowsNT, Unix, NetWare, OS/2 Warp. Все внутренние механизмы такой операционной системы оптимизированы для выполнения сетевых функций.

Другой вариант реализации сетевых служб - объединение их в виде некоторого набора (оболочки), при этом все службы такого набора должны быть между собой согласованы, то есть в своей работе они могут обращаться друг к другу, могут иметь в своем составе общие компоненты, например, общую подсистему аутентификации пользователей или единый пользовательский интерфейс. Для работы оболочки необходимо наличие некоторой локальной операционной системы, которая бы выполняла обычные функции, необходимые для управления аппаратурой компьютера, и в среде которой выполнялись бы сетевые службы, составляющие эту оболочку. Примером сетевой оболочки служат, например, LANServer, LANManager.

Естественно, что оболочка должна строится с учетом специфики той операционной системы, над которой она будет работать. Так LANServer, например, существует в различных вариантах: для работы над операционными системами VMS, VM, OS/400, AIX, OS/2.

Сетевые оболочки часто подразделяются на клиентские и серверные. Поскольку при реализации любого сетевого сервиса естественно возникает источник запросов (клиент) и исполнитель запросов (сервер), то и любая сетевая служба содержит в своем составе две несимметричные части - клиентскую и серверную. Оболочка, которая преимущественно содержит клиентские части сетевых служб называется клиентской. Например, типичным набором программного обеспечения рабочей станции в сети NetWare является MSDOS с установленной над ней клиентской оболочкой NetWare, состоящей из клиентских частей файлового сервиса и сервиса печати, а также компоненты, поддерживающие пользовательский интерфейс.

Серверная сетевая оболочка, примерами которой могут служить тот же LANServer и LANManager, а также NetWareforUnix, FileandPrintServiceforNetWare, ориентирована на выполнение серверных функций. Серверная оболочка, как минимум содержит серверные компоненты двух основных сетевых сервисов - файлового сервиса и печати, именно такой набор серверов реализован в упомянутых выше NetWareforUnix и FileandPrintServiceforNetWare. Некоторые же оболочки содержат настолько широкий набор сетевых служб, что их называют сетевыми операционными системами. Так, ни один обзор сетевых операционных систем не будет достаточно полным, если в нем отсутствует информация о LANServer, LANManager, ENS, являющихся сетевыми оболочками.

С одним типом ресурсов могут быть связаны разные сервисы, отличающиеся протоколом взаимодействия клиентских и серверных частей. Так, например, встроенный файловый сервис в WindowsNT реализует протокол SMB, используемый во всех ОС компании Microsoft, а дополнительный файловый сервис, входящий в состав оболочки FileandPrintServiceforNetWare для этой же WindowsNT, работает по протоколу NCP, "родному" для сетей NetWare. Кроме того, в стандартную поставку WindowsNT входит сервер FTP, реализующий файловый сервис Unix-систем. Ничто не мешает приобрести и установить для работы в среде WindowsNT и другие файловые сервисы, такие, например, как NFS, кстати имеющий несколько реализаций, выполненных разными фирмами. Наличие нескольких видов файлового сервиса, позволяет работать в сети приложениям, разработанным для разных операционных систем.

Сетевые оболочки создаются как для локальных операционных систем, так и для сетевых операционных систем. Действительно, почему бы не дополнить набор сетевых служб, встроенных в сетевую ОС, другими службами, составляющими некоторую сетевую оболочку. Например, сетевая оболочка ENS (EnterpriseNetworkServices) - содержащая базовый набор сетевых служб BanyanVines, может работать над сетевыми ОС Unix и NetWare (конечно, для каждой из этих операционных систем имеется соответствующий вариант ENS).

Существует и третий способ реализации сетевой службы - в виде отдельного продукта. Например, сервер удаленного управления WinFrame - продукт компании Citrix, предназначен для работы в среде WindowsNT, он дополняет возможности встроенного в WindowsNT сервера удаленного доступа RemoteAccessServer. Аналогичную службу удаленного доступа для NetWare также можно приобрести отдельно, купив программу NetWareConnect.

С течением времени сетевая служба может получить разные формы реализации. Так, например, компания Novell планирует поставлять справочную службу NDS, первоначально встроенную в сетевую ОС NetWare, для других ОС, для чего эта служба будет переписана в виде отдельных продуктов, каждый из которых будет учитывать специфику соответствующей ОС. Уже имеются версии NDS для работы в средах SCOUnix и HP-UX, а к концу года ожидаются версии для Solaris 2.5 и WindowsNT. А справочная служба StreetTalk уже давно существует и в виде встроенного модуля сетевой ОС BayanVines, и в составе оболочки ENS, и в виде отдельного продукта для различных операционных систем.

Таким образом, выбор сетевой операционной системы сводится к анализу возможностей всей совокупности сетевых служб (не только встроенных в ОС), способных работать в этой операционной среде. Насколько функционально полон этот набор, насколько он удовлетворяет требованиям пользователей и администраторов сети по производительности, по удобству использования, по безопасности - от всего этого и зависит выбор сетевой операционной системы.

### 6.2. Критерии выбора корпоративной ОС

Сетевые ОС могут быть разделены на две группы: масштаба отдела и масштаба предприятия (корпоративные ОС). От операционной системы отдела требуется, чтобы она обеспечивала некоторый набор сетевых сервисов, включая разделение файлов, приложений и принтеров. Она также должна обеспечивать свойства отказоустойчивости, такие как зеркальное отображение серверов и зеркальное отображение дисков. Обычно сетевые ОС отделов более просты в установке и управлении по сравнению с сетевыми ОС предприятия, но у них меньше функциональных свойств, они меньше защищают данные и имеют более слабые возможности по взаимодействию с другими типами сетей, а также худшую производительность. К числу наиболее популярных ОС для сетей отделов и рабочих групп могут быть отнесены ОС NetWare 3.x, PersonalWare, ArtisoftLANtastic.

В качестве корпоративных операционных систем чаще всего называют такие сетевые ОС, как BanyanVines, NovellNetWare 4.x, IBMLANServer, MicrosoftLANManager и WindowsNTServer, SunNFS, SolarisUnix и многие другие ОС семейства Unix.

Среди основных требований, которым должна отвечать корпоративная ОС можно указать следующие:

* функциональная полнота - разнообразие поддерживаемых сервисов;
* производительность - запросы к серверам должны обрабатываться с преемлемым уровнем задержек;
* масштабируемость - характеристики производительности сетевой ОС должны сохраняться неизменными в широком диапазоне изменения параметров системы, то есть сеть должна хорошо работать, и тогда, когда число пользователей и рабочих станций измеряется тысячами, число серверов - сотнями, объемы обрабатываемой информации - терабайтами;
* возможность работы на мощной аппаратной платформе: поддержка многопроцессорности, больших объемов оперативной и внешней памяти, а также широкой номенклатуры внешних устройств, включая разнообразные виды глобальных связей;
* способность работать в гетерогенной среде: поддержка разных стеков коммуника- ционных протоколов, поддержка разнообразных ОС на рабочих станциях, наличие средств взаимодействия с сервисами других сетевых ОС, аппаратная многоплатфор- менность, то есть способность работы на компьютерах разных типов;
* возможность использования в качестве сервера приложений, наличие большого коли- чества приложений для данной операционной среды;
* поддержка распределенных вычислений;
* эффективная поддержка удаленного доступа;
* развитая справочная служба;
* широкая поддержка Internet;
* стабильность и безопасность.

Ниже некоторые из этих требований обсуждаются более подробно.

**6.2.1. Поддержка многопроцессорности и многонитевости**

Непременным требованием для компьютера, претендующего на роль сервера корпоративной сети, является наличие нескольких процессоров. Чаще всего такое требование связано с необходимостью обеспечения высокой производительности, реже - для того, чтобы повысить надежность путем резервирования процессоров. В любом случае операционная система должна предоставить программные средства для управления несколькими процессорами.

Все популярные ОС корпоративного уровня: BanyanVines, NovellNetWare 4.x, IBMLANServer, MicrosoftWindowsNTServer, SunNFS, SolarisUnix, AIX, HP-UX - поддерживают мультипроцессорную обработку.

При выполнении некоторых приложений на многопроцессорной системе повышения производительности может и не произойти, необходимо, чтобы приложение допускало параллельное выполнение различных его частей. Внутренний параллелизм приложений наилучшим образом выявляется путем использования механизма нитей (или потоков). В большинстве современных ОС поддерживается многонитевость, которая позволяет выполнять приложение сразу на нескольких процессорах, в результате чего многонитевые серверы, например, могут одновременно обслуживать более одного клиента.

Отсюда следует, что для достижения высокой производительности мало приобрести мультипроцессорный компьютер и установить мультипроцессорную ОС, необходимо убедиться в том, что для этой мультипроцессорной программно-аппаратной платформы имеется достаточно многонитевых приложений, которые могут в полной мере реализовать имеющийся потенциал производительности.

Кроме повышения производительности, мультипроцессорность используется для обеспечения надежности за счет процессорной избыточности - отказ одного процессора снижает производительность, но не вызывает отказа системы. Однако из того, что ОС является мультипроцессорной вовсе не следует, что она обязательно поддерживает резервирование на уровне процессоров. Так, например, хотя в WindowsNT реализовано симметричное мультипроцессирование, при отказе одного из процессоров система останавливается и требует перезапуска.

Следует иметь ввиду, что разные ОС на одной и той же аппаратной платформе могут показать различную степень масштабируемости. В данном случае масштабируемость характеризуется тем, насколько линейно зависит производительность от числа процессоров. Известно, что наилучшей масштабируемостью обладают операционные системы семейства Unix, например, ОС Solaris демонстрировала линейный рост производительности при увеличении числа процессоров до 64.

Летом этого года было сообщено, что последняя корпоративная редакция WindowsNTServer 4.0 EnterpriseEdition будет стандартно поддерживать 8-процессорные SMP-конфигурации, но что OEM-производители смогут предлагать собственные изделия с большим числом процессоров. К сожалению нам не удалось получить количественные данные о масштабируемости новой версии, поэтому приводим результаты тестирования мультипроцессорных версий WindowsNT и NetWare, взятые из журнала DataCommunications годичной давности (рисунок 6.1).

Тестирование проводилось путем измерения числа транзакций, выполненных в минуту. В качестве приложения была выбрана СУБД Oracle, поскольку это типичное приложение, для которого может потребоваться мультипроцессорная платформа. Источником запросов являлись 32 клиента, работающих на ПК. В качестве сервера был выбран Tricord, имеющий следующую конфигурацию: 6 Pentium-процессоров, 1 Гб памяти, 24 Гб дисковой памяти. Измерения проводились для конфигураций с 2, 4 и 6 процессорами. За единицу производительности принято значение максимальной производительности, показанное в эксперименте.

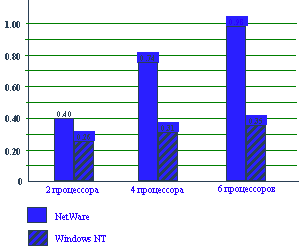


Рис. 6.1.

Из рисунка видно, что ОС NetWare показала хорошую масштабируемость, в то время как производительность WindowsNT с увеличением числа процессоров росла очень несущественно: при переходе с 2 процессоров к 4 она выросла только на 5%, а при переходе с 4 к 6 - на 4%. (Конечно, WindowsNT могла показать гораздо лучшие результаты, если бы в качестве базы данных был выбран MicrosoftSQLServer.)

Сравнивая возможности мультипроцессирования WindowsNT и NetWare, следует отметить также, что в то время как для приложений WindowsNT совершенно не важно, будут ли они выполняться на компьютере с одним или несколькими процессорами, приложения NetWare для того, чтобы иметь возможность выполняться в мультипроцессорной среде NetWare, должны быть написаны специальным образом, то есть уже существующие однопроцессорные приложения необходимо переписывать заново.

**6.2.2. Не только файловый сервер, но и сервер приложений**

Для того, чтобы ОС подходила на роль сервера приложений, она должна поддерживать несколько популярных универсальных API таких, которые позволяли бы, например, выполняться в среде этой ОС приложениям UNIX, Windows, MS-DOS, OS/2. Эти приложения должны выполняться эффективно, а это означает, что в данная ОС должна поддерживать многонитевую обработку, вытесняющую многозадачность, мультипроцессирование и виртуальную память. Сервер приложений должен базироваться на мощной аппаратной платформе (мультипроцессорные системы, часто на базе RISC-процессоров, специализированные кластерные архитектуры).

Учитывая все это, Unix и WindowsNT можно отнести к хорошим серверам приложений, а NetWare - только с большими оговорками. Действительно, сервер NetWare разрабатывался так, чтобы минимизировать среднее время доступа к данным на диске, что делает наиболее целесообразным его использование в качестве *специализированного файл-сервера*. Ради высокой производительности файловых операций разработчики пожертвовали некоторыми другими полезными свойствами, необходимыми для сервера приложений.

Так, в NetWare планирование процессов осуществляется в соответствии с алгоритмом *невытесняющей многозадачности*. Это означает, что каждое приложение берет на себя функции по распределению процессорного времени. Для этого в программах, написанных для выполнения в среде NetWare, должны быть предусмотрены специальные операторы, с помощью которых они по своей инициативе должны отдавать управление операционной системе. Такие программы, написанные с использованием специализированного прикладного программного интерфейса NetWare, называются загружаемыми модулями (NLM). Таким образом, для того, чтобы приложение могло быть выполнено на сервере NetWare, оно должно быть разработано в виде NLM. Приложения для NetWare нужно писать тщательно, осознавая последствия их совместной работы на сервере, чтобы одно приложение не подавляло другие из-за слишком интенсивного захвата процессорного времени. NLM, который слишком долго не отдает управление, может дезорганизовать работу сервера. Все это требует от разработчика приложений для NetWare высокой квалификации.

Универсальные прикладные программные интерфейсы в NetWare отсутствуют. Это затрудняет выполнение уже имеющихся и разработку новых приложений. Следует учесть, однако, что чрезвычайная популярность NetWare привела к тому, что к настоящему моменту имеется богатейший выбор программного обеспечения для NetWare.

Специализированность NetWare заключается также и в том, что все приложения выполняются в *0-ом кольце защиты* процессора, так же как выполняются все модули ядра NetWare. Другими словами, каждое приложение NetWare можно считать частью операционной системы. Любое некорректно разработанное приложение может испортить код ядра. Отсюда следует, что к приложениям NetWare должны предъявляться такие же высокие требования по надежности и качеству исполнения, что и к модулям операционной системы.

Вместе с тем выполнение всех программ в привилегированном режиме значительно ускоряет работу ОС, так как время переключения процессора из одного режима защиты в другой гораздо больше времени перехода на подпрограмму внутри одного и того же режима защиты.

Как уже было сказано, компьютер, на котором установлен сервер NetWare, не может быть использован для запуска обычных приложений. Для маленьких сетей, в которых каждый компьютер на счету, это, возможно и является недостатком, однако гораздо чаще администратор сети, не зависимо от того, какая операционная система установлена на файл-сервере, при возникновении малейшей возможности стремится разгрузить его от выполнения всех других системных и пользовательских приложений. Такая изоляция сервера от конечных пользователей повышает безопасность хранимых на сервере данных и надежность системы в целом.

Все выше перечисленные особенности ОС NetWare, связанные с ее специализированностью, делают эту операционную систему *не очень хорошимсервером приложений*.

Но поскольку необходимость в наличии сервера приложений все же возникает и в сетях NetWare, то для решения этой проблемы на какой-либо из рабочих станций сети устанавливается такая операционная система, которая хорошо подходила бы на роль сервера приложений. Для этой цели компанией Novell одно время позиционировалась операционная система UnixWare - вариант ОС Unix, совместимый с NetWare. В последнее время в качестве сервера приложений в сетях NetWare часто используют WindowsNT.

**6.2.3. Справочная служба - грозное оружие в борьбе за корпоративный рынок**

Для большой сети, состоящей из сотен серверов и тысяч клиентов, очень важно наличие мощной централизованной справочной службы (DirectoryServices), в которой в упорядоченном виде хранится информация о пользователях сети (их имена, пароли, разрешенные часы входа в сеть, разрешенные клиентские компьютеры) и ее ресурсах - серверах, томах файловой системы, базах данных, принтерах, системных приложениях. Необходимая справочная информация о ресурсах в зависимости от его типа включает разные данные, например, программный сервер может характеризоваться типом сервиса, версией, символьным именем и сетевым адресом.

Очевидно, что любая ОС ведет подобную справочную информацию, но она часто представляется в виде разрозненных файлов или баз данных. Эта разрозненность существует как в рамках отдельного компьютера (например, данные о пользователях хранятся и в ОС, и в различных приложения: СУБД, почте, сервисе удаленного доступа), так и в рамках всей сети, когда каждый сервер ведет свою базу данных пользователей и администрируется независимо от других, как это сделано, например, в NetWare 3.x. Из-за этого возникает избыточность и несогласованность справочной информации, которые усложняют работу администратора и приводят к ошибкам, создающим иногда прорехи в системе защиты сети от несанкционированного доступа.

Хорошая справочная служба предоставляет всем системным приложениям *общую* базу справочных данных, которой они могут пользоваться в своих целях: почта - для нахождения сетевых адресов ее абонентов, СУБД - для выполнения процедур аутентификации и авторизации, а также для нахождения таблиц распределенной базы данных.

Справочная служба должна обеспечивать *единую процедуру логического входа* в сеть. Аутентифицируясь с ее помощью один раз, пользователь получает доступ ко всем ресурсам сети. При этом пользователь избавляется от необходимости запоминать несколько идентификаторов, паролей и имен серверов, а администратор - поддерживать несколько пользовательских бюджетов на разных серверах.

Справочная служба должна обладать *масштабируемостью*, то есть хорошо работать при очень большом числе пользователей и ресурсов. Для этого необходимо, чтобы база справочных данных была *распределенной*, то есть части ее располагались на разных компьютерах. Действительно, хранение данных для большой сети в одном компьютере ненадежно и непроизводительно, так как обращения всех клиентов к одному компьютеру делают его узким местом, кроме того разбиение базы данных на части и хранение этих частей на разных компьютерах позволяет приблизить данные к источнику запросов, делая эти данные доступнее.

Поскольку справочная служба является централизованным элементом сети, она должна обладать отказоустойчивостью. Для этого справочная база данных должна быть не только распределенной, но и *реплицируемой*.

Примерами распределенных справочных служб, в наибольшей степени удовлетворяющих этим требованиям, являются службы NetWareDirectoryServices (NDS) компании Novell и служба StreetTalk компании Banyan.

Компания Banyan, не без основания предполагая, что ее справочная служба может составить сильную конкуренцию NDS компании Novell, даже отказалась от продвижения своей сетевой ОС Vines 7.0 и сконцентрировалась на развитии StreetTalk для WindowsNT, NetWare, Solaris и SCOUnix. Проблемы же StreetTalk заключаются в том, что на рынке пока ощущается дефицит приложений, поддерживающих эту справочную службу.

Известны также справочные службы NIS, NIS+, DCE, поддерживаемые ОС Unix, и доменная справочная служба DirectoryServicesWindowsNT. В виду того, что справочная служба, основанная на доменах плохо масштабируется, сложна в управлении и имеет ограниченные функциональные возможности, компания Microsoft предлагает новую глобальную справочную службу - ActiveDirectory . Эта служба должна появиться в начале 1998 года составе WindowsNT 5.0.

**6.2.4. Стабильность и отказоустойчивость**

Корпоративная ОС должна быть стабильной и устойчивой, для этого она, как минимум, должна работать в защищенном режиме и поддерживать вытесняющую многозадачность. Эти два условия не позволят некорректно работающим приложениям нарушить работу операционной системы и других приложений.

Стабильность ОС обуславливается также использованием прогрессивных технологий и архитектурных решений, таких как микроядерный подход к структурной организации ОС или использование объектов на уровне операционной системы.

Стабильности ОС способствует реализация *вытесняющей многозадачности*.Вытесняющая многозадачность - это такой способ разделения процессорного времени, который позволяет пользователю-программисту при разработке приложений не заботиться о том, чтобы приложение отдавало управление через определенный промежуток времени, что обязательно при использовании алгоритма невытесняющей многозадачности, реализованного, например, в NetWare. В системах с вытесняющей многозадачностью планирование процессов или нитейвыполняется централизованно, средствами ОС. Универсальность и независимость алгоритма планирования от конкретного набора прикладных программ делает систему более надежной и защищенной от неправильно работающих приложений, так как ни одно из них не сможет слишком долго занимать процессор и, следовательно, не сможет привести к сбою системы по этой причине.

На стабильность системы в целом влияет насколько хорошо ОС поддерживает аппаратные средства обеспечения надежности и отказоустойчивости. И прежде всего, необходимо поинтересоваться, обеспечивает ли ОС резервирование процессоров, то есть возможность продолжения работы при отказе одного из процессоров. Статистика показывает, что наиболее частой причиной отказа компьютера являются проблемы с источниками питания и отказы дисковых подсистем. Поэтому так важны для корпоративной ОС возможности по поддержке источников бесперебойного питания UPS, а также дисковых массивов типа RAID.

**6.2.5. Насколько важна сертифицированность ОС по критериям безопасности**

Каждая сетевая ОС имеет набор средств для обеспечения безопасности данных, их целостности, доступности и конфиденциальности. Для этого в ней предусмотрены средства проведения процедур аутентификации и авторизации пользователей, а также аудита (отслеживания) всех событий, потенциально опасных для системы. Для характеристики степени безопасности ОС используются различные стандарты.

Основы стандартов на безопасность были заложены изданными в 1983 году "Критериями оценки надежных компьютерных систем". Этот документ, изданный в США национальным центром компьютерной безопасности (NCSC - NationalComputerSecurityCenter), часто называют Оранжевой Книгой. Утвержденная в 1985 году в качестве правительственного стандарта, Оранжевая Книга определяет основные требования и специфицирует классы для оценки уровня безопасности готовых и коммерчески поддерживаемых компьютерных систем.

В соответствии с требованиями Оранжевой книги, безопасной считается такая система, которая "посредством специальных механизмов защиты контролирует доступ к информации таким образом, что только имеющие соответствующие полномочия лица или процессы, выполняющиеся от их имени, могут получить доступ на чтение, запись, создание или удаление информации".

Иерархия надежных систем, приведенная в Оранжевой Книге, помечает низший уровень безопасности как С, высший как А, промежуточный как B. В класс D попадают системы, оценка которых выявила их несоответствие требованиям всех других классов.

Основные свойствами, характерными для С-систем, являются: наличие подсистемы учета событий, связанных с безопасностью, избирательный контроль доступа. Избирательный контроль заключается в том, что каждый пользователь в отдельности наделяется или лишается привилегий доступа к ресурсам. Уровень С делится на 2 подуровня: С1 и С2. Уровень С2 предусматривает более строгую защиту, чем С1. В соответствии с этим уровнем требуется отслеживание событий, связанных с нарушениями защиты, детальное определение прав и видов доступа к данным, предотвращение случайной доступности данных (очистка освобожденной памяти). На уровне С2 должны присутствовать средства секретного входа*,* которые позволяют пользователям идентифицировать себя путем ввода уникального идентификатора (ID) входа и пароля перед тем, как им будет разрешен доступ к системе. Избирательный контроль доступа, требуемый на этом уровне, позволяет владельцу ресурса определить, кто имеет доступ к ресурсу и что он может с ним делать. Владелец делает это путем предоставляемых прав доступа пользователю или группе пользователей. Средства учета и наблюдения (auditing) - обеспечивают возможность обнаружить и зафиксировать важные события, связанные с безопасностью, или любые попытки создать, получить доступ или удалить системные ресурсы. Эти средства используют идентификаторы, чтобы зафиксировать того пользователя, который выполняет эти действия. Защита памяти - предохраняет от чтения информации, записанной кем-нибудь в память, после того, как структуры данных будут возвращены ОС. Память инициализируется перед тем, как повторно используется.

Требования уровней В и А гораздо строже и редко предъявляются к массово используемым продуктам.

Различные коммерческие структуры (например, банки) особо выделяют необходимость аудита, службы, соответствующей рекомендации С2. Любая деятельность, связанная с безопасностью, может быть отслежена и тем самым учтена. Это как раз то, что требует С2, и то, что обычно нужно банкам. Однако, коммерческие пользователи, как правило, не хотят расплачиваться производительностью за повышенный уровень безопасности. Уровень безопасности А занимает своими управляющими механизмами до 90% времени компьютера. Более безопасные системы не только снижают эффективность, но и существенно ограничивают число доступных прикладных пакетов, которые соответствующим образом могут выполняться в подобной системе. Например, для ОС Solaris (версия UNIX) есть несколько тысяч приложений, а для его аналога В-уровня - только сотня.

Операционные системы, которые сертифицировались в соответствии с требованиями Оранжевой книги, тестировались как локальные операционные системы, с отключенными сетевыми функциями. Для сетевых ОС предусмотрена сертификация по так называемой Красной книге, в которой критерии С2 конкретизируются по отношению к системам, работающим в сети. Таким образом, нет никаких формальных гарантий, что ОС, сертифицированные по Оранжевой книге, надежно защищают информацию и при работе в сети. Однако это не означает, что такие ОС не могут быть использованы для построения безопасных сетей.

К настоящему времени WindowsNT 3.51 прошла сертификацию по уровню С2 Оранжевой книги. Тестирование выполнялось с отключенными сетевыми функциями. Серверная ОС NetWare прошла сертификацию с учетом требований Красной книги (понятно, что как автономная операционная система она в принципе не может функционировать).

Отдавая должное наличию сертификата у того или иного продукта, не стоит все же преувеличивать важность этого факта, особенно в области безопасности. Сертификация вовсе не означает, что данный продукт никогда не сможет быть взломан; она просто говорит о том, что при использовании продукта, сертифицированного авторитетной организацией, риск может быть значительно уменьшен.

Сертификация в области безопасности всегда связана с тестированием продукта для вполне определенного и наиболее часто встречающегося набора угроз и атак. А так как человечество постоянно изобретает все новые и новые способы взлома информационных систем, то даже те ограниченные гарантии, которые были даны системе в момент сертификации, постепенно теряют свое значение.

Кроме того, необходимо учитывать, что процедура сертификации продолжается примерно 2 года, и каждая новая версия должна заново проходить эту процедуру.

**6.2.6. Способность работать в гетерогенной среде**

Это свойство операционной системы, называемое иногда совместимостью, имеет много аспектов, и одним из них является способность операционной системы поддерживать разнообразные аппаратные платформы: серверы и суперсерверы на основе процессоров Intel, RISC-серверы, миникомпьютеры и мэйнфреймы. Многоплатформенность позволяет уменьшить гетерогенность сети за счет использования на компьютерах разного типа одной и той же ОС.

Другим аспектом совместимости является интеграция сервисов различных операционных систем в рамках одной ОС. В корпоративной сети разные клиенты привыкли пользоваться разными вариантами реализации файлового сервиса. Клиенты, работающие с серверами NetWare, пользуются файловым сервисом NCP, клиенты Unix работают с файловым сервисом NFS. С другой стороны, в корпоративной сети у клиентов может возникнуть необходимость обращения к "чужим" файлам, то есть файлам, хранящимся на сервере, который предоставляет недоступный для них файловый сервис. Например, когда клиент Unix хочет получить доступ к файлам NetWare, он не может этого сделать, так как у него нет в распоряжении клиентской части этого сервиса. Поэтому очень важно, чтобы ОС могла в разной форме предоставлять доступ к хранящимся в ней файлам - и в стиле Unix, и в стиле NetWare.

Гетерогенность сети выражается и в многообразии операционных систем конечных пользователей. Корпоративная ОС должна иметь набор сетевых клиентских оболочек для широкого перечня операционных систем, устанавливаемых на клиентские станции, среди которых, конечно должны быть DOS, Windows, Windows 95, UNIX, OS/2, Mac.

В состав операционной системы входят программные средства, используемые при решении задачи транспортировки сообщений. К таким средствам относятся драйверы сетевых адаптеров, реализующие протоколы канального уровня, и разнообразные транспортные протоколы. Современная корпоративная ОС должна поддерживать сетевое оборудование стандартов Ethernet, TokenRing, FDDI, FastEthernet, 100VG-AnyLAN, ATM. На более высоких уровнях должны поддерживаться протоколы стеков TCP/IP, IPX/SPX, NetBIOS, AppleTalk.

### 6.3. Тенденции рынка сетевых ОС

631

**6.3.1. NT оттесняет Unix в мир 64-разрядных серверов**

В борьбе на рынке сетевых операционных систем можно выделить много фронтов, один из них пролегает между WindowsNT и Unix. В последнее время все чаще и острее проходят дискуссии на тему "Что лучше WindowsNT или Unix?" Не будучи фанатиком той или иной системы, нельзя дать однозначный ответ на этот некорректный вопрос, но вполне возможно очертить нишу, занимаемую каждой из систем, проследить тенденции, отметить достоинства и недостатки. Именно этому посвящен данный раздел.

Unix насчитывает 25 лет своей истории, а WindowsNT еще вполне молодая система. Отсюда во многом проистекают как достоинства, так и недостатки обоих систем. Надежность и стабильность любой технологии повышается с накоплением опыта использования: Unix прошел "шлифовку" в течение 25 лет. За четверть века трудом миллионов талантливых программистов был создан огромный багаж программного обеспечения, работающего в среде Unix. Но за эти годы многое, что изменилось в компьютерном мире, появлялись новые идеи, совершенствовалась технология, возникали новые задачи, появлялась новая аппаратура, которую необходимо было поддерживать, изменялись требования, предъявляемые к операционным системам. "Вызовы" времени отрабатывались, путем внесения изменений в программный код, создавались и согласовывались многочисленные версии. В результате Unix приобрел некоторый налет эклектичности, что в общем-то происходит со всеми развивающимися системами. В некоторых версиях Unix сосуществуют в одной системе сразу два вида программного кэша диска, иногда встречается реализация механизма нитей в виде библиотеки программ пользовательского режима, что не очень эффективно, можно привести примеры реализаций монолитного ядра с невыгружаемыми драйверами. (Утверждать или, не дай Бог, критиковать что-либо, связанное с внутренним устройством Unix - дело неблагодарное, ибо для любого недостатка одной версии, всегда можно найти другую версию из многочисленного семейства Unix, в которой этот недостаток отсутствует. Таким образом, противопоставляя какую-либо систему Unix'у мы противопоставляем ее огромному и разнообразному семейству операционных систем.) Тем не менее сейчас уже трудно говорить об академической стройности и логической ясности семейства Unix в целом.

С другой стороны, Unix привнесла в мир операционных систем множество новых идей, адаптацию которых мы наблюдаем во всех современных ОС, в том числе, конечно, и в WindowsNT. Эта операционная система с самого начала разрабатывалась с учетом переносимости, многопроцессорности и других свойств, которые должны быть присущи современной ОС. Разработчики WindowsNT уже имели в своем распоряжении опыт, наработанный Unix-системами. Но, к сожалению, не только в человеческой жизни невозможно научиться на чужих ошибках. За короткую жизнь WindowsNT ее разработчиками было сделано немало собственных ошибок, и сейчас для нормальной работы системы требуется постоянно следить за сообщениями, регулярно появляющимися на серверах Internet и в прессе о новых обнаруженных "багах" и не забывать вовремя перезапускать ServicePack соответствующей версии.

Конкуренция между этими двумя операционными системами наблюдается не только в области технических идей, но и прежде всего в коммерческой области. В 1996 году мировой объем продаж Unix-систем, включая ПК, рабочие станции, серверы ЛВС, системы среднего уровня и большие системы, вырос на 12%, достигнув 34,3 млрд. долл. Этот отрадный факт был омрачен для сторонников Unix-систем другим сообщением: по предварительным данным IDC мировые продажи лицензий на WindowsNTServer впервые превысили продажи лицензий на Unix: 725 и 602 тысячи соответственно. И хотя многие связывают рыночный успех WindowsNT не столько с техническими достижениями, сколько с маркетинговыми талантами Билла Гейтса, не возможно игнорировать тот факт, что на сцене сетевых операционных систем появился новый сильный игрок, претендующий на первые роли во всех секторах.

К настоящему моменту в результате борьбы WindowsNT и Unix на рынке сложилась ситуация, которую схематично можно представить в следующем виде:

|  |  |
| --- | --- |
| сетевые операционные системы персональных компьютеров | преимущественно WindowsNT |
| серверные системы младшего класса | Unix и все активнее внедряется WindowsNT |
| высокопроизводительные серверы | преимущественно Unix |

Именно в области высокопроизводительных серверов для самых крупных и требующих наивысшей надежности бизнес-приложений Unix доказала свою ценность. Эту нишу WindowsNT пока что не в состоянии занять. Unix остается операционной системой крупных серверов, хранилищ больших баз данных, мощных вычислителей, интеграционных машин крупных проектов и т. п. В одних случаях это обусловлено наличием соответствующего ПО, в других - тем простым фактом, что мощные серверы, построенные на базе RISC-процессоров, оснащаются только Unix. Ведь распространение WindowsNT вне мира Intel крайне незначительно и системе Unix здесь просто нет альтернативы.

В последние четыре года была проделана большая работа по расширению масштабируемости Unix. Поскольку Unix все шире применяется на машинах с симметричной мультипроцессорной обработкой (SMP), для массивно параллельных суперкомпьютеров и кластеров из большого числа машин не остается сомнений в возможностях дальнейшего развития Unix по мере роста требований промышленности. Аналитики предсказывают, что в ближайшие два года Unix-серверы останутся более масштабируемыми, чем NT-серверы. Они также будут лучше приспособлены для создания кластеров, применяемых для обеспечения высокого уровня надежности.

Важным этапом развития Unix является его перенос на машины с 64-разрядной архитектурой. Ожидается, что в ближайшее время Hewlett-Packard, IBM и Sun выпустят 64-разрядные версии своих Unix-систем для RISC-платформ. Для успеха 64-разрядных ОС Unix необходимо, чтобы независимые разработчики приложений усилили ее поддержку. Без качественных высокопроизводительных приложений преимущество любой ОС - только теоретическое.

В последнее время корпорация Microsoft ведет активное наступление на рынке средних и младших моделей серверов, затрагивая и традиционно принадлежащий Unix'у рынок рабочих станций. В результате давления серверов на платформе Intel под управлением WindowsNT на рынок младших моделей рабочих станций на базе RISC под Unix, в прошлом году произошел спад продаж в этом сегменте. Но в то же время объем продаж серверных Unix-систем среднего уровня значительно вырос, в основном за счет повышенного спроса на Unix-серверы как платформы для реляционных СУБД.

Хотя сегодня предлагается немало систем для архитектуры Intel - OpenDeskTop компании SCO, Solaris фирмы Sun, BSD/386 фирмы BSDI, бесплатные Linux, FreeBSD, Unix не сумела занять достойного места на рынке персональных компьютеров. Сказались недостаточное внимание к персональным компьютерам со стороны основных компаний-производителей Unix, недостаточно дружественный интерфейс приложений в среде Unix, ориентированный на профессионального пользователя. Система Unix изначально создавалась программистами для программистов. Однако в последнее время поставщики Unix-систем (IBM, в частности) стали серьезно относиться к удобству пользовательского интерфейса, к упрощению процедур администрирования.

Особая тема - историческая и нынешняя роль ОС Unix в Internet, но об этом в следующем разделе. 632

**6.3.2. Движение операционных систем в сторону Internet**

Рейтинг сетевых операционных систем во многом определяется тем, насколько полно они осуществляют поддержку Internet. Все наиболее популярные ОС в ответ на это требование рынка предложили целый спектр новых возможностей и продуктов. Как минимум, операционная система должна поддерживать стек коммуникационных протоколов TCP/IP (TCP, UDP, IP, OSPF и др.), основные сервисы прикладного уровня стека TCP/IP (FTP, Telnet, DNS, DHCP и др.), протокол HTTP и сервис WWW.

**Unix**

Всем известно, что Internet представляет собой сеть из сетей, построенных на Unix, поэтому позиции Unix в Internet непоколебимы. Internet сформировалась и завоевала миллионы подписчиков задолго до эпохи ее коммерциализации и во многом благодаря системе Unix и мировому сообществу ее разработчиков. Основные службы Internet приобрели современный вид именно в системе Unix.

С началом коммерческого использования Internet к ней стали подключаться миллионы и миллионы непрофессионалов. Поскольку непрофессиональные пользователи, как правило, использовали ПК на базе процессора Intel с операционной системой Windows, вся клиентская часть Internet совершенно естественно стала стремительно переходить на Windows. Мелкие узлы Internet на базе Intel-систем с операционной системой WindowsNT дешевле как по начальным затратам, так и в ежедневном обслуживании.

Несколько другая картина наблюдается с серверной частью Internet. Здесь использование Unix продолжается и оно оправдано. Во-первых, Internet-провайдеры, на своих серверах используют в основном Unix. Это обусловлено объективными причинами, такими, как большой объем сетевого трафика, для которого архитектура Unix более приспособлена, и субъективными - образованием и профессиональной подготовкой руководителей подобных компаний. Большие по сравнению с Windows затраты времени и средств на администрирование оправдываются значительно большей производительностью.

Во-вторых, крупные узлы Сети, опять-таки по соображениям производительности, реализуются в среде Unix.

**WindowsNT**

Поддержка стека TCP/IP была заложена в WindowsNT изначально и усиливалась от версии к версии. Существенные изменения в этом плане произошли и в последней версии. В комплект поставки вошли новые Internet-ориентированные компоненты:

* InternetInformationServer (IIS) версии 2.0 предоставляющий услуги Web-, ftp- и gopher-сервера. Возможности InternetInformationServer сравнимы, а по ряду тестов и превосходят аналогичный популярный продукт ServerNetscape;
* DNS/WINSServer, который позволяет легко находить в Internet или Intranet-сетях нужные Web-узлы;
* средства реализации технологии создания защищенного канала PPTP (point-to-pointtunnelingprotocol), которая предназначена для создания частных сетей (VPN) в Internet;
* программа FrontPage, которая позволяет создавать Web-страницы на основе разнообразных шаблонов, проверять правильность ссылок и осуществлять общее управление создаваемыми Web-узлами;
* индексный сервер MicrosoftSearchServer (кодовое название - "Tripoli"), который позволяет легко находить информацию на распределенных серверах Intranet-сети в рамках любых документов, в том числе и созданных в MicrosoftOffice.

Технология индексации и поиска информации для WindowsNTServer 4.0, разработанная Microsoft, осуществляет автоматическую индексацию содержимого HTML-страниц и других документов, хранящихся на корпоративных Intranet-серверах (например, документов созданных в MicrosoftOffice). SearchServer является одним из элементов технологии Microsoft, известной под кодовым названием "Cairo": таким образом, благодаря тесной интеграции с WindowsNTServerDirectory обеспечивается безопасность поиска, а благодаря интеграции с файловой системой WindowsNT - высокая эффективность. Этот продукт позволяет индексировать содержимое нескольких серверов, причем обеспечивается поддержка нескольких языков. SearchServer также поддерживает для каждого языка возможности углубленного лингвистического анализа, то есть пользователи могут находить документы или свойства документов с учетом грамматических форм ключевого слова.

