МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Северо-Западный государственный заочный технический университет

Г.И.Анкудинов
А.И.Стрижаченко

СЕТИ ЭВМ
И
ТЕЛЕКОММУНИКАЦИИ

АРХИТЕКТУРА И ПРОТОКОЛЫ

Утверждено редакционно-издательским советом университета
в качестве учебного пособия

Санкт-Петербург
2001

УДК 681.326(075)

Анкудинов Г.И., Стрижаченко А.И. Сети ЭВМ и телекоммуникации. Архитектура и протоколы: Учеб. пособие.- СПб.: СЗТУ, 2001, - 92 с.

Учебное пособие соответствует государственному образовательному стандарту дисциплины “Сети ЭВМ и телекоммуникации” направления подготовки дипломированных специалистов 654600 “*Информатика и вычислительная техника”* (Специальность 220100 “*Вычислительные машины, комплексы, системы и сети”*) и направления подготовки бакалавров 552800 “*Информатика и вычислительная техника”.*

Материал учебного пособия посвящен архитектуре вычислительных сетей: рассматривается классификация вычислительных сетей, сетевые топологии и методы доступа к среде передачи данных, эталонная модель взаимодействия открытых систем. Приведены сведения об устройствах объединения сетей: концентраторах, мостах, коммутаторах и маршрутизаторах. Приводится классификация сетевых протоколов и рассматриваются стандартные протоколы. Особое внимание уделяется протоколам Internet сетевого и транспортного уровней. Пособие содержит также вводный материал по сетевому программному обеспечению и администрированию компьютерных сетей.

Пособие предназначено для студентов 4 курса факультета информатики и систем управления, изучающих дисциплину “Сети ЭВМ и телекоммуникации” в рамках бакалаврской подготовки.

Рецензенты: кафедра процессов управления и информационных систем СЗТУ (**А.Б.Шадрин**, д-р техн.наук, проф.); **В.В.Лохмотко**, д-р техн.наук, проф., **М.О.Колбанев**, канд.техн.наук, доц. кафедры информационных управляющих систем Государственного университета телекоммуникаций им. проф. М.А.Бонч-Бруевича.

> Мы на заре века интеллектуальных сетевых технологий - века, рождающего новую экономику, новую политику и новое общество. Бизнес преобразуется, правительства будут обновлены, а люди смогут заново открыть себя - и во всем этом нам помогут новые информационные технологии.
>
> Don Tapscott, *Digital Economy: Promis and Peril in the Age of Networked Intelligence* (NY: McGraw-Hill, 1996)

Предисловие

Мы вступили в XXI век. Происходят фундаментальные изменения в экономике и технике, в том числе и в информатике - технике сбора, обработки, хранения и передачи информации. Эти изменения связаны не столько с новыми информационными технологиями, сколько с тем, что накопление нового в нашем поведении достигло критической массы. Миллионы людей дома и на работе общаются с помощью электронных средств, используя универсальные, открытые стандарты Internet. Этот взрывной рост коммуникаций - самая последняя, а для экономики - самая важная волна информационной революции. В грядущем десятилетии новая информационная экономика ускорит изменения структуры целых отраслей и методы конкуренции.

При написании данного учебного пособия авторы пользовались практически всей новой литературой, изданной к моменту завершения работы над пособием в 2001 г., и материалами, опубликованными в Internet. Много полезных материалов по компьютерным сетям опубликовано на сайте http://www.citforum.ru.

Список использованных источников содержит 18 *наименований. При этом неоднократно осуществлялось заимствование идей, методов изложения, определений и примеров, но отдельные ссылки в тексте не делались. Однако все использованные источники обязательно включены в список.*

В тексте приводятся многочисленные примеры, способствующие более успешному усвоению материала. Теория и особенно практика вычислительных сетей развиваются настолько быстро, что технические решения, признаваемые сегодня за наилучшие, завтра

могут оказаться морально устаревшими. Но в то же время, в вычислительной технике наблюдается спиралевидный характер развития, при котором старые решения возвращаются в новой реализации. Это относится, в частности, к рассмотрению в настоящем пособии вопроса об объединении сетей с помощью мостов.

В заключение, авторы выражают признательность рецензентам и редактору за внимательное прочтение рукописи и замечания, способствовавшие улучшению качества предлагаемого пособия.

Глава 1
Архитектура вычислительных сетей

1.1. Архитектура “клиент-сервер”

Сеть ЭВМ (*компьютерная сеть, или вычислительная сеть - ВС*) - это совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных, совместного использования общих информационных и вычислительных ресурсов.

Распределенные вычисления в компьютерных сетях основаны на архитектуре “клиент-сервер”, ставшей доминирующим способом обработки данных. Термины “клиент” и “сервер” обозначают роли, которые играют различные компоненты в распределенной среде вычислений. Компоненты “клиент” и “сервер” не обязательно должны работать на разных машинах, хотя обычно это так и есть – клиент-приложение находится на рабочей станции пользователя, а сервер - на специальной выделенной машине. Наиболее распространены следующие виды серверов: файл-серверы, северы баз данных, серверы печати, серверы электронной почты, WEB-сервер и другие. В последнее время интенсивно внедряются многофункциональные серверы приложений.

Клиент формирует запрос на сервер для выполннения соответствующих функций. Например, файл-сервер обеспечивает хранение данных общего пользования, организует доступ к ним и передает данные клиенту. Обработка данных распределяется в том или ином соотношении между сервером и клиентом. В последнее время долю обработки, приходящуюся на клиента, стали называть “толщиной” клиента.

Развитие архитектуры “клиент-сервер” происходит по спирали и в настоящее время намечается тенденция централизации вычислений (рис.1.1), т.е. замены “толстых” клиентов – рабочих станций на основе высокопроизводительных ПЭВМ, оснащенных мощным ПО для поддержки прикладных программ, мультимедийных средств, навигационного и графического интерфейса – “тонкими” клиентами. Характерный пример “тонкого” клиента – архитектура Sun Ray Hot Desk, предложенная компанией Sun Microsystems.

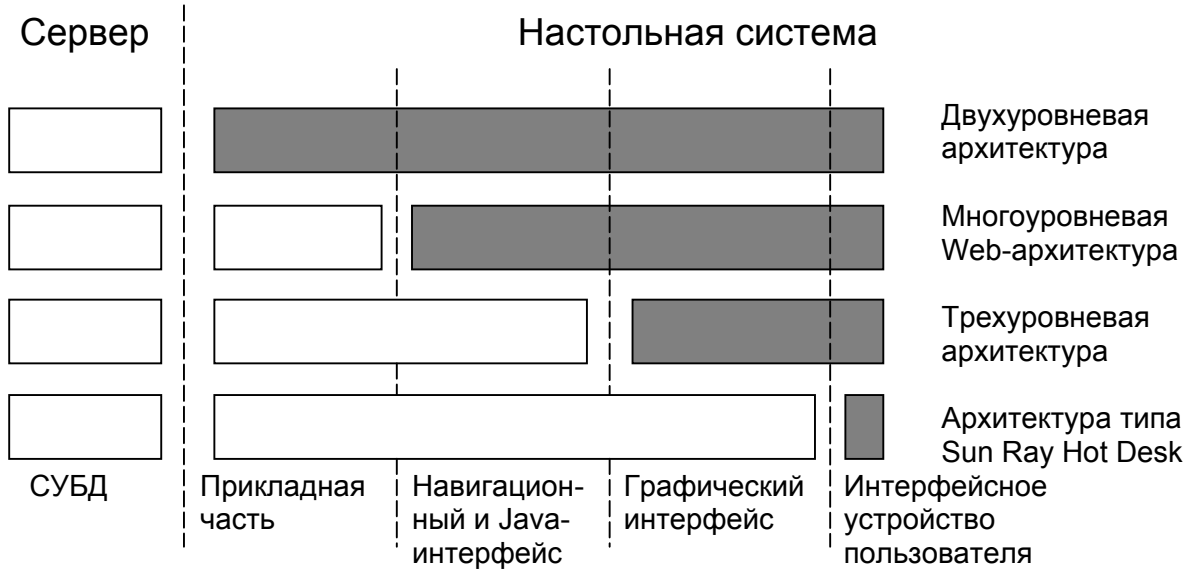


Рис. 1.1. Ранжирование клиентов по “толщине”:

- выполняется на центральном сервере;

- выполняется на компьютере пользователя

Архитектура Sun Ray Hot Desk предполагает использование настольных систем типа графических терминалов Sun Ray 1, имеющих минимум программных и аппаратных средств, но обладающих широкими возможностями работы с приложениями в соответствии с основной идеей “тонких” клиентов – вынести на сервер все, вплоть до виртуальных драйверов устройств, включая драйвер монитора. Историческими предшественниками “тонких” клиентов были алфавитно-цифровые терминалы, подключавшиеся к главным ЭВМ, или мэйнфреймам (mainframe) через специализированные интерфейсы или универсальные последовательные порты.

Мэйнфреймы – классический пример централизации вычислений, поскольку в едином комплексе были сконцентрированы все вычислительные ресурсы, хранение и обработка огромных массивов данных. Основные достоинства централизованной архитектуры - простота администрирования и защиты информации. Все терминалы были однотипными - следовательно, устройства на рабочих местах пользователей вели себя предсказуемо и в любой момент могли бы быть заменены, затраты на обслуживание терминалов и линий связи также легко прогнозировались.

Революция, вызванная появлением персональных компьютеров, сделала возможным иметь вычислительные и информационные ресурсы на рабочем столе пользователя и управлять ими по собственному разумению с помощью цветного оконного графического интерфейса. Увеличение производительности ПК позволило перенести части системы (интерфейс с пользователем, прикладную логику) для выполнения на персональном компьютере, непосредственно на рабочем месте, а функции обработки данных оставить на центральном компьютере. Система стала распределенной - одна часть функций выполняется на центральном компьютере, другая - на персональном, который связан с центральным посредством коммуникационной сети. Таким образом, появилась клиент-серверная модель взаимодействия компьютеров и программ в сети и на этой основе стали развиваться средства разработки приложений для реализации информационных систем.

Однако двухуровневая архитектура "клиент-сервер" (рис.1.1) имеет такие существенные недостатки, как сложность администрирования и низкая информационная безопасность, особенно заметные при сравнении ее с централизованной архитектурой мэйнфреймов (табл.1.1).

Таблица 1.1. Сравнение централизованной архитектуры мэйнфреймов и двухуровневой архитектуры "клиент-сервер”

Централизованная архитектура мэйнфреймов	Двухуровневая архитектура "клиент-сервер"
Вся информационная система на центральном компьютере	Систему, состоящую из большого числа разнотипных ПК, на которых работают разнородные приложения, трудно администрировать
На рабочих местах простые устройства доступа, дающие возможность пользователю управлять процессами в информационной системе	ПК сложны в конфигурировании и поиске неисправностей, стоимость обслуживания достигает от 3 до 7 тыс. долларов в год
Устройство доступа общается с центральным компьютером посредством простого, аппаратно реализованного протокола (передаются экраны и коды нажатых клавиш)	ПК весьма уязвим для вирусов, непродуманных или злонамеренных действий

1.2. Классификация вычислительных сетей

Протяженность связи, которую обеспечивает вычислительная сеть, может быть различной: в пределах одного помещения, здания, предприятия, региона, континента или всего мира. На рис. 1.2 показан вариант структуры глобальной вычислительной сети.

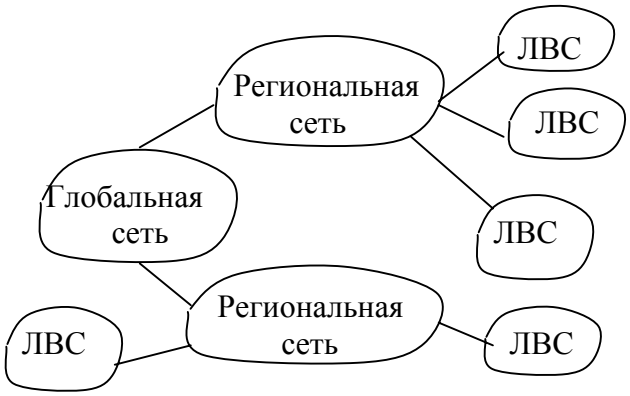


Рис. 1.2. Структура глобальной вычислительной сети

Локальные вычислительные сети, или ЛВС (LAN - Local Area Network), позволяют объединять компьютеры (рабочие станции), расположенные в ограниченном пространстве. Для локальных сетей прокладывается специализированная кабельная система, и положение возможных точек подключения абонентов ограничено этой кабельной системой. Локальные сети можно объединять в более крупномасштабные образования:

- корпоративная сеть (сеть корпорации, предприятия);
- кампусная сеть, объединяющая “кампус” (“лагерь”, городок), т.е. группу близко расположенных зданий (Campus Area Network - CAN);
- сеть городского масштаба (Metropolitan Area Network - MAN);
- региональная сеть, или широкомасштабная сеть (Wide Area Network - WAN);
- глобальная сеть (Global Area Network - GAN).

"Сетью сетей" в наше время называют глобальную сеть Интернет. Термин “Интернет” происходит от английского “Internetworking” – “межсетевое взаимодействие”.

1.3. Сетевые топологии и методы доступа к среде передачи данных

Топология сети характеризует взаимосвязи и пространственное расположение друг относительно друга компонентов сети – сетевых компьютеров (хостов), рабочих станций, кабелей и других активных и пассивных устройств. Топология влияет на:

- состав и характеристики оборудования сети;
- возможности расширения сети;
- способ управления сетью.

Все сети строятся на основе трех базовых топологий:

- шина (bus);
- звезда (star);
- кольцо (ring).

Метод доступа к среде передачи данных определяет, каким образом разделяемый ресурс – сетевой кабель – предоставляется узлам сети для для осуществления актов передачи данных. Основные методы доступа к среде передачи данных:

- состязательный метод (множественный доступ с контролем несущей и обнаружением коллизий - CSMA/CD);
- с передачей маркера;
- по приоритету запроса.

В табл. 1.2 приведены основные типы кабелей, используемых в ЛВС. Для высокопроизводительного обмена, но на ограниченном расстоянии, развивалось несколько направлений реализации локальных сетей - *Ethernet, ARCnet, TokenRing,* адаптеры которых широко используются в персональных компьютерах (ПК).

Шинную и звездообразную топологию использует самая популярная сетевая технология - *E t h e r n e t.* Эта технология представляет архитектуру сетей с состязательным доступом к среде и широковещательной передачей. Это означает, что все узлы сегмента сети получают пакет одновременно. Толстый коаксиальный кабель широко использовался в качестве базовой магистрали Ethernet. Базовая магистраль (backbone) нужна для того, чтобы соединять раз-

ные сети. Для новых компьютеров наиболее популярным стал кабель – витая пара 10BaseT.

Таблица 1.2. Сетевые кабели

Характеристика	Тонкий коаксиальный кабель	Толстый коаксиальный кабель	Витая пара	Оптоволоконный кабель
Стоимость	Дороже витой пары	Дороже тонкого коаксиального кабеля	Самый дешевый	Самый дорогой
Эффективная длина кабеля	185 м	500 м	100 м	2 км
Скорость передачи	10 Мбит/с	10 Мбит/с	≥ 100 Мбит/с	≥ 100 Мбит/с
Гибкость	Довольно гибкий	Менее гибкий	Самый гибкий	Не гибкий
Подверженность помехам	Хорошо защищен	Хорошо защищен	Подвержен помехам	Не подвержен помехам

В реализации *E t h e r n e t на витой паре* применяется звездообразная физическая топология, в центре которой располагается устройство – *концентратор, или хаб (hub).*

В результате развития появилась технология *Ethernet с коммутацией пакетов* (*Switched Ethernet*), реализуемая на звездообразной физической топологии. Здесь управление доступом к среде практически переносится с узлов в *центральное коммутирующее устройство* (*switched hub*), обеспечивающее установление временных (на время передачи одного пакета) виртуальных выделенных каналов между парами портов - источниками и получателями пакетов.

Технология Ethernet позволяет использовать скорости передачи данных 10Мбит/с, 100 Мбит/с и 1Гбит/с, причем высокая скорость доступна только для витой пары и оптоволокна.

Шинная топология (bus). При помощи кабеля каждая рабочая станция соединяется с другими рабочими станциями и с файловым сервером. Кабель проходит от узла к узлу, последовательно соединяя все рабочие станции и все файловые серверы (рис.1.3). На каждом конце кабеля подключается согласующая нагрузка (терминатор) для исключения эхоотражений.

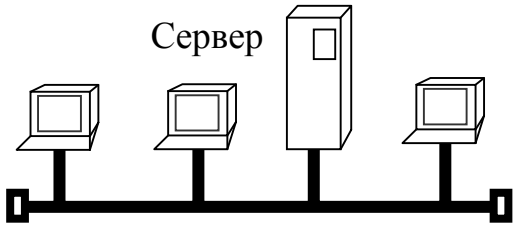


Рис.1.3. Шинная топология

Шинная топология использует состязательный метод доступа. Это означает, что информацию принимает только тот компьютер, адрес которого соответствует адресу получателя, зашифрованному в передаваемых сигналах. Остальные компьютеры отбрасывают сообщение. Перед передачей данных компьютер должен ожидать освобождения шины. В каждый момент времени отправлять сообщение может только один компьютер, поэтому число подключенных к сети машин значительно влияет на ее быстродействие.

Преимущества шинной топологии

- Надежно работает в небольших сетях, проста в использовании.
- Требует меньше кабеля для соединения компьютеров и потому дешевле, чем другие схемы соединений
- Легко расширяется за счет состыковки кабельных сегментов с помощью цилиндрического соединителя BNC и использования повторителей.

Недостатки шинной топологии

- Интенсивный сетевой трафик снижает производительность сети. При большом числе компьютеров в сети станции часто прерывают друг друга, и немалая часть полосы пропускания теряется понапрасну. При добавлении компьютеров к сети резко падает производительность.
- Цилиндрические соединители ослабляют электрический сигнал и большое их число вызывает нарушения в передаче информации по шине.

- Разрыв кабеля или неправильное функционирование одной из станций может привести к нарушению работоспособности всей сети**.** Сеть трудно диагностировать.

Звездообразная топология (Ethernet 10BaseT, 100BaseT). Каждый компьютер в сети с топологией типа "звезда" (“star”) взаимодействует с центральным *концентратором* (hub - устройство для повторения сетевых сигналов) (рис.1.4).

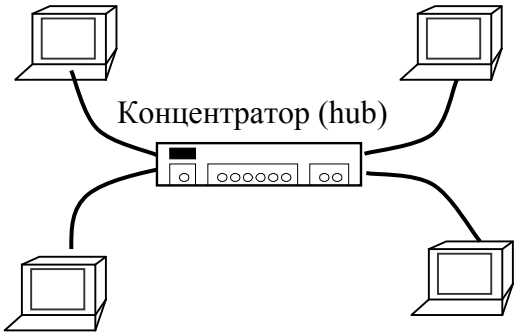


Рис. 1.4. Топология “звезда”

В звездообразной сети используется состязательный метод доступа к среде - концентратор (хаб) передает сообщение всем компьютерам. В *звездообразной сети с коммутацией* коммутатор передает сообщение только компьютеру-адресату.

Активный концентратор регенерирует электрический сигнал и посылает его всем подключенным компьютерам. Такой тип концентратора часто называют *многопортовым повторителем* (multiport repeater). Для работы активных концентраторов и коммутаторов требуется питание от сети. *Пассивные концентраторы*, например, коммутационная кабельная панель или коммутационный блок, действуют как точка соединения, не усиливая и не регенерируя сигнал. Электропитания пассивные концентраторы не требуют.