Два средства новой системы, предназначенные для работы в Internet, представляют особый интерес для администраторов. Во-первых, это интегрированная служба имен DNS/WINS. Теперь когда клиенту WINS нужно определить IP-адрес, соответствующий символьному NetBIOS-имени, он обращается сначала к базе данных WINS, а затем - собственно к DNS. Таким образом, в системе на равных можно применять и динамически распознаваемые имена WINS, и статические имена DNS.

Кроме того, в состав WindowsNT 4.0 вошла Web-ориентированная утилита администрирования, открывающая доступ к средствам администрирования WindowsNT из любого Web-браузера. Из соображений безопасности для удаленного администрирования следует использовать Web-браузеры, способные регистрировать пользователя непосредственно на сервере WindowsNT (т. е. такие, как InternetExplorer) или поддерживать протокол защищенного канала SSL.

Одно из усовершенствований связано с тем, что повышающаяся роль Internet'а и клиент-серверных систем ведет к росту числа мобильных пользователей. Microsoft в связи с этим улучшила RAS ( улучшила поддержку ISDN) и предоставила средства безопасной работы с RAS через Internet. В RAS реализованы протоколы PPTP (создает зашифрованный трафик через Internet) и MultilinkPPP (позволяет объединять несколько каналов в один). Клиентами могут быть WindowsNT 4.0 Workstation или Windows 95.

Использовать TCP/IP с NTServer или Workstation можно в сочетании с Point-to-PointTunnelingProtocol. PPTP позволяет туннелировать IPX, TCP/IP и/или NetBEUI через стандартные соединения InternetPoint-to-PointProtocol. Содержимое пакетов (например сетевые данные) шифруется в этих соединениях по умолчанию. В целом, PPTP дает пользователям шифрованный туннель сквозь Internet для удаленного доступа на NTServer. В настоящее время PPTP работает только между машинами Windows 95 и NT.

PPTP не является пока стандартным протоколом, и, хотя несколько поставщиков вместе с Microsoft обратились с RFC в Группу инженерной поддержки Internet, его принятие в качестве стандарта может занять от нескольких месяцев до нескольких лет.

Распределенная модель объектной компоновки (DistributedComponentObjectModel) - еще одно ключевое дополнение к WindowsNTServer 4.0. Модель объектной компоновки (COM) позволяет разработчикам программ создавать приложения, состоящие из отдельных компонент. Распределенная модель (DCOM) в WindowsNTServer 4.0 расширяет COM таким образом, что позволяет отдельным компонентам взаимодействовать через Internet. DCOM является растущим стандартом Internet, опубликованным в соответствии с форматом, определенным в спецификациях RFC 1543.

**NetWare**

Novell, по-прежнему являющаяся основным поставщиком серверов файлов и печати, по-прежнему обладающая крупнейшей инсталлированной базой клиентов службы каталогов и по-прежнему занимающая лидирующую позицию в разработке решений по обмену сообщениями и программного обеспечения коллективной работы,осознала важность Internet с опозданием.

Однако надо помнить, что до 1995 года компания имела крупнейшую в мире инсталлированную базу клиентов TCP/IP - продукты LANWorkplace и LANWorkgroup. Уже давно серверы NetWare умели маршрутизировать IP-трафик, поэтому пользователи могли иметь доступ в Internet и выполнять приложения Internet, даже когда они применяли IPX для доступа к файлам и печати, если стек протоколов IPX был установлен вместе со стеком TCP/IP. Продукты Novell поддерживали также и такие протоколы Internet, как NetworkFileSystem (NFS) и SNMP.

Но вот появление WorldWideWeb не было оценено в полной мере. Практически отсутствовали популярные приложения Internet для среды NetWare. Здесь отрицательно сказалась специализированность сервера NetWare.

Несмотря на катаклизмы реорганизации и бурю критики извне, в 1996 году Novell добилась значительных успехов в деле освоения Internet - была представлена IntranetWare, наследница NetWare 4.1.

Пакет IntranetWare состоит из набора средств создания сетей Intranet, а также доступа к Internet и интеграции с Unix-системами. Прежде всего, следует упомянуть NetWareWebServer 2.51, отвечающий большинству требований, предъявляемых к серверу Web. Novell уверяет, что по производительности данный продукт значительно опережает аналогичные серверы компаний Microsoft и Netscape. Обращает на себя внимание только отсутствие в составе сервера Web некоторых важных вспомогательных средств, в частности редактора HTML. Это тем более странно, поскольку Novell выпускает комплект InnerWebPublisher, имеющий такие инструменты. Особенность сервера Web компании Novell в том, что он дает возможность пользователям получать доступ к базе NDS, правда, в режиме просмотра. Другим недостатком NovellWebServer является отсутствие в нем поддержки протокола защищенного канала SecureSocketLayer (SSL), используемого при организации безопасного обмена данными между браузерами и серверами.

IntranetWare поддерживает удаленный и локальный общие шлюзовые интерфейсы (CGI) для создания динамических страниц Web, например посредством выполнения сценария поиска записи в базе данных. Инструментарий IntranetWare для написания сценариев включает интерпретаторы NetBasic (лицензирован у HiTecSoft), Perl и Basic. Набор инструментов для быстрого установления связи IntranetWareWebServer с другой программой пока отстает от возможностей аналогичных средств в средах NT и Unix, но все же Novell продвинулась здесь далеко вперед.

Важное значение имеют и другие добавления к NetWare:

* Доступ клиентов NetWare к сетям TCP/IP через шлюз IPX/IP. На клиентские машины загружается только стек IPX/SPX. Основным минусом такой схемы является недостаточная надежность подключения к TCP/IP, поскольку при выходе из строя шлюза клиенты лишаются Web- и FTP-сервисов. Возможны и перегрузки шлюзов, т.к. все потоки к Web и FTP идут через них. Кроме того, для клиентов становятся недоступны некоторые службы сетей TCP/IP. Однако такой подход обеспечивает более высокую, чем в других схемах, безопасность.
* Поддержка протокола динамической конфигурации хоста DHCP позволяет системе выделять IP-адреса клиентам по мере необходимости, благодаря чему адресное пространство используется более эффективно, а администрирование становится проще.
* С помощью нового сервера ftp можно разделять файлы с любым клиентом TCP/IP, а не только с клиентами IntranetWare.
* IntranetWare включает многопротокольный маршрутизатор NetWareMultiprotocolRouter (раньше он продавался отдельно), позволяющий серверу IntranetWare связываться непосредственно с Internet или другими глобальными сетями.

Большие перспективы для новой жизни Novell в Internet открывает ей справочная служба NDS. Действительно, с увеличением числа пользователей в любой сети возрастает потребность в хорошей справочной службе, что же говорить об Internet с ее миллионами пользователей. Весьма возможно, что именно NDS окажется по плечу роль службы каталогов Internet. NDS может оказаться полезной и для нахождения программных компонентов при организации распределенных вычислений в Internet.

# 7. Влияние Internet на мир корпоративных сетей

Стек TCP/IP сегодня представляет собой один из самых распространенных стеков транспортных протоколов вычислительных сетей. Стремительный рост популярности Internet привел и к изменениям в расстановке сил в мире коммуникационных протоколов - протоколы TCP/IP, на которых построен Internet, стали быстро теснить бесспорного лидера прошлых лет - стек IPX/SPX компании Novell. Сегодня общемировое количество компьютеров, на которых установлен стек TCP/IP, сравнялось с общим количеством компьютеров, на которых работает стек IPX/SPX, и это говорит о резком переломе в отношении администраторов локальных сетей к протоколам, используемым на настольных компьютерах, так как именно они составляют подавляющее число мирового компьютерного парка и именно на них раньше почти везде работали протоколы компании Novell, необходимые для доступа к файловым серверам NetWare. Процесс становления стека TCP/IP стеком номер один в любых типах сетей продолжается и сейчас любая промышленная операционная система обязательно включает программную реализацию этого стека в своем комплекте поставки.

Хотя протоколы TCP/IP неразрывно связаны с Internet, и каждый из многомиллионной армады компьютеров Internet работает на основе этого стека, однако, существует большое количество локальных, корпоративных и территориальных сетей, непосредственно не являющихся частями Internet, которые также используют протоколы TCP/IP. Чтобы отличать их от Internet, эти сети называют сетями TCP/IP или просто IP-сетями.

Локальные и корпоративные сети все шире используют протоколы TCP/IP для передачи своего внутреннего трафика. До недавнего времени это были в основном сети, построенные на основе операционной системы Unix. Причина заключалась в исторической связи Unix и TCP/IP - впервые протоколы стека TCP/IP были реализованы в среде UnixBSD в университете Berkeley. Однако сейчас, когда протоколы TCP/IP имеются в каждой сетевой операционной системе, появились локальные сети TCP/IP и на основе других операционных систем, например, WindowsNT, Windows 95, OS/2 Warp или NetWare. Конечно, одной из очевидных причин использования стека TCP/IP в локальных и корпоративных сетях является легкость присоединения таких сетей к Internet при первой необходимости. Однако, гибкость и открытость стека сами по себе являются достаточно вескими причинами для использования протоколов TCP/IP в автономных локальных и корпоративных сетях.

Параллельно с Internet существуют и другие публичные территориальные сети, работающие на основе протоколов TCP/IP. Это сети крупных провайдеров транспортных сервисов, таких как MCI, Sprint, AT&T и многих других, предоставляющих услуги по объединению локальных IP-сетей заказчика. Публичные IP-сети предоставляют заказчику более высокий уровень сервиса по сравнению с Internet - более низкий уровень задержек пакетов, защиту от несанкционированного доступа, высокий коэффициент готовности. С помощью сервисов публичных IP-сетей предприятие может строить транспортную магистраль своей корпоративной сети, не подвергая себя риску атак многочисленных хакеров, работающих и живущих в Internet.

### 7.1. Транспорт Internet приспосабливается к новым требованиям

В начале 90-х годов, после более чем десяти лет относительно спокойной жизни протокол IP столкнулся с серьезными проблемами. Именно в это время началось активное промышленное использование Internet: переход к построению сетей предприятий на основе транспорта Internet, применение Web-технологии для получения доступа к корпоративной информации, ведение электронной коммерции с помощью Internet, внедрение Internet в индустрию развлечений (распространение видеофильмов, звукозаписей, интерактивные игры).

Все это привело к резкому росту числа узлов сети, изменению характера трафика и к ужесточению требований, предъявляемых к качеству обслуживания сетью ее пользователей.

Динамика роста Internet такова, что сегодня трудно привести вполне достоверные сведения о числе сетей, узлов и пользователей Internet. Быстрый рост сети усугубляет уже давно ощущаемый дефицит IP-адресов, вызывает перегрузку маршрутизаторов, которые должны уже сегодня обрабатывать в своих таблицах маршрутизации информацию о нескольких десятках тысяч номеров сетей, а также ведет к резкому увеличению суммарного объема передаваемого по Internet трафика.

Все более широкое использование Internet в качестве общемировой широковещательной сети, заменяющей сети радио и телевидения, приводит к изменению характера передаваемого трафика. Наряду с традиционными компьютерными данными все большую долю трафика составляют мультимедийные данные. Синхронный характер мультимедийного трафика предъявляет нетрадиционные для Internet требования по качеству обслуживания - предоставления гарантированной полосы пропускания с заданным уровнем задержки передаваемых пакетов.

Промышленное использование Internet предполагает определенный уровень защиты данных - аутентификацию абонентов, обеспечение конфиденциальности и целостности данных. Особенно это важно при ведении в Internet электронной торговли, а также при построении частных виртуальных сетей.

Рост популярности Internet привел к появлению новых категорий пользователей. Во-первых, в сеть пришло много непрофессиональных пользователей - сотрудников предприятий, работающих на дому, людей, использующих Internet как средство развлечения или как неисчерпаемый источник информации. Многие из них желали бы работать как мобильные пользователи, не расставаясь со своим компьютером в поездках, вдали от своей "родной" сети. В этих случаях очень важной оказывается возможность автоконфигурирования стека TCP/IP, когда все параметры стека (в основном это касается IP-адресов компьютера, DNS-сервера и маршрутизаторов) автоматически сообщаются компьютеру при его подключении к сети.

Для того, чтобы в новых условиях протокол IP смог также успешно работать, как и в предыдущие годы, сообщество Internet после достаточно долгого обсуждения решило подвергнуть IP серьезной переработке.

**7.1.1. Защита данных**

Защита информации - ключевая проблема, которую нужно решить для превращения Internet в публичную всемирную сеть с интеграцией услуг. Без обеспечения гарантий конфиденциальности передаваемой информации бум вокруг Internet быстро утихнет, оставив ей роль поставщика интересной информации.

Сегодня защиту информации в Internet обеспечивают различные нестандартные средства и протоколы - firewall'ы корпоративных сетей и специальные прикладные протоколы, типа S/MIME, которые обеспечивают аутентификацию сторон и шифрацию передаваемых данных для какого-либо определенного прикладного протокола, в данном случае - электронной почты.

Существуют также протоколы, которые располагаются между прикладным и транспортным уровнями стека TCP/IP. Наиболее популярным протоколом такого типа является протокол SSL (SecureSocketLayer), предложенный компанией NetscapeCommunications, и широко используемый в серверах и браузерах службы WWW. Протоколы типа SSL могут обеспечить защиту данных для любых протоколов прикладного уровня, но недостаток их заключается в том, что приложения нужно переписывать заново, если они хотят воспользоваться средствами защиты, так как в приложения должны быть явно встроены вызовы функций протокола защиты, расположенного непосредственно под прикладным уровнем.

Проект IPv6 предлагает встроить средства защиты данных в протокол IP. Размещение средств защиты на сетевом уровне сделает их прозрачными для приложений, так как между уровнем IP и приложением всегда будет работать протокол транспортного уровня. Приложения переписывать при этом не придется.

В протоколе IPv6 предлагается реализовать два средства защиты данных. Первое средство использует дополнительный заголовок "AuthenticationHeader" и позволяет выполнять аутентификацию конечных узлов и обеспечивать целостность передаваемых данных. Второе средство использует дополнительный заголовок "EncapsulatingSecurityPayload" и обеспечивает целостность и конфиденциальность данных.

Разделение функций защиты на две группы вызвано практикой, применяемой во многих странах на ограничение экспорта и/или импорта средств, обеспечивающих конфиденциальность данных путем шифрации. Каждое из имеющихся двух средств защиты данных может использоваться как самостоятельно, так и одновременно с другим.

В проектах протоколов защиты данных для IPv6 нет привязки к определенным алгоритмам аутентификации или шифрации данных. Методы аутентификации, типы ключей (симметричные или несимметричные, то есть пара "закрытый-открытый"), алгоритмы распределения ключей и алгоритмы шифрации могут использоваться любые. Параметры, которые определяют используемые алгоритмы защиты данных, описываются специальным полем SecurityParametersIndex, которое имеется как в заголовке "AuthenticationHeader", так и в заголовке "EncapsulatingSecurityPayload".

Тем не менее, для обеспечения совместимой работы оборудования и программного обеспечения на начальной стадии реализации протокола IPv6 предложено использовать для аутентификации и целостности широко распространенный алгоритм хеш-функции MD5 с секретным ключом, а для шифрации сообщений - алгоритм DES.

Ниже приведен формат заголовка AuthenticationHeader.

|  |  |  |
| --- | --- | --- |
| Следующий заголовок | Длина аутентификационных данных | Зарезервированное поле |
| Индекс параметров безопасности (SPI) | | |
| Аутентификационные данные | | |

Протокол обеспечения конфиденциальности, основанный на заголовке "EncapsulatingSecurityPayload", может использоваться в трех различных схемах.

В первой схеме шифрацию и дешифрацию выполняют конечные узлы. Поэтому заголовок пакета IPv6 остается незашифрованным, так как он нужен маршрутизаторам для транспортировки пакетов по сети.

Во второй схеме шифрацию и дешифрацию выполняют пограничные маршрутизаторы, которые отделяют частные сети предприятия от публичной сети Internet. Эти маршрутизаторы полностью зашифровывают пакеты IPv6, получаемые от конечных узлов в исходном виде, а затем инкапсулируют (эта операция и дала название заголовку - Encapsulating) зашифрованный пакет в новый пакет, который они посылают от своего имени. Информация, находящаяся в заголовке "EncapsulatingSecurityPayload", помогает другому пограничному маршрутизатору-получателю извлечь зашифрованный пакет, расшифровать его и направить узлу-получателю.

В третьей схеме один из узлов самостоятельно выполняет операции шифрации-дешифрации, а второй узел полагается на услуги маршрутизатора-посредника.

**7.1.2. Гарантированная пропускная способность**

Многие аналитики считают, что сеть Internet сможет приспособиться к требованиям времени только в том случае, если она сможет предложить своим абонентам такие же гарантии по предоставляемой пропускной способности, которые сегодня являются обычными для пользователей сетей framerelay и ATM. Это значит, что сети IP должны достаточно тонко различать классы трафика и, в зависимости от класса, гарантировать либо определенную постоянную пропускную способность (например, для голосового трафика), либо среднюю интенсивность и максимальную пульсацию трафика (например, для передачи компрессированного видеоизображения), либо предоставлять полосу пропускания не ниже определенного уровня (для пульсирующего компьютерного трафика).

Для обеспечения заданного качества обслуживания в протокол IPv6 введено такое понятие как **"метка потока" (flowlabel).** Метка потока - это признак, размещаемый в основном заголовке IP-пакета и указывающий принадлежность данного пакета к последовательности пакетов - потоку, для которого требуется обеспечить определенные параметры обслуживания, отличные от принятых по умолчанию.

Для того, чтобы конечные узлы могли сообщить маршрутизаторам сети требования к качеству обслуживания своих потоков, необходим дополнительный протокол. Именно для этой цели разработан протокол резервирования ресурсов RSVP (ResourcereSerVationProtocol). Этот протокол имеет пока статус проекта стандарта (draft) и состоит в следующем.

Узел-источник, который собирается передавать данные, требующие определенного нестандартного качества обслуживания, например постоянной полосы пропускания для передачи видеоинформации, посылает по сети по протоколу RSVP специальное сообщение. Это сообщение, называемое сообщением о пути, содержит данные о том, какую информацию собирается пересылать по образуемому потоку узел-отправитель, и какая пропускная способность нужна для получения ее с хорошим качеством. Это сообщение передается от маршрутизатора к маршрутизатору, при этом определяется последовательность маршрутизаторов, в которых нужно зарезервировать определенную пропускную способность.

Когда узел назначения получает сообщение о пути, то он извлекает из него данные о том, какую пропускную способность он должен зарезервировать и в каких маршрутизаторах. Узлов назначения может быть и несколько, если узел-отправитель хочет начать мультивещательную сессию. На основании полученной информации каждый узел назначения отправляет по протоколу RSVP сообщение, с помощью которого он запрашивает у сети определенную пропускную способность для определенного потока. Это сообщение передается каждому маршрутизатору на пути от узла-отправителя до узла назначения.

Маршрутизатор, который получает такое сообщение, проверяет свои ресурсы для того, чтобы выяснить, может ли он выделить требуемую пропускную способность. Если нет, то маршрутизатор запрос отвергает. Если же да, то маршрутизатор настраивает алгоритм обработки пакетов таким образом, чтобы указанному потоку всегда предоставлялась требуемая пропускная способность, а затем передает запрос конечного узла следующему маршрутизатору вдоль пути.

По мере передачи запроса о резервировании пропускной способности по направлению к узлу-источнику он может слиться с аналогичным запросом от другого узла назначения (если сессия мультивещательная). Слияние запросов исключает ненужное дублирование потоков. При слиянии запросов удовлетворяются потребности в максимальной пропускной способности, найденной в нескольких запросах. При этом ни один из узлов-приемников не получит обслуживания с качеством ниже того, что он запросил.

Кроме протокола RSVP, существует также проект протокола RTP (Real-TimeProtocol), который должен использоваться на транспортном уровне вместо протоколов TCP и UDP в том случае, когда необходимо передавать по Internet трафик реального времени. Протокол RTP будет переносить в своем заголовке временные отметки, необходимые для успешного восстановления голоса или видеоизображения в приемном узле, а также данные о типе кодирования информации (JPEG, MPEG и т.п.).

### 7.2. Internet-провайдинг. Типовые услуги

В конце прошлого года число фирм, предоставляющих услуги доступа к информационным ресурсам Internet в нашей стране едва превышало десяток. При этом спектр услуг был достаточно однотипным (электронная почта, терминал и, как исключение, IP). Объявление компанией SovamTeleport своего проекта *Россия-OnLine* вызвало широкий резонанс, т.к. по сути впервые было заявлено о появлении настоящей информационной службы: построенной на основе технологии IP.

Прошло всего полгода и ситуация в корне изменилась. По всей стране созданы информационные центры, которые в настоящее время предлагают практически полный спектр услуг по информационному обслуживанию пользователей и подключению в различных режимах к сети Internet.

Прежде чем приступить к обзору этих фирм, рассмотрим предлагаемые технологии подключения и обслуживания. Это необходимо для отчетливого представления о том, что реально стоит за каждой строкой прайс-листа и на что пользователи могут рассчитывать.

**Виды сервиса и схемы подключения**

Для начала попробуем отклассифицировать схемы подключения и типы сервиса с точки зрения различных типов пользователей, протоколов взаимодействия (программного обеспечения) и технических средств подключения. Представить такую классификацию можно в виде таблицы:

|  |  |  |
| --- | --- | --- |
|  | **индивидуальные пользователи** | **коллективные пользователи** |
| **Тех. пом.** | установка и настройка оборудования | установка и настройка оборудования |
| **Обуч.** | консультации | учебные курсы для пользователей и администраторов |
| **UUCP** | почта (инд. почт.ящик), доступ к информационным ресурсам по почте | почта (группа почт. ящиков), подписка на ресурсы по почте для каждого пользователя |
| **Режим терминала** | почтовый ящик, telnet, FTP, news, WWW и др. ресурсы, но только в алфавитно-цифровом режиме |  |
| **IP** | почта telnet WWW News(Usenet) FTP | IP-сети, маршрутизация, DNS, рассылка почты. |
| **WWW** | домашняя страница пользователя | Домашняя страница организации, виртуальный www-сервер, реклама, регистрация www-серверов организаций, помощь в подготовке страниц. |

Конечно, никакая классификация не дает полного представления обо всем спектре услуг, но перед ней и не ставится такой задачи. Главное - получить некоторую структуру интересующей предметной области и потом заполнять клеточки этой структуры. Отчасти данная схема дает объяснение тому, что речь идет об Internet-провайдерах, а не об IP-провайдерах, так как круг предоставляемых услуг гораздо шире простого подключения и маршрутизации по протоколу IP.

Теперь рассмотрим основные схемы подключения пользователей к Internet-провайдерам. Начнем с индивидуальных пользователей.

**Подключение индивидуальных пользователей**

В общем виде схема подключения индивидуального пользователя показана на рисунке 7.1.

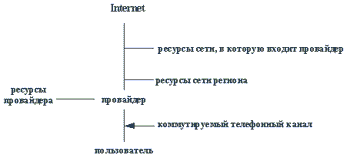


Рис. 7.1. Схема подключения индивидуального пользователя

На обоих концах коммутируемого канала ставятся модемы. Скорость обмена данными определяется качеством канала и модемом. Вообще говоря, можно использовать и другие линии связи, но реально индивидуальные пользователи в 99 случаях из ста используют обычный телефон. Аренда телефонных номеров модемного пула осуществляется в этом случае провайдером.

Индивидуальный пользователь обычно дозванивается на модемный пул провайдера, который в зависимости от числа пользователей и ресурсов последнего управляется либо персональным компьютером, либо маршрутизатором. Типовое решение для узла провайдера можно найти на домашней странице Relcom-Alpha *(http://www.dtk.net.kiae.su).*

Пользователь же имеет на своем конце персональный компьютер с модемом и больше ничего. В зависимости от типа протокола устанавливается либо программное обеспечение UUCP, либо стек TCP/IPc прикладными программами, либо, если доступ осуществляется в режиме удаленного терминала, не ставится вообще ничего. В последнем случае можно дозвониться на модемный пул провайдера при помощи любой коммуникационной программы, например, TERM95 из коллекции NortonCommander. Единственная проблема, которую приходится при этом решать - проблема кодировки. До последнего времени основной транспортной кодировкой оставалась кодировка KOI8, но мне уже приходилось получать почту и в кодировках Windows и ISO. И если в WorldWideWeb практически все провайдеры перешли к дублированию или перекодировке "на лету", то с почтой все еще приходится бороться самостоятельно. Тем не менее лучше работать в KOI8 и использовать пакеты провайдера.

Считается, что UUCP лучше подходит для работы по плохим линиям связи и для режимов обмена электронной почтой. При этом многие провайдеры не учитывают времени соединения по протоколу UUCP, полагая, что это время мало и практически полностью используется для передачи данных без простоя. Однако, при современных средствах связи и программном обеспечении такое суждение кажется несколько архаичным. Современные стеки TCP/IP можно настроить таким образом, что дозвон будет осуществляться только по требованию прикладной программы на момент передачи информации. Тем самым можно существенно сократить время простоев, а для режима электронной почты это время будет просто равно времени взаимодействия по UUCP. Кроме того, все большую популярность у наших зарубежных коллег получают *attachments*, т.е. закодированные в Latin 1 (USASCII) двоичные файлы. Это довольно большие послания, которые для своей пересылки требуют много времени, а платить, как правило, приходится и за приходящий трафик тоже.

Следует признать, что режим удаленного терминала - это полноценный доступ к ресурсам Internet. Единственным его недостатком является то, что вся информация получается пользователем в алфавитно-цифровом режиме. Ряд возможностей этого метода не используется нашими провайдерами вообще, например специальное программное обеспечение доступа с компьютера пользователя к ресурсам WorldWideWeb.

Но конечно самым полным из всех способов подключения является доступ по TCP/IP. Здесь возможно подключение либо по SLIP, либо по PPP. Провайдеры больше любят PPP, т.к. он дает возможность контроля за процессом установки соединения, раздаче IP-адресов по DHCP и аутентификации. Управлять этими параметрами на порте CISCO довольно легко, чего не скажешь о пользовательском конце соединения. Скажем, TCP/IP стек TrumpetWinsock позволяет использовать PPP, но параметров, которыми можно управлять там явно не хватает. Кроме этого довольно трудно протрассировать процесс соединения, что важно на медленных линиях. Гораздо проще это получается во FreeBSD при использовании демона pppd. Тем не менее в большинстве случаев PPP работает очень надежно.

Как правило, существует три типа ресурсов, которые доступны индивидуальному пользователю. Это локальные ресурсы провайдера, ресурсы сети или региона, например, региональный провайдер сети Релком предоставляет доступ к ресурсам московских узлов, и ресурсы всего Internet. Все сказанное относится к любому из протоколов доступа (UUCP, IP, ASCII (Терминал)). Разница заключается только в способах контроля и ограничения прав доступа, которые носят чисто административный характер, т.е., например, для подключения по протоколам TCP/IP маршрутизация пакетов присутствует всегда, но тариф на пересылку почты в рамках ответственности провайдера, сети или Internet - различный.

**Коллективные пользователи**

В качестве коллективных пользователей выступают обычно бюджетные организации и коммерческие фирмы. Общая схема подключения здесь выглядит несколько иначе (рисунок 7.2).

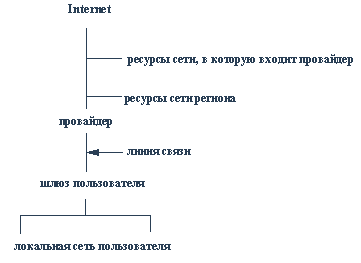


Рис. 7.2. Схема подключения коллективных пользователей

В качестве линии связи могут использоваться обычные телефонные каналы, выделенные телефонные линии, каналы системы Искра-2, наземные оптические линии связи и, в последнее время, спутниковые каналы связи (см. рекламу Demos и компании Контакт (Дубна)). При этом аренда канала связи, как и оплата оконечного оборудования связи, обычно осуществляется пользователем. Кроме того пользователь еще арендует и порт на маршрутизаторе провайдера. Стоимость этой аренды зависит от скорости обмена и типа порта.

Вообще говоря, наличие локальной IP-сети необязательно. Так в самом начале развития сети Relcom, когда ни о каком TCP/IP еще и речи не было, поддержка групповых пользователей уже осуществлялась. Программные средства UUPC и BML для персональных компьютеров позволяли (собственно прошедшее время применять для них еще рано) организовать иерархию почтовых ящиков для многих пользователей и специальный ящик администратора.

Конечно, рассматривать 286-ю машину с MS-DOS в качестве системы коллективного пользования несколько странно, но так было и во многих случаях есть до сих пор.

Существуют еще более интересные организационные реликты времен коллективного использования вычислительных ресурсов и пакетной обработки заданий. Обычно, когда речь идет о доступе к ресурсам Internet, то предполагается, что пользователь непосредственно работает с этими ресурсами со своего компьютера. В ряде случаев коллективные пользователи создают специальные компьютерные классы, из которых сотрудники этих организаций получают доступ к ресурсам сети, но организация такого сорта услуги самим провайдером - явление достаточно редкое. Тем не менее в Симбирске такого сорта услуга компанией MVC оказывается. Это следует из специального раздела прейскуранта: тарифы на базовые услуги пункта коллективного пользования ТТС. В этом прейскуранте предусмотрен набор текста в файл, распечатка текста письма, копирование письма на дискету пользователя. Вообще создается впечатление, что существует специальная служба, при которой пользователь реально не имеет доступа к компьютеру и общается только с оператором. Следует признать, что такая форма обслуживания в ряде случаев имеет право на существование и может быть выходом для пользователей, которые никогда не пользовались компьютером, но нуждаются в оперативном почтовом обслуживании.

Но все же самое интересное при обслуживании коллективных пользователей - это взаимодействие локальной сети пользователя и сети провайдера. В данном обзоре мы намеренно опускаем взаимодействие пользователей и провайдеров по протоколам, отличным от семейства TCP/IP, хотя такие формы доступа к ресурсам Internet и практикуются, например, РоСпринт или SovamTeleport и даже закладываются в качестве одного из базовых способов подключения в проект "Деловая сеть России" (ФАПСИ, ИНФОТЕЛ и AORelcom). Но при IP-подклю- чении возможны различные варианты.

Самый органичный вариант, когда локальная сеть пользователя и провайдера - это IP-сети. В этом случае пользователь должен иметь адрес в сети провайдера для своего шлюза и свою IP-сеть. Обычно это сеть класса C. Пример такого подключения показан на рисунке 7.3.

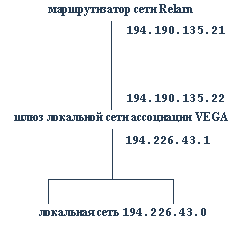


Рис. 7.3. Схема подключения локальной сети к провайдеру

Обычно, номер локальной сети следует получать у своего провайдера. При смене провайдера иногда номер остается у пользователя, но чаще возвращается провайдеру, как это делается, например, в Demos или в GlasNet. При этом одни провайдеры выдают эти номера бесплатно (Demos, AORelcom), а некоторые за дополнительную плату (Узел Sable в Красноярске).

После получения номера сети и подключения к провайдеру следует позаботится о маршрутизации. Так например, в компании МАРК-ИТТ (Ижевск) существует понятие "доступности". Существует доступность в пределах сети МАРК-ИТТ и подключение без ограничения доступности. В принципе, существуют технические возможности ограничить маршрутизацию сообщений из/в локальную сеть пользователя путем соответствующих настроек маршрутизаторов, чем некоторые провайдеры и пользуются.

Кроме маршрутизации необходимо позаботиться и о доменной адресации. В принципе можно переложить заботы о назначении доменных имен на плечи провайдера, но можно получить и свою зону. В последнем случае надо зарегистрировать у провайдера "прямую" и "обратную" зоны для своей сети. Провайдеры предлагают также поддерживать и вторичные серверы для зон пользователей. Какие могут быть причины, которые реально должны подвигнуть пользователя на регистрацию и размещение вторичных серверов DNS. Во-первых, неполадки со своим собственным сервером, а во-вторых, время разрешения запроса на IP-адрес по доменному имени. Регистрация обратных зон нужна хотя бы для того, чтобы нормально работать со всеми серверами FTP, например, для установки программного обеспечения FreeBSD с сервера АО Relcom.

**7.2.1. Режим электронной почты, подписка на телеконференции, файл-серверы и удаленный терминал**

Доступ по протоколу UUCP - это в первую очередь электронная почта. Вообще говоря, UUCP - это главным образом электронная почта, хотя есть возможность и доступа к Usenet. Однако, надо сказать, что электронная почта - это не мало. Современная электронная почта Internet позволяет работать не только с текстовой информацией, но и с графикой, и со звуком, и с изображением. Через электронную почту доступны все телеконференции Usenet (в данном случае конференции Relcom, Demos и т.п.). Через электронную почту можно передавать двоичные файлы, например, файлы программ или документы, подготовленные в форматах MS-Word или Postscript. По электронной почте доступны ftp-архивы, ресурсы WorldWideWeb и многое другое. Даже поисковые запросы к Wais можно отправлять по электронной почте. Таким образом, пользователю, подключенному по UUCP к Internet, доступен практически весь мир информационных ресурсов последней, но есть маленькое "но": все эти ресурсы доступны в режиме отложенного просмотра.

Фактически, пользователь получает их через промежутки времени равные времени заполнения его почтового ящика на почтовом сервере провайдера, если, конечно, связь с провайдером устойчивая. А эти промежутки равны, например, периодам просмотра очередей программой sendmail, которые обычно устанавливаются в пределах 30-60 минут. Конечно, при доступе к Usenet или файловым архивам это не имеет большого значения. В Usenet сообщения хранятся обычно в течении 5 дней, а в файловых архивах практически вечно. Но если пользователь предпринял путешествие по гипертекстовым ссылкам WorldWideWeb, то использование электронной почты для этой цели занятие довольно утомительное, порождающее при этом совершенно бесполезный трафик, за который еще надо платить.

Но следует также принять во внимание тот факт, что при качестве нашей телефонной сети многие отечественные пользователи не могут добиться устойчивой связи с сервером провайдера. В этом случае полный IP-сервис - это просто недоступная роскошь. Можно иметь даже собственный IP-адрес, но что в этом толку, если каждые 15-20 минут коммуникационная программа будет снова дозваниваться до сервера, а, например, ftp-соединение за это время будет оборвано из-за превышения лимита на неактивность пользователя.

При использовании протокола UUCP соединение устанавливается только на время передачи данных, и хотя другие способы подключения тоже дают эту возможность, например, и при IP-соединении можно добиться автоматического восстановления связи, с точки зрения передачи электронной почты реализовано оно в UUCP достаточно эффективно.

Очень часто можно встретить рекомендации по ограничению размера почтового сообщения. Цифры при этом называют разные, но все они обычно не превышают 1Мб. Здесь собственно следует руководствоваться следующими соображениями: во-первых, надежностью соединения, а во-вторых, ограничениями провайдера. Второй фактор важнее первого. Если провайдер ограничил размер сообщения до 1Мб, то как не старайся, больше передать просто не удастся. А на второе место поставили его по той причине, что во многих случаях ограничения установленные провайдером на много превосходят реальные возможности пользователей. За тот промежуток времени, пока держится связь, пользователь не успевает принять или отправить длинное сообщение, которое тем не менее не достигает установленного лимита. В свою очередь это вызывает новую попытку соединения, и отправка почты повторяется. Так будет происходить до тех пор пока, либо сообщение будет полностью передано, либо удалено.

На самом деле описанная процедура, с точки зрения бюджета, отведенного на оплату услуг электронной почты, не безобидна. Как только начинается прием/передача сообщений, сразу включается счетчик, как в такси. В результате платить за одну и ту же информацию придется многократно, а это просто расточительство.

Заговорив о затратах, мы вплотную подошли к вопросу о стоимости услуг по доступу к ресурсам Internet по электронной почте, к ее структуре и политике различных провайдеров в этой области.

# 8. Системы хранения и поиска данных

В основе всех современных информационных приложений находятся системы хранения данных во внешней памяти и их эффективного поиска. В зависимости от специфики приложений используются разные технологии хранения и поиска данных. Если проводить грубую классификацию таких технологий, то можно выделить два основных направления - системы управления базами данных (СУБД) и информационно-поисковые системы (ИПС). Эти направления взаимно дополняют друг друга и в совокупности обеспечивают возможность построения приложений в разных архитектурах, с разной функциональной ориентированностью, с разными требованиями к доступу к данным и т.д.

Основными характеристиками среды хранения данных, управляемой СУБД, являются:

* структурированная природа хранимых данных, причем описание структуры является частью самой базы данных, т.е. известна СУБД (соответствующую часть базы данных иногда называют метабазой данных);
* существенно частое обновление данных, эффективно поддерживаемое СУБД и производимое многими пользователями, одновременно подключенными к базе данных;
* автоматическое поддержание целостного состояния базы данных на основе механизма управления транзакциями и хранимого в метабазе данных набора ограничений целостности и/или триггеров;
* поддержка структурированных языков доступа к базе данных, в которых условие поиска представляет собой логическую связку разнообразных простых условий, накладываемых на содержимое требуемых записей.

Среду хранения данных, управляемую ИПС, характеризует в основном следующее:

* отсутствие структуризации данных, известной ИПС; как правило, в таких системах хранятся полнотекстовые документы, структура которых (если речь идет о структурированных документах) известна соответствующим приложениям;
* редкое обновление данных, уже включенных в среду хранения; в частности, во многих ИПС обновление среды хранения данных (изменение, удаление, добавление документов) производится только в специальном режиме, не допускающем одновременную выборку информации;
* отсутствие встроенного в систему механизма поддержания целостности данных; необходимые механизмы контроля вводимой информации должны являться частью приложения;
* наличие развитых механизмов выборки информации, основанных на логических связках простых условий, накладываемых на содержание искомых документов; примерами простых условий является указание ключевых слов, характеризующих документ, или контекста, относящегося к содержимому документа.

ИПС по естественным причинам никогда не претендовали на роль СУБД. Однако обратное неверно, и многие производители СУБД пытаются ввести в состав своих продуктов возможности, позволяющие использовать их в качестве систем информационного поиска. Нужно заметить, что особых успехов в этой деятельности не видно. Главным образом, это связано с реляционной спецификой современных серверов баз данных, с их ориентацией на обеспечение эффективного доступа к структурированной информации. С другой стороны, многие возможности серверов баз данных в действительности не требуются для поддержки полнотекстовой информации. К таким возможностям можно отнести развитые средства управления транзакциями, организацию развитых средств динамического обновления данных, реализацию структурированных языков запросов и т.д.

В дальнейшем в этом разделе курса мы не будем затрагивать специфические свойства ИПС. Это связано не с тем, что мы считаем это направление недостаточно существенным для организации информационных систем. Все дело в том, что в настоящее время основной областью применения ИПС являются распределенные информационные системы, базирующиеся на Web-технологии. Естественно, имеются смежные вопросы, касающиеся, например, возможностей доступа к традиционным базам данных в среде Internet. Тем не менее, однако, обе эти области достаточно широки по отдельности, чтобы можно было рассматривать их независимо.