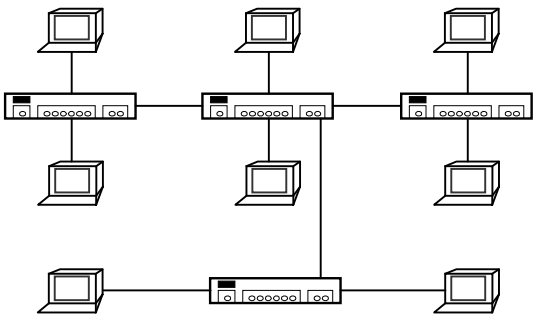


Рис. 1.5 Гибридно-звездообразная топология

Гибридный концентратор позволяет использовать в одной звездообразной сети разные типы кабелей. Расширить звездообразную сеть можно путем подключения вместо одного из компьютеров еще одного концентратора и подсоединения к нему дополнительных станций, в результате чего получается *гибридно-звездообразная сеть* (рис.1.5).

Преимущества топологии "звезда"(Ethernet 10BaseT, 100BaseT)

- Центральный концентратор звездообразной сети удобно использовать для диагностики. *Интеллектуальные концентраторы* (устройства с микропроцессорами, добавленными для повторения сетевых сигналов) обеспечивают также измерение параметров (мониторинг) и управление сетью.
- Отказ одного компьютера не обязательно приводит к останову всей сети. Концентратор способен выявлять отказы и изолировать такую машину или сетевой кабель, что позволяет остальной сети продолжать работу.
- В одной сети допускается применение нескольких типов кабелей (если их позволяет использовать концентратор).

Недостатки сети со звездообразной топологией

- При отказе центрального концентратора вся сеть становится неработоспособной .
- Все компьютеры должны соединяться с центральной точкой, это увеличивает расход кабеля, следовательно, такие сети обходятся дороже, чем сети с иной топологией.

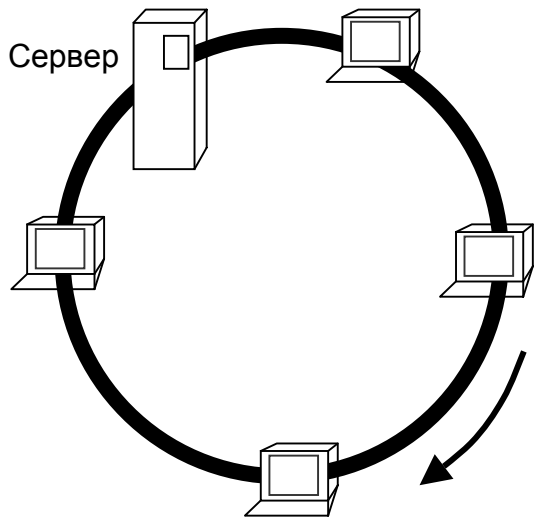


Рис. 1.6. Кольцевая топология

Кольцевая топология (Сети Token Ring). На рис. 1.6 показан пример топологии ЛВС, в которой каждая рабочая станция соединена с двумя другими рабочими станциями. Такая топология называется *кольцом (ring).* Кольцевая топология применяется преимущественно в США для сетей, требующих выделения определенной части полосы пропускания для критичных по времени средств (например, для передачи видео и аудио), в высокопроизводительных сетях, а также при большом числе обращающихся к сети клиентов (что тре-

бует ее высокой пропускной способности). В сети с кольцевой топологией каждый компьютер соединяется со следующим компьютером, ретранслирующим ту информацию, которую он получает от первой машины. Благодаря такой ретрансляции сеть является активной, и в ней не возникают проблемы потери сигнала, как в сетях с шинной топологией. Кроме того, поскольку «конца» в кольцевой сети нет, никаких оконечных нагрузок не нужно.

Некоторые сети с кольцевой топологией используют метод доступа к среде на основе маркера (метод эстафетной передачи). Специальное короткое сообщение-маркер циркулирует по кольцу пока компьютер не пожелает передать информацию другому узлу. Он модифицирует маркер, добавляет электронный адрес и данные, а затем отправляет его по кольцу. Каждый из компьютеров последовательно получает данный маркер с добавленной информацией и передает его соседней машине, пока электронный адрес не совпадет с адресом компьютера-получателя, или маркер не вернется к отправителю. Получивший сообщение компьютер возвращает отправителю ответ, подтверждающий, что послание принято. Тогда отправитель создает еще один маркер и отправляет его в сеть, что позволяет другой станции перехватить маркер и начать передачу. Маркер циркулирует по кольцу, пока какая-либо из станций не будет готова к передаче и не захватит его.

Все эти события происходят очень часто: маркер может пройти кольцо с диаметром в 200 м примерно 10000 раз в секунду. В некоторых еще более быстрых сетях циркулирует сразу несколько маркеров. В других сетевых средах применяются два кольца с циркуляцией маркеров в противоположных направлениях. Такая структура способствует восстановлению сети в случае возникновения отказов.

Преимущества сети с кольцевой топологией

- Поскольку всем компьютерам предоставляется равный доступ к маркеру, никто из них не сможет монополизировать сеть.
- Справедливое совместное использование сети обеспечивает постепенное снижение ее производительности в случае увеличения числа пользователей и перегрузки (лучше, если сеть будет продолжать функционировать, хотя и медленно, чем сразу откажет при превышении пропускной способности).

Недостатки сети с кольцевой топологией

- Отказ одного компьютера в сети может повлиять на работоспособность всей сети.
- Кольцевую сеть трудно диагностировать.
- Добавление или удаление компьютера вынуждает разрывать сеть.

Смешанные топологии. На основе трех базовых топологий можно создавать так называемые *гибридные* или *смешанные* топологии. К этим топологиям относятся:

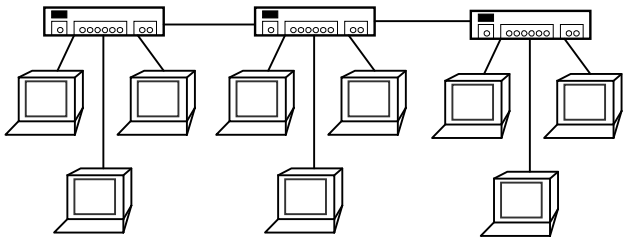


Рис. 1.7. Шинно-звездообразная топология

• Шинно-звездообразная.
• Звездообразно-кольцевая.

Шинно-звездообразная топология (рис.1.7) комбинирует сети типа «звезда» и «шина», связывая несколько концентраторов шинными магистралями. Если один из компьютеров отказывает, концентратор может выявить отказавший узел и изолировать неисправную машину. При отказе концентратора соединенные с ним компьютеры не смогут взаимодействовать с сетью, а шина разомкнется на два не связанных друг с другом сегмента.

В *звездообразно-кольцевой* топологии (которую называют также кольцом с соединением типа «звезда») сетевые кабели прокладываются аналогично звездообразной сети, но в центральном концентраторе реализуется кольцо (рис.1.8). С внутренним концентратором можно соединить внешние, тем самым расширив петлю внутреннего кольца.

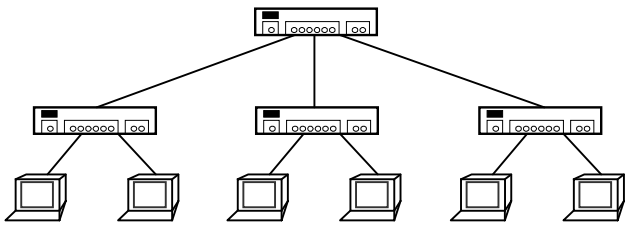


Рис. 1.8. Звездообразно-кольцевая топология

Большие объединенные ВС используют топологию самого общего вида – *ячеистую*. Узлами ячеистой топологии могут быть самые разнообразные сетевые устройства: повторители, мосты, концентраторы, маршрутизаторы, шлюзы.

1.4. Эталонная модель взаимодействия открытых систем

Обмен информацией между компьютерами, объединенными в сеть, очень сложная задача. Это связано с тем, что существует много производителей аппаратных и программных средств вычислительных систем. Единственный выход - унифицировать средства сопряжения систем, а именно использовать открытые системы. Открытая система взаимодействует с другими системами в соответствии с принятыми стандартами.

В 1984г. Международная Организация по Стандартизации (ISO) выпустила стандарт - *семиуровневую эталонную модель взаимодействия открытых систем* (Seven-layer Open System Interconnection Reference Model - OSI), чтобы помочь поставщикам создавать совместимые сетевые аппаратные и программные средства. В соответствии с этой моделью выделяются следующие иерархические уровни:

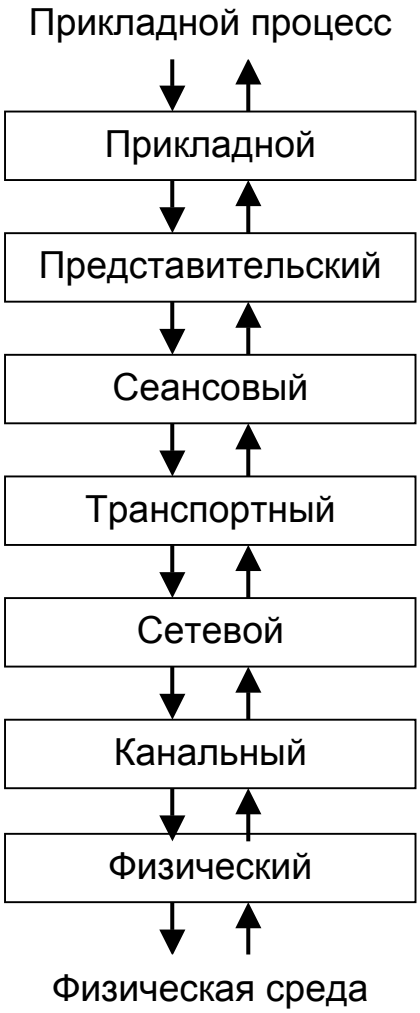


Рис.1.9. Модель OSI

- физический (Physical);
- канальный (Data Link);
- сетевой (Network);
- транспортный (Transport);
- сеансовый (Session);
- представительский (Presentation) ;
- прикладной (Application).

В соответствии с эталонной моделью OSI эти уровни взаимодействуют так, как показано на рис.1.9. Таким образом, сложная задача обмена информацией между компьютерами в сети разбивается на ряд относительно независимых и менее сложных подзадач взаимодействия между соседними уровнями. Каждая такая подзадача выполняется в соответствии с унифицированными правилами - протоколом взаимодействия.

Границу между сеансовым и транспортным уровнями можно рассматривать как границу между протоколами прикладного уровня и протоколами низших уровней. Если прикладной, представительный и сеансовый уровни обеспечивают прикладные процессы сеанса взаимодействия, то четыре низших уровня решают проблемы транспортировки данных.

Два самых низших уровня - физический и канальный - реализуются аппаратными и программными средствами, остальные пять более высоких уровней реализуются, как правило, программными средствами. При передаче информации от прикладного процесса в сеть на физический уровень происходит обра-

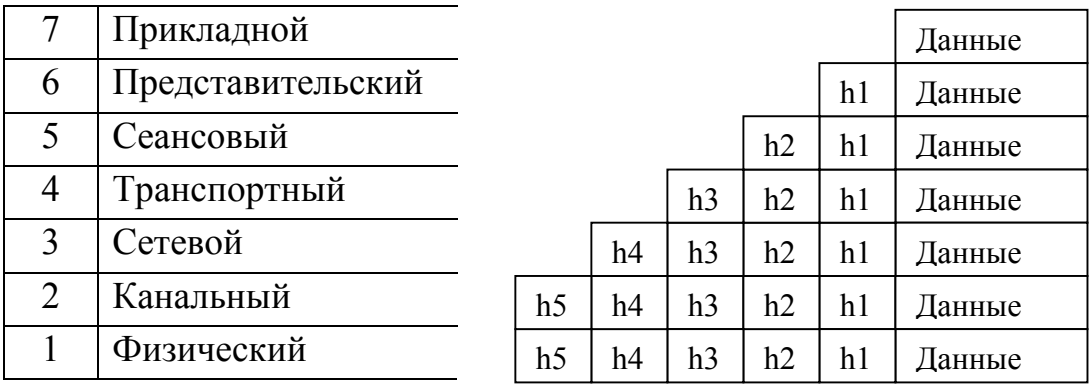


Рис. 1.10. Преобразование информации в соответствии с семиуровневой эталонной моделью OSI

ботка ее, которая заключается в разбиении передаваемых данных на отдельные блоки, преобразовании формы представления или кодировки данных в блоке и добавлении к каждому блоку заголовка $h_{7\text{-}L}$ (header), характеризующего соответствующий уровень L=1..5 (рис.1.10). Каждый заголовок характеризует используемый протокол обработки данных, причем каждый уровень L = 2…6 воспринимает в качестве данных весь блок, полученный от уровня L+1, включая присоединенный заголовок. Такое построение эталонной модели позволяет заложить в каждый передаваемый по физической среде информационный блок сведения, необходимые для выбора последовательности протоколов для осуществления обратных преобразований на принимающей информацию стороне.

Физический уровень

Этот уровень определяет механические, электрические, процедурные и функциональные характеристики установления, поддержания и размыкания

физического соединения между конечными системами. Физический уровень определяет такие характеристики соединения, как уровни напряжений, синхронизацию и физическую скорость передачи данных, максимальные расстояния передачи, конструктивные параметры разъемов и другие аналогичные характеристики. Известные стандарты RS-232-C, V.24 и IEEE 802.3 (Ethernet).

Канальный уровень

Канальный уровень (уровень звена данных, информационно-канальный уровень) отвечает за надежную передачу данных через физический канал, а именно:

- обеспечивает физическую адресацию (в отличие от сетевой или логической адресации);
- обеспечивает обнаружение ошибок в передаче и восстановление данных;
- отслеживает топологию сети и обеспечивает дисциплину использования сетевого канала конечной системой;
- обеспечивает уведомление о неисправностях;
- обеспечивает упорядоченную доставку блоков данных и управление потоком информации.

Для ЛВС канальный уровень разбивается на два подуровня:

- LLC (Logical Link Control) - обеспечивает управление логическим звеном, т.е. собственно функции канального уровня;
- MAC (Media Access Control) - обеспечивает специальные методы доступа к среде распространения.

Сетевой уровень

Этот уровень обеспечивает возможность соединения и выбор маршрута между двумя конечными системами, подключенными к разным подсетям (сегментам), которые могут быть разделены множеством подсетей и могут находиться в разных географических пунктах. Протоколы маршрутизации позволяют сети из маршрутизаторов выбирать оптимальные маршруты через связанные между собой подсети.

Транспортный уровень

Транспортный уровень обеспечивает высшим уровням услуги по транспортировке данных, а именно:

- обеспечивает надежную транспортировку данных через объединенную сеть;
- обеспечивает механизмы для установки, поддержания и упорядоченного завершения действия виртуальных каналов;
- обеспечивает обнаружение и устранение неисправностей транспортировки;
- следит за тем, чтобы конечная система не была перегружена слишком большим количеством данных.

Другими словами, транспортный уровень обеспечивает интерфейс между процессами и сетью, устанавливает логические каналы между процессами и обеспечивает передачу по этим каналам информационных блоков. Эти логические каналы называются транспортными.

Сеансовый уровень

Сеансовый уровень реализует установление, поддержку и завершение сеанса взаимодействия между прикладными процессами абонентов. Сеансовый уровень синхронизирует диалог между объектами представительного уровня , определяет точки синхронизации для промежуточного контроля и восстановления при передаче файлов. Этот уровень также позволяет производить обмен данными в режиме, заданном прикладной программой, или предоставляет возможность выбора режима обмена.

Кроме основной функции управления диалогом сеансовый уровень предоставляет средства для выбора класса услуг и уведомления об исключительных ситуациях (проблемах сеансового, представительного и прикладного уровней).

Представительный уровень

Представительный уровень (уровень представления данных) определяет синтаксис, форматы и структуры представления передаваемых данных (но не затрагивает семантику, значение данных). Для того, чтобы информация, посылаемая из прикладного уровня одной системы, была читаемой на прикладном уровне другой системы, представительный уровень осуществляет трансляцию между известными форматами представления информации за счет использования общего формата представления информации.

Таким образом, этот уровень обеспечивает служебные операции, выбираемые на прикладном уровне, для интерпретации передаваемых и получаемых данных: управление информационным обменом, отображение данных и управление структурированными данными. Эти служебные данные позволяют связывать воедино терминалы и вычислительные средства различных типов.

Прикладной уровень

В отличие от других уровней прикладной уровень - самый близкий к пользователю уровень OSI - не предоставляет услуги другим уровням OSI, однако он обеспечивает прикладные процессы, лежащие за пределами масштаба модели OSI.

Прикладной уровень обеспечивает непосредственную поддержку прикладных процессов и программ конечного пользователя (программ обработки крупномасштабных таблиц, текстовых процессоров, программ банковских терминалов и т.д.) и управление взаимодействием этих программ с сетью передачи данных:

- идентифицирует и устанавливает наличие предполагаемых партнеров для связи;
- синхронизирует совместно работающие прикладные программы;
- устанавливает соглашение по процедурам устранения ошибок и управления целостностью информации;
- определяет достаточность наличных ресурсов для предполагаемой связи.

Адресация компьютерных систем

Адресация компьютерных систем служит для определения их местонахождения в объединенной сети. Схема адресации, зависит от используемого семейства протоколов. В основе схем адресации лежит разделение понятий адресов канального уровня и адресов сетевого уровня.

Адреса канального уровня называются также физическими или аппаратными адресами. В соответствии с названием, адреса канального уровня существуют на уровне 2 эталонной модели ISO. Физический адрес уникален для каждого сетевого компьютера, и определяется сетевой интерфейсной картой (NIC - Network Interface Card), используемой для подключения компьютера к комму-

никационной среде ЛВС. Пример физического 6-байтного адреса: 211.023.101.001.199.067.

Уникальность адреса обеспечивается изготовителем интерфейсной карты. Таким образом, физический адрес определяет точку в некоторой адресной плоскости, координаты которой выбираются случайным образом, а именно зависят от конкретной сетевой карты, установленной в компьютер. Если компьютерная система имеет одно физическое сетевое соединение, то она имеет только один адрес канального уровня. Маршрутизаторы и другие системы, соединенные с множеством физических сетей, могут иметь множество адресов канального уровня.

Адреса сетевого уровня относятся к уровню 3 эталонной модели OSI и называются также виртуальными или логическими адресами. Если адрес канального уровня определяет некоторую уникальную точку адресной плоскости, в каком-то смысле случайную, то адреса сетевого уровня обычно иерархические и несут некоторую смысловую нагрузку. В этом смысле они похожи на почтовые адреса, которые описывают местонахождение человека, указывая имя человека, улицу, номер дома и квартиры, город, страну, почтовый индекс. Иерархическая структура адреса облегчает сортировку адресов и повторный вызов за счет исключения крупных блоков логически схожих адресов при выполнении последовательности операций сравнения. Например, адрес СЗТУ "nwpi.ru" позволяет исключить все другие страны, поскольку в адресе указана страна "ru" (Россия). Легкость сортировки и повторного вызова являются причиной того, что маршрутизаторы используют адреса сетевого уровня в качестве базиса маршрутизации.

Вопросы терминологии. Еще не сложилась достаточно устойчивая общепринятая терминология в области сетевых информационных технологий. Во многих случаях используются различные варианты перевода терминов с английского на русский, поэтому в данном пособии для основных понятий даются английские термины.

Например, рассмотрим термины, используемые для наименования единиц группирования информации, перемещаемой между абонентами и уровнями модели OSI. В литературе по сетевым технологиям можно видеть непоследовательность в наименовании таких единиц. Используются термины:

- "блок данных" (frame);
- "пакет" (packet);
- "блок данных протокола" (protocol data unit -PDU);
- "сегмент" (segment);
- "сообщение" (message).