### 8.1. Новые веяния в области серверов реляционных баз данных

Начнем с того, что текущее поколение программных продуктов, предназначенных для управления базами данных, практически полностью базируется на классической реляционной модели данных, которая в той или иной степени развивается и модифицируется в разных системах. Реляционная модель данных обладает большим числом достоинств и, конечно, многими недостатками. К числу достоинств можно отнести простую и вместе с тем мощную математическую основу модели, базирующейся на наиболее прочных аппаратах теории множеств и формальной логике первого порядка. Пожалуй, реляционная модель является исключительным примером соразмерности используемых математических средств и получаемых от этого преимуществ. Достоинством реляционного подхода является и его интуитивная ясность. Для того, чтобы начать грамотно использовать реляционную СУБД, совсем не требуется глубокое погружение в формальную математику. Достаточно понять житейский смысл объектов реляционных баз данных и научиться ими пользоваться. Проводятся интуитивные аналогии, не нарушающие смысл понятий, между отношениями-таблицами-файлами, атрибутами-столбцами-полями записей, кортежами-строками-записями файла и т.д.

Как всегда бывает в жизни, отрицательные качества реляционного подхода представляют обратную сторону его достоинств. Очень просто представлять информацию в виде регулярных плоских таблиц, в которых каждая строка имеет одну и ту же структуру, а в столбцах могут храниться только простые данные атомарной структуры. Но одновременно, при использовании реляционных баз данных для хранения сложной информации возникают сложности. Известный консультант и аналитик Эстер Дайсон сравнивает использование реляционных баз данных для хранения сложных объектов (объектов, для представления которых недостаточно использовать плоские таблицы) с необычным использованием гаража, когда каждый раз, устанавливая автомобиль в гараж, шофер полностью его разбирает, а утром выполняет полную процедуру сборки. На самом деле, соответствующие процедуры сборки (соединения нескольких таблиц) являются наиболее трудоемкими в реляционных СУБД. Другая проблема состоит в том, что упрощенная структура реляционных баз данных не позволяет сохранить в базе данных ту семантическую информацию (знания), которыми располагал ее создатель при проектировании. Реально ситуация выглядит следующим образом. На первой, наиболее интеллектуальной фазе проектирования базы данных имеется существенный запас данных, почерпнутых проектировщиком в процессе анализа предметной области. Однако по мере перехода к определению реально хранимых структур эти знания теряются, оставаясь в лучшем случае в виде бумажной или не привязанной к базе данных электронной информации, использование которой необходимо в жизненном цикле базы данных и соответствующей информационной системы, но сильно затруднено.

Перечисленные положительные и отрицательные качества реляционных баз данных в большой степени влияли на развитие индустрии баз данных в последние годы. Несколько лет тому назад казалось, что перевешивают отрицательные качества. В результате появилось несколько новых направлений архитектур баз данных, каждое из которых принесло новые методы и алгоритмы, а также экспериментальные реализации. Мы не будем глубоко вдаваться в обсуждение этих подходов, но все же приведем их краткую характеристику.

В классической реляционной модели данных содержимым столбцов могут быть только значения базовых типов данных (целые и плавающие числа, строки символов и т.д.). Не допускается хранение в столбце таблицы массивов, множеств, записей и т.п. Собственно, это и означает, что таблицы классической реляционной модели являются абсолютно плоскими (по-другому это называется представлением таблиц в первой нормальной форме). Вся сложность структур предметной области уходит в динамику; требуется непрерывная сборка сложных объектов из их элементарных составляющих. Один из подходов к расширению возможностей реляционной модели данных заключается в отказе от требования первой нормальной формы. Значением столбца таблицы может быть любой объект, представляемый в виде строки, массива, списка и даже таблицы. Понятно, что при использовании таким образом расширенной модели, в качестве строки таблицы может храниться уже собранный объект с произвольно сложной (иерархической) структурой. Причиной того, что реляционные системы с отказом от требования первой нормальной формы не вошли в широкую практику, является прежде всего необходимость полного пересмотра внутренней организации СУБД (начиная со структур хранения). Кроме того, если для классических реляционных баз данных давно и тщательно отработана технология и методология проектирования, то проектирование ненормализованных реляционных баз данных недостаточно хорошо изучено даже на уровне теории. Тем не менее, на рынке СУБД имеется продукт UniVerse компании VMark, в которой в ограниченных масштабах поддерживается хранение ненормализованными реляционными данными.

Понятно, что ограничением ненормализованных реляционных баз данных является обязательность иерархической организации хранимой информации. Конечно, это совсем другой уровень построения информации, чем тот, который поддерживался в дореляционных иерархических системах. Между элементами данных отсутствуют явно проводимые физические ссылки. Упрощена процедура реструктуризации данных (в частности, в ненормализованной реляционной модели предполагается наличие пары операций nest/unnest, позволяющих сделать существующую таблицу элементом, хранимым в столбце другой таблицы, или "вытянуть" на верхний уровень некоторую вложенную таблицу). Тем не менее, общая структура остается иерархической. Во многих случаях этого достаточно (в частности, для решения проблемы, упоминавшегося выше примера Эстер Дайсон).

Однако существуют потребности, выходящие за пределы возможностей иерархических организаций данных. Стандартным примером является необходимость в моделировании сложных объектов, в которые входит один и тот же подобъект. Например, объект "человек" может являться подобъектом объекта "отдел", подобъектом объекта "семья" и т.д. Для определения требуемых структур недостаточна чисто иерархическая организация, требуется возможность построения сетевых структур. Соответствующее направление получило название "базы данных сложных объектов". Поскольку трудно сказать, насколько сложные структуры потребуются при реальном моделировании предметных областей, должен поддерживаться механизм произвольно сложной структуризации с поддержкой возможности определения структурных типов данных, типов массивов, списков и множеств, а также развитыми средствами использования указателей. Конечно, с учетом опыта реляционных баз данных в базах данных сложных объектов никогда не предполагалось использование адресных указателей физического уровня. Одним из естественных требований было наличие возможности взятия из базы данных произвольно сложного объекта с возможностью его перемещения в другую область внешней памяти или вообще в память другого компьютера. Основной проблемой баз сложных объектов являлось отсутствие простого и мощного интерфейса доступа к данным, хотя бы частично сравнимого с простотой и естественностью реляционных интерфейсов.

Возникновение направления объектно-ориентированных баз данных (ООБД) определялось прежде всего потребностями практики: необходимостью разработки сложных информационных прикладных систем, для которых технология предшествующих систем баз данных не была вполне удовлетворительной. Как мы видели, определенными недостатками обладали и классические реляционные системы, и системы, основанные на ненормализованной реляционной модели, и базы данных сложных объектов.

Конечно, ООБД возникли не на пустом месте. Соответствующий базис обеспечивался как предыдущими работами в области баз данных, так и давно развивающимися направлениями языков программирования с абстрактными типами данных и объектно-ориентированных языков программирования.

Что касается связи с предыдущими работами в области баз данных, то наиболее сильное влияние на работы в области ООБД оказали проработки реляционных СУБД и следующего хронологически за ними семейства БД, в которых поддерживалось управление сложными объектами. Эти работы обеспечили структурную основу организации OOБД.

Среди языков и систем программирования наибольшее первичное влияние на ООБД оказал Smalltalk. Этот язык сам по себе не являлся полностью пионерским, хотя в нем была введена новая терминология, являющаяся теперь наиболее распространенной в объектно-ориентированном программировании. На самом деле, Smalltalk основан на ряде ранее выдвинутых концепций.

В наиболее общей и классической постановке объектно-ориентированный подход базируется на следующих концепциях:

1. объекта и идентификатора объекта;
2. атрибутов и методов;
3. классов;
4. иерархии и наследования классов.

Любая сущность реального мира в объектно-ориентированных языках и системах моделируется в виде объекта. Любой объект при своем создании получает генерируемый системой уникальный идентификатор, который связан с объектом во все время его существования и не меняется при изменении состояния объекта.

Каждый объект имеет состояние и поведение. Состояние объекта - набор значений его атрибутов. Поведение объекта - набор методов (программный код), оперирующих над состоянием объекта. Значение атрибута объекта - это тоже некоторый объект или множество объектов. Состояние и поведение объекта инкапсулированы в объекте; взаимодействие объектов производится на основе передачи сообщений и выполнении соответствующих методов.

Множество объектов с одним и тем же набором атрибутов и методов образует класс объектов. Объект должен принадлежать только одному классу (если не учитывать возможности наследования). Допускается наличие примитивных предопределенных классов, объекты-экземпляры которых не имеют атрибутов: целые, строки и т.д. Класс, объекты которого могут служить значениями атрибута объектов другого класса, называется доменом этого атрибута.

Допускается порождение нового класса на основе уже существующего класса - наследование. В этом случае новый класс, называемый подклассом существующего класса (суперкласса) наследует все атрибуты и методы суперкласса. В подклассе, кроме того, могут быть определены дополнительные атрибуты и методы. Различаются случаи простого и множественного наследования. В первом случае подкласс может определяться только на основе одного суперкласса, во втором случае суперклассов может быть несколько. Если в языке или системе поддерживается единичное наследование классов, набор классов образует древовидную иерархию. При поддержании множественного наследования классы связаны в ориентированный граф с корнем, называемый решеткой классов. Объект подкласса считается принадлежащим любому суперклассу этого класса.

Одной из более поздних идей объектно-ориентированного подхода является идея возможного переопределения атрибутов и методов суперкласса в подклассе (перегрузки методов). Эта возможность увеличивает гибкость, но порождает дополнительную проблему: при компиляции объектно-ориентированной программы могут быть неизвестны структура и программный код методов объекта, хотя его класс (в общем случае - суперкласс) известен. Для разрешения этой проблемы применяется, так называемый, метод позднего связывания, означающий, по сути дела, интерпретационный режим выполнения программы с распознаванием деталей реализации объекта во время выполнения посылки сообщения к нему. Введение некоторых ограничений на способ определения подклассов позволяет добиться эффективной реализации без потребностей в интерпретации.

Видимо, наиболее важным новым качеством ООБД, которое позволяет достичь объектно-ориентированный подход, является поведенческий аспект объектов. В прикладных информационных системах, основывавшихся на БД с традиционной организацией (вплоть до тех, которые базировались на семантических моделях данных), существовал принципиальный разрыв между структурной и поведенческой частями. Структурная часть системы поддерживалась всем аппаратом БД, ее можно было моделировать, верифицировать и т.д., а поведенческая часть создавалась изолированно. В частности, отсутствовали формальный аппарат и системная поддержка совместного моделирования и гарантирования согласованности этих структурной (статической) и поведенческой (динамической) частей. В среде ООБД проектирование, разработка и сопровождение прикладной системы становится процессом, в котором интегрируются структурный и поведенческий аспекты. Конечно, для этого нужны специальные языки, позволяющие определять объекты и создавать на их основе прикладную систему.

С точки зрения разработчиков информационных систем подход OOБД кажется очень заманчивым. Более того, на рынке программных продуктов управления базами данных сегодня существует около двух десятков коммерческих систем, которые более или менее успешно продаются (примерами таких систем являются O2 компании O2 Technology (www.o2tech.com), ObjectStore компании ObjectDesignInc., (www.odi.com), Objectivity/DB компании Objectivity, Inc., Versant компании VersantObjectTechnology (www.versant.com), ONTOSDB компании ONTOS, Inc., (www.ontos.com) и т.д.). Во всех этих системах поддерживается возможность распределенного хранения баз данных; имеется возможность написания приложений и/или методов объектов на одном или нескольких языках объектно-ориентированного программирования (как правило, в минимальный набор языков входят Си++ и Java); обеспечиваются удобные средства доступа к базам данных в среде Internet и т.д. Тем не менее, объектно-ориентированные системы оказались не в состоянии конкурировать с реляционными системами, поставляемыми ведущей шестеркой поставщиков программных средств управления базами данных.

Прежде чем перейти к краткому обзору современных продуктов ведущих компаний, попробуем понять, почему же большинство заказчиков предпочитает использовать именно эти продукты, а не объектно-ориентированные СУБД. Мы можем привести несколько соображений, некоторые из которых являются в большей степени эмоциональными, а другие - чисто техническими. Во-первых, компьютерное сообщество уже пережило техническую революцию при переходе от дореляционных СУБД к реляционным. Как и любая революция, эта техническая революция была пережита непросто. Хотя и очень простые, идеи реляционного подхода воспринимались широкими массами пользователей и разработчиков информационных систем на протяжении нескольких лет. Переход к новым технологиям вызвал необходимость в реинжиниринге, а иногда и полной переделке существующих и используемых практически информационных систем. В результате, конечно, были получены более качественные продукты, но одновременно с этим обострилась известная проблема "унаследованных" ("legacy") систем, которые являются морально устаревшими, плохо сопровождаемыми, но необходимыми для успешного функционирования предприятия. Переход к объектно-ориентированной технологии баз данных означал бы новую революцию. Потребовалась бы качественная, основанная на иных понятиях переделка прикладного программного обеспечения. Естественно, это отпугнуло пользователей и разработчиков от объектно-ориентированных баз данных.

Во-вторых, любая развитая система управления базами данных является предельно сложным программным продуктом, для эффективной реализации которого требуется привлечение правильным образом разработанного или выбранного из числа готовых набора методов, алгоритмов, протоколов и структур данных. Кроме того, для достижения должного уровня эффективности СУБД должна пройти длительный процесс отладки, обкатки и настройки. Ведущие производители реляционных СУБД в той или иной степени успешно решили эти проблемы за счет больших денежных и временных затрат. Конечно, сегодня невозможно говорить о какой-либо объектно-ориентированной СУБД, которая была бы настолько же хорошо отлажена и настроена, которая могла настолько же эффективно обрабатывать большие объемы данных, как продукты ведущей шестерки.

В-третьих, одним из основных преимуществ реляционного подхода по отношению к дореляционным системам является наличие ненавигационного интерфейса доступа к базам данных. Отсутствие явной навигации в базе данных позволяет освободить прикладную программу от технических деталей ассемблерного уровня (можно проводить аналогию между явным переходом по ссылке в структуре внешних данных и безусловным переходом в языках уровня ассемблера), а также дает возможность более эффективного по сравнению с ручным выполнения операций доступа к данным (когда способ выполнения непроцедурно заданного оператора выбирается в компиляторе соответствующего языка, то используется гораздо больший объем знаний, чем тот, которым располагает отдельно взятый человек). Отрицательным свойством ненавигационного, основанного на манипуляциях с таблицами способа доступа к базам данных является так называемая потеря соответствия (impedancemismatch) языков программирования и языков баз данных. Классические, наиболее используемые на практике языки программирования ориентированы на работу с атомарными значениями встроенных типов данных. Даже если в языке содержатся средства определения массивных типов данных (структур, массивов, множеств, списков и т.д.), то перед выполнением любой обрабатывающей операции необходимо выбрать атомарный элемент соответствующего массивного типа. Языки реляционных баз данных ориентированы на работу с таблицами: операндами любой операции являются таблицы, и в результате выполнения операции формируется новая таблица. Фактически, языки программирования и языки реляционных баз данных ортогональны:

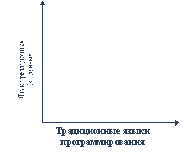


Рис. 8.1. Ортогональность языков программирования и языков баз данных

То, что предлагается в языке SQL относительно возможности встраивания его конструкций в традиционный язык программирования - это попытка сгладить эту ортогональность. (Имеются в виду средства встраиваемого SQL для развертывания операций обработки результата запроса к базе данных в последовательность или цикл обработки ее строк.)

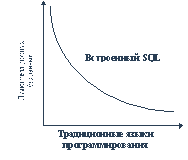


Рис. 8.2. Сглаживание ортогональности средствами языка SQL

Одной из задач, которую ставили перед собой разработчики подхода ООБД, состояла в том, чтобы добиться отсутствия потери соответствия между языками объектно-ориентированного программирования и языками объектно-ориентированных баз данных. Идеалом представлялось то, чтобы один и тот же язык использовался и для программирования приложений (и написания методов объектов), и для обеспечения доступа к базе данных. Поскольку природа объектно-ориентированных языков не изменилась по сравнению с их предшественниками, средства доступа к базе данных естественно стали снова навигационными. Конечно, это другой уровень навигации с использованием логических ссылок, но навигация есть навигация - за одно обращение к базе данных можно получить возможность работы с одним объектом. Естественно, и это отпугнуло разработчиков и пользователей.

Заметим, что в последнее время ситуация изменилась. В результате деятельности международного консорциума ObjectDatabaseManagementGroup (ODMG) был выработан стандарт языка запросов (OQL - ObjectQueryLanguage) к ООБД (в июле 1997 г. опубликована его вторая версия). Этот язык ненавигационный и синтаксически близок к языку SQL. Но для текущего поколения объектно-ориентированных СУБД язык OQL появился слишком поздно, и с этим связана вторая причина неудачи объектно-ориентированных СУБД на рынке.

До возникновения стандартов ODMG существовало столько же разных представлений о природе ООБД, сколько разных соответствующих продуктов. Если реляционные системы с самого начала основывались на одной и достаточно точной модели, то для ООБД такой общей модели не было вообще. Все попытки создать такую модель требовали привлечения весьма трудного математического аппарата, обычно непонятного даже профессиональным математикам. Группа ODMG, включающая всех основных поставщиков объектно-ориентированных СУБД, смогла предложить общую объектную модель баз данных, но и эта модель появилась слишком поздно, а кроме того, она слишком неточна и является результатом многочисленных компромиссов. Естественно, необоснованные теоретически объектно-ориентированные СУБД не могли вызвать слишком большого доверия.

Повторим, что ситуация меняется. Если раньше между ODMG и сообществом реляционных баз данных (в частности, комитетом по стандартизации языка SQL) существовали принципиальные разногласия, то сегодня обе стороны стремятся к сближению позиций. Тем не менее, на сегодняшний день рынок объектно-ориентированных систем по крайней мере на порядок уже рынка реляционных систем и, возможно, еще более сузится в связи с возникновением нового сектора коммерческих объектно-реляционных систем, поставку которых начинают ведущие компании.

**8.1.1. Обзор современного состояния рынка серверов баз данных**

Естественно, мы не можем в этом подразделе достаточно детально представить даже наиболее известных производителей серверных продуктов реляционных баз данных и собственно эти продукты. Это связано и с тем, что по поводу каждой компании можно сказать очень много, и с тем, что серверные продукты управления базами данных являются, по всей видимости наиболее сложными программными продуктами, присутствующими на рынке, и с тем, что компании-производители серверных продуктов очень не любят открывать технические детали реализации (это считается частью "know-how" компании). Мы постараемся охарактеризовать деятельность и продукты ведущей шестерки производителей (компаний Oracle *(www.oracle.com)*, Informix *(www.informix.com)*, Sybase *(www.sybase.com)*, ComputerAssociates *(www.cai.com)*, IBM *(www.ibm.com)* и Microsoft *(www.microsoft.com))*, касаясь только особенностей серверов баз данных и не затрагивая возможности громадного числа сопутствующих программных продуктов. Кроме того, мы не будем обсуждать продукты компаний "второго эшелона", хотя зачастую они обладают оригинальными и специфическими возможностями.

**8.1.1.1. История и серверные продукты компании Oracle**

Первая доступная заказчикам версия СУБД Oracle (OracleV.2) была выпущена компанией RelationalSoftwareInc. в 1979 г. Эта версия была ориентирована на использование в среде ОС RSX-11 для семейства миникомпьютеров PDP-11. Система была написана большей частью на языке ассемблера PDP-11, но включала также тексты на новом для того времени языке Си. СУБД могла функционировать не только в среде ОС RSX-11, но и в других операционных средах, поддерживаемых на PDP-11: IAS, RSTS и UNIX. Тогда же было принято решение о переносе Oracle в 32-разрядную операционную среду VAXVMS, что на долгие годы определило судьбу СУБД Oracle как ведущей системы управления базами данных на миникомпьютерах.

Наиболее сильное впечатление на пользователей Oracle производила возможность манипулирования базами данных на основе непроцедурного реляционного языка SQL, для которого в то время существовал только фирменный стандарт компании IBM. СУБД OracleV.2 обладала ограниченными возможностями; в частности, в системе отсутствовали средства управления транзакциями, что заставляло пользователей постоянно прибегать к использованию утомительной процедуры резервного копирования.

СУБД OracleV.3 была выпущена в свет в 1983 г. Она была полностью переписана на языке программирования Си, что в дальнейшем позволило решить проблему переноса системы на большое число аппаратно-программных платформ. Функции, доступные через SQL-интерфейс, были расширены. В обиход пользователей системы было введено понятие транзакции. В это же время компания RelationalSoftwareInc. была переименована в OracleCorp.

В 1985 г. на рынке появилась СУБД OracleV.5. В этой системе использовалась архитектура "клиент-сервер" и впервые появилась (ограниченная) возможность одновременного использования баз данных, расположенных в разных узлах сети.

Шестая версия системы представляла из себя инструмент построения корпоративных информационных приложений, выполняющихся в режиме OLTP. В системе были реализованы система блокировок на уровне записей (заметим, что хотя этот уровень блокировок кажется наиболее естественным, переход к его использованию происходил и происходит непросто; в частности, в очень развитых последних по времени продуктах компании Sybase до сих пор используются блокировки на уровне страниц), а также механизм непротиворечивых чтений из базы данных без потребности блокировок (это легко реализуемое оригинальное изобретение Oracle пока не воспроизведено в продуктах других компаний и, естественно, не стандартизовано). СУБД OracleV.6 была перенесена на ряд новых платформ, в том числе, OS/2 и Macintosh.

Важными нововведениями шестой версии было появление процедурного языка программирования PL/SQL, который можно было использовать как для определения процедур, хранимых на сервере баз данных, так и для разработки приложений в составе языка четвертого поколения SQL\*Forms. Кроме того, в реализованном варианте языка SQL появились средства определения ссылочной целостности (referenceintegrity), одного из наиболее фундаментальных механизмов поддержания целостности в реляционных базах данных.

Седьмой выпуск СУБД Oracle появился на рынке в середине 1994 г. Это один из наиболее серьезных серверных продуктов, предназначенных для управления реляционными базами данных. В OracleV.7 используется ряд новых архитектурных решений, направленных на повышение эффективности сервера, в том числе буферизация откомпилированных SQL-операторов на сервере баз данных, использование общего пула серверных процессов и нитей для выполнения операторов SQL, поступающих от разных транзакций, использование разнообразной статистической информации для оптимизации запросов и т.д. Заметим, что именно в седьмой версии начался переход от использования оптимизатора запросов, управляемого правилами, к применению более прогрессивной техники оптимизации запросов на основе статистических оценок.

Существенно расширены возможности использования языка PL/SQL. В OracleV.7 этот язык можно использовать для определения триггеров, хранимых процедур, вызываемых сервером автоматически при возникновении специфицированных событий (например, выполнении операций модификации таблицы в целом или обновлении конкретной строки таблицы). Функции PL/SQL могут вызываться как обычные встроенные функции в операторах SQL. Заметим, что относительным, но существенным недостатком языка PL/SQL является то, что он не входит в состав международного стандарта SQL, хотя многие его свойства нашли отражение в вышедшем в этом году стандарте языка хранимых модулей, являющемся частью будущего стандарта SQL-3.

В распределенном варианте OracleV.7 поддерживаются возможности репликации данных и имеется возможность асинхронного вызова удаленных процедур.

Летом 1997 г. был объявлен выпуск восьмой версии системы, которая должна обладать целым рядом новых возможностей: встроенными средствами для использования в Internet/Intranet, поддержкой хранения мультимедийной информации, приближением реализованного варианта языка SQL к разрабатываемому стандарту языка SQL-3 и т.д. Поскольку эта версия (Oracle 8.1) является переходной от реляционного к объектно-реляционному подходу.

**8.1.1.2. История и серверные продукты компании Informix**

24 сентября 1996 г. компания Informix отметила десятилетие своей официальной деятельности. За эти годы компания увеличила объем своих доходов в 33 раза и уверенно занимает второе место на мировом рынке продуктов, связанных с управлением реляционными базами данных. С самого начала своего существования компания ориентировалась на создание серверов баз данных и сопутствующих программных продуктов, функционирующих в среде ОС UNIX. В число основных стратегических партнеров Informix входят компании Sequent, HewlettPaсkard, SunMicrosystems, IBM, SiemensNixdorf, NCR, для продуктов которых в первую очередь обеспечиваются новые работоспособные версии систем Informix. Помимо UNIX-платформ продукты компании Informix могут работать в операционных средах DOS, NetWare, Windows и WindowsNT.

Характерной особенностью компании Informix является то, что она поддерживает, развивает и поставляет на рынок целое семейство серверов, отличающихся возможностями, эффективностью и, естественно, ценой. Все разновидности серверных продуктов Informix базируются на архитектуре "клиент-сервер" (мы приведем краткий обзор наиболее ярких представителей семейства).

Самым простым серверным продуктом является сервер баз данных Informix-SE. Он предназначен для использования в информационных системах со средним (или малым) объемом хранимой информации. Хранение данных поддерживается на уровне файловой системы, и на этом же уровне осуществляется синхронизация доступа со стороны параллельно выполняемых транзакций. На самом деле, в Informix-SE для каждой пользовательской транзакции образуется отдельный серверный процесс, и эти процессы взаимодействуют только при доступе к общим файлам базы данных. (Заметим, что это сильно напоминает организацию систем управления базами данных для персональных компьютеров.) Клиент и сервер могут располагаться в одном компьютере, но могут быть и разнесены на разные компьютеры, связанные сетью. Естественно, что при наличии выделенной аппаратуры, поддерживающей деятельность сервера, общая эффективность системы возрастает. Связь между клиентами и серверами поддерживается специальным модулем Informix-NET.

Базовым продуктом компании Informix является система Informix-OnLine, выпускаемая ныне в двух основных модификациях - Informix-OnLineDynamicServer и Informix-OnLineExtendedParallelServer. Эти серверы работают напрямую с дисковой памятью, обеспечивают выполнение транзакций в распределенной среде баз данных, поддерживают возможности хранения неструктурированных полей таблиц сверхбольшого размера (BLOBs - BinaryLargeObjects) и т.д.

Informix-OnLineDynamicServer ориентирован на применение симметричных мультипроцессорных компьютеров и опирается на параллельное использование процессоров с общей основной памятью. Поэтому в этом сервере широко используются приемы программирования, основанные на использование параллельных потоков управления, или нитей.

Informix-OnLineExtendedParallelServer может работать как в симметричных, так в несимметричных (sharingnothing) компьютерных архитектурах. При использовании несимметричных архитектур обещается наличие почти линейной масштабируемости.

В конце 1996 г. компания Informix объявила о выпуске объектно-реляционного сервера InformixUniversalServer. Поскольку этот продукт относится к новому поколению систем управления базами данных, отложим его обсуждение до п.10.1.4.

Informix утверждает, что особенностью стратегии компании является полное отсутствие конкуренции с любым из своих потенциальных партнеров. В отличие от Oracle, Informix производит только базовые продукты, не навязывая своей технологии разработки информационных приложений (это мнение компании Informix, а не автора данного раздела).

**8.1.1.3. Серверные продукты компании Sybase**

Компания Sybase является сравнительно новой на рынке конкурирующих производителей современных реляционных СУБД. Это одновременно дает компании ряд преимуществ и усложняет ее работу, хотя, несмотря на некоторые временные неудачи, продукты Sybase находятся на третьем месте в мире по числу продаж. Преимущества компании состоят в том, что она не настолько обремлена грузом предыдущих разработок и необходимостью их постоянной поддержки. Преимуществом является и то, что Sybase с меньшими потерями переходит к использованию новых архитектурных и технологических решений. Усложняет же работу компании тот факт, что при выпуске каждого очередного варианта сервера БД ей приходится решать множество новых архитектурных и технологических проблем (никуда не денешься: если компания провозглашает себя лидером в области архитектур и технологий серверов баз данных, то она должна поддерживать марку).

До выпуска в 1994 г. полномасштабного серверного продукта SybaseV.10 компания Sybase уверенно зарекомендовала себя в качестве ведущего производителя современных СУБД для применения в средних и малых информационных приложениях. Полностью основанная на архитектуре "клиент-сервер" SybaseV.10 могла использоваться на большинстве аппаратно-программных платформ: Sun, HP, IBMRS/6000, DigitalVAX/VMS, DigitalAlphaOpenVMS и AlphaOSF, NCR, NEC, Sequent, SiliconGraphics, NetWare, WindowsNT, OS/2, SCO и т.д. Архитектура SybaseV.10 обладала следующими характерными чертами:

* компонентная структура системы позволяла изменять отдельные компоненты, не нарушая работу других компонентов;
* в системе поддерживалось большинство принятых международных стандартов;
* поддерживалась работа как с другими реляционными источниками данных, так и с источниками данных унаследованных систем;
* обеспечивалась простая переносимость системы;
* система хорошо оптимизировалась для использования в данной предметной области, поскольку отдельные функциональные компоненты могли настраиваться независимо один от другого;
* гарантировалась высокая надежность системы: изменения, вносимые в один компонент не влияли на надежность других компонентов; были реализованы и расширены такие средства стандарта языка SQL-92, как хранимые процедуры, триггеры, средства поддержания ссылочной целостности, определяемые пользователем типы данных и т.д.;
* поддерживалось специфицированное X/Open управление распределенными транзакциями;
* были реализованы возможности адаптации к национальному языку, включая определения набора символов для выдачи сообщений, порядка сортировки и т.д.; появилась возможность русскоязычной идентификации таблиц и их столбцов.

В общем, по своим идеям система была правильной. К сожалению, как это свойственно компаниям, имеющим серьезных конкурентов, Sybase слишком поторопилась с выпуском на рынок SybaseV.10. Система появилась на рынке не вполне отлаженной, и это привело к тому, что в 1995-1996 гг. многие потенциальные и реальные покупатели перестали иметь с ней дело. Такого эффекта очень легко добиться, но его трудно устранить. В начале 1996 г. компания объявила о выпуске нового продукта, SybaseV.11.

В основной состав серверных продуктов SybaseV.11 входит следующее:

* Базовый сервер SybaseSQLServer - современная высокопроизводительная СУБД (более подробно по поводу этого продукта см. ниже);
* SybaseMPP - расширение архитектуры SybaseSQLServer, предназначенного для эффективного использования в массивно параллельных компьютерных архитектурах с поддержкой сверхбольших баз данных (VeryLargeDataBases - VLDB);
* SybaseIQ - серверное средство построения битовых индексов для высокоскоростного выполнения запросов к большим источникам информации;
* SybaseSQLAnywhere - полнофункциональная "облегченная" СУБД, приобретенная от компании Watcom и предназначенная для производства индивидуальных и групповых информационных систем на платформах Intel;
* SybaseReplicationServer - серверный продукт, поддерживающий репликацию данных;
* SybaseOmniServer - сервер, обеспечивающий "прозрачную" работу клиентов с несколькими серверами баз данных, вообще говоря, различных производителей: Sybase, Oracle, DB2 и т.д.

Имеется также ряд вспомогательных серверных средств, поддерживающих динамическую (на фоне выполнения производственных транзакций) загрузку и выгрузку данных, мониторинг действий пользователей и т.д. Как видно, компания Sybase продолжает проводить свою линию на компонентную организацию серверных средств. Далее мы обсудим только возможности базового сервера SybaseSQLServer 11, не вдаваясь в детали организации и возможностей дополнительных серверов (что было бы, кстати, нечестно по отношению к конкурентам компании Sybase).

В соответствии с утверждениями представителей компании Sybase, продукт SybaseSQLServer 11 обладает следующими основными возможностями:

1. Масштабируемость и эффективность SQLServer 11 основываются на тщательно проверенной технологии:

* сервер может работать на большом числе платформ, начиная от персональных компьютеров и заканчивая мощными мультипроцессорными серверами;
* на каждой платформе обеспечивается очень высокая эффективность (без настройки на конкретную платформу обойтись нельзя!) благодаря тесному взаимодействию с производителями аппаратуры и базового программного обеспечения;
* в ядре СУБД используется полностью симметричная многопотоковая архитектура, позволяющая использовать возможности аппаратуры и поддерживающая большое число пользователей.

2. SQLServer 11 обеспечивает надежность хранения и целостность данных:

* поддерживаются механизмы триггеров и хранимых процедур, декларативной ссылочной целостности, управления транзакциями и т.д.;
* как и полагается SQL-ориентированной СУБД, SQLServer 11 поддерживает уровень безопасности данных C2 в соответствии с требованиями Оранжевой Книги Министерства обороны США.

3. Обеспечивается повышенная доступность данных:

* на программном уровне поддерживаются зеркальные копии журнала и самой базы данных;
* для восстановления базы данных после сбоев применяются специально разработанные механизмы высокоскоростной перезагрузки.

4. В SQLServer 11 обеспечивается соответствие основным принятым формально или фактически стандартам:

* реализованный в системе вариант языка SQL полностью соответствует стандарту SQL-89, а также ядерному уровню (entrylevel) стандарта SQL-92;
* поддерживается выполнение приложений, выполненных в стандартах ODBC и X/OpenXA;
* применяются различные сетевые протоколы, что позволяет соединить клиента и сервера практически на любой платформе.

5. Гарантируется простота управления системой и ее поддержки:

* продуманная многопотоковая архитектура системы обеспечивает то, что на однопроцессорном компьютере запускается только один процесс сервера;
* при использовании симметричной мультипроцессорной аппаратной организации можно указывать количество процессоров, которые должны использоваться для целей СУБД;
* можно управлять распределением областей внешней памяти, контролировать доступ пользователей к БД и т.д. в масштабах индивидуальной системы, масштабах ограниченного предприятия или масштабах реальной корпоративной сети.

В целом, набор серверных продуктов одиннадцатого выпуска компании Sybase представляет собой основательный, хорошо продуманный комплект инструментов, которые можно с пользой применять в разнообразных приложениях. По отзывам, которые успели поступить с момента выпуска SybaseV.11, серверные средства работают достаточно надежно.

В сентябре 1997 г. компания Sybase выпустила на рынок продукт под новым названием - SybaseAdaptiveServer, который на самом деле является пятым выпуском версии 11. В этом продукте еще более развита компонентная организация, улучшены возможности организации распределенных баз данных, внедрены более развитые средства интеграции с технологией Internet и т. д. Пока неизвестны какие-либо планы компании по переходу к объектно-реля- ционным архитектурам.

**8.1.1.4. Линия серверных продуктов CA-OpenIngres компании ComputerAssociates**

Проект и экспериментальный вариант СУБД Ingres были разработаны в университете Беркли под руководством одного из наиболее известных в мире ученых и специалистов в области баз данных Майкла Стоунбрейкера (MichaelStonebraker). С самого начала СУБД Ingres разрабатывалась как мобильная система, функционирующая в среде ОС UNIX. Первая версия Ingres была рассчитана на 16-разрядные компьютеры и работала главным образом на машинах серии PDP. Это была первая СУБД, распространяемая бесплатно для использования в университетах. Впоследствии группа Стоунбрейкера перенесла Ingres в среду ОС UNIXBSD, которая также была разработана в университете Беркли. Семейство СУБД Ingres из университета Беркли принято называть "университетской Ingres" (соответствующие программные продукты вместе с исходными текстами и документацией до сих пор доступны в секторе publicdomainInternet).

В начале 80-х была образована компания RTI (RelationalTechnologyInc.) для сведения университетских прототипов до уровня коммерческих продуктов. С этого момента стали различать университетскую и коммерческую СУБД Ingres. В настоящее время коммерческая Ingres поддерживается, развивается и продается компанией ComputerAssociates. Сейчас это одна из наиболее развитых коммерческих реляционных СУБД.

Мы коснемся главным образом особенностей базового серверного продукта компании ComputerAssociates линии CA-OpenIngres - CA-OpenIngres/Server. Сервер базируется на следующих пяти ключевых архитектурных принципах компании ComputerAssociates:

* использование многопоточной и мультипроцессорной организации сервера для систем оперативной обработки транзакций (OLTP - OnLineTransactionProcessing);
* применение более развитых методов обработки данных для систем уровня OLAP (OnLineAnalyticalProcessing);
* безопасное и надежное управление данными информационных приложений;
* обеспечение расширенных инструментальных средств администрирования серверной части системы (это вообще конек компании; у них больше всего внимания обращается на администрирование и управление систем);
* возможность гибкой настройки сервера в соответствии с конкретными потребностями корпорации.

Архитектура CA-OpenIngresServer поддерживает совместное использование многочисленных серверов баз данных на основе поддержки совместного буферного кэша, в котором хранятся данные, объекты, откомпилированные запросы, хранимые процедуры и информация, характеризующая состояние транзакций. Средства администрирования и управления позволяют выделить некоторые серверы для целей оперативного управления транзакциями, в то время как другие будут использоваться для генерации отчетов с более низким приоритетом.

В соответствии с начальным подходом Стоунбрейкера, в сервере поддерживается широкий набор допустимых способов хранения данных: куча (heap), B-деревья, таблицы хеширования и т.д. Допускается поддержание индексов с избыточными столбцами данных для эффективного выполнения особо критичных запросов. Применяются способы сжатия таблиц и индексов.

При оптимизации запросов учитываются разнообразные допустимые формы хранения. Используются истинные распределения данных за счет динамического построения гистограмм распределения значений. Для особо критических запросов поддерживаются административные способы оптимизации.

В соответствии с традициями Ingres в CA-OpenIngres поддерживается встроенная система правил (расширенный вариант более или менее известного механизма триггеров). Расширенные языковые возможности определения правил позволяют решать на основе этого механизма не только задачи поддержания целостности баз данных, но и полностью разрешать производственные задачи.

В CA-OpenIngres поддерживаются стандарты SQL-89 и ядерный уровень языка SQL-92. (Обратите внимание, что никто из ведущих производителей не гарантирует полной совместимости с SQL-92. Это очень плохо, поскольку уже на протяжении более чем 4 лет, ни одна компания не может или не хочет произвести продукт, полностью соответствующий требованиям хорошего международного стандарта. Правда, и стандарт несколько сложноват.)

Очень интересным направлением развития линии CA-OpenIngres явилось приобретение японской объектно-ориентированной СУБД Jasmin. Это оригинальная объектно-ориентированная система, модель данных которой основывается одновременно на идеях Smalltalk и Си++. Компания ComputerAssociates считает, что в принципе невозможно сочетать в одной системе объектно-ориентированный и реляционный подходы (в частности, отметим по-прежнему существующую проблему потери соответствия объектных и реляционных операций, присущую объектно-реляционным системам).

Поэтому в ближайших планах компании содержатся намерения одновременного поддержания объектно-ориентированного и реляционного интерфейсов доступа к базам данных, среда хранения которых будет поддерживаться OpenIngres. Сама же СУБД OpenIngres поддерживает возможности шлюзования для доступа к унаследованным системам баз данных (в частности, IDMS), что обеспечивает полную преемственность по отношению к ранее разработанным и накопленным хранилищам данных. К сожалению, в текстах, распространяемых компанией CA, содержится слишком мало технической информации относительно Jasmin. Однако достоверно известно, что объектно-ориентированные возможности управления данными широко используются в новом продукте компании, предназначенном для управления и администрирования глобально распределенных корпоративных приложений и называемом TNG (TheNextGenerationofUniCenter).

В июне 1997 г. компания CA объявила о выпуске новой версии OpenIngres 2.0. Кроме улучшенных возможностей репликации и наличия развитых средств интеграции с Internet, в OpenIngres появились следующие новые возможности:

1. использование блокировок на уровне записи с возможностью выбора вида блокировок для указанных таблиц или для указанных пользователей;
2. поддержка правил (триггеров) уровня оператора со срабатыванием правила для данного оператора один раз независимо от числа затрагиваемых им строк;
3. реализация всех четырех уровней изоляции транзакций, специфицированных в стандарте языка SQL (READCOMMITTED, REPEATABLEREAD, SERIALIZABLE и READUNCOMMITTED);
4. наличие средств параллельного копирования базы данных, хранимых на дисковых накопителях с разными контроллерами;
5. улучшенные средства управления пространственными данными, включая индексацию на основе R-деревьев;
6. возможность выбора размера страницы для хранения индивидуальной таблицы;
7. более строгая оптимизация на основе развитой статистической информации, позволяющей, в частности, оценить размер результата соединения таблиц.

Кроме того, в версии 2.0 улучшены средства визуального администрирования базы данных.

О конкретных работах, связанных с интеграцией с объектно-ориентированной СУБД Jasmine, пока не сообщается.

**8.1.1.5. Серверные продукты линии DB2 компании IBM**

Серьезные практические исследования в области систем управления реляционными базами данных компания IBM начала с проекта экспериментальной системы SystemR, которая разрабатывалась в исследовательской лаборатории фирмы IBM в 1975-1979 г.г. Эта работа оказала революционизирующее влияние на развитие теории и практики реляционных систем во всем мире. Именно SystemR практически доказала жизнеспособность реляционного подхода к управлению базами данных.

После успешного завершения работ по созданию этой системы и получения экспериментальных результатов ее использования был разработан целый ряд коммерчески доступных реляционных систем, в том числе и на основе непосредственного развития SystemR (возможности одной из коммерчески доступных реляционных систем - DB2 описываются в переведенной на русский язык книге К. Дейта "Руководство по реляционной СУБД DB2", хотя книга существенно отстала от практики; ее прекрасный перевод на русский язык вышел в свет в 1988 г.). Исключительно важен опыт, приобретенный при разработке этой системы. Практически во всех более поздних реляционных СУБД в той или иной степени используются методы, примененные в SystemR.

На наш взгляд, компания IBM много потеряла, ориентируя DB2 только на использование своих аппаратно-программных платформ. Первый вариант DB2 работал на IBM/370 в операционной среде OS/370. В связи с развитием аппаратуры и программного обеспечения мейнфреймов система была перенесена в операционную среду MVS. Программное обеспечение специализированного аппаратно-программного комплекса AS/400 также во многом основано на DB2. После разработки операционной системы OS/2 появился вариант DB2, пригодный для использования на персональных компьютерах. СУБД DB2 всегда представляла собой развитый программный продукт управления реляционными базами данных. В ней всегда присутствовали, в частности, такие возможности как управление транзакциями, журнализация изменений и восстановление согласованного состояния базы данных после сбоев программного обеспечения и/или аппаратуры. Заметим, что именно IBM выпустила первый корпоративный стандарт языка SQL.

Развитие системы и сферы ее применений ограничивало то, что отсутствовал мобильный текст системы. Ориентируясь на использование DB2 на своих аппаратных платформах, компания IBM исторически использовала для программирования DB2 смесь языка ассемблера и языка PL/1. Прорывом как для DB2, так и для IBM в целом стало появление Unix-ориентированной серии серверов и рабочих станций RS/6000. Именно при создании варианта системы DB2/6000 компания была вынуждена переписать систему на языке Си. После этого появилась очевидная возможность простого переноса СУБД на другие аппаратные платформы. В последнее время IBM объявила выпуск DB2 для аппаратно-программных платформ Sun и HP. По мнению автора курса, этот шаг означает появление на рынке независимых серверных продуктов управления реляционными базами данных очень серьезного и достойного конкурента. Коротко охарактеризуем основные возможности наиболее распространенной в настоящее время версии DB2 Version 2:

1. Возможности, соответствующие требованиям реляционной модели данных:

* Поддерживаются почти соответствующие полному стандарту SQL-92 средства обеспечения ссылочной целостности. Для таблицы-предка можно объявить правила удаления строк Restrict, NoAction, Cascade или SetNulls, в соответствии с которыми будут обрабатываться соответствующие строки таблицы-потомка. (При применении правила Restrict будет запрещено удаление строки таблицы-предка, если на эту строку ссылается хотя бы одна строка таблицы-потомка; задание правила Cascade приводит к автоматическому удалению ссылающихся строк таблицы-потомка; при указании правила SetNulls в поле ссылки ссылающихся строк автоматически устанавливаются неопределенные значения; правило NoAction, которое действует подобно правилу Restricted, но с другим диагностическим сообщением, устанавливается при определении ограничений ссылочной целостности по умолчанию).
* Возможно определение пользователем значений полей таблицы по умолчанию. Вообще-то эта возможность определена в SQL-92, и ее реализация не доставляет особого труда, но тем не менее в большинстве систем такая возможность отсутствует.
* Наконец-то реализовано средство, задаваемое фразой WITHCHECKOPTION при определении представления. При указании такого требования становится невозможным занести строку в представляемую таблицу или изменить существующую строку таким образом, чтобы полученная строка не была видима через представление.

2. Объекты базы данных:

* Помимо стандартных типов данных DB2 допускает хранение BLOBs размером до 2 Гб. Соответствующие поля предназначены для сохранения мультимедийной информации, структура которой известна только приложению.
* Поддерживается развитый механизм триггеров с возможностью указания степени гранулированности триггера, например, должно ли срабатывать заданное действие один раз при выполнении операции строки из заданной таблицы, или его следует выполнять для каждой удаляемой строки.
* Обеспечивается механизм хранимых процедур, которые программируются на смеси языков SQL и одного из процедурных языков третьего поколения.

3. Возможности запросов:

* Реализованная версия языка SQL является расширенным множеством ядерного уровня языка SQL-92 и включает ряд конструкций, наличие которых предполагается в SQL-3.
* Поддерживаются все разновидности соединений, определенные в SQL-92, но синтаксис соответствующих конструкций отличается от стандартного.

4. Поддержка доступа из Internet:

* Специальный шлюз, встроенный в сервер, обеспечивает доступ к данным DB2, транслируя команды HTML в операторы языка SQL.

IBM активно ведет работу по созданию собственного универсального сервера. С августа 1997 г. доступна бета-версия DB2 UniversalDatabase. Этот продукт относится к категории объектно-реляционных, и мы кратко обсудим его возможности позже.

**8.1.1.6. Серверные продукты управления базами данных компании Microsoft**

Первые работы компании Microsoft, относящиеся к области баз данных, проводились совместно с компанией Sybase, причем Microsoft поддерживала линию OS/2, а Sybase - UNIX. Так продолжалось до 1992 г., когда компания Microsoft приняла решение о переносе SQL-сервера на платформу WindowsNT. (Заметим, кстати, что недаром серверные продукты Sybase и Microsoft называются SybaseSQLServer и MicrosoftSQLServer соответственно; у этих продуктов общие корни.) Перенос SQL сервера в среду NT сопровождался существенными переделками ядра системы и в основном был выполнен специалистами Microsoft. Первая по-настоящему работоспособная версия сервера (MSSQLServer 4.21) вышла в свет в 1994 г. для использования в среде NT 3.5. Особое внимание обращалось на развитие средств администрирования, внедрение (в час- тности, и для собственного использования) механизма хранимых процедур и т.д. В 1995 г. была выпущена версия 6.0, в которой был внедрен ряд средств, обеспечивающих удобные взаимодействия с другими продуктами Microsoft. Наконец, в 1996 г. появилась наиболее современная версия 6.5.

Перечислим основные возможности MicrosoftSQLServer 6.5:

1. в ядре сервера может выполняться несколько нитей, что позволяет эффективно использовать симметричные мультипроцессорные компьютеры (SMP - SymmetricMultiprocessorProcessors);
2. поддерживается асинхронное выполнение обменов с несколькими дисковыми устройствами;
3. обеспечивается параллельное выполнение пользовательских транзакций, операций построения индексов и загрузки данных, а также резервного копирования;
4. при оптимизации запросов используется статистическая информация;
5. для синхронизации параллельного доступа применяется механизм блокировок на уровне записи с автоматическим распознаванием и разрешением тупиковых ситуаций;
6. поддерживается автоматическая репликация данных на основе журнальной информации или мгновенных снимков, включая доступность репликации в базы данных, доступные через интерфейс ODBC, и репликации "массивных" данных (типов TEXT и IMAGE);
7. обеспечивается двухфазная фиксация распределенных транзакций;
8. реализованы операции CUBE и ROLLUP для работы с многомерными данными в системах оперативной аналитической обработки (OnLineAnalyticalProcessing - OLAP);
9. доступны средства интеграции с системами электронной почты, в частности, с MicrosoftExchangeServer;
10. имеются средства, обеспечивающие доступ к SQL-серверу из Web-серверов и формирование результатов запросов в формате HTML (WebConnector и WebAssistant);
11. развиты возможности администрирования баз данных: SQLEnterpriseManager позволяет с одного терминала управлять произвольным числом SQL-серверов; дополнительные административные процедуры могут разрабатываться на VisualBasic со встроенным SQL;
12. реализован входной уровень стандарта SQL-92 и важные компоненты промежуточного уровня;
13. можно использовать национальные символы, в том числе символы русского языка.

Microsoft не объявляет о планах перехода к использованию объектно-реляционного подхода. Пока развитие SQL-сервера происходит в рамках все большего внедрения в этот продукт компонентов, основанных на общей объектной архитектуре COM (ComponentObjectArchitecture - компонентная объектная архитектура). Основываясь на спецификациях OLE (ObjectLinkingandEmbedding - связывание и встраивание объектов), которые обеспечивают объектное представление различных сервисов операционной системы, а также развитие OLE - OLEDB, компания Microsoft пытается обеспечить пользователям общую объектную среду компонентов (включая данные, поддерживаемые SQL-сервером, которые могут разнообразным образом комбинироваться).

**8.1.2. Насколько возможности серверов баз данных соответствуют потребностям приложений?**

Как мы отмечали в вводной части этого раздела, реляционные базы данных обладают достоинствами и недостатками. Для некоторых приложений перевешивают достоинства, для других - недостатки. Можно сказать, что современные серверы реляционных баз данных (такие как Infor- mixOnLine версии 6 и выше, Oracle 7.x, SybaseV.11) обладают качествами, близкими к предельно возможным при использовании традиционной технологии. Еще возможны совершенствование оптимизации запросов, применение более эффективных методов распараллеливания, реализация более высоких уровней стандарта SQL и т.д., но принципиально при этом ничего не изменится. Реляционные СУБД могут управлять очень большими базами данных; эффективно используют возможности симметричных и массивно параллельных компьютеров; позволяют хранить в таблицах текстовую и графическую информацию произвольного размера; поддерживают достаточно развитое подмножество языка SQL; обеспечивают возможности частичного переноса логики приложений на сторону сервера за счет использования ограничений целостности, триггеров и хранимых процедур.

Существует масса приложений, для которых возможностей современных серверов баз данных хватает с запасом. Это традиционные банковские приложения, системы резервирования билетов и мест в гостиницах, системы управления предприятиями, библиотечные информационные системы и т.д. Основными претензиями со стороны разработчиков таких систем к поставщикам СУБД является не то, что СУБД не может что-то сделать, а то, что она может сделать слишком много, в том числе вещи, абсолютно ненужные данному приложению.

Например, очевидно, что библиотечная информационная система должна быть в состоянии обслуживать одновременно многих пользователей. Но эти пользователи почти всегда только читают информацию, а изменения данных происходят сравнительно редко и не обязательно на фоне продолжающегося оперативного доступа к данным. Конечно, требуется развитый механизм эффективно реализуемых запросов. Тем не менее, используемая СУБД будет включать общие средства управления транзакциями со всеми разновидностями блокировок. И это общая ситуация. Чем большими возможностями обладает СУБД, тем большему числу приложений не потребуется хотя бы часть этих возможностей. При этом сервер остается монолитом, стоит одних и тех же денег и требует для своего использования практически одни и те же ресурсы.

Существует много приложений, для которых средства, предоставляемые традиционными реляционными серверами баз данных, недостаточны. Основной характеристикой таких приложений является потребность в обработке сложноструктурированных объектов, для представления которых средствами реляционных систем требуется использование многочисленных таблиц, над которыми при выполнении приложения приходится выполнять операции соединения. Классическими примерами подобных приложений являются системы автоматизированного проектирования, системы поддержки принятия решений и т.д. Известно, что операция соединения является наиболее трудоемкой в реляционных системах. Время выполнения операции и количество требуемых операций возрастают нелинейно при увеличении размера и числа соединяемых таблиц. Кроме того, и для приложений этой категории часто не требуются все возможности, поддерживаемые сервером баз данных.

Таким образом, мы видим, что несмотря на наличие высокопроизводительных серверов реляционных баз данных, обладающих развитыми возможностями в соответствии с реляционным подходом, в мире информационных приложений существует определенная неудовлетворенность. Компании, производящие СУБД, ощущают эту неудовлетворенность, и чтобы не потерять рынок в будущем, пытаются придать своим продуктам новые качества.

**8.1.3. Преобразование реляционного подхода к организации баз данных в объектно-реляционный подход**

Мы уже отмечали в начале этого раздела, что радикальным решением всех проблем, свойственных реляционным системам, был бы революционный переход к объектно-ориентирован- ной организации как баз данных, так и систем управления ими. Мы перечислили ряд причин, по которым массовых потребителей баз данных не устроила эта возможная революция. Но дело не только в этом. В конце 1989 г. группа ведущих специалистов в области объектно-ори- ентированных баз данных опубликовала "Манифест объектно-ориентированных баз данных" (перевод на русский язык опубликован в журнале "СУБД", N 4, 1995 г.).

В этом манифесте утверждается, что современная ситуация с ООБД напоминает ситуацию с реляционными системами середины 1970-х гг. При наличии большого количества экспериментальных проектов (и коммерческих систем) отсутствует общепринятая объектно-ориенти- рованная модель данных, и не потому, что нет ни одной разработанной полной модели, а по причине отсутствия общего согласия о принятии какой-либо модели. На самом деле, имеются и более конкретные проблемы, связанные с разработкой декларативных языков запросов, выполнением и оптимизацией запросов, формулированием и поддержанием ограничений целостности, синхронизацией доступа и управлением транзакциями и т.д.

Объектно-реляционный подход возник как эволюционная альтернатива революционному объектно-ориентированному подходу. Заметим, что термин "объектно-реляционные СУБД" вошел в обиход далеко не сразу, и даже сейчас системы соответствующего класса часто характеризуются другими терминами. Точкой отсчета можно считать опубликование в 1990 г. "Манифеста систем баз данных третьего поколения" (перевод на русский язык опубликован в журнале "СУБД", N 2, 1995 г.). В этом манифесте выдвигалась идея и обосновывались преимущества эволюционного развития возможностей СУБД без коренной ломки предыдущих подходов и с сохранением преемственности с системами предыдущего поколения. Частично требования к системам следующего поколения включали просто необходимость реализации давно известных свойств, которые отсутствовали в большинстве текущих реляционных СУБД (ограничения целостности общего вида, триггеры, модификация БД через представления и т.д.). В число новых требований входили полнота системы типов, поддерживаемых в СУБД; поддержка иерархии и наследования типов; возможность управления сложными объектами и т.д.

Другим термином, используемым по отношению к СУБД третьего поколения является термин "расширяемые СУБД". Под этим словосочетанием понимается то, что компания-производи- тель поставляет некоторый базовый остов системы, функции которой в дальнейшем могут на- ращиваться сторонними компаниями или даже пользователями.

Термин "объектно-реляционная СУБД" был внедрен в обиход Майклом Стоунбрейкером, идейным руководителем проектов свободно распространяемых систем Ingres и Postgres и основателем компании Illustra, выпустившей одноименный коммерческий продукт. (На самом деле, система Illustra является хорошо отлаженным и документированным вариантом Postgres). В Postgres были реализованы многие интересные средства, свойственные системам третьего поколения: поддерживалась темпоральная модель хранения и доступа к данным и в связи с этим был абсолютно пересмотрен механизм журнализации изменений, откатов транзакций и восстановления БД после сбоев; обеспечивался мощный механизм ограничений целостности; имелась возможность хранения в столбцах таблицы вложенных таблиц. Допускалось хранение в полях таблицы данных абстрактных, определяемых пользователями типов, что обеспечивало возможность внедрения в базу данных поведенческого аспекта.

Наконец, ведущие компании-производители СУБД стали использовать для идентификации своих серверных продуктов нового поколения термин "универсальный сервер", имея в виду, что в этих продуктах используется объектно-реляционный подход, а также обеспечиваются возможности расширяемости, интеграции с Internet и т.д.

**8.1.4. Первые серверные продукты, поддерживающие объектно-реляционный подход**

С конца 1996 г. три компании объявили о выпуске серверных продуктов, соответствующих названию "универсальный сервер": Oracle с продуктом Oracle 8, Informix с продуктом InformixUniversalServer и IBM с продуктом DB2 UniversalDatabase. Далее мы приведем краткие характеристики этих систем.

**8.1.4.1. Oracle 8**

Компания Oracle заявляла, что Oracle 8 будет уметь делать с объектами все, что умеет делать InformixUniversalServer, и даже больше. Однако это не так в первом выпуске Oracle 8 (этот выпуск был официально объявлен 24 июня 1997 г.). В фокусе Oracle 8 находятся расширенная система типов и поддержка бизнес-объектов; появление других возможностей расширяемости ожидается в версии 8.1. Oracle концентрируется на реализации своей сетевой вычислительной архитектуры (NCA - NetworkComputingArchitecture), и в Oracle 8 вносятся улучшенные возможности производительности, масштабируемости, доступности, репликации, разделения дан- ных и т.д.

NCA - это трехзвенная архитектурная структура, основанная на CORBA (CommonObjectRequestBrokerArchitecture - Общая архитектура брокера объектных заявок). В NCA используются расширяющие компоненты, называемые "картриджами", которые могут разрабатываться Oracle или сторонними поставщиками. Впервые использование картриджей приложений и картриджей баз данных потребовалось в OracleWebApplicationServer для организации связи на основе CORBA. В контексте расширяемой среды данных Oracle обеспечивает компоненты на уровне промежуточного программного обеспечения приложений (например, компонент управления транзакциями в WebApplicationServer) и на уровне универсального сервера (Oracle 8). На объектном уровне Oracle 8 поддерживает объектные представления данных с использованием новых объектных типов и существующих реляционных данных. Кроме того, Oracle 8 поддерживает объектный кэш на стороне клиентов и навигацию между объектами по ссылкам. Транслятор объектных типов отображает объекты базы данных в соответствующие конструкции языка Си.

Далее приводится сводка свойств, имеющихся в Oracle 8 и ожидаемых в версии 8.1:

* Расширяемая система типов. Поддерживаются объектные типы, типы коллекций (массивы переменного размера и вложенные таблицы) и ссылочные типы. Объектный тип может применяться либо к столбцу, либо к строке и может быть семантически эквивалентен именованному строчному типу SQL-3. Кроме того, Oracle 8 явно связывает с объектными типами методы. Пока поддерживается только один уровень вложенности таблиц (в 8.0) и не реализована полная инкапсуляция определяемых пользователями типов данных. В версии 8.1 будет поддерживаться одиночное наследование. Не предполагается возможность репликации расширенных данных.
* Определяемые пользователями функции. Скалярные функции, перегрузка и разрешение имени на основе типов параметров, параллельное выполнение функций с определенными пользователями агрегатами (функции столбцов) находятся в стадии обсуждения.
* Расширяемая система индексации. Поддержка определяемых пользователями индексных структур и индексов на выражениях появится вместе с компонентом DatabaseExtensibilityServices.
* Расширяемый оптимизатор. Возможность указывать стоимость выполнения определенных пользователями функций и обеспечивать статистику об использовании определенных пользователями данных ожидается с появлением компонента DatabaseExtensibilityServices. К расширенным типам пока не применяются параллельные операции.
* Большие объекты и внешние данные. LOB могут храниться внутри базы данных или во внешних файлах. Oracle 8 не поддерживает доступа по записи к внешним данным и не гарантирует их целостность. Большие объекты могут быть реплицированы, но таблицы с LOB реплицировать нельзя.
* Расширяемая языковая поддержка. Определяемые пользователями типы данных и хранимые процедуры пишутся на PL/SQL или Си/Си++, причем подпрограммы на Си/Си++ выполняются вне адресного пространства сервера. Поддержка Java в сервере будет обеспечиваться в DatabaseExtensibilityServices (версия 8.1).
* Предопределенные расширения. Сервер Oracle 8 поддерживает хранение текстовых, пространственных и видео данных; ожидается появление нового типа данных для хранения графической информации. Компания работает также с несколькими партнерами для тестирования и формализации API для DatabaseExtensibilityServices и разработки картриджей данных.

Как видно, в Oracle 8 компания главным образом сосредоточилась на реализации собственных расширений, которых может оказаться достаточно для начала моделирования бизнес-объектов. Возможности реализации расширений сторонними компаниями менее развиты, чем в продуктах Informix и IBM. Это означает, что пользователи Oracle будут вынуждены более долго ожидать наличия полных наборов строительных блоков для построения новых или улучшения существующих приложений. С другой стороны, развитие продуктов Oracle происходит на основе единой архитектуры NCA, которая закладывает базу расширяемости.

**8.1.4.2. InformixUniversalServer**

Компания InformixSoftware приобрела реальные шансы стать лидером сектора объектно-реляционных систем после приобретения компании Illustra в начале 1996 г. (Заметим, что сотрудниками компании стали все основные разработчики СУБД Illustra, включая М.Стоунбрейкера.) Многие авторитетные специалисты компьютерной индустрии скептически относились к решению компании произвести слияние продуктов Informix-OnLine 7.2 и IllustraServer к концу того же года. Однако компания Informix действительно смогла выпустить в конце декабря устойчивую версию InformixUniversalServer, работающую на трех платформах. К лету 1997 г. UniversalServer был доступен на 10 платформах.

Компания сконцентрировалась прежде всего на общей архитектуре универсального сервера. Внимание обращается на средства, обеспечивающие интеграцию с технологией Internet/Web (с применением продукта WebConnect). Затрагиваются вопросы поддержки сложных данных в средствах разработки и приложениях с использованием недавно объявленного продукта DataDirector (приобретенного вместе с компанией CenterViewSoftware в начале этого года). Этот продукт обеспечивает возможность использования развитых сред разработки (Java, PowerBuilder, VisualBasic и др.), доступ к расширениям в духе SQL-3 и, кроме того, доступ к новым типам данных и функциям UniversalServer.

В своем первом выпуске UniversalServer объединяет средства сервера InformixOnLine 7.2 с расширениями, обеспечиваемыми механизмом DataBlade сервера, и языковыми расширениями, основанными на идеях SQL-3. Новые возможности включают следующее:

* Расширяемую систему типов для интеграции определяемых пользователями типов данных, уточненных типов, абстрактных типов данных, строчных типов и типов коллекций, включая возможность неограниченной вложенности таблиц. Поддерживается простое наследование в иерархии именованных строчных типов. Каждая таблица, основанная на строчном типе, становится типизированной таблицей и может использоваться в этой иерархии. В будущем ожидается появление ссылочных типов и абстрактных типов данных в стиле SQL-3, а также средств репликации данных определяемых пользователями типов.
* Определяемые пользователями функции - полный диапазон типов функций с поддержкой перегрузки, распознавания нужного тела функции на основе типов параметров и параллельного выполнения функций, когда это оправдано. (Параллельные возможности UniversalServer применимы на симметричных мультипроцессорных платформах; при использовании расширенной параллельной версии InformixOnLine эти возможности распространяются на кластерные архитектуры; на массивно-параллельных платформах объектные расширения пока недоступны.)
* Расширяемую систему индексирования - R-деревья, применение индексов на основе B-деревьев к определяемым пользователями типам данных, определяемые пользователями индексные структуры.
* Расширяемый оптимизатор - таблично-управляемый оптимизатор, позволяющий интегрировать определяемые пользователями типы данных, новые индексные структуры и новые методы доступа. Разработчик может указать стоимость выполнения функций определяемого им типа и зарегистрировать новый тип индекса, что включает определение функций, которые используют индекс, интерфейсных подпрограмм для доступа к индексу и стоимости использования индекса. Индексы хранятся под управлением СУБД с соответствующим транзакционным управлением; сторонние поставщики должны сами следить за целостностью внешних индексов. Кроме того, оптимизатор применяет параллельные операции к разделенным данным и индексам.
* Возможность хранения больших объектов (LOB - LargeObjects) внутри базы данных или во внешних файлах. Пока UniversalServer не гарантирует целостность данных, хранимых во внешних файлах, но Informix имеет на этот счет планы. Интерфейс виртуальной таблицы позволяет регистрировать внешние данные в каталогах базы данных и обращаться к ним таким же образом как если бы они были таблицами, непосредственно поддерживаемыми универсальным сервером.
* Наличие предопределенных расширений: Informix подрядил много сторонних поставщиков для написания DataBlades - упакованных наборов расширений, относящихся к некоторым прикладным областям (текстовому поиску, управлению графической информацией, финансовому анализу и т.д.). DataBlade интегрируется с ядром IUS на основе специального интерфейса уровня вызовов (DataBladeAPI). К июню 1997 г. планировалось иметь в поставке 20 DataBlade, еще 10 находятся в стадии бета-тестирования, к концу года ожидается иметь около 50 готовых к поставке DataBlade. В настоящее время имеется два режима выполнения DataBlade - в адресном пространстве ядра (это дает наибольшую эффективность) и в отдельном виртуальном процессоре (более безопасный режим).
* Расширенный набор языков приложений - в дополнение к поддержке разработки пользовательских типов данных на основе собственной среды NewEraInformix теперь обеспечивает новое средство разработки клиентских приложений DataDirector, которое дает возможность связи между данными, контролируемыми приложением, и данными, хранимыми в среде UniversalServer. В этом средстве используются собственные интерфейсы с универсальным сервером (JavaAPI и Си++ API), а не ODBC. Кроме того, DataDirector поддерживает языковые расширения в духе SQL-3.

Одной из проблем компании Informix, связанной с форсированным внедрением объектно-реляционных средств, является сохранение позиции на бизнес-рынке, которую компании удалось занять на основе InformixOnLine 7.x. До слияния с Illustra основное внимание уделялось производительности и масштабируемости на основе разных видов параллелизма, и именно это определяло лицо Informix на рынке. С появлением UniversalServer эти важные качества продуктов отошли на второй план. Кроме того, компания должна принимать во внимание наличие трех разных серверов баз данных - UniversalServer, OnLine и ExtendedParallelServer.

Однако компании первой удалось продемонстрировать возможность эволюционного перехода к объектно-реляционной технологии с сохранением надежного и высокоэффективного управления базами данных. UniversalServer является развитием OnLine, и именно так следует представлять его на рынке.

**8.1.4.3. IBMDB2 UniversalDatabase**

Компания IBM ведет исследования в области инфраструктуры расширяемых реляционных баз данных в течение более чем десяти лет. Компания была первой из основных поставщиков, выпустившим на рынок реляционный сервер баз данных с объектными расширениями: DB2 CommonServer 2 (в июле 1995 г.). Теперь IBM готова начать поставки обновленного продукта с новым названием - DB2 UniversalDatabase, представляющего собой слияние начальных объектно-реляционных свойств DB2 CommonServer 2.1 с возможностями параллельной обработки DB2 ParallelEdition 1.2 и доступного на разнообразных мультипроцессорных платформах (симметричных, кластерных и массивно-параллельных). Ключевым моментом является то, что продукты IBM основаны на едином исходном тексте сервера. Это обеспечивает основу новых разработок, тем более что расширения с самого начала базируются на полностью параллельных средах.

Стратегия IBM в области объектно-реляционных баз данных включает четыре основных компонента. Во-первых, универсальная база данных DB2 (UniversalDatabase) находится в стадии бета-тестирования и должна начать поставляться в третьем квартале 1997 г. Во-вторых, разновидность определяемых пользователями типов данных - сильные связи с файлами (robustfilelinks) позволяет DB2 UniversalDatabase активно управлять внешне хранимыми данными с соблюдением требований безопасности и целостности. Реализация этой возможности ожидается к концу 1997 г. Третий компонент стратегии компании - DataJoiner - промежуточное программное обеспечение для доступа к неоднородным базам данных. Этот компонент включает все функциональные возможности сервера DB2, глобальный оптимизатор с расширяемыми знаниями о поддерживаемых источниках данных, возможность компенсировать функциональные различия между этими источниками. В планы компании входит расширение возможностей DataJoiner для работы с объектно-реляционными расширениями. Наконец, IBM разрабатывает компонент объектного слоя, называемый ClientObjectSupport, который обеспечит единое логическое представление всех доступных данных, а также их транзакционную согласованность, управление кэшами клиентов и интеграцию с объектно-ориентированными языками.

Вот сводка расширенных возможностей DB2 UniversalDatabase и средств, которые планируется включить в будущие выпуски:

* Расширяемая система типов - уточненные типы, абстрактные типы данных и объекты OLE с поддержкой в будущем абстрактных типов данных в стиле SQL-3, строчных типов, типов коллекций, ссылок, множественного наследования и репликации данных.
* Определяемые пользователями функции - скалярные функции с поддержкой перегрузки функций и нахождения нужного тела функции на основе типов параметров, параллельное выполнение. Ожидается появление табличных функций. Функции могут выполняться внутри адресного пространства сервера для повышения производительности или вне него для обеспечения безопасности.
* Расширяемая система индексации. IBM обеспечивает свои собственные индексы специального назначения для типов данных, поддерживаемых расширителями DB2 (DB2 RelationalExtenders). В будущих версиях будут поддерживаться определяемые пользователями индексные структуры, навигационный доступ через ссылки и возможность строить индексы на выражениях, на результатах вызовов функций и на атрибутах абстрактных типов данных.
* Расширяемый оптимизатор. DB2 уже включает основанный на правилах оптимизатор с возможностью выполнять глобальную оптимизацию и преобразования запросов. Планируется расширить интерфейс оптимизатора с тем, чтобы можно было добавлять новые правила. Разработчик может указать стоимость определенной пользователем функции, включая информацию о том, производит ли функция внешние действия. При интеграции с DB2 ParallelEdition оптимизатор также применяет параллельные операции для разделенных данных и индексов.
* Большие объекты и внешние данные. LOB может храниться внутри базы данных или во внешних файлах. На сегодняшний день DB2 обеспечивает доступ к данным, хранимым вне сервера; в будущих выпусках (к концу 1997 г.) будет гарантироваться и целостность на основе механизма сильных связей с файлами.
* Расширяемая языковая поддержка. Определяемые пользователем функции могут быть написаны на языках Си, Си++, VisualBasic или Java; хранимые процедуры могут писаться на языках третьего и четвертого поколения и Java. Ожидается появление языковых расширений в стиле SQL-3 для обеих целей.
* Предопределенные расширения. IBM обеспечивает реляционные расширители для текста, графической информации, видео, аудио и т.д. Компания образует партнерские отношения со сторонними поставщиками для разработки пакетов расширений. Имеется соответствующий набор инструментальных средств.

Текущее состояние продуктов и стратегические планы IBM могут позволить ей занять лидирующие позиции на рынке объектно-реляционных систем. Для этого прежде всего требуется добиться признания и высокого уровня продаж в секторе платформ других компаний (в настоящее время DB2 UniversalDatabase готовится к выпуску на всех платформах IBM, а также HewlettPackard и SunMicrosystems).

**8.1.5. Решения ведущих производителей серверов баз данных для их интеграции с технологией Internet/Intranet**

Широкое внедрение в практику технологий Internet и, в особенности, Web-технологии существенно повлияло на общую организацию современных информационных систем. Простой и легко осваиваемый универсальный интерфейс Web-браузеров в ряде случаев позволяет не задумываться о программном обеспечении клиентских рабочих мест. Особенности организации Web-серверов позволяют достаточно естественно расширять их возможности, перенося на сторону сервера логику приложений.

Естественной потребностью информационных систем является использование баз данных. Присущие Web-технологии возможности гипермедийной организации документов, естественно, нужны и полезны, но недостаточны. Конечно, новые требования заказчиков не могли не учесть производители СУБД. Сегодня каждая компания ведущей шестерки предлагает собственные решения для интеграции технологии баз данных и Web-технологии. Имеется масса соответствующих продуктов, производимых другими софтверными компаниями, но мы ограничимся кратким обзором средств интеграции, предлагаемых ведущей шестеркой. Заметим, что иногда средства интеграции включаются в состав сервера баз данных, а иногда предлагаются в виде отдельных необязательных компонентов.

**8.1.5.1. Решения компании Oracle**

Компания Oracle предлагает семейство продуктов WebSystem, предназначенных для облегчения разработки приложений, в которых используются средства Web и базы данных. Семейство включает следующие продукты: OracleWebServer, OracleWebServerOption и OraclePowerBrowser. Продукты основаны на стандартах Web-технологии (HTML, HTTP, SHTTP - SecureHypertextTransferProtocol), сетевых стандартах (TCP/IP, ISDN и т.д.) и стандартах объектных технологий (OLE, CORBA, OpenDoc).

WebServer объединяет сервер баз данных, сервер HTTP и OracleWebAgent. Для поддержки мультимедийной информации WebServer имеет интерфейсы с OracleTextServer и OracleMediaServer. В совокупности эти продукты создают полную среду мультимедиа для построения корпоративных Web-приложений.

Ключевой частью OracleWebServer является WebAgent, который дает возможность Web-клиентам вызывать хранимые процедуры и создавать "на лету" динамические HTML-страницы. Фактически, WebAgent представляет собой шлюз, подключаемый к Web-серверу через CGI (CommonGatewayInterface) и обеспечивающий расширяемые возможности формирования страниц HTML.

Компания уделяет большое внимание безопасности, поддерживая в WebServer развитый механизм аутентификации. В будущем будут поддерживаться все расширения HTML, относящиеся к безопасности.

WebServerOption содержит только сервер HTTP и WebAgent и предназначен для тех заказчиков, у которых уже имеется сервер баз данных и которые желают пользоваться Web-техно- логией.

Наконец, OraclePowerBrowser представляет собой среду разработки Web-приложений на рабочих станциях. Пакет включает развитый браузер, реляционную СУБД Blaze, средства разработки HTML-страниц, локальный Web-сервер и другие средства. Доступны версии PowerBrowser для платформ Microsoft, Macintosh, OS/2 и Motif/UNIX.

**8.1.5.2. Решения компании Informix**

Компания Informix предлагает продукт UniversalWebConnect, обеспечивающий возможность разработки Web-приложений, которые могут иметь доступ к любой информации, хранимой в базах данных Informix. Для разработки приложений на стороне клиента или сервера промежуточного уровня могут использоваться различные языки: Си, Си++, Java, ActiveX и т.д. Кроме того, WebConnect позволяет обеспечивать широковещательную рассылку информации подписавшимся на нее пользователям.

Для построения интеллектуальных Web-приложений WebConnect обеспечивает следующие сервисы:

* URL-интерфейс (UniversalResourceLocator). Этот интерфейс дает разработчикам Web-приложений единую точку доступа к логике приложения и информации, содержащейся в базе данных. Такая информация может включать динамически конструируемые или статические HTML-страницы и соответствующие мультимедийные данные. Для быстрой и персонализированной доставки пользователям этой информации WebConnect передает URL серверу баз данных.
* Страницы приложений (AppPages). AppPage - это HTML-страница со встроенными операторами SQL, на основании которой конструируется документ, отображаемый браузером. AppPage может включать скрипты JavaScript и апплеты Java.
* Программный интерфейс. Набор соответствующих средств позволяет просто переносить приложения "клиент-сервер" в среду Web или разрабатывать новые такие приложения. С помощью этого интерфейса обеспечивается стыковка с объектными брокерами (в частности, основанными на CORBA).
* Подписка и оповещение. Этот сервис дает возможность посетителям Web-узла подписаться на конкретную информацию, в частности, пользователи смогут получать электронные сообщения об изменениях базы данных.
* Безопасность. Обеспечивается возможность авторизации и аутентификации на уровне сервера баз данных. Это дает возможность распространить систему безопасности баз данных на Web-сервер.

**8.1.5.3. Решения компании Sybase**

Компания Sybase и ее дочерняя компания Powersoft поставляют несколько продуктов, предназначенных для производства Web-приложений, которые имеют доступ к базам данных. Кратко охарактеризуем некоторые из них. (Заметим, что в силу общей ориентации Sybase на компонентную архитектуру, в большинстве случаев эти продукты представляют собой компоненты, которые могут встраиваться в разные продукты.)

NetImpactDynamo - компонент, содержащий средства для построения сложных, динамических, связанных с базами данных Web-узлов. В настоящее время этот компонент поставляется в составе продукта SybaseSQLAnywhereProfessional.

jConnect - полная реализация на языке Java стандарта JDBC. Продукт дает возможность непосредственного доступа к базам данных (без использования промежуточного Web-сервера) из Java-приложений.

PowerSite - полная среда разработки в среде Web. Продукт обеспечивает разнообразные средства разработки, управления и тестирования динамических управляемых данными Web-приложений. PowerSite состоит из редактора и браузера HTML, персонального Web-сервера и средств поддержки активных языков сценариев.

Продукт Sybaseweb.sql дает возможность встраивать операторы SQL и скрипты, написанные на языке Perl, в HTML-страницы, что позволяет создавать персонализированные и кастомизированные страницы.

**8.1.5.4. Решения компании IBM**

Для интеграции технологии баз данных и Web-технологии компания IBM предлагает продукт DB2 WorldWideWebConnection. Этот продукт представляет собой шлюз с Internet и предназначен для работы с семейством серверов реляционных баз данных DB2 и семейством Web-серверов и защитных экранов (firewall). Кроме того, DB2 WWWConnection работает с продуктами IBM категории промежуточного программного обеспечения, что дает возможность Web-доступа к другим источникам данных.

Web-приложения для DB2, построенные с помощью DB2 WWWConnection, позволяют пользователю использовать для доступа к базам данных из стандартного браузера без потребности в каких бы то ни было изменениях в существующих структурах данных. На основе стандартных языков HTML и SQL строятся приложения, которые взаимодействуют с DB2 с помощью обычных операторов SQL.

Возможно построение двух- или трехзвенных архитектур "клиент-сервер". При двухзвенной организации используются клиентские рабочие места с Web-браузерами и Web-сервер, на котором и производится доступ к DB2. В трехзвенной среде Web-сервер обращается к другим серверам баз данных. Связующие возможности обеспечиваются средствами DistributedDatabaseConnectionServices (DDCS), ClientApplicationEnabler (CAE) и DistributedRelationalDatabaseArchitectureApplicationRequester (DRDAAR).