В настоящем пособии будем придерживаться следующих определений этих терминов:

- "блок данных" (frame) - блок информации, источником и пунктом назначения которого являются объекты канального уровня;
- "пакет" (packet) - блок информации, у которого источник и пункт назначения - объекты сетевого уровня;
- "сообщение" (message) - обозначает информационный блок, у которого объекты источника и места назначения находятся выше сетевого уровня, а также для обозначения отдельных информационных блоков низших уровней, которые имеют специальное, хорошо сформулированное назначение.

Глава 2
Объединение сетей с помощью мостов, коммутаторов и маршрутизаторов

2.1. Устройства объединения сетей

Устройства объединения сетей обеспечивают связь между сегментами локальных сетей, отдельными ЛВС и подсетями любого уровня. Эти устройства в самом общем виде могут быть отнесены к определенным уровням эталонной модели взаимодействия открытых систем. Существуют следующие классы устройств для объединения сегментов ЛВС и сетей (рис. 2.1):

- повторители (repeaters) объединяют сети на физическом уровне
- мосты (bridges) и коммутаторы (switches) объединяют сети на канальном уровне и используют функциональные возможности физического уровня. Мосты выполняются на основе компьютера, оснащенного соответствующим ПО. Отличие коммутаторов от мостов в том, что они реализуют свои функ-

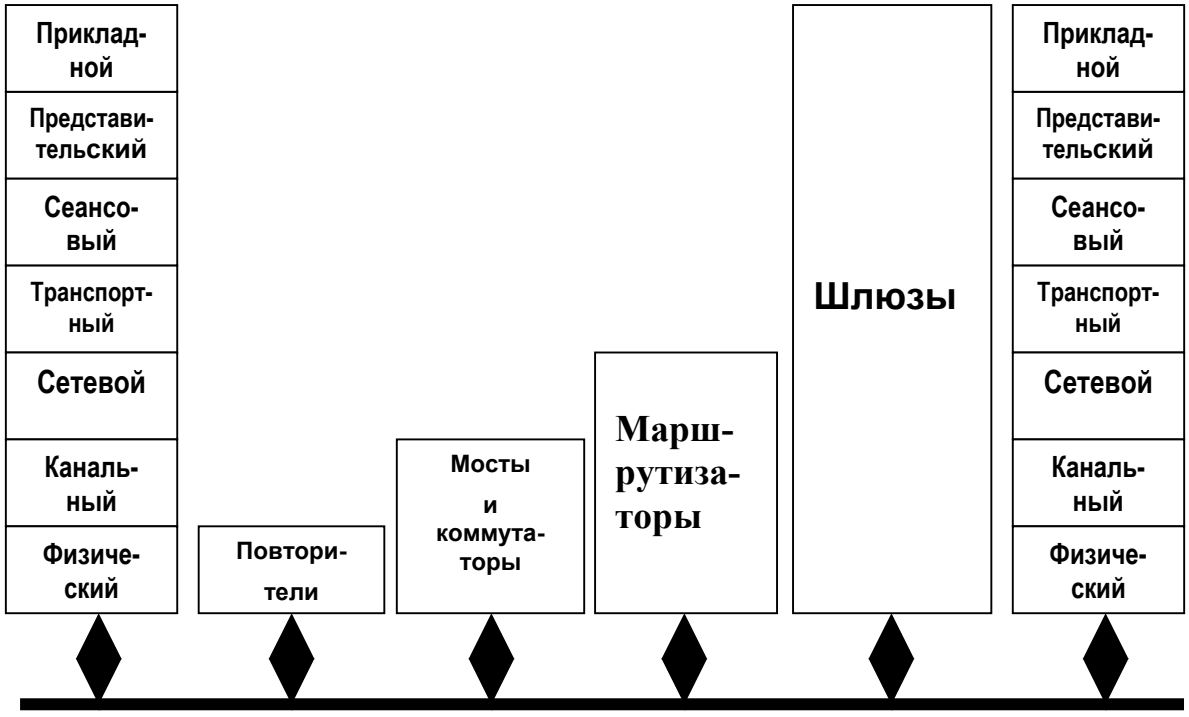


Рис.2.1. Устройства объединения сетей

ции аппаратными средствами и поэтому обладают значительно более высоким быстродействием;

- маршрутизаторы (routers) объединяют сети на сетевом уровне и используют функциональные возможности уровней 1 и 2;
- шлюзы, или межсетевые интерфейсы (gateways) объединяют сети на прикладном уровне и используют функциональные возможности всех нижележащих уровней.

Мосты появились и использовались для объединения однородных сетей с начала 1980-х гг. В последнее время уменьшились цены на маршрутизаторы и многие из них получили возможность включения по мостовой схеме, что привело к сокращению выпуска чистых мостов. Появились мосты, которые осуществляют сложные схемы фильтрации, псевдоинтеллектуальный выбор маршрута и имеют высокую производительность. Функции мостов определяются стандартом IEEE (Институт инженеров по электротехнике и радиоэлектронике). Существуют следующие основные варианты объединения сетей с помощью мостов:

- прозрачное соединение (transparent bridging) используется в среде Ethernet;
- соединение маршрут-источник (source-route bridging) используется в среде Token Ring ;
- трансляционное соединение (translational bridging) обеспечивает трансляцию между форматами и принципами передачи различных типов сред (обычно Ethernet и Token Ring);
- прозрачное соединение маршрут-источник (source-route transparent bridging) объединяет алгоритмы прозрачного соединения и соединения маршрут- источник, что позволяет передавать сообщения в смешанных средах Ethernet/Token Ring.

2.2. Сегментация сетей с помощью мостов

Мосты применяются на канальном уровне, который контролирует поток информации, обрабатывает ошибки передачи, обеспечивает физическую (в отличие от логической) адресацию и управляет доступом к физической среде. Мосты обеспечивают выполнение перечисленных функций путем поддержки

различных протоколов канального уровня. В качестве примеров распространенных протоколов канального уровня можно назвать Ethernet, Token Ring и FDDI (Fiber Distributed Data Interface). Эти протоколы предписывают определенный поток информации, обработку ошибок, адресацию и алгоритмы доступа к физической среде передачи.

В соответствии со стандартом IEEE канальный уровень OSI подразделяется на два подуровня:

- подуровень управления доступом к носителю (MAC = Media Access Channel), который управляет доступом к среде передачи (разрешает конфликтные ситуации, организует эстафетную передачу и т.д.);
- подуровень управления логическим каналом (LLC = Logical Link Channel), который осуществляет адресацию подуровня MAC, выделяет кадры и управляет потоком информации, контролирует неисправности.

Мосты подуровня MAC соединяют гомогенные (однородные) сегменты (сети) с одинаковыми протоколами, например, сети стандарта IEEE 802.3 (Ethernet) и IEEE 802.5 (Token Ring). Другие мосты, как показано на рис.2.2, могут осуществлять трансляцию между различными протоколами канального уровня (например, IEEE 802.3 и IEEE 802.5).

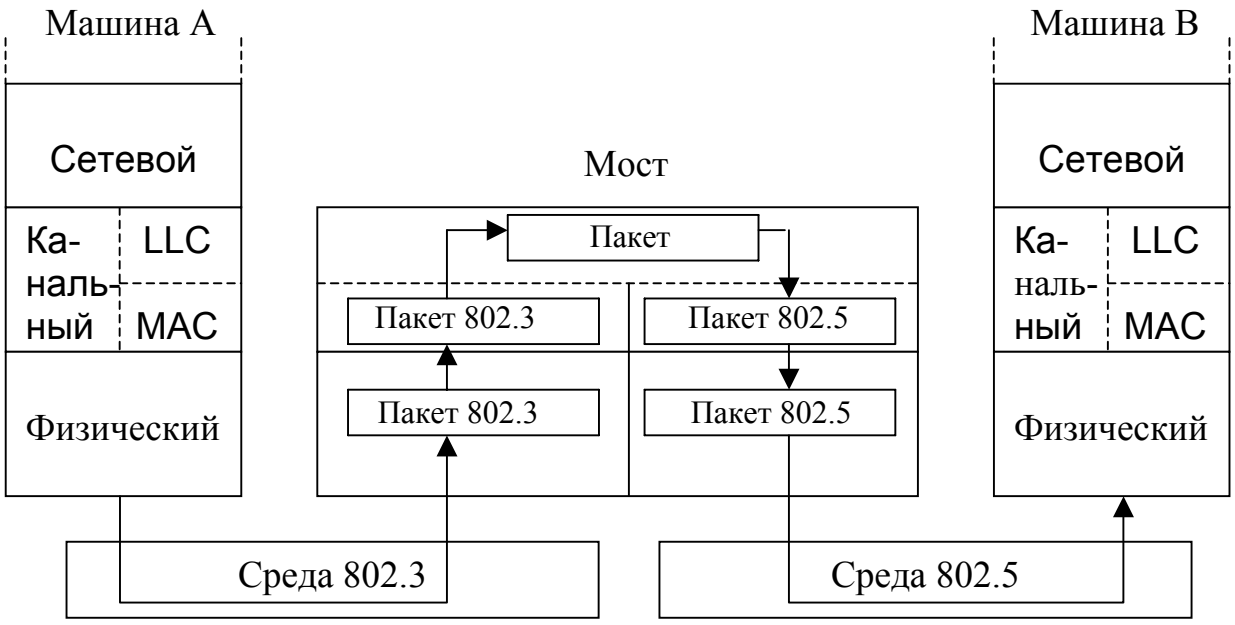


Рис. 2.2. Мост, связывающий сегменты IEEE 802.3 и IEEE 802.5

На этом рисунке вычислительная машина А работает в сегменте сети с протоколом IEEE 802.3. Эта машина формирует пакет, содержащий прикладную информацию, и погружает этот пакет в совместимый с IEEE 802.3 фрейм, который через среду IEEE 802.3 поступает в мост. Внутри моста фрейм освобождается от заголовка IEEE 802.3 в подуровне MAC канального уровня и затем передается выше в подуровень LLC для дальнейшей обработки. После обработки пакет снова передается вниз в реализацию IEEE 802.5, которая добавляет к пакету заголовок IEEE 802.5 для передачи пакета через среду IEEE 802.5 в вычислительную машину В. Следует учитывать, что всегда имеется вероятность, что одна сеть поддержит определенный фрейм, который не поддерживается другой сетью.