В трехзвенной организации допускается использование продукта DataJoiner, при применении которого обеспечивается возможность Web-доступа к базам данных Oracle, Sybase, MicrosoftSQLServer и другим реляционным и нереляционным источникам данных. Для повышения уровня безопасности приложений используется криптография.

В настоящее время продукт DB2 WWWConnection доступен на платформах OS/2, AIX, OS/400, MVS/ESA, WindowsNT, Solaris, HP/UX.

**8.1.5.5. Решения компании ComputerAssociates**

Компания ComputerAssociatesInternational включила средства интеграции с Web-технологией в новый выпуск OpenIngres 2.0. В дополнение к существовавшей в предыдущих версиях системы возможности устанавливать связь между Web-сервером и сервером баз данных на основе CGI, в выпуске 2.0 реализованы встроенные интерфейсные средства для связи с Web-серверами компаний Microsoft, Netscape и Spyglass. Новый макропроцессор позволяет встраивать операторы SQL прямо в HTML-страницы. Во время работы Web-сервер обращается с такими операторами к OpenIngres.

**8.1.5.6. Решения компании Microsoft**

Компания Microsoft поставляет целый ряд продуктов, предназначенных для разработки Web-приложений, работающих с базами данных. Большая часть этих продуктов входит в состав крупных серверных средств.

В состав MicrosoftSQLServer входит средство SQLServerWebAssistant. Основное назначение этого средства - формировать динамическую HTML-страницу на основе результата SQL-оператора выборки из базы данных.

Внутренним компонентом MicrosoftInternetInformationServer является InternetDatabaseConnector (IDC). Фактически, этот компонент является встроенным шлюзом с SQL-сервером. Получая от браузера HTML-страницу (например, заполненную форму), IDC обращается к SQL-серверу с соответствующим запросом. После получения результатов IDC формирует возвращаемую пользователю HTML-страницу. Другим ключевым компонентом InformationServer является ActiveServerPages (ASP). Это средство позволяет встраивать в HTML-страницы скрипты, написанные на языках VisualBasicScriptingLanguage и Jscript, которые могут производить доступ к ресурсам (приложениям, базам данных и т.д.), расположенным на локальном InformationServer или на других серверах.

### 8.2. Склады данных и системы оперативной аналитической обработки

В этом разделе мы рассмотрим вопросы организации специального класса информационных приложений, ориентированных не на оперативную обработку транзакций (onLineTransactionProcessing - OLTP), а на оперативную аналитическую обработку (OnLineAnalyticalProcessing - OLAP). Значимость аналитических систем непрерывно возрастает. Любая серьезная компания независимо от вида ее бизнеса нуждается не только в непрерывной оперативной транзакционной поддержке, но и в средствах анализа и прогнозирования как своей собственной деятельности, так и деятельности своих поставщиков, потребителей, партнеров и конкурентов.

**8.2.1. Чем отличаются системы оперативной обработки транзакций и системы оперативной аналитической обработки?**

У этих двух разновидностей систем принципиально разные задачи. Корпоративные информационные OLTP-системы создаются для того, чтобы способствовать повседневной деятельности корпорации, и опираются на актуальные для текущего момента данные. OLAP-системы служат для анализа деятельности корпорации или ее компонентов и прогнозирования будущего состояния. Для этого требуется использовать многочисленные накопленные данные о деятельности корпорации в прошлом, а также внешние источники данных, формирующие контекст, в котором работала корпорация.

Система оперативной аналитической обработки данных отличается от статической системы поддержки принятия решений (DecisionSupportSystem - DSS) тем, что OLAP-система позволяет аналитику динамически формировать класс вопросов, который требуется для решаемой им текущей аналитической задачи. DSS обеспечивает выдачу отчетов в соответствии с заранее сформулированными правилами. Для удовлетворения нового запроса нужно формально его описать, запрограммировать и только потом выполнить.

Тематика OLAP-систем очень широка и специальна. Мы не будем обсуждать соответствующие вопросы на глубоком уровне, а в основном (и тоже не очень глубоко) сосредоточимся на проблемах обеспечения OLAP-системы данными. Мы будем говорить о складах данных (Datawarehouse).

**8.2.2. Что вызвало появление понятия склада данных?**

Любая крупная и давно существующая корпорация обладает несколькими базами данных, относящимися к разным видам деятельности. Данные могут иметь разные представления, а иногда могут быть даже несогласованными (например, из-за ошибки ввода в одну из баз данных). Это нехорошо даже для OLTP-систем (в частности, с этой проблемой связаны потребности в интеграции корпоративных информационных OLTP-систем) и в принципе непригодно для OLAP-систем, которые должны обрабатывать общие исторические согласованные корпоративные данные. Более того, для оперативной аналитической обработки требуется привлечение внешних источников данных, которые тем более могут обладать разными форматами и требовать согласования. Видимо, на подобных рассуждениях и возникла концепция склада данных как предметно-ориентированного, интегрированного, неизменчивого, поддерживающего хронологию набора данных, организованного для целей поддержки управления.

Заметим, что подход построения склада данных для интеграции неоднородных источников данных принципиально отличается от подхода динамической интеграции разнородных баз данных. В случае склада данных реально строится новое крупномасштабное хранилище, управление данными в котором происходит, вообще говоря, по другим правилам, нежели в исходных оперативных базах данных.

Итак, в основе концепции склада данных лежат две основные идеи:

1. Интеграция разъединенных детализированных данных (детализированных в том смысле, что описывают некоторые конкретные факты, свойства, события и т.д.) в едином хранилище. В процессе интеграции должно выполняться согласование рассогласованных детализированных данных и, возможно, их агрегация. Данные могут поступать из исторических архивов корпорации, оперативных баз данных, внешних источников.
2. Разделение наборов данных, используемых для оперативной обработки, и наборов данных, применяемых для решения задач анализа.

**8.2.3. Необходимые свойства склада данных**

Обычно выделяют следующие основные свойства, которыми должен обладать склад данных:

* неоднородность программной среды;
* распределенный характер организации;
* повышенные требования к безопасности данных;
* необходимость наличия многоуровневых справочников метаданных;
* потребность в эффективном хранении и обработке очень больших объемов информации.

Склад данных практически никогда не создается на пустом месте. Почти всегда конечное решение будет разнородным, т.е. в нем будут использоваться автономно разработанные программные средства. Прежде всего это касается формирования интегрированного согласованного набора данных, которые могут поступать из разнородных баз данных, электронных архивов, публичных и коммерческих электронных каталогов, справочников, статистических сборников. При построении склада данных приходится решать задачу построения единой, согласованно функционирующей информационной системы на основе неоднородных программных средств и решений. При выборе средств реализации склада данных приходится учитывать множество факторов, включающих уровень совместимости различных программных компонентов, легкость их освоения и использования, эффективность функционирования и т.д.

В концепции склада данных предопределено то, что операционная аналитическая обработка может выполняться в любом узле сети независимо от места расположения основного хранилища. Хотя при аналитической обработке данные только читаются, и потребность в синхронизации отсутствует, для достижения эффективности необходимо поддерживать репликацию данных в разных узлах сети. (На самом деле, все не так просто. Одним из требований к складам данных является то, чтобы свежая информация поступала на склад как можно быстрее. Т.е. потенциально любая модификация оперативной базы данных может инициировать добавление данных к складу данных, а тогда потребуется обновить и все реплики, для чего синхронизация все-таки нужна).

Собранная вместе согласованная информация об истории развития корпорации, ее успехах и неудачах, о взаимоотношениях с поставщиками и заказчиками, об истории и состоянии рынка дает возможность анализа прошлой и текущей деятельности корпорации и построения прогнозов для будущего. Эта информация настолько ценна для корпорации, что нельзя допустить возможности ее утечки (на самом деле, если склад данных одной корпорации попадет в руки аналитиков другой корпорации, то все аналитические прогнозы первой корпорации сразу станут неверными). В системах, основанных на складах данных, оказывается недостаточной защита данных в стиле языка SQL, которую обеспечивают обычные коммерческие СУБД (этот уровень защиты соответствует классу C2 в соответствии с классификацией Оранжевой Книги Министерства обороны США). Для обеспечения должного уровня защиты доступ к данным должен контролироваться не только на уровне таблиц и их столбцов, но и на уровне отдельных строк (это уже соответствует классу B1 Оранжевой Книги). Приходится также решать вопросы аутентификации пользователей, защиты данных при их перемещении в склад данных из оперативных баз данных и внешних источников, защиты данных при их передаче по сети.

Если роль метаданных (обычно содержащихся в таблицах-каталогах) в оперативных информационных системах достаточно ограничена, то для OLAP-систем наличие развитых метаданных и средств их предоставления конечным пользователям является одним из основных условий успешной реализации. Например, прежде, чем менеджер корпорации задаст системе свой вопрос, он должен понять, какая информация имеется, насколько она актуальна, можно ли ей доверять, сколько времени может занять формирование ответа и т.д. Для пользователя OLAP-системы требуются метаданные, по крайней мере, следующих типов:

* Описания структур данных, их взаимосвязей.
* Информация о хранимых на складе данных и поддерживаемых им агрегатах данных.
* Информация об источниках данных и о степени их достоверности. Одна и та же информация могла попасть в склад данных из разных источников. Пользователь должен иметь возможность узнать, какой источник был выбран основным, и каким образом производились согласование и очистка данных.
* Информация о периодичности обновлений данных. Желательно знать не только то, какому моменту времени соответствуют интересующие его данные, но и когда они в следующий раз будут обновлены.
* Информация о владельцах данных. Пользователю OLAP-системы может оказаться полезной информация о наличии в системе данных, к которым он не имеет доступа, о владельцах этих данных и о действиях, которые он должен предпринять, чтобы получить доступ к данным.
* Статистические оценки времени выполнения запросов. До выполнения запроса полезно иметь хотя бы приблизительную оценку времени, которое потребуется для получения ответа, и объема этого ответа.

Уже сейчас известны примеры складов данных, содержащих терабайты информации. По данным консалтинговой компании MetaGroup, около половины корпораций, использующих или планирующих использовать склады данных предполагает довести их объем до сотен гигабайт. Проблемой таких больших хранилищ является то, что накладные расходы на внешнюю память возрастают нелинейно при возрастании объема хранилища. Исследования, проведенные на основе тестового набора TPC-D, показали, что для баз данных объемом в 100 гигабайт потребуется внешняя память объемом в 4.87 раза большая, чем нужно собственно для полезных данных. При дальнейшем росте баз данных этот коэффициент увеличивается.

Последнее, на чем мы остановимся в этом разделе, - это рынки данных (DataMart; кстати ведущий специалист Московского отделения компании Informix Ховард Залкин предпочитает называть их "лавками данных"). Рынок данных по своему исходному определению - это набор тематически связанных баз данных, которые содержат информацию, относящуюся к отдельным аспектам деятельности корпорации. По сути дела, рынок данных - это облегченный вариант склада данных, содержащий только тематически объединенные данные. Целевая база данных максимально приближена к конечному пользователю и может содержать тематически ориентированные агрегатные данные. Рынок данных, естественно, существенно меньше по объему, чем корпоративный склад данных, и для его реализации не требуется особо мощная вычислительная техника.

**8.2.4. Насколько склады данных могут поддерживаться существующими серверами реляционных баз данных?**

В последнее время все более популярной становится идея совместить концепции склада и рынка данных в одной реализации и использовать склад данных в качестве единственного источника интегрированных данных для всех рынков данных. Тогда естественной становится такая трехуровневая организация OLAP-системы:

* На первом уровне реализуется корпоративный склад данных на основе одной из развитых современных реляционных СУБД. Это хранилище интегрированных в основном детализированных данных. Реляционные СУБД обеспечивают эффективное хранение и управление данными очень большого объема, но не слишком хорошо соответствуют потребностям OLAP-систем, в частности, в связи с требованием многомерного представления данных.
* На втором уровне поддерживаются рынки данных на основе многомерной системы управления базами данных (примером такой системы является OracleExpressServer). Мы не будем рассматривать здесь особенности организации многомерных СУБД (это отдельная большая тема), но заметим, что такие СУБД почти идеально подходят для целей разработки OLAP-систем, но пока не позволяют хранить сверхбольшие объемы данных (предельный размер многомерной базы данных составляет 10-20 гигабайт). В данном случае это и не требуется, поскольку речь идет о рынках данных. Заметим, что рынок данных не обязательно должен быть полностью сформирован. Он может содержать ссылки на склад данных и добирать оттуда информацию по мере поступления запросов. Конечно, это несколько увеличивает время отклика, но зато снимает проблему ограниченного объема многомерной базы данных.
* Наконец, на третьем уровне находятся клиентские рабочие места конечных пользователей, на которых устанавливаются средства оперативного анализа данных.

**8.2.5. Характеристика интегрированных продуктов ведущих компаний для организации складов данных**

В этом разделе мы коротко охарактеризуем продукты ведущих поставщиков, имеющие связь с технологией складов данных.

**8.2.5.1. Компания IBM**

Решение компании IBM называется ADataWarehousePlus. Целью компании является обеспечение интегрированного набора программных продуктов и сервисов, основанных на единой архитектуре. Основой складов данных является семейство СУБД DB2. Преимуществом IBM является то, что данные, которые нужно извлечь из оперативной базы данных и поместить в склад данных, находятся в системах IBM. Поэтому естественна тесная интеграция программных продуктов.

Предлагаются три решения для складов данных:

1. Изолированный рынок данных. Предназначен для решения отдельных задач вне связи с общим хранилищем корпорации.
2. Зависимый рынок данных. Аналогичен изолированному рынку данных, но источники данных находятся под централизованным контролем.
3. Глобальный склад данных. Корпоративное хранилище данных, которое полностью централизовано контролируется и управляется. Глобальный склад данных может храниться централизовано или состоять из нескольких распределенных в сети рынков данных.

**8.2.5.2. Oracle**

Решение компании Oracle в области складов данных основывается на двух факторах: широкий ассортимент продуктов самой компании и деятельность партнеров в рамках программы WarehouseTechnologyInitiative. Возможности Oracle в области складов данных базируются на следующих составляющих:

* наличие реляционной СУБД Oracle 7 (а теперь и Oracle 8), которая постоянно совершенствуется для лучшего удовлетворения потребностей складов данных;
* существования набора готовых приложений, обеспечивающих возможности разработки склада данных;
* высокий технологический потенциал компании в области анализа данных;
* доступность ряда продуктов, производимых другими компаниями.

**8.2.5.3. HewlettPackard**

Работы, связанные со складами данных, выполняются в рамках программы OpenWarehouse. Выполнение этой программы должно обеспечить возможность построения складов данных на основе мощных компьютеров HP, аппаратуры других производителей и программных компонентов. Основой подхода HP являются Unix-платформы и программный продукт IntelligentWarehouse, который предназначен для управления складами данных. Основа построения складов данных, предлагаемая HP, оставляет свободу выбора реляционной СУБД, средств реинжиниринга и т.д.

**8.2.5.4. Sybase**

Стратегия компании в области складов данных основывается на разработанной ей архитектуре WarehouseWORKS. В основе подхода находится реляционная СУБД SybaseSystem 11, средство для подключения и доступа к базам данных OmniCONNECT и средство разработки приложений Powerbuilder. Компания продолжает совершенствовать свою СУБД для лучшего удовлетворения потребностей складов данных (например, введена побитная индексация).

**8.2.5.5. InformixSoftware**

Стратегия компании в отношении складов данных направлена на расширение рынка для ее продукта OnLineDynamicParallelServer. Предлагаемая архитектура склада данных базируется на четырех технологиях: реляционные базы данных, программном обеспечении для управления складом данных, средствах доступа к данным и платформе открытых систем. Три последние компонента разрабатываются партнерами компании. После выхода Универсального Сервера, основанного на объектно-реляционном подходе, можно ожидать, что и он будет использоваться для построения складов данных.

**8.2.5.6. AT&TGIS**

Решение компании направлено на решение проблем корпораций, у которых одинаково сильны потребности и в системах поддержки принятия решений, и в системах оперативной аналитической обработки данных. Предлагаемая архитектура называется EnterpriseInformationFactory и основывается на опыте использования системы управления базами данных Teradata и связанных с ней методах параллельной обработки.

**8.2.5.7. SASInstitute**

Компания считает себя поставщиком полного решения для организации склада данных. Подход основан на следующем:

* обеспечение доступа к данным с возможностью их извлечения из самых разнообразных хранилищ данных и реляционных, и нереляционных;
* преобразование данных и манипулирование ими с использованием 4GL;
* наличие сервера многомерных баз данных;
* большой набор методов и средств для аналитической обработки и статистического анализа.

**8.2.5.8. SoftwareAG**

Деятельность компании в области складов данных происходит в рамках программы OpenDataWarehouseInitiative. Программа базируется на основных продуктах компании ADABAS и Natural 4GL, собственных и приобретенных средствах извлечения и анализа данных, средстве управления складом данных SourcePoint. SourcePoint позволяет автоматизировать процесс извлечения и пересылки данных, а также их загрузки в склад данных.

Существует еще целый ряд компаний, которые прямо или косвенно связаны с технологией складов данных, но мы ограничимся перечисленными, поскольку их продукты и подходы кажутся наиболее продвинутыми.

# 9. Разновидности и архитектуры информационных приложений

В зависимости от конкретной области применения, информационные системы могут очень сильно различаться по своим функциям, архитектуре, реализации. Однако можно выделить по крайней мере два свойства, которые являются общими для всех информационных систем.

Во-первых, любая информационная система предназначена для сбора, хранения и обработки информации. Поэтому в основе информационной системы лежит среда хранения и доступа к данным. Среда должна обеспечивать уровень надежности хранения и эффективность доступа, соответствующие области применения информационной системы.

Во-вторых, информационные системы ориентируются на конечного пользователя, например, банковского клерка. Такие пользователи могут быть очень далеки от мира компьютеров. Для них терминал, персональный компьютер или рабочая станция представляют собой всего лишь орудие их собственной профессиональной деятельности. Поэтому информационная система обязана обладать простым, удобным, легко осваиваемым интерфейсом, который должен предоставить конечному пользователю все необходимые для его работы функции, но в то же время не дать ему возможность выполнять какие-либо лишние действия. Иногда этот интерфейс может быть графическим с меню, кнопками, подсказками и т.д. Сейчас очень популярны графические интерфейсы, и многие современные средства разработки информационных приложений прежде всего ориентированы на разработку графических интерфейсов. Наличие развитых интерфейсных средств является обязательным для любой современной информационной системы.

Тематика прикладных информационных систем исключительно широка. В этой части курса мы рассмотрим только некоторые вопросы, связанные с организацией, разработкой и использованием таких систем, причем не будем слишком углубляться в технические детали.

### 9.1. Обзор рынка готовых информационных приложений

Сегодня трудно найти компанию, производящую компьютерную технику или программное обеспечение, которая не поставляла бы средства разработки информационных приложений, компоненты информационных систем или законченные готовые к использованию решения. Одним из немногих исключений из этого правила является компания InformixSoftware, которая (по крайней мере, пока) производит только базовые программные системы и средства разработки, но не предлагает собственных готовых приложений, оставляя их создание своим партнерам.

Вообще, в компьютерном мире понятие стороннего поставщика (third-partycompany) имеет очень большое значение, поскольку значительная часть прикладного программного обеспечения производится именно небольшими независимыми софтверными компаниями.

**9.1.1. Информационные приложения, поставляемые крупными компаниями производителями вычислительной техники и СУБД (Oracle, Hewlett-Packard, IBM, Microsoft и т.д.)**

Исключительно в качестве примера приведем очень краткую характеристику прикладных программных продуктов, производимых и распространяемых некоторыми ведущими компаниями.

Начнем с компании Oracle. Представители этой компании любят представлять набор производимых продуктов в виде перевернутой пирамиды с вершиной внизу и основанием наверху. Вершине пирамиды соответствует базовый комплект серверов баз данных, предлагаемых компанией. В середине пирамиды располагаются средства проектирования и разработки баз данных и информационных систем (в частности, Designer/Developer 2000). Наконец, основанию пирамиды соответствуют готовые компоненты информационных приложений и законченные решения. Такая картина правильно отражает соотношения объемов и числа продуктов. Конечно, серверы баз данных - это наиболее сложные и ответственные продукты. Но их число и объем кода не очень велики. Средства проектирования и разработки опираются на использование серверов; эти продукты менее сложны, но более объемны. Наконец, приложения создаются с использованием средств проектирования и разработки; приложений много, и суммарный объем кода очень велик.

Oracle предлагает приложения (естественно, основанные на использовании серверов баз данных Oracle) для использования в следующих предметных областях:

* управление финансами (распределенные многовалютные бухгалтерские системы, системы финансового планирования, системы финансового анализа и т.д.);
* управление человеческими ресурсами (автоматизация процедуры найма на работу, планирование обучения персонала, удержания его в компании, продвижения по службе и т.д.);
* автоматизация производства (автоматизированные гибридные производственные системы, системы поддержки новых методов инженерии, системы планирования и моделирования и т.д.);
* системы поддержки ведения проектов (организация и мониторинг процесса проектирования, отслеживание проектных расходов и т.д.);
* автоматизация деятельности по продажам (анализ рынка, управление рыночной деятельностью, управление продажами);
* системы сетевого планирования.

Другая картина представлена компанией HewlettPackard. Как один из ведущих производителей вычислительной техники, компания в основном зарабатывает деньги за счет продажи компьютеров. Но с другой стороны, для увеличения спроса желательно предлагать заказчикам готовые решения. Это и делает компания. В кооперации с компаниями-разработчиками программного обеспечения (в частности, с Oracle) разрабатываются законченные аппаратно-программные конфигурации, пригодные для использования в различных предметных областях.

В частности, имеются решения HP для применения в следующих сферах:

* проверка качества окружающей среды;
* управление финансами;
* автоматизация государственной деятельности (федерального уровня, уровня штата, уровня города и т.д.);
* автоматизация деятельности в области здравоохранения;
* системы фармацевтического анализа;
* автоматизация производства;
* управление розничной торговлей и т.д.

Компания IBM, являясь крупнейшей (и одной из старейших) в мире производителей компьютеров и программного обеспечения, предлагает компоненты информационных приложений и готовые решения для использования во всех сферах человеческой деятельности. Естественно, как и в случае HewlettPackard, эти решения опираются на аппаратные средства IBM (и базовое программное обеспечение этой же компании).

Совершенно необъятное число информационных приложений предлагает компания Microsoft. Следуя своей технологии компонентной организации программных продуктов, компания Microsoft предлагает компоненты для горизонтального рынка - набор полуфабрикатов, пригодных для построения бухгалтерских систем, систем документооборота, систем планирования ресурсов и т.д., а также готовые продукты для вертикального рынка - финансовые системы, здравоохранение, автоматизация работы гостиниц, издательские системы и т.д.

**9.1.2. Приложения, предлагаемые третьими компаниями (пример: Catalyst компании SunMicrosystems)**

Мы уже отмечали, что большинство прикладных программных продуктов производится небольшими самостоятельными софтверными компаниями. Такие компании в силу ограниченности своих ресурсов часто не могут обеспечить разработку и сопровождение продуктов на разных платформах и сосредотачиваются на поддержке платформ одного поставщика. Компании этого рода недаром называют "сторонними поставщиками", поскольку они ведут свою собственную политику, сами рекламируют и продают свою продукцию и не связаны со своими "старшими братьями" партнерскими отношениями. Тем не менее, основные поставщики также заинтересованы в том, чтобы их потенциальные заказчики знали о существовании продуктов третьих компаний, поскольку это является дополнительным доводом при принятии решения о покупке. Большинство крупных компаний собирает и распространяет информацию о доступности продуктов сторонних поставщиков.

Например, компания SunMicrosystems каждый год издает специальный каталог под названием "Catalyst", содержащий исключительно информацию о продуктах независимых фирм, работающих на платформах Sun. Catalyst обычно имеет объем более 1000 страниц. Прикладные продукты разбиты на предметно-ориентированные категории. По поводу каждого продукта приводятся его краткая характеристика и адрес и другая контактная информация производителя.

### 9.2. Инструментальные средства создания пилотных версий приложений и разработки их законченных вариантов

Понятное дело, что любую программу можно написать на любом языке программирования. Однако, когда речь идет об информационных приложениях, существенно возрастает сложность проектирования и программирования интерфейсов с конечным пользователем (как мы отмечали, эти интерфейсы должны быть предельно просты, понятны и удобны), а также требуется большая скорость получения хотя бы как-нибудь работающей версии программы (чтобы понять, устраивает ли это приложение хотя бы в первом приближении).

**9.2.1. Что такое "быстрая разработка приложений" (rapidapplicationdevelopment)**

Все равно, если Вы хотите получить правильно работающую, настроенную под Ваши потребности, грамотно написанную программу, обращайтесь к профессионалам. Но необходимо знать, какие средства разработки Вашими коллегами применяются. Возможны два подхода: RapidApplicationDevelopment - быстрая разработка приложения и традиционный способ программирования. Быстрая разработка - это, как правило, создание работающего прототипа приложения (прототипа в том смысле, что это все-таки не законченное приложение, а некоторый его прообраз). Этот прототип может быть полностью функционален. (Иногда не полностью; это зависит от сложности приложения.) Что Вам могут честно обещать - это полная отработка интерфейсов. Экранные формы, разного рода меню, подсказки делаются быстро и качественно. Можно почувствовать, как внешне будет выглядеть приложение.

Но это не значит, что внутренность прикладной программы будет достаточно эффективной и качественной. Как правило, за небольшим числом исключений (к ним, в частности, относится язык компании BorlandDelphi) языки быстрого прототипирования являются интерпретируемыми. Это не обязательно означает пошаговую пооператорную обработку программы. Иногда, как в случае языка Java, вся программа подвергается предварительной обработке, в результате которой образуется промежуточный машинно-независимый код, который в дальнейшем исполняется с помощью встроенного машинно-независимого интерпретатора. Но в любом случае интерпретатор остается интерпретатором; он не может так же эффективно выполнять программу как компьютер.

Почему же так часто используются языки быстрой разработки приложений? Основной ответ состоит в том, что они действительно быстро дают возможность получить работающий вариант программы со всеми ее внешними интерфейсами (применительно к системам, связанным с базами данных, такие языки часто называют 4GL, языками четвертого поколения).

**9.2.2. Пригодны ли средства быстрой разработки только на стадии пилотного проекта?**

Конечно же, нет. Можно найти массу простых информационных приложений, основное назначение которых состоит в том, чтобы формировать отчеты на основе заранее подготовленных запросов пользователей к базам данных. Общая эффективность подобных приложений главным образом определяется эффективностью используемой СУБД, и на нее мало влияет интерпретируемость выполнения клиентской части приложения.

Заметим, что большая часть современных языков и инструментальных средств быстрой разработки приложений (например, Delphi компании Borland, PowerBuilder компании Sybase и т.д.) имеют несколько собственных интерфейсов с распространенными серверами баз данных, либо могут работать с ними через драйверы ODBC. Поэтому с архитектурной точки зрения быстро разработанное приложение вполне может соответствовать модели "клиент-сервер". Если же в дополнение к средствам быстрой разработки применяются такие серверные средства как хранимые процедуры, ограничения целостности и триггеры, то часть логики приложения может быть перенесена на сервер баз данных (фактически, это позволяет сформировать третье звено общей цепочки - сервер приложений).

**9.2.3. Тенденции к сближению языков быстрой разработки и языков программирования. Что это дает?**

При использовании средств быстрой разработки приложений для создания сложных информационных систем с необходимостью существенных вычислений на стороне клиента обычно применяется комбинированный подход. Сначала создается пилотный проект, выполненный целиком на языке быстрой разработки, а затем наиболее ответственные части приложения переписываются на некотором традиционном языке программирования (как правило, на Си/Си++), компилируются и компонуются с остальной частью системы. В большинстве средств быстрой разработки такая возможность явно предусматривается и поддерживается. Ясно, что так работать можно. Но возникают некоторые сомнения.

По каким причинам для большинства языков быстрой разработки используется техника интерпретации? Кажется, что можно дать два основных ответа. Во-первых, в случае применения интерпретации упрощаются компиляторы. В частности, в компиляторе исчезает существенный по сложности компонент генерации машинных кодов. В результате средство быстрой разработки может быть реализовано быстрее. Быстрее можно получить и работоспособный вариант приложения. Во-вторых, применение техники интерпретации машинно-независимых промежуточных кодов снимает (или, по крайней мере, облегчает) проблему переноса на новую платформу как самого средства разработки, так и созданных на его основе информационных систем. Но на самом деле, оба приведенных довода являются чисто техническими.

Все мы знаем примеры высококачественных компиляторов традиционных языков программирования (например, того же семейства Си/Си++), которые легко переносятся, умеют генерировать машинный код для многих платформ и обеспечивают простоту переноса приложений. К тому же, эти компиляторы обеспечивают высокое качество генерируемого кода и соответственно высокую эффективность выполнения приложений.

В настоящее время имеется явная тенденция к переходу на новое поколение средств быстрой разработки, которые не утратят достоинства существующих систем, но будут выполняться в режиме компиляции. Одним из ярких примеров таких систем является BorlandDelphi, которая позволяет быстро производить высокоэффективные информационные приложения (ограни- чиваясь, правда, платформами Intel). В компьютерной прессе обсуждаются перспективы к пе- реходу на такую технологию для Powerbuilder и других средств быстрой разработки.

**9.2.4. Особенности инструментальных средств, предназначенных для разработки Intranet-приложений**

Об Intranet-приложениях уже достаточно много говорилось в этом курсе. Тем не менее, для полноты мы немного обсудим особенности инструментальных средств Intranet-приложений и в этом пункте.

Не будем обсуждать базовые механизмы организации Web-ориентированных Intranet-приложений и, в частности, средства их интеграции с другими серверами (включая, естественно, серверы баз данных). Коснемся только языка Java, наиболее популярного на сегодня инструмента Internet/Intranet, и сопоставим особенности реализации и использования Java с языками быстрой разработки, упоминавшимися в предыдущих пунктах этого раздела.

Краткой характеристикой языка Java может быть следующее: более безопасный по сравнению с Си++ объектно-ориентированный язык с постоянно развивающимися библиотеками классов. Компания SunMicrosystems разработала и ввела в обиход этот язык специально для операционной поддержки клиентов Всемирной Паутины. Полная машинная независимость языка Java дала возможность создать ряд интерпретаторов, которые сегодня существуют практически для всех платформ и в состоянии выполнять программы (Java-апплеты), передаваемые клиенту с Web-серверов. Хотя много раз начинались разговоры о реализации компиляторов Java в машинные коды, для "гуляющих" по Сети Java-апплетов более естественно применение интерпретации промежуточных кодов.

### 9.3. Обзор применяемых архитектур современных информационных приложений

В начале этой части курса мы кратко рассмотрели основные требования, которым должна удовлетворять информационная система, и задачи, которые должны решаться такой системой. При этом мы подчеркивали, что строгость соблюдения требований и фиксированность набора решаемых задач во многом являются условными в зависимости от конкретных целей, для достижения которых разрабатывается прикладная информационная система. Соответственно, проектирование и разработка информационной системы может базироваться на разных архитектурных решениях.

В данном разделе приводится классификация возможных архитектур информационных систем. Мы начинаем с традиционных архитектурных решений, основанных на использовании выделенных серверов баз данных и, возможно, серверов приложений. Затем рассматриваются варианты архитектур корпоративных информационных систем, базирующихся на технологии Internet (Intranet-приложения). Следующая разновидность архитектуры информационной системы (еще не вполне установившаяся) относится к приложениям оперативной аналитической обработки данных. Наконец, последняя выделяемая нами архитектура предназначена для построения глобальных распределенных информационных приложений с интеграцией информационно-вычислительных компонентов на основе объектно-ориентированного подхода.

Замечание по поводу терминологии. С терминологией в области информационных систем вообще, а русскоязычной терминологией в особенности, дела обстоят неважно. Область информационных систем очень быстро развивается. Практически каждый год возникают новые технологии и архитектурные решения, для которых в маркетинговых целях придумываются оригинальные, привлекающие внимание названия, далеко не всегда точно отражающие смысл технологии и/или архитектуры. На самом деле, все подходы к организации информационных систем, рассматриваемые в этом курсе базируются на общей архитектуре "клиент-сервер". Различие состоит только в том, что делают клиенты и серверы. Тем не менее, чтобы избежать путаницы, далее мы вынуждены применять русскоязычные эквиваленты соответствующих англоязычных терминов.

Следует заметить, что как и любая классификация, наша классификация архитектур информационных систем не является абсолютно жесткой. В архитектуре любой конкретной информационной системы часто можно найти влияния нескольких общих архитектурных решений. Тем не менее, при архитектурном проектировании системы кажется полезным иметь хотя бы частично ортогонализированный архитектурный базис.

**9.3.1. Информационные системы, использующие серверы приложений**

Под клиент-серверным приложением в узком смысле мы будем понимать информационную систему, основанную на использовании серверов баз данных. Общее представление информационной системы в архитектуре "клиент-сервер" показано на рисунке 9.1.

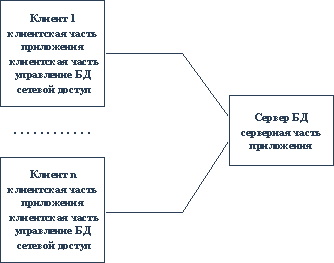


Рис. 9.1. Общее представление информационной системы в архитектуре "клиент-сервер"

* На стороне клиента выполняется код приложения, в который обязательно входят компоненты, поддерживающие интерфейс с конечным пользователем, производящие отчеты, выполняющие другие специфичные для приложения функции.
* Клиентская часть приложения взаимодействует с клиентской частью программного обеспечения управления базами данных, которая, фактически является индивидуальным представителем СУБД для приложения.

(Здесь опять проявляются недостатки в терминологии. Обычно, когда компания объявляет о выпуске очередного сервера баз данных, то неявно понимается, что имеется и клиентская составляющая этого продукта. Сочетание "клиентская часть сервера баз данных" кажется несколько странным, но нам придется пользоваться именно этим термином.)

Интерфейс между клиентской частью приложения и клиентской частью сервера баз данных, как правило, основан на использовании языка SQL. Поэтому такие функции, как, например, предварительная обработка форм, предназначенных для запросов к базе данных, или формирование результирующих отчетов выполняются в коде приложения.

* Наконец, клиентская часть сервера баз данных, используя средства сетевого доступа, обращается к серверу баз данных, передавая ему текст оператора языка SQL.

Здесь необходимо сделать еще два замечания:

1. Обычно компании, производящие развитые серверы баз данных, стремятся к тому, чтобы обеспечить возможность использования своих продуктов не только в стандартных на сегодняшний день TCP/IP-ориентированных сетях, но и в сетях, основанных на других протоколах (например, SNA или IPX/SPX). Поэтому при организации сетевых взаимодействий между клиентской и серверной частями СУБД часто используются не стандартные средства высокого уровня (например, механизмы программных гнезд или вызовов удаленных процедур), а собственно функционально подобные средства, менее зависящие от особенностей сетевых транспортных протоколов.
2. Когда мы говорим об интерфейсе на основе языка SQL, нужно отдавать себе отчет в том, что несмотря на титанические усилия по стандартизации этого языка, нет такой реализации, в которой стандартные средства языка не были бы расширены. Необдуманное использование расширений языка приводит к полной зависимости приложения от конкретного производителя сервера баз данных.

Посмотрим теперь, что же происходит на стороне сервера баз данных. В продуктах практически всех компаний сервер получает от клиента текст оператора на языке SQL.

* Сервер производит компиляцию полученного оператора. На основе информации, содержащейся в таблицах-каталогах базы данных производится преобразование непроцедурного представления оператора в некоторую процедуру его выполнения.
* Далее (если компиляция завершилась успешно) происходит выполнение оператора. Рассмотрим возможные действия операторов SQL:
* Оператор может относиться к классу операторов определения (или создания) объектов базы данных (точнее и правильнее было бы говорить про элементы схемы базы данных или про объекты метабазы данных). В частности, могут определяться домены, таблицы, ограничения целостности, триггеры, привилегии пользователей, хранимые процедуры. В любом случае, при выполнении оператора создания элемента схемы базы данных соответствующая информация помещается в таблицы-каталоги базы данных (в таблицы метабазы данных). Ограничения целостности обычно сохраняются в метабазе данных прямо в текстовом представлении. Для действий, определенных в триггерах, и хранимых процедур вырабатывается и сохраняется в таблицах-каталогах процедурный выполняемый код. Заметим, что ограничения целостности, триггеры и хранимые процедуры являются, в некотором смысле, представителями приложения в поддерживаемой сервером базе данных; они составляют основу серверной части приложения.
* При выполнении операторов выборки данных на основе содержимого затрагиваемых запросом таблиц и, возможно, с использованием поддерживаемых в базе данных индексов формируется результирующий набор данных. Серверная часть СУБД пересылает результат клиентской части, и окончательная обработка производится уже в клиентской части приложения.
* При выполнении операторов модификации содержимого базы данных (INSERT, UPDATE, DELETE) проверяется, что не будут нарушены определенные к этому моменту ограничения целостности (те, которые относятся к классу немедленно проверяемых), после чего выполняется соответствующее действие (сопровождаемое модификацией всех соответствующих индексов и журнализацией изменений). Далее сервер проверяет, не затрагивает ли данное изменение условие срабатывания какого-либо триггера, и если такой триггер обнаруживается, выполняет процедуру его действия. Эта процедура может включать дополнительные операторы модификации базы данных, которые могут вызвать срабатывание других триггеров и т.д. Можно считать, что те действия, которые выполняются на сервере баз данных при проверке удовлетворенности ограничений целостности и при срабатывании триггеров, представляют собой действия серверной части приложения.
* При выполнении операторов модификации схемы базы данных (добавления или удаления столбцов существующих таблиц, изменения типа данных существующего столбца существующей таблицы и т.д.) также могут срабатывать триггеры, т.е., другими словами, может выполняться серверная часть приложения.
* Аналогично, триггеры могут срабатывать при уничтожении объектов схемы базы данных (доменов, таблиц, ограничений целостности и т.д.).
* Особый класс операторов языка SQL составляют операторы вызова ранее определенных и сохраненных в базе данных хранимых процедур. Если хранимая процедура определяется с помощью достаточно развитого языка, включающего и непроцедурные операторы SQL, и чисто процедурные конструкции (например, языка PL/SQL компании Oracle), то в такую процедуру можно поместить серьезную часть приложения, которое при выполнении оператора вызова процедуры будет выполняться на стороне сервера, а не на стороне клиента.
* При выполнении оператора завершения транзакции сервер должен проверить соблюдение всех, так называемых, отложенных ограничений целостности (к таким ограничениям относятся ограничения, накладываемые на содержимое таблицы базы целиком или на несколько таблиц одновременно; например, суммарная зарплата сотрудников отдела 999 не должна превышать 150 млн. руб.). Снова к проверке отложенных ограничений целостности можно относиться как к выполнению серверной части приложения.