Мосты выполняют несложные функции: анализируют поступающие фреймы и, базируясь на информации, содержащейся в фреймах, принимают решения о их пересылке к месту назначения. При объединении типа "источник-маршрут" вся информация о пути к месту назначения содержится в каждом фрейме. В случае прозрачного объединения с помощью мостов фреймы продвигаются к месту назначения отдельными пересылками от узла к узлу, причем каждый фрейм содержит только адрес следующего узла.

Поскольку мосты функционируют на канальном уровне, они могут быстро продвигать трафик, представляющий любой протокол сетевого уровня, не проверяя информацию высших уровней. Это свойство прозрачности мостов для протоколов верхних уровней позволяет, например, продвигать трафик протоколов Apple Talk, DECnet, TCP/IP, XNS и других между двумя и более сетями и является основным преимуществом использования мостов для создания объединенных сетей.

Вместе с тем, мост можно запрограммировать так, чтобы он не пропускал все фреймы, посылаемые из определенной сети. Для этого в соответствующем поле фрейма для канального уровня должна быть ссылка на протокол высшего уровня, что позволяет фильтровать фреймы по этому параметру. Если в соответствующее поле фрейма включить признак широковещательных пакетов, то анализ этого признака позволяет отвергать необязательную информацию широкой рассылки.

Таким образом, достоинства использования мостов:

- мосты увеличивают число связанных сетью устройств и эффективную длину ЛВС, позволяя подключать дополнительные отдаленные станции и сетевые сегменты;
- разделяя крупные сети на автономные блоки, мосты уменьшают трафик в отдельных сегментах и создают преграду для распространения некоторых потенциально опасных для сети неисправностей.

Можно выделить два основных типа мостов:

- Локальные мосты обеспечивают прямое соединение множества сегментов ЛВС, находящихся на одной территории.
- Дистанционные мосты соединяют множество сегментов ЛВС на различных территориях, обычно через телекоммуникационные линии.

Дистанционное соединение с помощью мостов имеет один недостаток: сложность согласования скоростных ЛВС и региональных сетей с низкими скоростями передачи. Мосты могут компенсировать несоответствия в скоростях путем использования достаточных буферных мощностей. Если устройство ЛВС, работающей со скоростью 3 Мбит/с связывается с устройством отдаленной ЛВС, то локальный мост должен регулировать с помощью буферной памяти поток информации, передаваемой со скоростью 3Мбит/с, чтобы не переполнить последовательный канал, который пропускает 64 кбит/с.

2.3. Прозрачные мосты (transparent bridges)

Прозрачные мосты используются в сетях Ethernet/IEEE 802.3 и названы так потому, что они в определенном смысле являются "прозрачными" для машин сети. После подачи питания на прозрачный мост он анализирует адрес назначения каждого поступающего блока данных и определяет топологию сети следующим образом: если, например, блок данных отправителя - машины с физическим адресом 15 - поступил через порт PN 1, то это фиксируется в таблице (рис. 2.3). Предполагается, что порт PN1

Адрес машины	Номер сегмента
15	1
17	1
12	2
13	2
18	1
9	1
4	3

Рис. 2.3. Таблица связи физических адресов машин и номеров сегментов

связан с сегментом 1. Таким образом, после включения питания заполняются аналогичные таблицы во всех прозрачных мостах многосегментной сети.

Информация, содержащаяся в таблице, используется для продвижения трафика. Предположим, через порт PN2 моста принят блок данных для пункта назначения - машины с адресом 4. Используя таблицу, мост определяет, что полученный блок надо отправить в сегмент 3 и направляет этот блок данных в порт PN3, обслуживающий сегмент 3.

Если таблица не содержит адреса пункта назначения, то принятый блок данных отправляется лавинной адресацией во все порты, кроме порта, через который получен блок данных. Аналогичным образом пересылаются широковещательные сообщения и сообщения многопунктовой адресации.

Достоинством прозрачных мостов является следующее:

- они изолируют внутрисегментный трафик, пропускают только необходимый транзитный трафик и тем самым сокращают суммарный трафик в каждом отдельном сегменте. Для сетей работающих на пределе пропускной способности среды это заметно улучшает время реакции сети;
- прозрачные мосты позволяют создавать сети с резервными избыточными связями между сегментами. Граф такой сети содержит циклы, поэтому в активном состоянии должны быть только связи, формирующие топологию сети без циклов - древовидный граф, или дерево. Дело в том, что при наличии циклов сначала возникает циркуляция пакетов, которая приводит к нарушению всей работы сети.

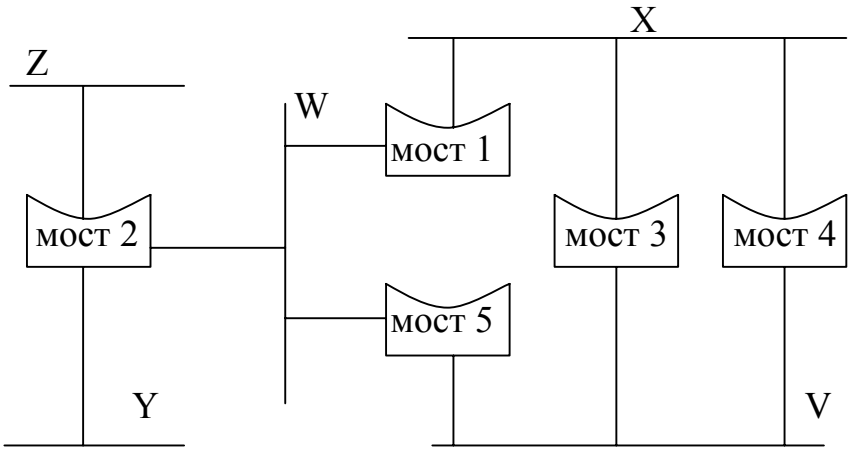


Рис. 2.4. Пример сети до прогона алгоритма построения остовного дерева

Для любого связного графа можно построить связный древовидный подграф, содержащий все вершины исходного графа - остовное дерево. Для этого используется алгоритм построения остовного дерева (STA = Spanning-Tree Algoritm).

На рис. 2.4 изображен пример сети, содержащей циклы “мост 1 - мост 3- мост 5” ,“мост 1 - мост 4 - мост 5” и “мост 3 - мост 4” до прогона STA. На рис. 2.5 показана та же сеть после прогона STA. Таким образом, устраняются все мосты, непосредственно соединенные с каждым сегментом, кроме одного, и, следовательно, разрываются все циклы исходного графа.

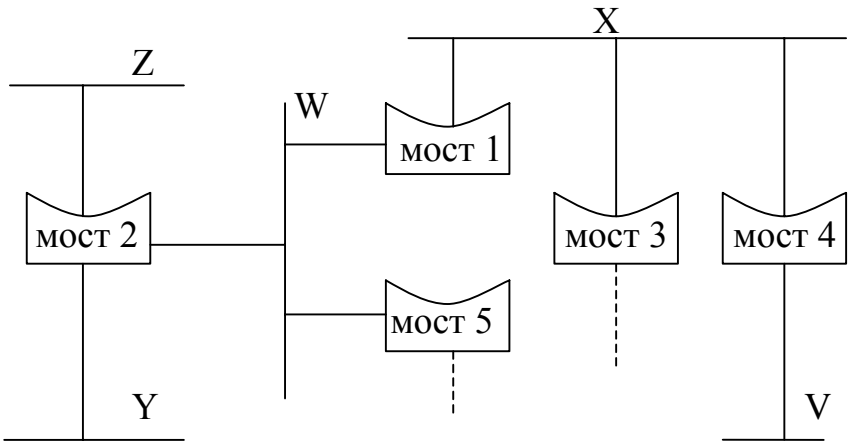


Рис. 2.5. Сеть после прогона алгоритма построения остовного дерева

Построение остовного дерева начинается при подаче питания на мост, а также во всех случаях, когда обнаруживается изменение топологии сети, вызванное отказом какого-либо моста. Для этого мосты через регулярные интервалы времени (1-4 секунды) обмениваются так называемыми сообщениями конфигурации, формат которых приведен на рис. 2.6. Если какой-нибудь мост отказывает, то соседние мосты, не получившие ожидаемое сообщение, инициируют процесс перестроения топологии сети, чтобы восстановить ее связность.

Сообщения о топологических изменениях содержат 4 байта:

- поле идентификатора протокола (2 байта);

- поле версии (1 байт);
- поле типа (1 байт).

1	2	3	4	5	6	7	8	9	10	11	12

1 - идентификатор протокола (2 байта)
2 - поле версии (1 байт)
3 - тип сообщения (1 байт)
4 - поле флагов (1 байт): бит TC сигнализирует об изменении в топологии, бит TCA устанавливается для подтверждения приема сообщения конфигурации с установленным битом TC
5 - идентификатор корневого моста (8 байт): идентифицирует корневой мост путем перечисления его 2-байтового приоритета, за которым следует его 6-байтовый ID
6 - затраты тракта от моста, который отправляет конфигурационное сообщение, до корневого моста (4 байта)
7 – поле идентификатора моста (8 байт), которое идентифицирует приоритет и ID моста, отправляющего сообщение
8 - поле идентификатора порта (2 байта) идентифицирует порт, из которого отправлено конфигурационное сообщение. Это поле позволяет обнаруживать и устранять циклы
9 - поле возраста сообщения (2 байта) определяет промежуток времени, прошедшего с момента отправки корневым мостом конфигурационного сообщения
10 - поле максимального возраста (2 байта) указывает, когда текущее конфигурационное сообщение должно быть стерто
11 - поле времени приветствия (2 байта) обеспечивает период времени между конфигурационными сообщениями корневого моста
12 - поле задержки продвижения (2 байта) обеспечивает промежуток времени, в течение которого мосты должны выжидать, прежде чем перейти в новое состояние после изменения в топологии

Рис. 2.6. Формат сообщения конфигурации прозрачного моста

Прозрачные мосты разработаны компанией Digital Equipment Corporation в начале 1980-х гг. и включены в стандарт IEEE 802.1.