Как видно, в клиент-серверной организации клиенты могут являться достаточно "тонкими", а сервер должен быть "толстым" настолько, чтобы был в состоянии удовлетворить потребности всех клиентов (рисунок 9.2).

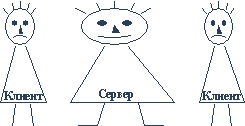


Рис. 9.2. Тонкий клиент и толстый сервер в клиент-серверной архитектуре

С другой стороны, разработчики и пользователи информационных систем, основанных на архитектуре "клиент-сервер", часто бывают неудовлетворены постоянно существующими сетевыми накладными расходами, которые следуют из потребности обращаться от клиента к серверу с каждым очередным запросом. На практике распространена ситуация, когда для эффективной работы отдельной клиентской составляющей информационной системы в действительности требуется только небольшая часть общей базы данных. Это приводит к идее поддержки локального кэша общей базы данных на стороне каждого клиента.

Фактически, концепция локального кэширования базы данных является частным случаем концепции реплицированных (или, как иногда их называют в русскоязычной литературе, тиражированных) баз данных. Как и в общем случае, для поддержки локального кэша базы данных программное обеспечение рабочих станций должно содержать компонент управления базами данных - упрощенный вариант сервера баз данных, который, например, может не обеспечивать многопользовательский режим доступа. Отдельной проблемой является обеспечение согласованности (когерентности) кэшей и общей базы данных. Здесь возможны различные решения - от автоматической поддержки согласованности за счет средств базового программного обеспечения управления базами данных до полного перекладывания этой задачи на прикладной уровень. В любом случае, клиенты становятся более толстыми при том, что сервер тоньше не делается (рисунок 9.3).

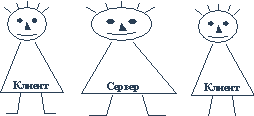


Рис. 9.3. Потолстевший клиент и толстый сервер в клиент-серверной архитектуре с поддержкой локального кэша на стороне клиентов

Другой идеей, позволяющей сократить сетевой трафик между компонентам клиент-серверной организации, является выделение дополнительного сервера (сервера приложений), на который перекладывается часть прикладной обработки и в котором, в свою очередь, может поддерживаться локальный кэш базы данных. Тогда клиенты взаимодействуют напрямую только с сервером (или несколькими серверами) приложений (естественно, что при этом могут использоваться протоколы более высокого уровня, чем при взаимодействии с сервером баз данных, и интенсивность этих взаимодействий может быть снижена), а сервер приложений по мере необходимости обращается к серверу баз данных (рисунок 9.4).

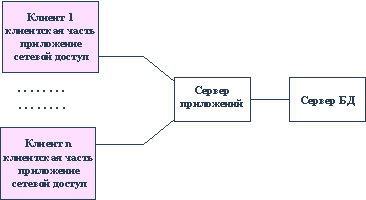


Рис. 9.4. Информационная система в архитектуре "клиент-сервер" с выделенным сервером приложений

Информационные системы в трехзвенном (или многозвенном исполнении) могут создаваться на основе использования промежуточного программного обеспечения мониторов распределенных транзакций, например, Tuxedo компании BEASystems, Inc. или Encina компании TransarcCorp.

Громадным преимуществом клиент-серверной архитектуры любого рода является ее масштабируемость и вообще способность к развитию. При проектировании информационной системы, основанной на этой архитектуре, большее внимание следует обращать на грамотность общих решений. Технические средства пилотной версии могут быть минимальными (например, в качестве аппаратной основы сервера баз данных может использоваться одна из рабочих станций). После создания пилотной версии нужно провести дополнительную исследовательскую работу, чтобы выяснить узкие места системы. Только после этого необходимо принимать решение о выборе аппаратуры сервера, которая будет использоваться на практике.

Увеличение масштабов информационной системы не порождает принципиальных проблем. Обычным решением является замена аппаратуры сервера (и, может быть, аппаратуры рабочих станций, если требуется переход к локальному кэшированию баз данных). В любом случае практически не затрагивается прикладная часть информационной системы. В идеале, которого конечно же не бывает, информационная система продолжает нормально функционировать даже после смены аппаратуры.

**9.3.2. Влияние intranet-технологии**

В предыдущих разделах курса уже говорилось довольно много о технологии Internet/Intranet. В этом пункте мы сосредоточимся на возможных архитектурных решениях Intranet-систем.

Возникновение и внедрение в широкую практику высокоуровневых служб Всемирной Сети Сетей Internet (e-mail, ftp, telnet, Gopher, WWW и т.д.) естественным образом повлияли на технологию создания корпоративных информационных систем, породив направление, известное теперь под названием Intranet. По сути дела, информационная Intranet-система - это корпоративная система, в которой используются методы и средства Internet. Такая система может быть локальной, изолированной от остального мира Internet, или опираться на виртуальную корпоративную подсеть Internet. В последнем случае особенно важны средства защиты информации от несанкционированного доступа.

Хотя в общем случае в Intranet-системе могут использоваться все возможные службы Internet, наибольшее внимание привлекает гипермедийная служба WWW (WorldWideWeb - Всемирная Паутина). Видимо, для этого имеются две основные причины. Во-первых, с использованием языка гипермедийной разметки документов HTML можно сравнительно просто разработать удобную для использования информационную структуру, которая в дальнейшем будет обслуживаться одним из готовых Web-серверов. Во-вторых, наличие нескольких готовых к использованию клиентских частей - браузеров, или "обходчиков", избавляет от необходимости создавать собственные интерфейсы с пользователями, предоставляя им удобные и развитые механизмы доступа к информации. В ряде случаев такая организация корпоративной информационной системы (рисунок 9.5) оказывается достаточной для удовлетворения потребностей компании.

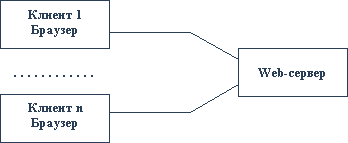


Рис. 9.5. Простая организация Intranet-системы с использованием средств WWW

Однако при всех своих преимуществах (простота организации, удобство использования, стандартность интерфейсов и т.д.) эта схема обладает сильными ограничениями. Прежде всего, как видно из рисунка 9.5, в информационной системе отсутствует прикладная обработка данных. Все, что может пользователь, это только просмотреть информацию, поддерживаемую Web-сервером. Далее, гипертекстовые структуры трудно модифицируются. Для того, чтобы изменить наполнение Web-сервера, необходимо приостановить работу систему, внести изменения в HTML-описания и только затем продолжить нормальное функционирование. Наконец, далеко не всегда достаточен поиск информации в стиле просмотра гипертекста. Базы данных и соответствующие средства выборки данных по-прежнему часто необходимы.

На самом деле, все перечисленные трудности могут быть разрешены с использованием более развитых механизмов Web-технологии. Эти механизмы непрерывно совершенствуются, что одновременно и хорошо и плохо. Хорошо, потому что появляются новые возможности. Плохо, потому что отсутствует стандартизация.

Что касается логики приложения, то при применении Web-технологии существует возможность ее реализации на стороне Web-сервера. Для этого могут использоваться два подхода - CGI (CommonGatewayInterface) и API (ApplicationProgrammingInterface). Оба подхода основываются на наличии в языке HTML специальных конструкций, информирующих клиента-браузера, что ему следует послать Web-серверу специальное сообщение, при получении которого сервер должен вызвать соответствующую внешнюю процедуру, получить ее результаты и вернуть их клиенту в стандартном формате HTTP (рисунок 9.6).

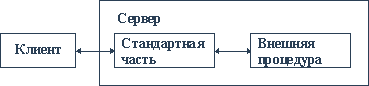


Рис. 9.6. Вызов внешней процедуры Web-сервера

Заметим, что подход CGI является более надежным (внешняя программа выполняется в отдельном адресном пространстве), но менее эффективным, чем подход API (в этом случае внешние процедуры компонуются совместно со стандартной частью Web-сервера).

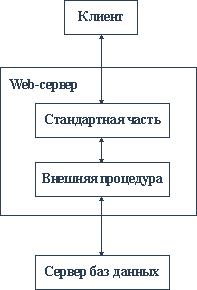


Рис. 9.7. Доступ к базе данных в Intranet-системе

Аналогичная техника широко используется для обеспечения унифицированного доступа к базам данных в Intranet-системах. Язык HTML позволяет вставлять в гипертекстовые документы формы. Когда браузер натыкается на форму, он предлагает пользователю заполнить ее, а затем посылает серверу сообщение, содержащее введенные параметры. Как правило, к форме приписывается некоторая внешняя процедура сервера. При получении сообщения от клиента сервер вызывает эту внешнюю процедуру с передачей параметров пользователя. Понятно, что такая внешняя процедура может, в частности, играть роль шлюза между Web-сервером и сервером баз данных. В этом случае параметры должны специфицировать запрос пользователя к базе данных. В результате получается конфигурация информационной системы, схематически изображенная на рисунке 9.7.

На принципах использования внешних процедур основывается также возможность модификации документов, поддерживаемых Web-сервером, а также создание временных "виртуальных" HTML-страниц.

Даже начальное введение в Intranet было бы неполным, если не упомянуть про возможности языка Java. Java - это интерпретируемый объектно-ориентированный язык программирования, созданный на основе языка Си++ с удалением из него таких "опасных" средств как адресная арифметика. Мобильные коды (апплеты), полученные в результате компиляции Java-программы, могут быть привязаны к HTML-документу. В этом случае они поступают на сторону клиента вместе с документом и выполняются либо автоматически, либо по явному указанию. Апплет может быть, в частности, специализирован как шлюз к серверу баз данных (или к какому-либо другому серверу). При применении подобной техники доступа к базам данных схема организации Intranet-системы становится такой как на рисунок 9.8.

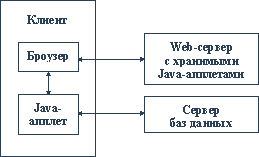


Рис. 9.8. Доступ к базе данных на стороне клиента Intranet-системы

На наш взгляд, Intranet является удобным и мощным средством разработки и использования информационных систем. Как мы уже отмечали, единственным относительным недостатком подхода можно считать постоянное изменение механизмов и естественное отсутствие стандартов. С другой стороны, если информационная система будет создана с использованием текущего уровня технологии и окажется удовлетворяющей потребностям корпорации, то никто не будет обязан что-либо менять в системе по причине появления более совершенных механизмов.

**9.3.3. Особенности архитектур приложений, ориентированных на оперативную аналитическую обработку**

Специфика архитектур приложений, ориентированных на оперативную аналитическую обработку, связана с особенностями организации требуемых хранилищ информации - складов данных (datawarehouse) и рынков данных (datamarts). Мы кратко обсудили возможные архитектурные решения и не будем больше возвращаться к этому вопросу (хотя упомянуть его нужно было обязательно, поскольку это отдельное направление архитектурной организации информационных систем).

**9.3.4. Перспективные архитектуры глобальных распределенных информационных приложений**

Нет никаких проблем, если с самого начала информационное приложение проектируется и разрабатывается в духе подхода открытых систем: все компоненты являются мобильными и интероперабельными, общее функционирование системы не зависит от конкретного местоположения компонентов, система обладает хорошими возможностями сопровождаемости и развития. К сожалению, на практике этот идеал является трудно достижимым. По разным причинам (мы перечислим некоторые из них ниже) возникают потребности в интеграции независимо и по-разному организованных информационно-вычислительных ресурсов. Видимо, ни в одной действительно серьезной распределенной информационной системе не удастся обойтись без применения некоторой технологии интеграции. К счастью, теперь существует путь решения этой проблемы, который сам лежит в русле открытых систем, - подход, предложенный крупнейшим международным консорциумом OMG (ObjectManagementGroup).

Остановимся на некоторых факторах, стимулирующих использование методов интеграции разнородных информационных ресурсов:

* Неоднородность, распределенность и автономность информационных ресурсов системы. Неоднородность ресурсов может быть синтаксической (при их представлении используются, например, разные модели данных) и/или семантической (используются разные виды семантических правил, детализируются и/или агрегируются разные аспекты предметной области). Возможна и чисто реализационная неоднородность информационных ресурсов, обусловленная использованием разных компьютерных платформ, операционных систем, систем управления базами данных, систем программирования и т.д.).
* Потребности в интеграционном комплексировании компонентов информационной системы. Очевидно, что наиболее естественным способом организации сложной информационной системы является ее иерархически-вложенное построение. Более сложные фун- кционально-ориентированные компоненты строятся на основе более простых компонентов, которые могли проектироваться и разрабатываться независимо (что порождает неоднородность; ниже мы приведем примеры).
* Реинжинерия системы. После создания начального варианта информационной системы неизбежно последует процесс ее непрерывных переделок (реинжинерии), обусловленный развитием и изменением соответствующих бизнес-процессов корпорации. Реконструкция системы не должна быть революционной. Все компоненты, не затрагиваемые процессом реинжиниринга, должны сохранять работоспособность.
* Решение проблемы унаследованных (legacy) систем. Любая компьютерная система (на- деюсь, что это не относится к открытым системам в теперешнем понимании; только надеюсь, поскольку неизвестно, как отнесутся к нашим взглядам будущие поколения) со временем становится бременем корпорации. Постоянно (и чем раньше, тем лучше) приходится решать задачу встраивания устаревших информационных компонентов в систему, основанную на новой технологии. Нужно, чтобы эта задача была разрешимой, т.е. чтобы компоненты унаследованных систем сохраняли интероперабельность.
* Повторно используемые (reusable) ресурсы. Технология разработки информационных систем должна способствовать использованию уже существующих компонентов, что в конечном итоге должно перевести нас от экстенсивного ручного программистского труда к интенсивным методам сборки ориентированной на конкретную область применения информационной системы.
* Продление жизненного цикла информационной системы. Чем дольше живет и приносит пользу информационная система, тем это выгоднее для корпорации. Естественно, что для этого должна существовать возможность добавления в нее компонентов, спроектированных и разработанных, вообще говоря, в другой технологии.

Решение проблемы интеграции неоднородных информационных ресурсов началось с попыток интеграции неоднородных баз данных. Направление интегрированных или федеративных систем неоднородных БД и мульти-БД появилось в связи с необходимостью комплексирования систем БД, основанных на разных моделях данных и управляемых разными СУБД.

Основной задачей интеграции неоднородных БД является предоставление пользователям интегрированной системы глобальной схемы БД, представленной в некоторой модели данных, и автоматическое преобразование операторов манипулирования БД глобального уровня в операторы, понятные соответствующим локальным СУБД. В теоретическом плане проблемы преобразования решены, имеются реализации.

При строгой интеграции неоднородных БД локальные системы БД утрачивают свою автономность. После включения локальной БД в федеративную систему все дальнейшие действия с ней, включая администрирование, должны вестись на глобальном уровне. Поскольку пользователи часто не соглашаются утрачивать локальную автономность, желая тем не менее иметь возможность работать со всеми локальными СУБД на одном языке и формулировать запросы с одновременным указанием разных локальных БД, то развивается направление мульти-БД. В системах мульти-БД не поддерживается глобальная схема интегрированной БД и применяются специальные способы именования для доступа к объектам локальных БД. Как правило, в таких системах на глобальном уровне допускается только выборка данных. Это позволяет сохранить автономность локальных БД.

Как правило, интегрировать приходится неоднородные БД, распределенные в вычислительной сети. Это в значительной степени усложняет реализацию. Дополнительно к собственным проблемам интеграции приходится решать все проблемы, присущие распределенным СУБД: управление глобальными транзакциями, сетевую оптимизацию запросов и т.д. Очень трудно добиться эффективности. Как правило, для внешнего представления интегрированных и мульти-БД используется (иногда расширенная) реляционная модель данных. В последнее время все чаще предлагается использовать объектно-ориентированные модели, но на практике пока основой является реляционная модель. Поэтому, в частности, включение в интегрированную систему локальной реляционной СУБД существенно проще и эффективнее, чем включение СУБД, основанной на другой модели данных.

Основным недостатком систем интеграции неоднородных баз данных является то, что при этом не учитываются "поведенческие" аспекты компонентов прикладной системы. Легко заметить, что даже при наличии развитой интеграционной системы, большинство из указанных выше проблем не решается. Естественным развитием взглядов на информационные ресурсы является их представление в виде набора типизированных объектов, сочетающих возможности сохранения информации (своего состояния) и обработки этой информации (за счет наличия хорошо определенного множества методов, применимых к объекту). Наиболее существенный вклад в создание соответствующей технологии внес международный консорциум OMG, выпустивший ряд документов, в которых специфицируются архитектура и инструментальные средства поддержки распределенных информационных систем, интегрированных на основе общего объектно-ориентированного подхода.

В базовом документе специфицируется эталонная модель архитектуры (OMA - ObjectManagementArchitecture) распределенной информационной системы (рисунок 9.9). Согласованная с архитектурой OMA прикладная информационная система представляется как совокупность классов и экземпляров объектов, которые взаимодействуют при поддержке брокера объектных заявок (ORB - ObjectRequestBroker). ORB, общие средства (CommonFacilities) и объектные службы (ObjectServices) относятся к категории промежуточного программного обеспечения (middleware) и должны поставляться вместе. Объектные службы представляют собой набор услуг (интерфейсов и объектов), которые обеспечивают выполнение базовых функций, требуемых для реализации прикладных объектов и объектов категории "общие средства" (например, специфицированы служба именования объектов, служба долговременного хранения объектов, служба управления транзакциями и т.д.). Общие средства содержат набор классов и экземпляров объектов, поддерживающих функции, полезные в разных прикладных областях (например, средства поддержки пользовательского интерфейса, средства управления информацией и т.д.).

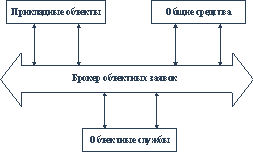


Рис. 9.9. Эталонная модель OMA

В основе OMA лежит базовая объектная модель COM (CoreObjectModel), в которой специфицированы такие понятия, как объект, операция, тип, подтипизация, наследование, интерфейс. Определены также способы согласованного расширения COM в разных объектных службах.

Интерфейсы объекта-клиента и объекта-сервера должны быть определены на специальном языке IDL (InterfaceDefinitionLanguage), который очень напоминает компонент спецификации класса (без реализации) языка Си++. Обращения к ORB могут быть сгенерированы статически при компиляции спецификаций IDL или выполнены динамически с использованием специфицированного в документах OMGAPI брокера объектных заявок. Правила построения и использования ORB определены в документе OMGCORBA (CommonObjectRequestBrokerArchitecture).

Заметим, что архитектура, предложенная OMG, не является единственно возможной. В частности, альтернативные (и очень популярные) решения предлагает компания Microsoft со своей моделью SOM и продуктами OLE, ActiveX, OLEDB.

Основным выводом из материала этого пункта является то, что проблемы интеграции неоднородных информационных ресурсов являются актуальными для корпораций и существуют технологии, позволяющие решать эти проблемы.

В заключение данного раздела курса повторим, что приведенная классификация методов и технологий не является полностью ортогональной. В реальной жизни требуется использовать разумные сочетания технологических и архитектурных решений. Наша цель не состоит в том, чтобы выдать готовые рецепты построения информационной системы. Мы стремимся погрузить слушателей в мир информационных технологий с тем, чтобы в дальнейшем было проще ориентироваться в этом мире и понимать значение его сущностей.

### 9.4. Проблемы проектирования и разработки приложений

В этом разделе мы кратко обсудим проблемы построения информационных приложений, которые, с одной стороны, примыкают к чисто техническим проблемам, а с другой стороны, тес- но связаны с проблемами организационными. Отметим, что в данном разделе многое является спорным и отражает личную точку зрения автора (которая совпадает с точкой зрения многих авторитетных специалистов).

**9.4.1. Как правильно оценить текущие и будущие потребности организации**

На самом деле, это является одной из наиболее сложных задач. Недооценка потребностей приводит к размножению мелких информационных систем, обеспечивающих отдельные информационные нужды предприятий (например, бухгалтерия, складское хозяйство, отдел кадров, служба маркетинга и т.д.). В дальнейшем может оказаться, что эти системы частично дублируют одна другую, а их информация сильно перекрывается. Наличие нескольких информационных систем, построенных на разных технологиях, - это и источник "унаследован- ных" систем. Переоценка потребностей может привести к созданию чрезмерно масштабной системы с неразумным расходованием средств и потребностью в квалифицированном штате администраторов.

Самое главное, что здесь невозможны общие рекомендации. Руководители организации обладают (или, по крайней мере, должны обладать) развитой интуицией относительно перспектив развития. Сторонние компании-интеграторы имеют больший опыт создания информационных систем, могут лучше оценить доступность недорогих и эффективных решений.

**9.4.2. Как выбрать базовые аппаратно-программные средства, чтобы не потратить слишком много денег и не ограничить будущее развитие приложения**

Естественно, что требования к аппаратно-программным средствам определяются требованиями к информационной системе в целом. Какие бы информационные возможности не требовались служащим корпорации, окончательное решение всегда принимается ее руководством, которое корректирует требования к информационной системе и формирует окончательное представление об аппаратной среде (что является исходной точкой и проектов, выполняемых сторонними интеграторами). На наш взгляд, имеются четыре возможных позиции руководства по поводу места информационной системы в корпорации: пессимистическая, пессимистически-оптимистическая, оптимистически-пессимистическая и оптимистическая.

Руководитель-пессимист рассуждает следующим образом. Корпорации нужно продержаться хотя бы какое-то время. Без информационной системы это невозможно. Нужно выбрать самое дешевое решение, которое может быть реализовано максимально быстро. Руководитель не думает, что будет с корпорацией через два года (вернее, поскольку он пессимист, то думает, что, скорее всего, через два года корпорация просто не будет существовать или у нее сменится руководитель). При такой позиции наиболее подходящим является некоторое закрытое и законченное техническое решение. Например, это может быть полностью сбалансированная локальная сеть Novell с выделенным файл-сервером и фиксированным числом рабочих станций. Жесткость решения затем закрепляется соответствующим программным обеспечением информационной системы. Возможности расширения системы отсутствуют, реинжиниринг требует практически полной переделки системы.

Пессимистически-оптимистический руководитель не ожидает краха корпорации или своего собственного увольнения. Но он не надеется на изменение статуса компании, например, на появление зарубежных филиалов. Возможно, корпорация будет несколько развиваться, возможно, появятся новые виды бизнеса, возможно, увеличится число служащих. Но поскольку руководитель все-таки более пессимист, чем оптимист, то он не очень высоко оценивает шансы на развитие (дай-то Бог, чтобы за два года мы выросли на 20%). Такой позиции руководства больше всего подходит закрытое решение, обладающее ограниченными возможностями расширения. Например, это может быть локальная сеть Novell, в которой пропускная способность превосходит потребности имеющихся рабочих станций, а файл-сервер может быть оснащен дополнительными магнитными дисками. Если пессимизм руководителя окажется неоправданным, то корпорация встретится с потребностью сложного реинжиниринга.

Руководитель с оптимистически-пессимистической позицией ставит своей целью развитие корпорации. Он учитывает, что при развитии корпорации потребуется соответствующее развитие информационной системы, для чего, вообще говоря, может понадобиться сменить сервер баз данных. Он учитывает, что при развитии корпорации могут образоваться территориально разнесенные офисы, в результате чего, возможно, потребуется перейти к использованию распределенной базы данных. Он учитывает, в конце концов, что корпорации могут потребоваться развитые средства телекоммуникации с удаленными филиалами и/или партнерами. Пессимизм руководителя состоит только в том, что он заранее делает ставку на одного производителя (эти-то продукты я знаю). Например, может быть сделана установка на использование только Intel-платформ в среде Microsoft или только Alpha-платформ с VAX/VMS. Вообще говоря, это здоровый пессимизм, поскольку однородность аппаратно-программной среды существенно облегчает ее администрирование. Но как обидно будет этому руководителю, если ему предложат дешево купить прекрасный аппаратный продукт другого производителя. Придется либо отказаться, либо снова производить изнурительный реинжиниринг.

Наконец, руководитель-оптимист разделяет позицию оптимистически-пессимистического руководителя по поводу перспектив развития корпорации, но при этом желает сохранить возможность использования различных аппаратных платформ. Он стимулирует построение открытой корпоративной информационной системы, которая может неограниченно наращиваться за счет подключения новых сегментов сети, включения новых серверов и рабочих станций. Оптимистический подход, естественно, требует применения международных стандартов, что облегчает комплексирование аппаратного комплекса и обеспечивает реальное масштабирование информационной системы. Если начать построение корпоративной системы с сети Ethernet с использованием стека протоколов TCP/IP, то это начало оптимистического решения. В этом случае проблемы реинжиниринга практически отсутствуют (пока не сменятся стандарты).

Конечно, приведенная классификация является несколько утрированной. В жизни все гораздо сложнее. В частности, нельзя не учитывать влияние на руководителя технических специалистов. Чем грамотнее составляется обоснование на приобретение технических средств, тем обоснованнее оптимизм или пессимизм руководителя. Конечно, многое определяется возможными денежными затратами. Но стоит заметить, что оптимистический подход (по сути дела, это подход открытых систем) требует минимальных затрат на начальном этапе становления корпоративной системы (если, конечно, не учитывать необходимость приобретения хорошего программного сервера баз данных).

Главный вывод этого пункта состоит в том, что выбор технических средств для построения корпоративной информационной системы - это непростая задача, включающая технические, политические и эмоциональные аспекты. Ни в коем случае комплексирование аппаратуры нельзя пускать на самотек. Простой пример. Многие считают, что чем больше тактовая частота аппаратного сервера баз данных, то тем быстрее будет работать СУБД. Вообще говоря, это неправильно. Быстродействие любого сервера баз данных в основном определяется объемом основной памяти и/или числом процессоров. Другими словами, с одинаковой тщательностью нужно относиться и к выбору общей аппаратной архитектуры системы, и к выбору конфигурации каждого из ее компонентов.

**9.4.3. Какие этапы разработки проекта приложения являются наиболее дорогостоящими и почему**

Это в большой степени зависит от масштабности проекта и вида проектируемого информационного приложения. Например, если используется традиционная двухзвенная архитектура системы с применением одного из современных сервера баз данных, то, по всей видимости, самой дорогой частью проекта будет проект базы данных. И понятно, почему. Во-первых, для проектирования сложной базы данных трудно обойтись без дорогостоящей системы автоматизации проектирования (CASE-системы). Во-вторых, корректность схемы базы данных и правильность ее построения являются залогом работоспособности и эффективности будущей информационной системы. Поэтому к проектированию базы данных должны применяться максимальные усилия.

Если же по архитектурным соображениям выбирается многозвенная архитектура "клиент-сервер" с несколькими серверами баз данных и серверами приложений, работающими под управлением общего монитора распределенных транзакций, то трудоемкость проекта несколько перераспределяется. Конечно, по-прежнему критично важен проект (распределенной) базы данных. Но не менее важно понять, где и какие серверы приложений будут использоваться, будет ли применяться репликация и т.д.

Для организации прикладной системы оперативной аналитической обработки информации, основанной на использовании склада данных, наибольшие накладные расходы требуются именно на организацию склада данных, включая механизмы источников данных, очистки и согласования данных, возможности многомерного представления информации и т.д.

**9.4.4. Что такое "унаследованная система" (legacysystem) и как поступить с этим "наследством"**

Особенно трудно создавать современную высокотехнологичную информационную систему в организации, где уже используются информационные системы, основанные на технологии предыдущих поколений. Такие системы называются унаследованными. С ними связаны следующие проблемы: (1) они не приспособлены к интеграции с системами нового поколения; (2) системы часто используются на морально устаревших аппаратно-программных платформах и не могут быть легко перенесены на новые платформы; (3) недоступны разработчики систем, а сами они написаны на старых и трудно постигаемых языках программирования (в частности, на языках ассемблера); (4) без этих систем организация не может существовать, причем не допускается даже временный выход из строя.

Отмеченные проблемы кажутся настолько сложными и противоречивыми, что первым является желание закрыть на них глаза. Собственно, так многие и делают. Пока унаследованная система работает, пускай работает, а когда перестанет функционировать, тогда и подумаем, что делать. Решение простое, но очень ненадежное. Известны примеры, когда мультимиллиардные компании несли колоссальные убытки из-за выхода из строя унаследованной системы.

Так что же делать? Прежде всего нужно оценить важность унаследованной системы для организации. Если окажется, что некоторое время без нее можно обойтись, то самым дешевым решением будет воспроизводство системы на новой технологии, которая в будущем не воссоздаст аналогичные проблемы. Если же остановка функционирования системы недопустима, то при проектировании нового поколения информационной системы предприятия потребуется применять некоторую промежуточную технологию типа той, которую мы уже упоминали.

**9.4.5. Как перенести существующее приложение на другую аппаратно-программную платформу**

Конечно, возможность подобного переноса должна быть предусмотрена при первоначальной разработке приложения. Очевидно, что без участия компании-разработчика или ее ответственного представителя невозможно произвести перенос сервера баз данных. Обычно крупные платформы для своих заказчиков оказывают услугу по поставке аналогичного или усовершенствованного продукта для другой платформы (естественно, не бесплатно, но существенно дешевле, чем обошлась бы покупка полностью заново).

Так что основные проблемы могут быть связаны с портированием клиентских частей информационной системы и серверов приложений (а также Web-серверы, если система является Intranet-ориентированной и используется свободно доступный Web-сервер). Опять же, если покупается готовая информационная система или ее сборкой/разработкой занимается компания-интегратор, то условия возможности переноса должны быть точно оговорены в контракте (главное не забыть, что этот перенос может понадобиться).

Наиболее сложным, с одной стороны, и наиболее естественно решаемым является случай, когда организация сама занимается проектированием и разработкой информационной системы. Тогда прежде всего нужно решить, какая операционная система будет использоваться на клиентских местах. Если это какая-то разновидность ОС UNIX (что встречается все реже), то клиентская часть приложения должна разрабатываться с использованием некоторого мультиплатформенного средства разработки, опирающегося на стандарты этой ОС. Тогда с большой вероятностью особые проблемы при переносе не возникнут. Если же будет использоваться операционная система компании Microsoft (Windows 95 или NT), то с большой вероятностью аппаратной платформой клиентской части будет Intel, и проблемы с переносом могут проявиться при смене версии операционной системы. Аналогичные соображения применимы к серверам приложений.

# 10. Проектирование корпоративных сетей

В этом разделе будут рассмотрены разные аспекты проектирования сетей. Прежде всего, это методология проектирования, то есть последовательность и содержание работ, которые необходимо выполнить при создании или модернизации корпоративной сети. Далее рассматривается проблема выбора средств анализа работы сети, без которых трудно сформулировать требования к проектируемой сети, провести натурные эксперименты на пилотной сети и, наконец, провести приемно-сдаточные испытания новой только что спроектированной сети. Следующую группу проблем, рассматриваемых в данном разделе, составляют вопросы взаимодействия заказчика с интегратором.

### 10.1. Особенности проектирования корпоративных сетей

При проектировании корпоративной сети полезно ее представление в виде многослойной пирамиды. Хотя слои этой пирамиды связаны и оказывают непосредственное влияние друг на друга, обычно каждый слой проектируется достаточно автономно, специалистами и фирмами соответствующего профиля.

В зависимости от направления движения по этой пирамиде: сверху вниз - от бизнес-приложений к аппаратной платформе, или снизу вверх - от аппаратуры к приложениям, или от середины - от конкретной СУБД, - все фирмы, работающие в области сетевой интеграции, можно условно разделить на три группы:

1. Фирмы-производители или дистрибьюторы аппаратуры, выступающие в роли интеграторов. У этих интеграторов пирамида опирается на очень узкое основание из одной платформы от одного-двух производителей. Минусы и некоторые плюсы в работе такого интегратора достаточно очевидны.
2. Фирмы, ориентирующиеся на одну из СУБД, например, только на Oracle или Informix. В этом случае узким местом пирамиды является середина: при попытке использовать несколько аппаратных платформ и широкий спектр прикладного программного обеспечения, ограничения диктуются используемой СУБД.
3. Наконец, третья группа - независимые интеграторы, которые могут предлагать любые решения на каждом из уровней пирамиды и которых нельзя уличить в особой привязанности к определенной платформе, сетевым конфигурациям или приложениям. У таких интеграторов единственным критерием выбора каждого конкретного решения в идеале является требование достижения максимального эффекта в рамках заданных ресурсов. В таком случае есть возможность гибко строить любые конфигурации, что позволяет достаточно просто решать проблемы, связанные с тем, что заказчик уже использует, например, какую-либо СУБД и не хочет переучивать свой персонал для работы с другой базой данных.

При этом, при проектировании какого-либо слоя характеристики других слоев, оказывающих влияние на принятие проектных решений, берутся в виде исходных данных, чаще всего в весьма обобщенном виде. Например, при проектировании приложений учитываются скорости, которые может обеспечить сегодняшнее коммуникационное оборудование вполне определенного диапазона стоимости - того диапазона, который имеется в распоряжении предприятия. И наоборот, разработчики транспортной системы ориентируются на усредненные данные о трафике, который могут создать имеющиеся на предприятии приложения и те приложения, которые намечено ввести в действие в ближайшие год-два.

**10.1.1. Этапы проектирования**

Следует оговориться, что рассматриваемые в этом разделе вопросы методологии проектирования часто не вполне соответствуют существующей сейчас практике реализации проектов фирмами-интеграторами. Нельзя однозначно сказать, насколько это хорошо или плохо, но одновременно с произошедшими за последние пять лет изменениями отношений собственности и глубокой организационной перестройкой предприятий была утеряна и культура выполнения проектных работ, включавшая в себя следование строго регламентированному перечню этапов, непременное сопровождение каждого этапа стандартной документацией, ведение протоколов рабочих совещаний и всевозможных видов актов "приемно-сдаточных испытаний ".

Сейчас каждый сетевой интегратор выполняет проекты согласно своим собственным представлениям о рациональной организации труда, и методика проектирования каждой фирмы является неким "ноу-хау". Тем не менее, с учетом более богатого (но, может быть, не всегда нам подходящего) западного опыта можно сформулировать некоторые типовые этапы выполнения сетевых проектов:

*Анализ требований*. На этом этапе формулируются основные деловые цели предприятия, для которого разрабатывается проект, например, сокращение производственного цикла, более оперативный прием заказов или повышение производительности труда за счет более эффективного взаимодействия сотрудников, то есть те цели предприятия, которые в настоящий момент, при существующих средствах и технологиях не вполне достигаются. Осуществляется поиск аналогичных систем, анализируются их сильные и слабые стороны, определяется возможность использования удачного опыта для проектируемой системы.

Разработка *бизнес-модели.* Бизнес-модель можно по-другому назвать функциональной моделью производства, она описывает деловые процедуры, последовательность и взаимозависимость всех выполняемых на предприятии работ. При этом внимание концентрируется не на компьютерной системе, а на деловой практике.

Разработка *технической модели*. Техническая модель описывает в достаточно общих терминах, какое компьютерное оборудование нужно использовать, чтобы достичь целей, определенных в бизнес-модели. Для построения технической модели необходимо провести инвентаризацию всего имеющегося оборудования, определить требования к новой системе (при этом требования должны быть сформулированы не с технической точки зрения, а с позиций руководителей и конечных пользователей сети), на основании этого определить, что из существующего оборудования может быть использовано в новой системе. Далее необходимо определить полный функциональный набор необходимых аппаратных средств без конкретизации марок и моделей оборудования.

После того, как выбрана техническая модель, описывающая сеть в общих терминах, создается так называемая *физическая модель*, которая является подробным описанием конкретных продуктов, их количества, технических параметров и способов взаимодействия.

*Установка и наладка системы.* Данный этап подразумевает координирование поставок от субподрядчиков, управление конфигурированием, инсталляцию и наладку оборудования, обучение персонала.

*Тестирование системы.* На этом этапе должны проводиться приемочные испытания, оговоренные в контракте с интегратором.

*Сопровождение и эксплуатация системы*. Этот этап не имеет четко определенных временных границ, а представляет собой непрерывный процесс.

Для каждого из упомянутых этапов и даже для отдельных более мелких задач может быть разработано *техническое задание*. Постановка задачи зависит от того, какую часть работы решено отдать внешнему интегратору, а какую часть - выполнить своими силами. Этот этап тесно связан с этапами *выбора интегратора* и заключения с ним *контракта*.

**10.1.2. Роль системных интеграторов**

Задачи системного интегратора могут меняться в зависимости от условий контракта, в наиболее же общей форме в его обязанности входят следующие:

* Принятие и обоснование решений на всех этапах разработки сети, начиная от анализа требований к системе и кончая сопровождением и эксплуатацией системы. Эти решения должны обеспечивать подбор и взаимоувязку всех разнородных компонентов корпоративной сети, исходя из конечных потребностей пользователя, то есть наиболее рационально отвечать деловым целям организации-заказчика. Интегратор должен стремиться к вовлечению заказчика в процесс принятия проектных решений для уменьшения вероятности создания системы, не отвечающей целям заказчика, которые могут не вполне четко формулироваться на начальных этапах разработки.
* Управление действиями, направленными на решение конечной задачи. При наличии нескольких субподрядчиков генеральный подрядчик должен обеспечивать их взаимодействие и взаимопонимание. Обычно интеграторы специализируются на проектировании какого-либо одного слоя вычислительной системы, например, слоя сетеобразующего коммуникационного оборудования или же слоя баз данных и т.д. Поэтому для разработки общего проекта корпоративной сети заказчик может либо сам заключать отдельные контракты на раздельное проектирование каждого слоя, либо заключить контракт с генеральным подрядчиком, который в свою очередь, нанимает субподрядчиков на выполнение отдельных проектов. В первом случае заказчик берет на себя роль координатора работ и, в конечном счете, ответственность за успех системы, а во втором это делает интегратор.
* Координация работы всех поставщиков оборудования и программного обеспечения.
* Обеспечение выполнения работ в указанные сроки и в пределах установленной сметы расходов. Чем сложнее замысел и крупнее система, тем больше вероятность несоблюдения сроков или сметы.

### 10.2. Этапы проектирования корпоративных сетей

**10.2.1. Анализ требований**

Нельзя построить хорошую корпоративную сеть без ясного понимания деловых целей предприятия и без четкого плана достижения этих целей. Первый шаг, заключающийся в определении проблем предприятия и, следовательно, целей проекта называется *анализом требований*.

Анализ требований к сети поможет оценить деловую значимость информационно-техно- логических решений, определить главные цели и выбрать приоритеты для отдельных частей компьютерной системы, которую вы хотите улучшить или расширить. Четкое определение требований к функциям сети поможет избежать реализации не нужных свойств сети, что сэкономит средства вашего предприятия. Тщательный анализ требований к сети является основой для написания хорошего *технического задания,* на базе которого системные интеграторы смогут разработать *проект* сети. Наконец, ясное понимание целей поможет сформулировать критерии качества для *оценки и тестирования* реализованной сети.