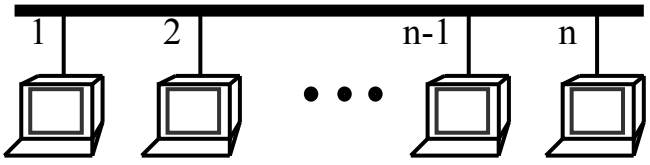
2.4. Сегментация сетей с помощью коммутаторов

Использование разделяемой среды передачи между всеми узлами сегмента при большом числе станций и интенсивном трафике резко снижает производительность сети. Мосты обрабатывают поступающие кадры последователь-

но и поэтому не могут удовлетворить возрастающие требования к пропускной способности ЛВС. Поэтому в 1990 году фирма Kalpana разработала технологию коммутации сегментов Ethernet, основанную на использовании многопортовых коммутаторов, позволяющих одновременно передавать пакеты между всеми парами портов. Поскольку технология Ethernet больше других страдает от повышения времени ожидания доступа к среде при повышении загрузки сегмента, узкие места крупных сетей Ethernet в певую очередь нуждаются в средствах разгрузки .

Многопортовый коммутатор работает как многопортовый мост, то есть работает на канальном уровне, анализирует заголовки кадров, автоматически строит адресную таблицу и на основании этой таблицы перенаправляет кадр в один из своих выходных портов или фильтрует его, удаляя из буфера. Отличие коммутаторов от мостов заключается в параллельной обработке поступающих кадров. Для этого коммутатор должен иметь несколько внутренних процессоров для обработки кадров, каждый из которых выполняет алгоритм моста. Таким образом, коммутатор можно рассматривать как мультипроцессорный мост, имеющий за счет внутреннего параллелизма высокую производительность.

а)



б)

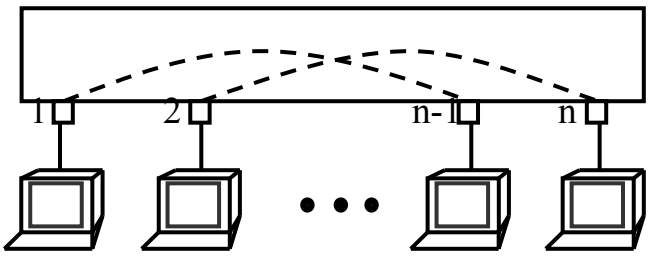


Рис. 2.7. Топология ЛВС:
а) на основе шины;
б) на основе многопортового коммутатора

На рис. 2.7 для сравнения показан сегмент сети, объединяющей n станций на основе шинной топологии (а) и n станций, объединенных с помощью многопортового коммутатора (б).

Если используется протокол Ethernet со скоростью 10 Мбит/с, то шинная топология обеспечивает пропускную способность отдельного виртуального канала между парами узлов значительно менее 10 Мбит/с / n, а суммарную производительность значительно менее 10 Мбит/с. В то же время использование комму-татора для n/2 пар одновременно взаимодействующих узлов, при ус-

ловии, что коммутатор успевает обрабатывать кадры, поступающие на входные порты с максимальной интенсивностью, дает производительность отдельного виртуального канала порядка 10 Мбит/с, а суммарную производительность порядка (n/2)*10 Мбит/с.

Таким образом, коммутатор предоставляет каждой станции или сегменту, подключенным к его портам, выделенную пропускную способность протокола. Повышение производительности сети при установке коммутатора в общем случае не будет таким значительным, как в рассмотренном примере. На эффективность работы коммутатора влияет много факторов, и в первую очередь – сбалансированность трафика между портами коммутатора.

Технология коммутации для повышения производительности используется и в других технологиях ЛВС, таких как TokenRing и FDDI. Принципы работы коммутатора в сетях любых технологий одинаковы и не зависят от физической среды передачи, формата пакета и других деталей:

- обеспечивается одновременное продвижение кадров между парами портов коммутатора;
- используется алгоритм работы прозрачного моста, т.е. коммутатор изучает на основании проходящего через него трафика адреса конечных узлов сети, строит адресную таблицу сети и затем на ее основании производит межкольцевые передачи в сетях TokenRing или FDDI.

2.5. Маршрутизация и маршрутизаторы

Маршрутизация решает две задачи:

- выбор оптимального по некоторому критерию пути продвижения информации от источника к пункту назначения через объединенную сеть;
- транспортировка информационных блоков (пакетов) по выбранному маршруту, или коммутация.

Как правило, также как при использовании мостов предполагается, что на пути встречается по крайней мере один узел. Основное отличие маршрутизации от объединения с помощью мостов в том, что мосты работают на уровне 2 эталонной модели ISO, в то время как маршрутизация используется на уровне 3.

Определение оптимального маршрута

Критерий оптимальности маршрута может использовать различные показатели (длину, стоимость маршрута и т.д.). Алгоритмы маршрутизации заполняют и поддерживают таблицы маршрутизации, в которых содержится информация необходимая для выбора маршрута. Таблица маршрутизации, кроме различных показателей, необходимых для оптимизации маршрутов, содержит также результаты расчета оптимальных маршрутов в виде пар "Сеть назначения/Следующий узел". Приняв очередной пакет, маршрутизатор по таблице маршрутизации определяет следующий узел, т.е. направление пересылки пакета.

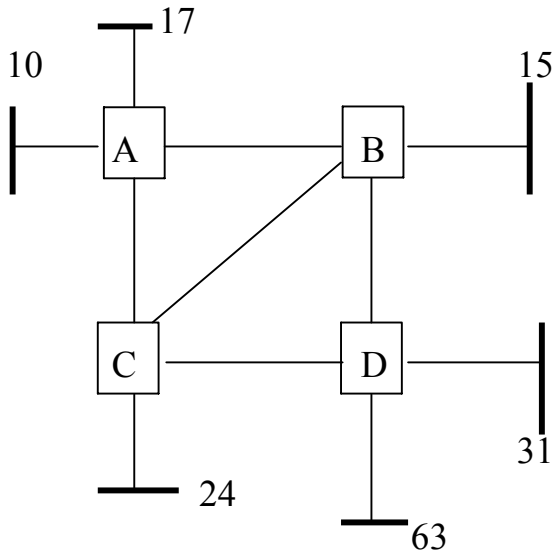


Рис. 2.8. Пример фрагмента сети, содержащего четыре узла

На рис. 2.8 приведен пример фрагмента сети, содержащего четыре узла A, B, C и D. Каждый узел имеет маршрутизатор и одну или несколько главных вычислительных машин, обслуживающих компьютерные сети, номера которых указаны рядом с каждым узлом. На рис. 2.9 приведены маршрутные таблицы для узлов A и B. Для определения оптимальных маршрутов к пунктам назначения, а также для поддержания и обновления своих маршрутных таблиц необходима полная информация о топологии сети. Для этого маршрутизаторы общаются друг с

Таблица маршрутизации узла A

Сеть назначения	Следующий узел
15	B
24	C
31	B
63	C

Таблица маршрутизации узла B

Сеть назначения	Следующий узел
10	A
17	A
24	C
63	D
31	D

Рис.2.9. Таблицы маршрутизации для узлов A и B

другом путем обмена специальными сообщениями:

- сообщениями об обновлении маршрутизации, включающими всю маршрутную таблицу или ее часть, что позволяет каждому маршрутизатору построить полную картину топологии сети;
- объявлениями о состоянии канала, содержащими информацию о состоянии каналов отправителя, которая также необходима маршрутизаторам для построения детальной топологии сети.

Алгоритмы коммутации. Как правило, хост-источник определяет необходимость отправки пакета в другой хост и поэтому отправляет пакет, адресованный в физический адрес своего маршрутизатора (уровень MAC), однако с сетевым адресом (адресом протокола) хоста пункта назначения.

Маршрутизатор отсылает пакет к следующему маршрутизатору путем замены физического адреса пункта назначения на физический адрес следующего маршрутизатора. Маршрутизатор, как правило, игнорирует пакет, если не знает, как переслать его дальше.

Следующая пересылка может быть в хост-пункт назначения или в очередной промежуточный маршрутизатор. По мере продвижения пакета через объединенную сеть, его физический адрес меняется, однако адрес протокола остается неизменным.

В соответствии со стандартами ISO, в больших сетях маршрутизация и коммутация организованы по иерархическому принципу. Для этого вводятся понятия:

- *конечная система* (End System - ES) - любой узел сети, который не занимается маршрутизацией, т.е. устройство сети, не обладающее способностью пересылать пакеты между подсетями;
- *промежуточная система* (Intermediate System - IS) - маршрутизатор, т.е. устройство сети, способное пересылать пакеты между подсетями;
- *область* (Area) - группа смежных сетей и подключенных к ним хостов, которые определяются как область администратором сети или другим аналогичным лицом;
- домен (Domain) - набор соединенных областей. Домены маршрутизации обеспечивают полную связность со всеми конечными системами, находящимися в их пределах.

Промежуточные системы далее подразделяются на следующие виды:

- *внутридоменные* IS, т.е. системы, которые могут сообщаться в пределах *автономных систем* (Autonomous System - AS) или *доменов маршрутизации*;
- *междоменные* IS, т.е. системы, которые могут сообщаться как в пределах домена маршрутизации, так и с другими доменами маршрутизации.

Как правило, домен маршрутизации представляет часть объединенной сети под общим административным управлением, базирующемся на определенных принципах. Домены маршрутизации в свою очередь могут быть подразделены на *участки маршрутизации*, внутри которых также используются внутридоменные протоколы маршрутизации.