Для большинства администраторов локальных сетей, особенно масштаба отдела, анализ требований к сети является чем-то экзотическим. Очень часто такой администратор покупает сетевое оборудование, операционные системы и приложения в ближайшем компьютерном магазине. В лучшем случае он консультируется с дистрибьютором или дилером. Очевидно, что администратор сети отдела скорее всего не захочет заниматься анализом требований, так как он покупает простую сеть для отдела, а не для предприятия в целом. Такой произвольный выбор вполне допустим для масштаба отдела, так как эта сеть не оказывает значительного влияния на работу предприятия. Если эта сеть откажет, то это повлияет только на пользователей одного отдела. Кроме того, в связи с тем, что цена такой сети сравнительно невелика, не очень велики и потери в случае ошибки.

Работы по проектированию и установке корпоративной сети лучше всего поручить сотрудникам отдела автоматизации, а не администраторам сетей групп и отделов. Хорошо, если сотрудники этого отдела имели дело с мейнфреймами; в этом случае у них есть опыт тщательного информационного анализа системы, планирования и установки ответственных приложений для больших машин. Вместе с тем совершенно необходим и опыт работы с сетями, для которых (в отличие от централизованных систем на базе мейнфреймов) характерно большее разнообразие оборудования, программного обеспечения и протоколов, а также разнообразие поставщиков.

Для выполнения анализа требований к корпоративной сети необходимо:

* оценить текущее состояние локальных сетей и парка компьютеров на предприятии, что поможет выявить, какие проблемы требуют решения;
* определить цели и выгоды от корпоративной сети, что поможет вам правильно спроектировать сеть;
* обосновать перед руководством предприятия необходимость покупок;
* написать эффективное техническое задание;
* определить критерии для оценки качества сети.

Различные компании выполняют анализ требований к сети различными способами с различной степенью детализации в соответствии с традициями предприятия и его технической политикой. Документ, описывающий требования к сети, может быть объемом от 10 до 150 страниц. Некоторые предприятия выполняют такой анализ сами. Другие предприятия прибегают к помощи консультантов или системных интеграторов. Это делается в тех случаях, когда персонал отдела автоматизации не имеет достаточного опыта в проектировании и установке сетей. Кроме того, использование независимых консультантов или системных интеграторов в силу их объективности поможет взглянуть по-новому на проблемы и на их решения.

**10.2.2. Построение функциональной модели производства**

Сеть предприятия предназначена для того, чтобы выполнять производственные функции, поэтому следует оценить ее роль в производственной структуре предприятия. Для успешного построения корпоративной сети нужно построить функциональную модель (или, по-другому, бизнес-модель), из которой потом получить техническую и физическую модели сети. Большинство крупных системных интеграторов придерживаются такой стратегии.

Архитектура приложений и вычислительной системы играет ключевую роль в деловой архитектуре предприятия. Бизнес предприятия базируется на архитектуре управления данными, на приложениях и архитектуре сети. Успешный анализ требований и успешное построение корпоративной сети требуют от технического специалиста умения думать как бизнесмен. Общей ошибкой некоторых руководителей проектов является мышление только в технических и технологических терминах.

Перед тем, как начать оценивать требования к корпоративной сети, нужно получить общее представление о том, что происходит в каждом отделе. Именно бизнес-модель описывает, как делаются дела на предприятии. В ней обычно не упоминается компьютерная система, она скорее концентрируется на деловой практике и последовательности работ. Сначала постройте модель, в которой отражается последовательность работ всего предприятия, а затем постройте модель для последовательности работ в каждом отделе. Детально опишите, как выполняются работы, кто выполняет эти работы и каковы взаимосвязи между рабочими группами и отделами.

Для разработки бизнес-модели необходимо собрать бригаду, состоящую из руководителей отделов, ведущих специалистов и сотрудников отдела автоматизации. Обратите внимание на следующие моменты:

* необходимо назначить руководителя работы;
* нужно опросить руководителей отделов и конечных пользователей корпоративной сети, чтобы определить их функции и выяснить, как их компьютерные системы помогают им в работе;
* необходимо выяснить, как работа переходит из одного отдела в другой, и каким образом информация и задачи передаются от одного сотрудника к другому;
* необходимо узнать, в чем заключаются зависимости - кто утверждает какой-либо этап работы и в какой последовательности должны завершаться этапы;
* нужно понять, какие узкие места имеются у системы - слишком большое время ответа или же неэффективная обработка данных.

**10.2.2.1. Работа с руководящим составом**

Опросите руководителей подразделений и пользователей для того, чтобы определить, что они хотят получить от корпоративной сети. Их ответы будут лежать в диапазоне от "Я хочу в точности то, что имею сегодня" до прогрессивных идей на переднем плане науки и техники таких, как полностью автоматизированное производство и электронный документооборот. Действительно ли руководители хотят уменьшить количество денег, затрачиваемых на экспресс-доставку и обмен сообщениями за счет улучшения коммуникаций? Действительно ли они хотят взаимодействовать электронным способом с покупателями и поставщиками?

На этой ранней стадии проекта есть время для воображения и мозговых атак. Спросите руководителей, какие сервисы они хотели бы иметь, если бы цена не имела никакого значения. Расспросите руководителей и пользователей о существующей компьютерной системе. Не слишком ли она много простаивает из-за отказов? Не слишком ли она медленная? К каким источникам информации они обычно обращаются? Затем попросите руководителей отделов упорядочить свои требования в порядке их приоритетности. Этот список поможет вам определить этапы развертывания сети.

На этом этапе руководители подразделений предприятия должны оценить эффективность своих ручных и компьютеризованных операций. Автоматизация неэффективных деловых процедур приводит к неэффективным компьютерным системам. Действительно ли руководители готовы изменить процедуры своей деятельности или же эти изменения чересчур нарушат их деловую практику?

После того, как вы уяснили функции подразделений, вы должны объяснить их руководителям, как корпоративная сеть может кардинально изменить и улучшить их работу. Необходимо потратить большое количество времени, чтобы объяснить, как сеть может изменить ведение дел на предприятии.

При этом необходимо пользоваться терминами, понятными руководителям. Общей ошибкой является употребление технических терминов, а не производственных. Руководителей не волнуют преимущества 100 Мбитной сети FDDI по сравнению с 16 Мбитной сетью TokenRing. Им нет дела до разницы между обнаружением коллизий и передачей токена. Однако их беспокоит, сможет ли сеть помочь им производить больше телевизоров, продавать больше автомобилей, уменьшить страховые запасы сырья. Объясните, что руководители отделов и менеджеры теперь смогут уменьшить время ожидания потенциальных заказчиков за телефоном. А, например, руководителю издательской фирмы необходимо объяснить, что, имея сеть, сотрудники могут не только передавать текст, но также и звуки, и изображение. Информация будет передаваться как единое целое, а не как разрозненный набор чисел, слов, картинок. Использование конкретных содержательных примеров поможет деловым людям понять, какое влияние сеть окажет на организацию работ и на производительность.

Построение бизнес-модели предприятия - не простая задача. Это объясняется дефицитом технических специалистов, понимающих производственные проблемы, и наоборот, отсутствием понимания технических аспектов у многих управленцев. Технические специалисты и управленческие работники должны взаимно обучаться для того, чтобы построить эффективную и полезную корпоративную сеть.

Построение бизнес-модели поможет также получить поддержку руководства предприятия, так как покажет, что разработчики сети понимают и производственные моменты предприятия, а не только технические. Общая стратегия заключается в том, чтобы найти руководителя или менеджера, который увидит выгоду для себя от внедрения корпоративной сети. Этот человек потом будет выступать как ваш "спонсор" перед высшим руководством.

**10.2.2.2. Работа с пользователями сети**

Корпоративная сеть строится для удовлетворения производственных потребностей. Конечные пользователи будут работать с ней каждый день, поэтому обязательно нужно сделать, чтобы им было удобно работать с сетью. К сожалению, о конечных пользователях часто забывают во время планирования, проектирования и установки корпоративной сети. И когда сеть полностью установлена, пользователи начинают говорить "Она не работает" или "Она всегда отказывает". Они говорят, что печать теперь занимает больше времени, чем раньше, и что сеть им не нравится вообще. Вместо того, чтобы использовать сеть как инструмент, они смотрят на нее как на помеху.

Одним из главных условий успешной работы является общение. Как правило, сотрудники отдела автоматизации слишком мало разговаривают не только друг с другом, но и с конечными пользователями. Конечные пользователи часто больше других знают о компьютерных системах, и именно они создают наибольшую нагрузку на компьютеры. Так как они работают с сетью каждый день, то часто обладают наилучшим пониманием того, как взаимосвязаны деловые функции и компьютерные системы. Обязанностью сотрудников отдела автоматизированных информационных систем является выбор такого оборудования, которое наилучшим образом удовлетворяет потребностям конечного пользователя, а не просто самой новой, самой быстрой продукции.

Необходимо назначить по крайней мере одного конечного пользователя, ответственного за связь между пользователями и отделом автоматизации, чтобы пользователи были уверены, что их интересы кто-то представляет. Этот "связной" должен периодически встречаться с другими пользователями, оповещать их о ходе разработки и служить постоянной обратной связью.

Недовольство пользователей иногда основывается на неверных, часто завышенных, ожиданиях. Необходимо дать пользователям реалистичную картину того, что сеть может делать. Необходимо объяснить, какое ожидается среднее время простоев из-за отказов. (Во время анализа требований необходимо выяснить, какое время простоев пользователи считают допустимым.) Нужно честно рассказать пользователям об этапности установки сети. Часто сотрудники отдела автоматизированных информационных систем не говорят о том, что, например, при установки первой очереди корпоративной сети связь с компьютерами VAX и не планировалась, и что она вступит в строй только через год. Пользователи, не предупрежденные об этом, будут очень разочарованы.

Необходимо иметь ввиду, что на предприятии имеется две группы сотрудников, которые будут работать с корпоративной сетью: сотрудники отдела автоматизации, которые будут поддерживать сеть и управлять ею, и конечные пользователи, которые будут использовать ее как инструмент. Цели и потребности этих двух групп часто противоположны. В то время, как конечные пользователи заинтересованы в расширении функций и сервисов сети, сотрудники отдела автоматизации больше озабочены простотой эксплуатации и поддержания надежной работы сети. И те, и другие требуют обучения, хотя и по разным причинам, и по разным программам. Конечных пользователей надо обучать, чтобы работа в корпоративной сети не показалась им странной и неудобной, сотрудников отдела автоматизации необходимо обучать обслуживанию новой современной техники, чтобы они не боялись потерять в будущем работу и не противились внедрению корпоративной сети.

Нужно иметь ввиду, что большинство людей не любит перемен. Они привыкли к используемым программам и компьютерам, многие не хотят отказываться от твердо укоренившихся в памяти приемов. При разработке взаимодействия пользователя с будущей системой будет полезно промоделировать существующее взаимодействие при условии, что оно будет иметь смысл в новом окружении. Необходимо также помнить, что нельзя уменьшать существующие функциональные возможности системы - их можно только наращивать.

**10.2.3. Построение технической модели**

После разработки бизнес-модели предприятия и определения того, какие процедуры требуют изменения или улучшения, необходимо построить техническую модель сети. Техническая модель описывает в достаточно общих терминах, какое компьютерное оборудование нужно использовать, чтобы достичь целей, определенных в бизнес-модели. Чтобы построить техническую модель, нужно проанализировать существующее оборудование, определить системные требования, оценить сегодняшнее и завтрашнее состояния техники.

**10.2.3.1. Инвентаризация существующего оборудования и приложений**

Не многие предприятия могут себе позволить роскошь построения корпоративной сети с нуля. Большинство же вместо этого строят свои компьютерные системы с использованием имеющегося оборудования, которое уже эксплуатируется какое-то время. Необходимо оценить, какое существующее оборудование продолжает иметь стратегическое значение для предприятия, а какое может быть безболезненно списано. Однако сначала необходимо все же определить, что же вы имеете.

Для каждого отдела и офиса необходимо провести инвентаризацию существующего компьютерного оборудования и выяснить, какое действительно используется. Например, действительно ли еще используются 286-е IBMPC-совместимые персональные компьютеры, сколько их? Сколько имеется других компьютеров? Выполняются ли какие-либо приложения мейнфреймом IBM 370? Какие используются типы сетевого оборудования и протоколов, а также сетевых операционных систем: SNA, IPX/SPX, TCP/IP, NetWare, WindowsforWorkgroups, Unix, WindowsNT и т.п. Где это оборудование находится, и в каком оно состоянии? Есть ли экономический смысл продолжать эксплуатировать эту технику или же более эффективно перейти на новую технику? Например, если в программное обеспечение мейнфрейма вложено значительное количество денег, и система работает без проблем, то, очевидно, нет смысла ее заменять. В этом случае корпоративная сеть должна уметь взаимодействовать и с этим мейнфреймом.

После того, как проведена ревизия оборудования, необходимо проделать то же самое с приложениями. Необходимо выяснить:

* Какие приложения используются в каждом офисе и отделе, сколько людей пользуются каждым приложением, и трафик какой интенсивности они создают?
* Могут ли эти приложения быть использованы в корпоративной сети?
* Где хранятся эти приложения и каким образом пользователи смогут получать к ним доступ в новой операционной среде?
* Существуют ли более эффективные приложения аналогичного назначения и захотят ли сотрудники перейти на них?

Инвентаризация - очень скучная процедура, но тем не менее ее нужно осуществить, чтобы точно знать, чем вы располагаете и что нужно приобрести. Для такой большой системы, как корпоративная сеть, очень важно, чтобы каждый элемент, будь то кабель или плата памяти, был промаркирован и учтен. Процесс инвентаризации можно и нужно автоматизировать. Существуют программы, которые могут автоматически исследовать состав аппаратного и программного обеспечения уже работающих в сети компьютеров.

В общем случае такие программы могут выяснить тип CPU, имеющуюся память, тип диска и свободное пространство на нем, имеющиеся дополнительные контроллеры - такие как сетевые адаптеры, факс-модемы и т.п. Для программного обеспечения можно узнать наименование и версию приложений, версии операционной системы, установленные сетевые драйверы.

Программы-исследователи создают базу данных подобной информации, которая может использоваться для справок или при устранении неисправностей. Администратор может периодически запускать исследовательскую программу и получать недельную, месячную или квартальную справку о текущем состоянии сетевых ресурсов.

**10.2.3.2. Определение системных требований**

После инвентаризации существующей вычислительной системы необходимо определить требования к новой системе. Для определения технических параметров сети рассматривайте системные требования не с технической точки зрения, а с позиций руководителей, менеджеров и конечных пользователей.

Для выяснения системных требований необходимо ответить на следующие вопросы:

* Что нужно соединять? Требуется ли сотрудникам какого-либо подразделения общаться с небольшим (большим) количеством людей в пределах небольшой территории или же им нужно общаться с небольшим (большим) количеством людей в пределах географически обширной области? Объем и распределение трафика поможет определить требуемую мощность компьютеров, а также типы и скорости коммуникационного оборудования и сервисов.
* Что из существующего аппаратного и программного обеспечения будет использоваться в новой системе? Какие системы нужно оставить в разрабатываемой корпоративной сети? Нужно ли эти системы соединять в сеть? Будут ли существующие системы нормально работать в новой сети? Существуют ли какие-либо стандарты предприятия, существуют ли преобладающие приложения? Какое оборудование и приложения нужно добавить, чтобы достигнуть поставленных производственных целей?
* Какие объемы информации будут передаваться по сети? Объем передаваемой информации определяет требуемую пропускную способность сети. По корпоративной сети будет передаваться больше или меньше информации? Определите это подсчетом количества пользователей сети, среднего количества выполняемых транзакций в день каждым из пользователей и среднего объема транзакции. Такой подсчет поможет определить технологию доступа к среде передачи данных (Ethernet, FDDI, ...) и требования к глобальным сервисам.
* Какое время реакции сети является приемлемым? Будут ли пользователи ждать одну секунду, полсекунды или две секунды? Такие измерения помогут определить требования к скорости оборудования, приложений и коммуникационных связей.
* В течение какого времени сеть существенно необходима для работы предприятия? Нужна ли сеть 24 часа в день и 7 дней в неделю или же только в течение 8 часов в день и 5 дней в неделю? Нужно ли увеличить сегодняшние параметры использования сети?
* Какие требования предъявляются к среднему времени устранения неисправностей? Как отражаются операции по обслуживанию и ремонту сети на эффективности ведения дел предприятием? Потеряет ли предприятие 5 миллионов долларов или же 100 тысяч долларов, если сеть будет неисправна в течение одного часа? Каков будет ущерб от простоя сети в течение двух часов?
* Каков планируемый рост системы? Каков текущий коэффициент использования сети и как он может измениться в течение ближайших 6 месяцев, одного года, двух лет? Даже если вы тщательно спланировали сеть, но не учли возможности ее роста и развития, то системные требования придется изменить и увеличить. Рост сети нужно планировать заранее, а не просто реагировать на фактический рост ее нагрузки.

**10.2.3.3. Разработка технической модели**

После того, как системные требования определены, можно описать техническую модель корпоративной сети. На этом этапе нужно определить, каким образом предполагается удовлетворить производственные требования с технической точки зрения. Большинство проектировщиков сетей хорошо знакомы с методами разработки технической модели.

Прежде всего подвергните анализу существующую систему. Что нужно из нее сохранить, а что отбросить? Будут ли согласовываться различное сетевое оборудование, операционные системы и приложения, используемые сегодня?

После этого оцените доступную технологию. Что на сегодня является технической вершиной? Хотите ли вы использовать самую передовую технику и технологию? Нужна ли вам на самом деле самая передовая техника? Например, если вы строите сеть для отдела закупок, то технология FastEthernet очевидно будет излишней, даже если в отделе обрабатывается большое число документов. В то же время в инженерном отделе технология FastEthernet будет более целесообразна, так как там имеют дело с большими файлами. Можно также рассмотреть целесообразность сети Ethernet с отдельными сегментами для каждого пользователя. Построение хорошей сети означает постоянное сопоставление ваших желаний и потребностей предприятия. Нужно использовать только такие технические решения, которые необходимы.

Тщательно обдумайте выбор между передовой технологией и технологией, проверенной временем. Например, Ethernet и TokenRing являются проверенными технологиями, а ATM - сравнительно новой. Не многие проектировщики хорошо с ней знакомы, а капиталовложения необходимо сделать значительные.

Далее нужно оценить, какое семейство технических средств удовлетворяет производственным потребностям. Например, нужны ли вашей сети мосты и маршрутизаторы? (Это зависит от интенсивности трафика, количества сегментов, распределения сегментов и уровня квалификации администраторов.) Будете ли вы использовать смешанную топологию или топологию звезды? (Это зависит от структуры и организации вашего предприятия.) Какие коммуникационные связи и с какими скоростями нужны для вашей сети? (Это зависит от того, что будет передаваться по сети, в том числе от того, хотите ли вы комбинировать передачу голоса и данных.) Для принятия подобных решений вы должны узнать сравнительные характеристики различных технологий и технических средств, а также тщательно проанализировать интенсивность и типы передаваемых по сети данных.

Проектировщик также должен обеспечить нужный набор функций и требуемое время доступности сети. Например, если сеть должна быть доступна по ночам и в выходные дни, в ответственных файл-серверах нужно предусмотреть избыточные диски и источники бесперебойного питания. Необходимо решить, достаточно ли применение способа зеркального отображения дисков или требуется использовать дисковый массив. Какой запас аккумуляторов нужен для источников бесперебойного питания?

Далее нужно выяснить, какие технологии и технические средства станут доступными в ближайшее время, а также каковы долгосрочные перспективы этих новшеств. Оцените, сможет ли проектируемая сеть принять завтрашние технологические новинки.

Искусство проектировщика заключается в оценке имеющихся на сегодня решений, предвидении того, что станет доступным завтра, и объединении этих решений в элегантную и эффективную сеть. Например, сегодня неэкранированная витая пара категории 5 устраивает практически все протоколы локальных сетей, в том числе и АТМ на 155 Мб/с. Поэтому можно сделать несложный вывод о том, что внутри зданий на вашем предприятии лучше использовать витую пару, а не коаксиальный кабель.

**10.2.4. Построение физической модели**

После того, как для сети выбрана техническая модель, необходимо оценить, насколько она удовлетворяет производственным требованиям. Нужно вернуться к бизнес-модели и сопоставить ее требования с техническими решениями. Вряд ли технические решения будут полностью удовлетворять требованиям бизнес-модели, но к этому надо стремиться.

Например, если на вашем предприятии сотрудники часто перемещаются из отдела в отдел, то требованием бизнес-модели является высокая мобильность. Техническая модель должна в таком случае обеспечивать быстрое присоединение и отсоединение рабочей станции.

После того, как вы убедились, что техническая модель соответствует требованиям, нужно построить физическую модель. Физическая модель конкретизирует специфику технической модели. Физическая модель является очень подробным описанием сети, в то время как техническая модель использует для ее описания более общие термины.

Хотя обычно проектировщик детально знаком с каждой технологией, характеристики конкретных продуктов он знает только в общих чертах. Этап физического моделирования требует более детального знакомства с имеющимися продуктами. Теперь проектировщик должен оценить свойства и функции подходящих продуктов и решить, какие из них наилучшим образом удовлетворяют требованиям разрабатываемой системы. После определения доступных продуктов проектировщик должен получить их "в натуре", чтобы оценить, насколько их характеристики соответствуют характеристикам, объявленным производителем. Для того, чтобы определить подходящие продукты, проектировщику нужно воспользоваться опытом консультантов, знакомых из отделов автоматизации других предприятий, системных интеграторов, а также такими вторичными источниками информации, как обзоры продуктов в журналах и т.п.

На стадии физического моделирования проектировщик должен точно описать, какие компоненты нужны, в каком количестве, где они будут расположены и как эти компоненты соединяются друг с другом в корпоративную сеть. Этот этап завершается разработкой технического задания.

**10.2.5. Разработка технического задания**

После того, как были разработаны бизнес-модель и техническая модель, необходимо разработать техническое задание. Техническое задание базируется в основном на информации, собранной на этапе анализа требований. В сущности, техническое задание говорит интегратору: "Вот производственные проблемы нашей фирмы и вот проблемы нашей вычислительной сети. Каким образом вы как проектировщик сети и интегратор можете их решить? Какие проблемы вы можете решить сами, и для решения каких проблем вам необходимо заключить контракт с другими исполнителями?" Это техническое задание необходимо раздать нескольким (по крайней мере трем) сетевым интеграторам, чтобы выяснить их мнение о том, как нужно спроектировать вашу корпоративную сеть. Сложность проектируемой корпоративной сети и опыт вашей фирмы определят глубину и широту охвата вашего технического задания. Техническое задание может иметь объем и 5, и 50, и более страниц.

Как и на этапе анализа требований, вы можете подготовить техническое задание сами или с помощью консультанта. Если вы привлекли консультанта, то, очевидно, что техническое задание будут более полными и завершенными, чем если бы вы его готовили сами. Техническое задание, подготовленное консультантом, должно включать полное описание проекта сети и список оборудования, которое нужно купить, так что поставщикам останется только назвать цену. Если вы сами готовите техническое задание, то вы можете больше полагаться на предложения системных интеграторов и меньше думать о стоимости.

В приведенном ниже примере технического задания подразумевается, что вы пользуетесь услугами консультантов и интеграторов при разработке вашей корпоративной сети. Даже если вы этого не делаете, все равно вам следует написать техническое задание для себя, и не только для обоснования решений перед руководством, но и для того, чтобы помочь уяснить себе все технические проблемы, требующие решения.

**Пример технического задания**

*Введение*

Во введении дается обзор содержания технического задания. Оно коротко описывает вашу компанию, ее сеть, производственные цели и примерные этапы установки сети:

1. Обзор - является ли данное техническое задание документом, полностью описывающим вопросы планирования, реализации, поддержки и управления корпоративной сетью? Или в нем исключены некоторые вопросы, например, касающиеся кабельной системы? Здесь вы должны также указать, что вы хотите от системных интеграторов: участие в поставках, инсталляции, конфигурировании, в пуско-наладке, тестировании, управлении и/или поддержке.
2. План - включает сроки выполнения этапов реализации технического задания: срок выпуска технического задания, крайний срок получения ответа, срок оценки предложений системных интеграторов, срок проведения презентаций и демонстраций интеграторами своих предложений, срок заключения контракта, сроки выполнения работ по установке сети.
3. Конфиденциальность - в техническое задание можно включить пометку о конфиденциальном статусе, так как это техническое задание может содержать информацию, которой очень интересуются ваши конкуренты.
4. Описание предприятия и его целей - необходимо определить в общих терминах, чем занимается ваше предприятие. Чтобы правильно построить вашу сеть, интегратор должен вполне представлять ваш бизнес. Этот раздел должен также содержать данные о количестве сотрудников вашего предприятия и местах расположения филиалов, которые нужно соединить сетью.

*Цели создания сети на вашем предприятии*

Этот раздел описывает, как вы представляете использование сети для решения производственных проблем вашего предприятия. Здесь следует также указать, имеется ли какой-либо проект автоматизированной информационной системы, связанный с данной сетью, но не отраженный в данном техническом задании:

* Требования к сети. Здесь отмечаются общие цели создания сети, полученные из анализа потребностей вашего предприятия. Например, предприятию нужны коммуникации со всеми подразделениями и филиалами, при этом нужно включить в сеть существующее оборудование и используемые в настоящее время приложения, связь персоналок с миникомпьютерами, новые приложения, использующие механизм транзакций; сеть должна быть проста в использовании и управлении, обладать высокой отказоустойчивостью и производительностью.
* Используемое в настоящее время оборудование. Перечислите типы и количество компьютеров каждого типа в каждом подразделении и филиале. Включите все данные - от состава оборудования мейнфреймов и устройств межсетевой связи до операционных систем персональных компьютеров и их приложений. Возьмите эту информацию из списка, составленного на этапе анализа требований.
* Кабельная система. Укажите, какие кабели проложены на предприятии. Будет ли корпоративная сеть использовать существующие кабели, или системный интегратор должен предложить новую кабельную систему?
* Соответствие стандартам предприятия. Опишите стандарты предприятия на оборудование и программное обеспечение. Например, ваше предприятие может использовать компьютеры CompaqSystemPro как стандартные для файловых серверов. Укажите, может ли интегратор делать исключения из этих стандартов.
* Этапы внедрения. Большинство корпоративных сетей внедряются поэтапно из-за их сложности и ограничений финансирования. Необходимо описать порядок включения в корпоративную сеть подразделений предприятия.

*Требования к предложениям системных интеграторов*

В этом разделе необходимо указать, что бы вам хотелось увидеть в предложениях системных интеграторов. Это поможет вам сравнить в дальнейшем предложения разных системных интеграторов:

* Технические требования. Здесь можно указать требуемый перечень характеристик (название, номер модели, цена и т.п.) для программного и аппаратного обеспечения. Можно попросить также представить схему проектируемой сети, а также сроки поставки оборудования.
* Сведения о интеграторах. Наряду с техническими требованиями можно запросить сведения о самих интеграторах-поставщиках. В этом случае интеграторы должны включить в свои предложения описания своих предыдущих разработок, данных о сотрудниках, которые будут проектировать и устанавливать сеть, информацию о финансовом состоянии фирмы-интегра-тора, а также указать клиентов, для которых выполнялись аналогичные работы.

*Сетевые цели*

В этом разделе технического задания описывается ваше видение сети:

* Требования к квалификации системного интегратора. Опыт системного интегратора в построении корпоративных сетей, особенно в той области, в которой работает ваше предприятие, является очень существенным. К другим критериям относятся качество проекта, услуг по установке, обучению и поддержке. Сведения о финансовой стабильности и конкурентоспособных ценах также очень важны. Кроме того, укажите, хотите ли вы работать сразу с несколькими интеграторами, или вам нужен один интегратор в качестве генерального подрядчика.
* Требования к техническим аспектам сети. Этот раздел должен основываться на вашей технической модели. Какие компоненты содержит сеть? Например, хотите ли вы использовать в качестве устройства межсетевого взаимодействия маршрутизаторы, или интеграторы должны предложить какой-либо другой вариант? Если к разработке технического задания или к анализу требований привлекались консультанты, этот раздел будет достаточно подробным. Если же вы сами готовили техническое задание, эта часть может оказаться весьма общей, так как вы обращаетесь к интеграторам за экспертной помощью в создании сети, отвечающей вашим потребностям.
* Требования к приложениям. Для каждого типа приложения сформулируйте прежде всего следующие требования. Какое приложение вам нужно: электронная почта, база данных, средства автоматизации документооборота или средства коммуникации? Вам требуются приложения, использующие графический пользовательский интерфейс? Должны ли эти приложения работать в разных операционных системах и в разных сетевых средах? Намерены ли вы использовать шлюзы для интеграции пакетов других производителей? Какова приемлемая скорость?
* Требования к средствам коммуникаций. Нужно ли пользователям взаимодействовать с другими офисами, мейнфреймами, миникомпьютерами или другими источниками информации в режиме on-line. Должны ли они стать клиентами публичной электронной почты, такой как MICMail, или же системы поставки оперативной коммерческой информации. Определите, есть ли у вас предпочтение по отношению к определенной клиентской операционной системе, к сетевой операционной системе или к пользовательскому интерфейсу.

*Системные спецификации*

Раздел системных спецификаций технического задания описывает технические спецификации компонентов сети:

* Уровень отделов. Техническое задание описывает спецификации для рабочих станций, принтеров, файл-серверов, приложений, утилит печати, коммуникационных утилит и утилит электронной почты. Если требуется высокая отказоустойчивость, то опишите желаемые компоненты для ее обеспечения, такие как зеркальные серверы, источники бесперебойного питания и устройства архивирования. Техническое задание должно определить требуемое среднее время безотказной работы компонентов сети.
* Уровни кампусов и предприятия. Техническое задание определяет требуемые характеристики мостов, маршрутизаторов, модемов, факс-серверов, шлюзов к миникомпьютерам и мейнфреймам, коммуникационных программ и программного обеспечения широкого применения, такого как электронная почта. Должно быть определено приемлемое среднее время доступности сети.

*Сетевые проблемы*

Этот раздел содержит информацию из физической модели сети:

* Сетевая операционная система. В техническом задании следует запросить системного интегратора определить тип корпоративной сетевой операционной системы, если на предприятии она не стандартизована. Системный интегратор должен обосновать свой выбор.
* Сетевое и коммуникационное оборудование. Будут ли в корпоративной сети использоваться методы доступа к среде, отличные от тех, которые используются в "позвоночнике" сети? В техническом задании следует запросить обоснование выбора методов доступа Ethernet, TokenRing, FDDI или иных, с обсуждением скорости передачи данных и возможного влияния технического прогресса в будущем.
* Концентраторы. Попросите интегратора описать характеристики предлагаемых концентраторов. Интегратор должен обосновать выбор производителя, обращая особое внимание на наращиваемость, управляемость и цену.
* Межсетевое взаимодействие. Каким образом будут соединены сети отделов? Техническое задание требует от интегратора обоснования выбора метода межсетевого взаимодействия, а также выбора изготовителя оборудования.
* Доступ к миникомпьютерам и мейнфреймам. В техническом задании должно быть определено, какое оборудование используется в настоящий момент для доступа к миникомпьютерам и мейнфреймам, также должен содержаться вопрос о том, какие другие варианты могут быть предложены. Предложения должны полностью определять программные и аппаратные средства, необходимые для хост-машин и для рабочих станций. Должно быть определено, сколько одновременных сессий может поддерживаться и каковы ограничения.
* Глобальные связи. Какие глобальные связи будут использованы для соединения с другими офисами? В предложении должны быть описаны требования для каждого филиала, причем они должны быть обоснованы по характеристикам пропускной способности, стоимости и управляемости. Будут ли они публичными или частными сервисами?
* Удаленный доступ. В предложении интегратора должен быть описан способ удаленного доступа. Это особенно важно, если сотрудники предприятия используют ноутбуки и лаптопы или же много перемещаются и в то же время нуждаются в доступе к сети. Требуются ли только входящие соединения удаленного доступа или и исходящее также? Будут ли сотрудники звонить по публичным сетям данных, таким как BT/Tymnet? Будут ли сотрудники использовать такие публичные компьютерные сети, как CompuServe или Internet? Будут ли они использовать сети других корпораций?
* Безопасность. Какую схему безопасности необходимо использовать, чтобы защитить сеть от случайного или преднамеренного вторжения? Какая защита от вирусов предусмотрена в сети?
* Доступность сети. В течение какого времени сеть должна быть доступной? Какая избыточность оборудования предусматривается для обеспечения надежной работы сети в течение всего периода доступности? Какие требования предъявляются к среднему времени между отказами для наиболее ответственных компонентов сети? В течение какого времени поставщики должны устранять проблемы с их оборудованием?
* Управление сетью. Каким образом будет управляться сеть? Как будут устраняться неисправности? Какой инструментарий требуется для этого? Будет ли управление осуществляться удаленно? Сколько людей потребуется для управления сетью?
* Масштабируемость сети. В предложениях должно быть указано, как сеть сможет удовлетворять потребности предприятия в будущем. Для этого в техническом задании должна быть дана оценка возможного роста предприятия и корпоративной сети на год вперед. В предложениях, в свою очередь, должно быть указано, какое максимальное количество пользователей, серверов, мостов, маршрутизаторов и шлюзов сможет обслуживать данная сеть без значительного уменьшения производительности.

*Обслуживание сети*

Обслуживание включает в себя:

* Обслуживание аппаратуры. Интегратор должен описать, где он будет заниматься обслуживанием аппаратуры. Если у него нет центра обслуживания в каждом филиале вашего предприятия, то будет ли он заключать субподряд на обслуживание с каким-либо другим интегратором? Определите правила этих взаимоотношений. Должен ли интегратор сам заботиться о наличии самых важных запасных частей в центрах обслуживания? Сколько технических специалистов имеется в каждом центре обслуживания? Каково гарантированное время ремонта? Сможет ли интегратор осуществлять удаленную диагностику проблем с аппаратурой в ваших филиалах? Каковы гарантии на предлагаемое оборудование?
* Поддержка программного обеспечения. В чем заключаются гарантии на используемое программное обеспечение? Какова стоимость поддержки программного обеспечения после истечения срока гарантии? Доступен ли исходный код? И опять - сможет ли интегратор выполнять удаленную диагностику и разрешение проблем программного обеспечения? Сколько сотрудников фирмы-интегратора могут поддерживать программное обеспечение в каждом из мест расположения филиалов?
* Надежность. Будут ли различные изменения, добавления и перемещения требовать остановки сети? Если да, то на сколько? Можно ли этим заниматься ночью или в выходной день? Имеется ли возможность модульного наращивания сети без возникновения помех для работы пользователей?

*Обучение*

* Обзор. Определите, сколько сотрудников предприятия нуждается в обучении, каков уровень их подготовки, какие виды учебы им нужны, может ли обучение проводиться с отрывом от производства или нет. Оцените количество часов подготовки. Интегратор в своих предложениях должен указать, каким образом он собирается обучать конечных пользователей и сотрудников отдела АИС и сколько это будет стоить.
* Документация. Определите количество инструкций пользователя, которые должны быть поставлены без дополнительной оплаты. Какова стоимость дополнительного количества инструкций пользователя? Каким образом будут документироваться особые функции вашей сети и кем?
* Оперативная поддержка. В техническом задании необходимо сделать запрос о стоимости оперативной поддержки интегратором и стоимости оперативной диагностики.

*Инсталляция сети*

Проблемы инсталляции включают план поставок оборудования, физические требования к оборудованию и стоимость каждого этапа инсталляции:

* План поставок. В техническом задании необходимо запросить предварительный план поставок для предлагаемой системы.
* Физическое размещение. Предложения интегратора должны описывать физическое размещение оборудования, включая требования к необходимой площади, электрической проводке, типам электрических разъемов, параметрам воздушной среды, теплообмена, возможной нагрузки на этаж. Для каждого используемого оборудования в предложениях должно быть указано: его размеры, вес, нагрузка на пол, потребляемые ток, напряжение и мощность, типы электрических разъемов, выделение тепла в час, требования к температуре и влажности помещения.
* Поэтапная стоимость. В техническом задании необходимо запросить поэтапную стоимость инсталляции сети с учетом стоимости работы по ночам и в выходные дни.

*Оценка предложений интеграторов*

В техническом задании необходимо указать критерии, которыми вы будете пользоваться при выборе оборудования и программного обеспечения. Критерии для оценки оборудования могут включать: стоимость, скорость, поддержку при установке, удобства работы, надежность, масштабируемость, данные о производителе, стоимость обслуживания. Критерии для оценки программного обеспечения могут включать: стоимость, надежность, поддержку, опыт интегратора по данному продукту, состав документации, быстродействие.

### 10.3. Средства анализа и оптимизации сетей

**10.3.1. Анализаторы протоколов**

В ходе проектирования новой или модернизации старой сети часто возникает необходимость в количественном измерении некоторых характеристик сети таких, например, как интенсивности потоков данных по сетевым линиям связи, задержки, возникающие на различных этапах обработки пакетов, времена реакции на запросы того или иного вида, частота возникновения определенных событий и других характеристик.

Для этих целей могут быть использованы разные средства и прежде всего - средства мониторинга в системах управления сетью, которые уже обсуждались в предыдущих разделах. Некоторые измерения на сети могут быть выполнены и встроенными в операционную систему программными измерителями, примером тому служит компонента ОС WindowsNTPerformanceMonitor. Даже кабельные тестеры в их современном исполнении способны вести захват пакетов и анализ их содержимого.

Но наиболее совершенным средством исследования сети является анализатор протоколов. Процесс анализа протоколов включает захват циркулирующих в сети пакетов, реализующих тот или иной сетевой протокол, и изучение содержимого этих пакетов. Основываясь на результатах анализа, можно осуществлять обоснованное и взвешенное изменение каких-либо компонент сети, оптимизацию ее производительности, поиск и устранение неполадок. Очевидно, что для того, чтобы можно было сделать какие-либо выводы о влиянии некоторого изменения на сеть, необходимо выполнить анализ протоколов и до, и после внесения изменения.

Анализатор протоколов представляет собой либо самостоятельное специализированное устройство, либо персональный компьютер, обычно переносной, класса Notebook, оснащенный специальной сетевой картой и соответствующим программным обеспечением. Применяемые сетевая карта и программное обеспечение должны соответствовать топологии сети (кольцо, шина, звезда). Анализатор подключается к сети точно также, как и обычный узел. Отличие состоит в том, что анализатор может принимать все пакеты данных, передаваемые по сети, в то время как обычная станция - только адресованные ей. Программное обеспечение анализатора состоит из ядра, поддерживающего работу сетевого адаптера и декодирующего получаемые данные, и дополнительного программного кода, зависящего от типа топологии исследуемой сети. Кроме того, поставляется ряд процедур декодирования, ориентированных на определенный протокол, например, IPX. В состав некоторых анализаторов может входить также экспертная система, которая может выдавать пользователю рекомендации о том, какие эксперименты следует проводить в данной ситуации, что могут означать те или иные результаты измерений, как устранить некоторые виды неисправности сети.

**10.3.2. Программные пакеты имитационного моделирования**

Анализаторы протоколов незаменимы для исследования реальных сетей, но они не позволяют получать количественные оценки характеристик для еще не существующих сетей, находящихся в стадии проектирования. В этих случаях проектировщики могут использовать средства имитационного моделирования, с помощью которых разрабатываются модели, воссоздающие информационные процессы, протекающие в сетях.

Существуют специальные, ориентированные на моделирование вычислительных сетей программные системы, в которых процесс создания модели как таковой вообще отсутствует. Такие программные системы сами генерируют модель сети на основе исходных данных о ее топологии и используемых протоколах, об интенсивностях потоков запросов между компьютерами сети протяженности линий связи, о типах используемого оборудования и приложений. Программные системы моделирования могут быть узко специализированными и достаточно универсальными, позволяющие имитировать сети самых различных типов. Качество результатов моделирование в значительной степени зависит от точности исходных данных о сети, переданных в систему имитационного моделирования.

Программные системы моделирования сетей - инструмент, который может пригодиться любому администратору корпоративной сети, особенно при проектировании новой сети или внесении кардинальных изменений в уже существующую. Продукты данной категории позволяют проверить последствия внедрения тех или иных решений еще до оплаты приобретаемого оборудования. Конечно, большинство из этих программных пакетов стоят достаточно дорого, но и возможная экономия может быть тоже весьма ощутимой.

Программы имитационного моделирования сети используют в своей работе информацию о пространственном расположении сети, числе узлов, конфигурации связей, скоростях передачи данных, используемых протоколах и типе оборудования, а также о выполняемых в сети приложениях.

Рынок систем имитационного моделирования представлен продуктами различного класса - от простых программ, предназначенных для установки на персональном компьютере, до мощных сетевых пакетов. Стоимость систем высшего класса может доходить до нескольких десятков тысяч долларов. Так, популярная система COMNETIII компании CACIProducts составляет $35000 - $40000.

**10.3.3. Натурные эксперименты**

Если для задания информации о топологии сети не нужно иметь реальную сеть, то для сбора исходных данных об интенсивности источников сетевого трафика могут потребоваться измерения на пилотных сетях, представляющих собой натурную модель проектируемой сети. Эти измерения могут быть выполнены различными средствами, в том числе и с помощью анализаторов протоколов. Имеются и специальные программные системы, которые автоматизируют процесс сбора данных о временных характеристиках работы приложений. Примером может служить InfoVista - система для расчета времен реакции приложений в локальных сетях (Network, July, с. 18). Эта система работает совместно с системой управления сетью Transcend компании 3Com. Transcend собирает данные о работе сети от агентов RMON, а InfoVista рассчитывает времена для различных приложений, работающих в сети. Если на основании этих измерений нужно изменить режимы работы коммуникационного оборудования, то это можно сделать опять же с помощью системы Transcend, которая управляет коммуникационным оборудованием 3Com. Компания InfoVista предлагает адаптацию своей системы измерения и для других платформ управления.

Помимо получения исходных данных для имитационного моделирования пилотная сеть может использоваться для решения самостоятельных важных задач. Она может дать ответы на вопросы, касающиеся принципиальной работоспособности того или иного технического решения или совместимости оборудования. Натурные эксперименты могут потребовать значительных материальных затрат, но они компенсируются высокой достоверностью полученных результатов.

### 10.4. Взаимодействие с сетевыми интеграторами

Реализация проекта может осуществляться либо сторонней организацией, привлеченной специально для выполнения проекта, либо собственными силами организации, либо в тесном контакте заказчика с исполнителем сетевых проектов - сетевым интегратором. Для осуществления крупных сетевых проектов корпорации все чаще прибегают к помощи сторонних организаций. Причина очевидна: в современных условиях ожесточенной конкуренции у менеджеров и администраторов сетей слишком много дел и слишком мало свободных ресурсов. Когда собственный персонал работает с предельной нагрузкой, а с набором квалифицированного персонала имеются сложности, целесообразно прибегнуть к услугам фирм-интеграторов, которые скомпонуют и отладят систему, а иногда и займутся ее эксплуатацией.

Имеется по крайней мере три области, где подряды на сетевые проекты становятся популярными:

1. проектирование и установка систем с целью расширения, модернизации и замены существующих сетей;
2. удаленное управление локальными сетями;
3. разработка сетевых приложений.

Хотя на западном рынке 56% от всей суммы контракта уходило именно на различного рода услуги, а только 44% - на программное и аппаратное обеспечение, в нашей стране большинству заказчиков эти затраты по-прежнему кажутся совершенно излишними.

Уровни сотрудничества со сторонними фирмами могут быть самыми различными - от оказания масштабных консалтинговых услуг, приводящих к коренным изменениям функционирования предприятия, до выполнения небольших конкретных заданий типа инсталляции одного программного продукта.

**10.4.1. Консалтинг, системная и сетевая интеграция**

Среди фирм, занимающихся *консалтингом* в области информационных технологий, до недавнего времени доминировали вовсе не компьютерные фирмы и поставщики компьютерных услуг. Теперь же этот бизнес притягивает многие компьютерные фирмы, которые стремятся в процессе оказания консалтинговых услуг установить тесные отношения с руководителями предприятий, с теми, кто вырабатывает стратегию и принимает решения. Консультант, помогающий определить стратегию фирмы, может предложить свои услуги и в области системной интеграции или проведения работ по информационному обслуживанию. При этом увеличивается вероятность получения заказа на внеконкурсной основе.

К мировым лидерам консалтинга информационных технологий относятся компании IBM, Andersen, ComputerSciences, OracleConsulting, ElectronicDataSystems, Unisys, Hewlett-Packard,DigitalEquipment и некоторые другие компании. Область деятельности компаний, работающих в этом бизнесе, как правило, включает широкий перечень услуг. Например, компания IBM консультирует по применению информационных технологий в финансовой и банковской сферах, в областях страхования, химии, образования, развлечений, здравоохранения, фармацевтики. Кроме того, в ее функции входит поддержка бизнеса, промышленного производства и оптово-розничной торговли, консультации правительственным организациям и оказание юридических услуг. Все ведущие консалтинговые фирмы участвуют в реинжиниринге бизнес-процессов ("реинжиниринг" - радикальная перестройка деловых процессов предприятия).

В России рынок консалтинга только зарождается, однако можно уже назвать несколько отечественных фирм, в задачу которых входит анализ, формализация и структуризация бизнес-задач клиента, например "Метатехнология", IBS и ряд других.

Примером приближения компьютерных фирм к рынку консалтинговых услуг служит альянс известной российской компьютерной фирмы ЛВС и компании PriceWaterhouse, занимающей 16 место в мировом рейтинге компаний, занимающихся консалтингом в области информационных технологий.

Проблемы предприятия или организации могут возникать и решаться на разных уровнях. Одни из них требуют привлечения консалтинговых фирм, другие могут быть разрешены системными или сетевыми интеграторами. В результате оказания консалтинговых услуг могут возникнуть проблемы, относящиеся к системной интеграции, а из этих проблем в свою очередь вырастают задачи сетевой интеграции.

Если задачей является только разработка сети: ее транспортной подсистемы или компьютерной платформы, реализация гетерогенной операционной среды или разработка приложений, работающих с базами данных - это может быть отнесено к сетевой интеграции.

Задачи, требующие комплексного, "системного" подхода, учитывающие взаимосвязи и взаимозависимости разных аспектов функционирования предприятия - это область действия системных интеграторов. В деятельности системного интегратора наряду с техническими вопросами не менее важными является решение вопросов организационного, юридического и управленческого характера. Термин "системный интегратор" в настоящий момент не имеет полной определенности.

Существует несколько моделей работы с сетевыми интеграторами. Не обязательно каждый проект сетевой интеграции должен быть крупным. Напротив, в последнее время в западном бизнесе сетевой интеграции преобладает тенденция *предоставления услуг по прейскуранту*, вместо более традиционного подхода, когда крупные интеграторы систем берут на себя всю работу по созданию и эксплуатации сетевой системы заказчика. Для таких небольших проектов характерны короткие сроки исполнения и низкая стоимость. Услуги по прейскуранту известны как подряд на уровне задач, когда контракт составляется на отдельную работу, такую как инсталляция программы, проводка кабельной системы, оптимизация программного или аппаратного обеспечения, установка сети NovellNetWare или разработка приложения.

Подряд на уровне задач рассматривается как менее рискованный подход, позволяющий организации сохранять контроль, не отдавая все на откуп фирме-интегратору.

**10.4.2. Сопровождение проекта заказчиком**

При работе со сторонней организацией-интегратором очень важно правильно осуществить выбор такой организации и активно участвовать в осуществлении проекта.

Полезно учесть, что по результатам анализа неудачных проектов, причинами такого их завершения чаще всего являются:

1. Недостаточно тесный контакт между интегратором и организацией-заказчиком и пассивное участие в проекте конечных пользователей. Всячески поощряйте общение между участниками проекта (во всяком случае, не препятствуйте ему): менеджеры и пользователи должны знать о выбранном направлении развития проекта. Постоянные консультации помогут избежать досадных недоразумений.
2. Недостаточно четкая постановка задачи. Один из основных источников недоразумений между корпоративными менеджерами и разработчиками решений по организации бизнеса состоит в том, что они по-разному понимают то, к чему следует стремиться. Четкое определение задач в самом начале - например, на этапе анализа требований - позволяет избежать последующих взаимных обвинений в недобросовестности.
3. Плохо составленный проект.
4. Поверхностное руководство, оставляющее неясными вопросы относительно оценки этапов проекта. Определите цели на каждом из этапов, согласуйте время их выполнения (это поможет контролировать процесс реализации проекта в целом) и непременно договоритесь о критериях оценки текущих работ и четких правилах определения момента завершения проекта.

Сопровождение проекта заказчиком должно осуществляться на всех этапах проектирования, начиная от составления технического задания и заканчивая поддержкой эксплуатации системы.

Важно хорошо отслеживать временную последовательность запланированных работ. Реакция на несоответствие между запланированной и реальной датой выполнения очередного этапа должна следовать незамедлительно. Не следует ждать завершения проекта - требуйте отчета в случае поэтапных задержек немедленно.

Еженедельные встречи с системным интегратором позволят вам всегда держать руку на пульсе событий. Во время этих встреч узнавайте о ходе работ, о выполнении субподрядчиками своих договоров и, при необходимости, корректируйте планы.

Каждая встреча должна иметь свою повестку. Лучше всего начинать с подведения итогов (обязательно обратите внимание на то, как выполнялись рекомендации, данные на предыдущей встрече), а затем уже переходите к дальнейшему обсуждению возможных путей решения той или иной проблемы. Кроме того, не забывайте держать руководство в курсе еженедельных встреч; ничто так не помогает завоевать доверие и поддержку на высшем уровне, как осведомленность - проект должен находиться под пристальным вниманием и надзором.

Если в реализации проекта участвуют несколько поставщиков, то на встречах должны присутствовать представители каждого. Вряд ли стоит говорить о том, насколько важны не только совместный контроль над осуществлением проекта, но и совместное выявление потенциальных проблем, а также постоянный обмен идеями и предложениями об улучшениях.

При реализации проекта очень важна степень квалификации как сотрудников интеграционной фирмы, так и фирмы-заказчика. Причем разные фирмы-интеграторы могут по-разному относиться к наличию высокой квалификации у сотрудников, сопровождающих проект со стороны заказчика.

Для "слабого" интегратора ответ на поставленный вопрос очевиден - для него хорош только тот заказчик, который не разбирается в сути проблемы и готов довольствоваться любым решением, лишь бы оно было не очень дорогим и немного работало. Не нужен квалифицированный заказчик и той фирме-однодневке, для которой весьма соблазнительным кажется "впарить" клиенту самое дорогое оборудование, которое есть у него на складе.

Для "сильного" интегратора ответ на этот вопрос требует более тщательного обдумывания, и здесь полезным оказывается деление заказчиков по уровню квалификации на три группы. В первую группу входят заказчики, квалифицированные "на самом деле", то есть такие, квалификация которых признается и системными интеграторами. Вторую группу составляют заказчики, которые сами считают себя квалифицированными, но интеграторы сильно в этом сомневаются. И, наконец, к третьей группе относятся заказчики, которые абсолютно не претендуют на роль сетевых специалистов и дают интегратору "карт-бланш".

Для сильного интегратора работа с действительно квалифицированным заказчиком имеет много преимуществ:

* Работая с грамотным заказчиком, интегратору легче обосновать необходимость неочевидных, но благоприятно влияющих на качество проекта затрат. К таким затратам можно отнести, например, затраты на предпроектное обследование системы, очень кропотливую работу, стоящую на Западе больших денег, а у нас выполняемую редко. Другим примером служат затраты на реализацию не основных, а вспомогательных функций системы, например, обеспечивающих повышенную отказоустойчивость. Неспециалисту трудно объяснить, зачем, приобретая модульный концентратор стоимостью в несколько десятков тысяч долларов, желательно купить еще и программу управления им примерно такой же стоимости - "Ведь он же и без этой программы нормально работает?" Или же доказать, в чем состоят преимущества интеллектуального маршрутизатора перед коммутирующим на лету мостом - "Когда еще этот broadcast-шторм произойдет, может их вообще не бывает, а это только выдумки для накрутки цены?"
* При сравнении разных проектных вариантов у грамотного заказчика помимо стоимости и красоты оформления технических предложений имеются и другие, более объективные, критерии оценки. В случае проведения таким заказчиком тендера, у сильного интегратора появляются хорошие шансы на победу, так как оценка предложений выполняется квалифицированными специалистами.
* Профессионально подготовленный заказчик чаще всего принимает непосредственное участие во всех этапах процесса проектирования - от разработки технического задания до тестирования системы. Отсюда проистекает несколько положительных моментов для интегратора. Во-первых, упрощается работа по постановке задачи, так как никто лучше сотрудников предприятия не знает его специфических проблем, технологии выполнения деловых процедур, требований к системе конечных пользователей. Во-вторых, при совместной работе над подробным техническим заданием, в котором намечаются конкретные пути решения задач, делится ответственность за принимаемые решения между заказчиком и интегратором. К примеру, если в техническом задании в качестве исходных данных фигурируют выделенные цифровые линии для реализации глобальных связей, то это в дальнейшем, при реализации проекта, ограждает интегратора от возможных упреков в том, что экономически выгоднее было бы для данного предприятия использовать сети общего пользования X.25. В-третьих, активный заказчик, познакомившийся со всеми тонкостями проекта, в дальнейшем успешнее эксплуатирует готовую систему, что также работает на репутацию интегратора.

Именно по этим причинам некоторые дальновидные сетевые интеграторы начали проводить различные семинары и курсы, целью которых является не столько продвижение поставляемой продукции на рынок за счет рекламы их потребительских свойств, сколько обучение своих потенциальных клиентов более фундаментальным понятиям - принципам работы оборудования, обоснованию типовых решений, анализу проектных вариантов.

Если же заказчик только мнит себя квалифицированным специалистом, то это может сильно осложнить работу интегратора - заказчик постоянно вмешивается в процесс проектирования, предлагает и навязывает нерациональные решения, которые явно приведут к неработающей или плохо работающей системе, и, соответственно, к снижению репутации интегратора. Это плохой вариант для интегратора, так как малоплодотворные дискуссии, отнимающие много времени, редко приводят к хорошему результату. Хуже может быть только заказчик-"злоумышленник". Такой заказчик подолгу ходит по разным фирмам-интеграторам, подробно обсуждая детали своего будущего проекта, эксплуатируя свою роль потенциального заказчика и "вытягивая" из интеграторов конкретные решения. В результате всей этой бурной деятельности, измучив несколько интеграторов и не заключив договора ни с одним, он собирает полученные решения в общий проект и самостоятельно пытается его реализовать. Ясно, что решения при этом вряд ли получатся работоспособными, но для интегратора это уже не важно - заказ не состоялся.

Третий тип заказчика - это заказчик, честно признающий себя непрофессионалом в области компьютеризации и разрешающий интегратору делать все, что ему хочется для достижения поставленной цели в пределах выделенного бюджета. Такой беззаботный заказчик не мешает, но и не помогает. Интегратор должен потратить много времени на уяснение специфики организации, ее деловых целей, структуры, опрашивая конечных пользователей, оценивая трафик, анализируя приложения. При этом существует большой риск, что из-за недостатка времени не будет учтен какой-нибудь важный фактор.

Нет сомнений, что заказчику всегда желательно иметь некоторую квалификацию, и чем яснее заказчик представляет возможные пути решения своих задач, тем для него это лучше. Грамотный заказчик никогда не свяжется со слабым интегратором, так как быстро поймет, что тот из-за своего уровня просто не может решить поставленную задачу, какие бы благие намерения он не имел. Он также легко распознает квалифицированного, но недобросовестного интегратора, который, преследуя свои корыстные цели, может действовать в ущерб интересам клиента.

Но еще более важна квалификация заказчика при реализации больших проектов, выполняемых на переднем крае технологических достижений. В этом случае даже обращение к сильным интеграторам, обладающим хорошей деловой репутацией, не гарантирует отсутствия проблем.

Фирма IBMConsultingGroup изучила опыт 24 крупных американских компаний и обнаружила, что 68% проектов превысили плановые сроки, 55% превысили плановый бюджет и 88% были переделаны заново. Все эти проекты были задуманы в новой для того времени архитектуре клиент-сервер, и даже у авторитетных интеграционных фирм почти не было опыта работы с этой архитектурой. Кроме новизны, осложнить реализацию проекта могут и его масштабы. Исследовательская фирма SoftwareProductivityResearch, анализируя опыт корпораций США и Великобритании, нашла, что 24% больших программных проектов (объемом свыше 5000 условных функциональных единиц, что соответствует 500 000 операторам процедурного языка программирования) завершаются провалом.

Важным фактором, оказывающим влияние на успех проекта, является степень участия заказчика в разработке. Фирма WiilbernGroup оказывала помощь в спасении 80 сверхбольших и 400 не очень больших проектов. В большинстве неудачных проектов заказчик полностью перекладывал ответственность по принятию всех технических решений на внешнего исполнителя. Так, BankofAmerica хотел создать самую лучшую в мире сеть по управлению трастовыми операциями под названием MasterNet. Этот амбициозный проект с первоначальной стоимостью в 20 миллионов долларов был полностью доверен консультантам и внешним поставщикам. В результате, банк был вынужден отказаться от этого проекта, истратив 60 миллионов долларов. Ему пришлось свернуть почти все трастовые операции, а два высокопоставленных сотрудника, отвечавших в банке за проект, были уволены. Другим характерным примером может служить опыт фирмы WhirlpoolCorp., решившей создать сеть OneCallNet для обслуживания своих покупателей. Работы были поручены чикагскому филиалу IBM - фирме ProfessionalServicesGroup, получившей полную свободу в выборе решений. После того, как этот солидный интегратор израсходовал 20 миллионов долларов и вдвое превысил первоначальную смету, проект был еще далек от завершения и его пришлось передать другой фирме, которая и завершила его в тесном сотрудничестве с заказчиком.

Чтобы не обидеть отечественных интеграторов, ограничимся только этими, зарубежными примерами грандиозных провалов, но и они ясно показывают, что главными причинами неудач крупных проектов являются не технические просчеты, а плохое управление и отсутствие контроля со стороны заказчика за ходом проектирования. Заказчик должен отслеживать этот процесс хотя бы для того, чтобы вовремя остановить его, и принять решение о том, стоит ли продолжать задуманное, несколько скорректировав цели и средства, или же следует вовсе отказаться от проекта, пока не поздно. Очевидно, что такая ответственная роль требует от заказчика достаточно высокого уровня специальных знаний.

**10.4.3. Оценка и выбор предложений интеграторов. Организация тендера**

При выборе интегратора прежде всего нужно правильно организовать тендер - конкурс развернутых предложений фирм-интеграторов на право реализации вашего проекта.

Предложения интеграторов, в зависимости от их опыта, будут содержать различные конкретные варианты решения вашей задачи. Сравните предлагаемые варианты сетей и оцените, насколько они соответствуют конечной вашей бизнес-цели. Выполните анализ стоимость/воз- можности, чтобы определить, какое из предложений дает лучшее соотношение стомость/про- изводительность. Различают грубый анализ и детальный анализ. Грубый анализ определяет стоимость (в денежных единицах) установки сети и экономический эффект, который она принесет компании. Детальный анализ проводить сложнее, и он, вероятно, более важен. Этот анализ экстраполирует деловые преимущества установки сети, а также оценивает стоимостные последствия ее *не* установки.

Так как цены на сетевые продукты изменяются почти каждый день, то гораздо важнее уяснить стоимость реализации различных технических решений. Это надо делать еще на этапе анализа требований к сети, а вот цены на продукты, независимо от того, даны ли они системным интегратором или отделом снабжения вашего предприятия, должны анализироваться в последнюю очередь.

Конечно, для заказчика лучшим способом оценить предложение интегратора была бы демонстрация последним небольшой пилотной сети.

При принятии решения о выборе интегратора желательно ознакомиться и проверить сведения, которые сообщают интеграторы о себе в ответ на ваш запрос в техническом задании. Изучите опыт и историю работы интеграторов. Продемонстрировали ли они техническую компетенцию и другие способности при установке таких сетей, которые они предлагают? Выдерживали ли они сроки в прошлом?

С хорошей стороны характеризует интегратора высокая культура организации собственной работы. Эта культура косвенно проявляется, например, в использовании интегратором специального программного обеспечения управления проектом типа MicrosoftProject, которое позволяет создавать диаграммы и сетевые графики для представления результатов проделанной работы и описания шагов на последующих этапах - сведения, кстати говоря, незаменимые при реализации сложных проектов.

Для успеха любого проекта чрезвычайно важно правильное ведение документации.

Этот принцип применим и к аппаратуре, и к программному обеспечению, и к переносу данных. По окончании работ заказчик должен иметь исчерпывающий отчет со всеми необходимыми данными. Менеджеры фирмы-заказчика, не задающие вопросов предполагаемому системному интегратору о составлении документации, оказывают себе и своим подчиненным медвежью услугу, хотя бы потому, что когда происходит что-либо неприятное, а оно всегда случается без предупреждения (например, когда ключевые сотрудники оставляют компанию или программное и аппаратное обеспечение обновляется), все заканчивается катастрофой.

Эта одна из множества причин, по которой следует потребовать от интегратора исчерпывающих отчетов, зафиксированных на бумаге. Не стесняйтесь попросить документацию, написанную для других заказчиков, с тем, чтобы убедиться в ее ясности, полноте и полезности. Настаивайте на включении в документацию проекта локальных и глобальных адресов, документации по проводке и топологии, диаграмм потоков данных в систему и из нее, а также описаний конфигурационных файлов для файловых серверов, шлюзов, маршрутизаторов и других серверных систем. Кроме того, основные возможные неисправности и способы их исправления должны быть описаны, и должен быть составлен список гарантийных мастерских по каждому из видов оборудования.

Иначе говоря, интегратор должен предоставить всю относящуюся к делу информацию, чтобы сотрудники вашей компании смогли самостоятельно решить любую возникшую проблему. Включение сотрудников в процесс составления сетевой документации дает им возможность заранее ознакомиться с новой сетью и сокращает время на переподготовку.

Хорошая документация поможет поддерживать систему в рабочем состоянии, предотвратить сбои, а если они произойдут, быстро ликвидировать их, да и просто использовать все возможности сети. Интегратор, пренебрегающий документацией, должен вызывать большие сомнения.

Реальные критерии выбора интегратора изменяются от компании к компании. Если вы сами занимались анализом требований к сети, ваш выбор интегратора будет основываться на предложенном проекте, опыте интегратора и цене. Если провести анализ ваших требований вам помогал консультант, то выбор скорее будет сделан на основании цены.

Закупку сетевых компонентов можно поручить той же фирме-интегратору, которая будет устанавливать сеть. Или же можно использовать интегратора в качестве консультанта, а закупки поручить отделу снабжения предприятия.

Закупка своими собственными силами вынуждает ваше предприятие действовать самому в качестве интегратора. Предприятию придется взаимодействовать с большим количеством интеграторов вместо одного. Составление большого количества контрактов и распределение ответственности - вещь более сложная, чем сотрудничество с одной фирмой-интегратором, но при этом можно сэкономить большое количество денег.

**10.4.4. Разработка контракта с интегратором**

После выбора интегратора и его предложений, необходимо разработать и согласовать с ним *контракт*. В контракте приводятся взаимные обязательства между фирмой-интегратором и вашим предприятием. Контракт устанавливает и определяет детали гарантий, обслуживания, оплаты, лицензирования продуктов, определяет права собственности и условия поставок. Контракт - это своего рода "страховой полис" вашей сети. Если на его разработку и согласование требуется 6 недель - потратьте эти 6 недель, так как это лучше, чем после установки сети обнаружить, что в контракте не оговорены обсуждавшиеся проблемы.

Контракт формализует предложения интегратора. В нем юридически устанавливаются термины для приемки оборудования и программного обеспечения, оговариваются виды обслуживания, документации, сроки внедрения сети, создания проекта сети и ее реализации, цены и порядок тестирования.

Рассмотрим некоторые типовые пункты контракта.

*Сетевое оборудование, системное программное обеспечение и приложения.* В контракте перечисляются количество и характеристики сетевого и коммуникационного оборудования, которое должно быть установлено. Контракт фиксирует условия окружающей среды в офисах, где будет установлено оборудование, включая требования к температуре, влажности, и требования к электрическим параметрам. В контракт также включается кабельная система.

Контракт включает лицензионные соглашения на системное программное обеспечение и приложения, которые будут использоваться. Следует включить в контракт условия взаимодействия при возникновении проблем с программным обеспечением, так как здесь всегда имеется большое количество поставщиков и разработчиков. Если вам нужен исходный код, это условие нужно записать в контракт.

*Обслуживание.* Если вы хотите, чтобы обслуживание сети производилось фирмой-интегратором, это должно быть зафиксировано в контракте, включая количество технических специалистов, обслуживающих сеть на вашем предприятии, и гарантированное время обслуживания. Обсуждению подлежит и следующий вопрос: относится ли контракт на обслуживание и к обслуживанию сетевой аппаратуры, коммуникационных средств, системного программного обеспечения и приложений.

*План поэтапного внедрения.* Включите в контракт сроки поэтапного внедрения. Как распределены сроки внутри каждого этапа внедрения? Что произойдет, если намеченный план не будет выполнен?

*Документация.* Контракт должен определять состав документации. Она должна включать в себя описание работ в виде схемы логической сети, включая месторасположения таких основных компонентов, как маршрутизаторы и мосты. Она должна включать также документацию о программно-аппаратных средствах третьих сторон. Будет ли осуществляться интеграторами бесплатное исправление документации, если во время инсталляции произойдут какие-либо изменения? Определите стоимость внесения поправок после этого периода.

*Реализация.* Контракт описывает технические спецификации проекта сети. Он должен содержать информацию о том, какие компоненты будут связаны и каким образом. Следует основывать эту часть контракта на технических аспектах предложений интегратора. Контракт может определять, что интегратор должен гарантировать определенное время ответа (время реакции системы), возможность наращивания и, если это необходимо, совместимость с некоторыми наиболее важными приложениями. Например, вы можете захотеть, чтобы, если потребуется, сеть могла работать со старыми DEC'овскими миникомпьютерами. В контракте должно быть также указано, кто со стороны интеграторов является главным руководителем работ по установке вашей сети.

*Тестирование.* Инсталляционные и приемочные испытания являются очень важными, но часто слишком поверхностными элементами периода анализа требований и заключения контракта. Инсталляционные испытания являются основными, они гарантируют, что сетевые компоненты работают по отдельности и все вместе. Они обычно выполняются небольшой группой людей, использующих пилотный вариант сети. Приемочные испытания гарантируют, что система, построенная интеграторами, отвечает всем техническим и бизнес-требованиям. Опишите в контракте сроки инсталляционных и приемочных испытаний:

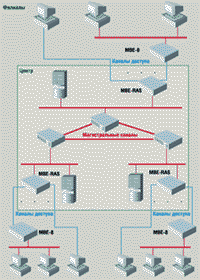
*Стоимость*. Контракт определяет стоимость, сроки согласований стоимости и сроки оплаты. После подписания контракта стоимость может измениться, если вы потребуете дополнительных работ. Понятно, что вами может быть упущен ключевой элемент, или может потребоваться подправить концепцию. Оставьте себе путь для внесения изменений во время процесса планирования либо в план вашей сети, либо в контракт. Что касается сроков оплаты, то вы можете заплатить интегратору 30% при подписании контракта, 40% - после поставки оборудования и оставшиеся 30% после успешного завершения приемочных испытаний. Запишите все дополнительные (сверхурочные) обязанности в контракт. В контракте следует также указать, намерены ли вы приобретать оборудование для инсталляции у других, отличных от вашего интегратора, фирм.

***топологии***

Термин "интрасеть" (или intranet) вошел в моду. И хотя о точном его определении - технология передачи данных, документооборота или что-то другое - еще можно поспорить, популярность интрасетей уже давно не вызывает сомнений. Пристальный интерес пользователей к решениям для интрасетей, в свою очередь, повлиял на развитие рынка телекоммуникационного оборудования.

Эту тенденцию можно проследить на примере семейства мостов/маршрутизаторов LAN RANger от RAD Data Communications. Устройства LAN RANger изначально были рассчитаны на территориально-распределенные корпоративные сети среднего масштаба. Наиболее распространенным типом топологии таких сетей является иерархическая "звезда" с делением на "центр" и "филиалы". В общем случае "центр" состоит из нескольких объединенных магистральными каналами узлов, но каждый "филиал" подключается только к одному узлу "центра" (рис. 1). Задача определения оптимальных маршрутов передачи данных целиком решается на "центральных" узлах (если "центр" состоит лишь из одного узла, то единственно возможный маршрут и является оптимальным).

(1x1)



**Рисунок 1.**  
Типичная топология территориально-распределенной корпоративной сети среднего масштаба: иерархическая "звезда" с делением на "центр" и "филиалы"

На первом этапе в семейство LAN RANger входили лишь элементарные мосты: MBE-RAS, TRE-RAS - серверы удаленного доступа (до 8 синхронных/асинхронных портов); MBE-8, TRE-8 - мосты для подключения локальных сетей (ЛС) филиалов; MBE-1 - мост для подключения отдельных рабочих станций. Выбор технологии коммутации кадров ЛС был обусловлен тем, что она не накладывала ограничений на выбор протоколов верхних уровней (интрасети тогда еще не получили широкого распространения)

Рост популярности интрасетей привел к появлению устройств LAN RANger, имеющих расширенные функциональные возможности. Они применяются для маршрутизации захвативших лидерство протоколов IP и IPX, а для интеграции ЛС с помощью иных протоколов по-прежнему могут быть использованы простейшие мосты. Такая модификация позволила существенно повысить конкурентоспособность семейства, сохранив, что немаловажно, изначально низкую стоимость оборудования.

Интрасети позволяют не только эффективно решать корпоративные задачи, но и предоставлять пользователям корпоративных сетей непосредственный доступ к необъятным информационным ресурсам Internet. Всемирная сеть имеет неоспоримые преимущества - быстрая связь с зарубежными партнерами, продажи через Inetnet-магазины, возможность оперативного получения последних версий ПО, поиск документации по интересующей тематике и др. Однако в ней таится и существенный недостаток: при подключении к Internet внутриведомственные сети теоретически становятся доступными для всех пользователей Internet, которых можно условно разделить на четыре группы:

* добропорядочные пользователи, способные забраться в вашу корпоративную сеть разве что по ошибке;
* хакеры-любители, либо набирающиеся опыта работы в Internet с помощью попытки обойти защитные механизмы, либо просто привлеченные запретным плодом. Не надейтесь, что "любители" не станут перебирать все возможные лазейки: главная сила этой группы - многочисленность;
* хакеры-профессионалы, предлагающие свои услуги, например, для ведения промышленного шпионажа;
* откровенно криминальные структуры.

Системным администраторам и раньше приходилось сталкиваться с проблемами обеспечения безопасности. Для блокирования данных давно и активно используются битовые маски. Например, для передачи по сети, работающей по протоколу IP, данные должны быть упакованы в пакеты, каждый из которых снабжается адресами отправителя и получателя, а также дополнительной служебной информацией. Таким образом, разрешая/запрещая определенные комбинации битов в маске, можно поставить обмен данными под довольно жесткий контроль.

Фильтрация до сих пор служит верой и правдой для разграничения доступа и оптимизации нагрузки каналов связи. Однако из-за широкой доступности программ (в том числе предназначенных для отладки работы сетей), позволяющих "подделывать" адреса и служебные поля IP-пакетов, фильтрация перестала быть эффективным средством обеспечения безопасности доступа в ведомственную сеть.

Чтобы исключить возможность подбора заветного адреса, между ЛС и Internet можно установить брандмауэр, который возьмет на себя функции посредника. В этом случае процесс обмена данными между ЛС и удаленными абонентами состоит из трех этапов:

* удаленный абонент посылает запрос брандмауэру;
* убедившись, что абонент является тем, за кого себя выдает, и что ему разрешен доступ к запрошенной информации, брандмауэр запрашивает из ЛС требуемые данные;
* полученные из ЛС данные брандмауэр отсылает удаленному пользователю, подставляя в поле адреса отправителя свой адрес.

Таким образом, в рассмотренном случае брандмауэр выступает в качестве сервера для подключенных к Internet удаленных абонентов и в качестве клиента для ресурсов ЛС. Одним из примеров реализации подобной системы может служить компьютер с установленным на нем специальным ПО. Как правило, такое ПО позволяет определить не только список допустимых адресов, но и виды услуг, предоставляемых прикладными программами удаленным пользователям. Соответственно, появляется возможность исключить доступ к некоторым ресурсам ЛС через Internet.

Брандмауэры, в отличие от фильтров, могут руководствоваться довольно сложной системой правил разграничения доступа удаленных пользователей. Однако фильтры имеют значительно более высокую производительность. Это привело к появлению нового типа устройств, сочетающих в себе преимущества рассмотренных решений. Компромисс между требованиями к скорости обработки данных и функциональностью позволяет существенно снизить цену оборудования.

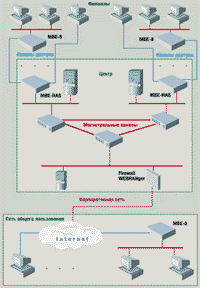
Одним из представителей таких устройств является WEB RANger фирмы RAD Data Communications. Ряд функций WEB RANger был включен и в мосты/маршрутизаторы семейства LAN RANger. Роль этих брандмауэров сводится лишь к блокированию запросов к ресурсам, недоступных для удаленных абонентов. Обработка же разрешенных запросов выполняется непосредственно серверами корпоративной сети.

Было бы ошибкой считать, что WEB RANger или какое-либо иное решение способно обеспечить 100-процентную защиту корпоративной сети от взлома. Специалисты по компьютерной безопасности сходятся во мнении, что существующие реализации протоколов TCP/IP оставляют слишком много лазеек для вторжения извне. Однако многоуровневая защита позволяет практически исключить угрозу взлома защиты хакерами-любителями и существенно усложнить жизнь их более профессиональным "коллегам". Поэтому использование продуктов типа WEB RANger в качестве первой линии обороны интрасети от нежелательных посетителей не только целесообразно, но и крайне желательно даже для корпоративных сетей с полноценными серверами-брандмауэрами.

Наиболее логичным сценарием дальнейшего развития интрасетей представляется их расширение до сетей extranet. "Филиалы" по-прежнему обмениваются данными преимущественно с "центром", но подключенным к Internet предприятиям среднего размера выгоднее использовать услуги Internet-провайдеров, нежели организовывать и обслуживать собственные каналы.

Конечно, совсем без таких каналов не обойтись - передаваемую через Internet информацию легко перехватить, а механизмы криптографической защиты зачастую становятся государственной тайной, как только начинают обеспечивать достаточно надежный уровень безопасности. Поэтому можно ожидать, что в ближайшем будущем extranet-сети унаследуют все ту же широко распространенную звездообразную топологию (рис. 2): центральные узлы будут объединяться выделенными магистральными каналами друг с другом и с наиболее важными "филиалами", а остальные "филиалы" и отдельные удаленные пользователи будут подключаться к центральной интрасети через Internet.

(1x1)



**Рисунок 2.**  
Один из вариантов топологии сети extranet, в которой для ограничения доступа удаленных пользователей к центральной интрасети используется брандмауэр WEB RANger фирмы RAD

1. Введение. В чем состоит планирование сети 1

1.1. Многослойное представление корпоративной сети 3

1.2. Стратегические проблемы построения транспортной системы корпоративной сети 4

1.2.1. Создание транспортной инфраструктуры с масштабируемой производительностью для сложных локальных сетей 5

1.2.2.Предоставление индивидуального качества обслуживания для различных типов трафика и различных приложений в локальных сетях 7

1.2.3. Выбор технологии магистрали для крупных локальных сетей предприятия 14

1.3. Стратегические проблемы выбора сетевой операционной системы и СУБД 17

1.4. Стратегические проблемы создания корпоративных приложений 18

1.5. Защита корпоративной информации при использовании публичных глобальных сетей (в том числе и Internet) 18

1.6. Создание интегрированной системы управления 21

1.7. Планирование этапов и способов внедрения новых технологий в существующие сети 22

1.8. Выбор интеграторов, производителей и поставщиков программных и аппаратных продуктов, провайдеров услуг территориальных сетей 22

1.9. Обучение и набор персонала 23

2. Главные тенденции развития локальных сетей 24

2.1. Обеспечение иерархии скоростей и качества обслуживания 24

2.1.1. От Ethernet к Fast и Gigabit Ethernet 25

2.5. Выбор технологии для построения магистрали крупной локальной сети 36

2.6. Перспективы развития кабельных систем 42

4. Стратегии защиты данных 45

4.1. Комплексный подход - необходимое условие надежной защиты корпоративной сети 45

4.2. Усугубление проблем безопасности при удаленном доступе. Защитные экраны - firewall'ы и proxy-серверы 46

4.3. Использование сертификатов для аутентификации массовых пользователей при ведении бизнеса через Internetи другие публичные сети 48

4.4. Технологии защищенного канала 50

4.5. Правовая регламентация деятельности в области защиты информации 52

4.6. Продукты, сертифицированные для использования в России 54

6. Борьба сетевых операционных систем за корпоративный рынок 57

6.1. Что такое сетевая операционная система 57

6.2. Критерии выбора корпоративной ОС 59

6.3. Тенденции рынка сетевых ОС 67

7. Влияние Internet на мир корпоративных сетей 73

7.1. Транспорт Internet приспосабливается к новым требованиям 74

7.2. Internet-провайдинг. Типовые услуги 77

8. Системы хранения и поиска данных 83

8.1. Новые веяния в области серверов реляционных баз данных 84

8.2. Склады данных и системы оперативной аналитической обработки 112

9. Разновидности и архитектуры информационных приложений 118

9.1. Обзор рынка готовых информационных приложений 118

9.2. Инструментальные средства создания пилотных версий приложений и разработки их законченных вариантов 120

9.3. Обзор применяемых архитектур современных информационных приложений 123

9.4. Проблемы проектирования и разработки приложений 133

10. Проектирование корпоративных сетей 137

10.1. Особенности проектирования корпоративных сетей 137

10.2. Этапы проектирования корпоративных сетей 141

10.3. Средства анализа и оптимизации сетей 157

10.4. Взаимодействие с сетевыми интеграторами 160