Частные критерии оценки алгоритмов маршрутизации. Критерии оценки алгоритмов маршрутизации характеризуют отдельные частные цели их разработки:

1. *Оптимальность маршрута*. Оптимальность маршрута характеризует способность алгоритма маршрутизации выбирать "наилучший" маршрут. Наилучший маршрут в свою очередь зависит от таких показателей, как время задержки, стоимость пересылки, и от весов, выражающих важность этих показателей.
2. *Сложность*. Алгоритм маршрутизации должен иметь минимальную сложность, эффективно выполнять свои функции с минимальными затратами вычислительных ресурсов, особенно в том случае, когда программа, реализующая алгоритм маршрутизации, должна работать на компьютере с ограниченными физическими ресурсами.
3. *Живучесть и стабильность*. Поскольку маршрутизаторы расположены в узлах сети, вероятность их отказа должна быть минимальной, т.е. алгоритмы маршрутизации должны четко функционировать в случае неординарных или непредвиденных обстоятельств, таких как отказы аппаратуры, условия высокой нагрузки и некорректные данные. Живучесть и стабильность выявляется длительной надежной работой в различных условиях эксплуатации сети.
4. *Сходимость.* Алгоритмы маршрутизации должны быстро сходиться. Сходимость достигается в процессе согласования между всеми маршрутизато-

рами. Процесс согласования запускается такими событиями, как изменение доступности некоторых маршрутов. В таких случаях маршрутизаторы рассылают сообщения об обновлении маршрутизации. Процесс согласования заключается в пересчете согласованных оптимальных маршрутов. Алгоритмы маршрутизации, имеющие плохую сходимость, могут привести к образованию циклов в маршрутизации и серьезным нарушениям работы сети. Возможность зацикливания демонстрирует следующий пример.

Маршрутизация в узле С до отказа звена C-D

Сеть назначения	Следующий узел
10	A
15	B
17	A
31	D
63	D

Маршрутизации в узле С после отказа звена C-D

Сеть назначения	Следующий узел
10	A
15	B
17	A
31	B
63	A

Рис. 2.10. Маршрутизация в узле С до и после отказа звена C-D

Рис.2.10 представляет таблицы маршрутизации в узле С (для примера см. рис. 2.9) до и после отказа звена C-D. До отказа звена C-D пакет, адресованный из А в сеть 63, направляется в узел С, а из С этот же пакет в D. После отказа звена C-D маршрутизатор в С обновляет свою таблицу как показано на рис. 2.10. Если в некоторый момент времени, когда узел С уже обновил таблицу маршрутизации, а узел А еще не успел обновить свою таблицу, А направит в узел С пакет для сети 63, то достигнув узла С, этот пакет вернется в А, т.е. возникнет зацикливание.

5. *Гибкость.* Алгоритмы маршрутизации должны быстро и точно адаптироваться к изменениям топологии и параметров элементов сети - полосам пропускания и задержкам каналов, длинам очередей к маршрутизаторам и т.д.

Классификация алгоритмов маршрутизации. Алгоритмы маршрутизации могут быть классифицированы по следующим признакам:

1. Динамичность (статические или динамические);
2. Число маршрутов (одномаршрутные или многомаршрутные);
3. Число уровней (одноуровневые или иерархические);
4. Интеллектуальность (с интеллектом в хосте или в маршрутизаторе);

5. Масштаб (внутридоменные и междоменные);
6. Принцип вычисления маршрута (алгоритмы состояния канала или вектора расстояний).

Динамичность

Статические алгоритмы используют таблицы маршрутизации, заполняемые администратором сети до начала маршрутизации.

Поскольку статические алгоритмы маршрутизации не могут оперативно реагировать на изменения в сети, они непригодны для современных крупных, постоянно изменяющихся сетей. Статические алгоритмы просты и могут быть использованы в небольших сетях.

Динамические алгоритмы, анализируя приходящие сообщения об обновлении маршрутизации, способны реагировать на изменения состояния сети в реальном масштабе времени. При изменениях состояния сети, динамический алгоритм пересчитывает маршруты и в свою очередь рассылает сообщения о корректировке маршрутизации. Такие сообщения вызывают лавинообразный процесс корректировки таблиц маршрутизации.

Число маршрутов

Одномаршрутные алгоритмы обеспечивают единственный маршрут к пункту назначения. Эти алгоритмы просты в реализации, но не всегда способны обеспечить требуемую пропускную способность и надежность доставки.

Многомаршрутные алгоритмы обеспечивают мультиплексную передачу трафика по многочисленным путям. Преимущества многомаршрутных алгоритмов в том, что они могут обеспечить значительно большую пропускную способность и надежность доставки.

Однако, многомаршрутные алгоритмы сложнее в реализации.

Число уровней

Одноуровневые алгоритмы маршрутизации основаны на том, что все маршрутизаторы равны по отношению друг к другу и, в этом смысле, оперируют в плоском пространстве.

Иерархические алгоритмы маршрутизации основаны на том, что часть маршрутизаторов формируют базу (backbone) маршрутизации.

Для этого, как уже было сказано выше, выделяются логические группы узлов: домены, автономные системы (AS) и области. В иерархических системах часть маршрутизаторов какого-либо домена могут общаться с маршрутизаторами других доменов, в то время как другие маршрутизаторы этого домена могут поддерживать связь с маршрутизаторами только в пределах своего домена. В очень крупных сетях могут существовать дополнительные иерархические уровни. Маршрутизаторы наивысшего иерархического уровня образуют базу маршрутизации. Подробнее об иерархической маршрутизации см. ниже.

Интеллектуальность алгоритмов маршрутизации

Алгоритмы маршрутизации с интеллектом в маршрутизаторе предполагают, что хосты не обладают информацией о маршрутах. В таких системах маршрутизаторы определяют маршрут через объединенную сеть, базируясь на своих собственных расчетах.

Алгоритмы маршрутизации с интеллектом в хосте, или *алгоритмы маршрутизации от источника*, предполагают, что хост-источник определяет весь маршрут. В таких системах маршрутизаторы используются просто как устройства буферизации и пересылки пакетов, не выполняющие каких-либо расчетов по определению маршрута.

Системы с интеллектом в хосте способны выбрать наилучший маршрут по критерию оптимальности, принятому для данной конкретной системы. Для этого они, как правило, осуществляют в том или ином виде перебор всех возможных маршрутов к пункту назначения и вычисляют значение критерия для каждого маршрута. Однако определение оптимального маршрута часто требует значительных накладных затрат времени на вычисления и увеличение трафика поиска. Выбор между маршрутизацией с интеллектом в хосте и маршрутизацией с интеллектом в маршрутизаторе достигается путем сопоставления выигрыша от оптимальности маршрута с непроизводительными затратами трафика.

Масштаб

Внутридоменные алгоритмы маршрутизации действуют только в пределах доменов.

Междоменные алгоритмы маршрутизации действуют как в пределах доменов, так и между ними. Оптимальный алгоритм междоменной маршрути-

зации не обязательно будет оптимальным алгоритмом внутридоменной маршрутизации .

Принцип вычисления маршрута

Алгоритмы состояния канала (алгоритмы "первоочередности наикратчайшего маршрута") направляют потоки маршрутной информации во все узлы объединенной сети. Однако каждый маршрутизатор посылает только ту часть таблицы маршрутизации, которая описывает состояние его собственных каналов.

Алгоритмы вектора расстояния (алгоритмы Беллмана-Форда) требуют от каждого маршрутизатора посылки всей или части своей таблицы маршрутизации, но только своим соседям. По сравнению с алгоритмами состояния канала, которые направляют небольшие корректировки по всем направлениям, алгоритмы вектора расстояний отсылают более крупные корректировки только в соседние маршрутизаторы.

Алгоритмы состояния каналов характеризуются более сложными расчетами и имеют более быструю сходимость, чем алгоритмы вектора расстояния. Поэтому они обеспечивают меньшую вероятность образования петель маршрутизации, однако требуют большей процессорной мощности и памяти, чем алгоритмы вектора расстояний. Оба типа алгоритмов маршрутизации хорошо функционируют при самых различных обстоятельствах.

Показатели и критерии, используемые в алгоритмах маршрутизации

В данном разделе рассматриваются частные показатели (метрики), используемые при построении таблиц маршрутизации и вычислении оптимальных маршрутов, а также рассматривается вопрос о построении интегрального (глобального или обобщенного) критерия для определения предпочтительности одного маршрута по сравнению с другими по совокупности частных показателей.

Перечислим частные показатели, которые используются в алгоритмах маршрутизации:

1. Длина маршрута.
2. Надежность.
3. Задержка.

4. Ширина полосы пропускания.
5. Нагрузка.
6. Стоимость связи.

Длина маршрута

Могут использоваться следующие варианты определения (задания) длины маршрута:

- администратор сети назначает произвольные цены на каждый канал сети. В этом случае длина маршрута равна сумме цен (расходов), связанных с каждым каналом, который входит в маршрут;
- учитывается количество пересылок, т.е. показатель, характеризующий число проходов, которые пакет должен совершить на пути от источника до пункта назначения через устройства объединения сетей (такие как маршрутизаторы).

Надежность

Надежность алгоритмов маршрутизации складывается из нескольких факторов:

- вероятность сбоя для каждого канала сети (может измеряться в числе правильно переданных бит на одну ошибку – бит/ошибка);
- вероятность отказа для каждого канала сети (может измеряться в длительности наработки на один отказ – час/отказ);
- трудоемкость устранения последствий сбоя или отказа.

Администратор сети обычно назначает числовые оценки надежности для отдельных каналов сети. При назначении таких оценок администратор сети может принимать в расчет любые факторы надежности.

Задержка маршрутизации

Задержка маршрутизации - это отрезок времени, необходимый для продвижения пакета от источника до пункта назначения через объединенную сеть. Задержка маршрутизации зависит от следующих факторов:

- полоса пропускания (Мбит/с, кбайт/с) промежуточных каналов сети; полоса пропускания является оценкой максимально достижимой пропускной

способности канала, т.е. характеризует мощность трафика, который он способен пропустить;
- длина очереди в порт каждого маршрутизатора на пути продвижения пакета;
- нагруженность всех промежуточных каналов сети;
- физическое расстояние, на которое необходимо переместить пакет.

2.6. Иерархическая маршрутизация

Основное преимущество иерархической маршрутизации заключается в том, что она согласуется с организацией большинства компаний и, следовательно, очень хорошо поддерживает их схемы трафика. Большая часть сетевых коммуникаций осуществляется в пределах групп небольших подразделений компании (доменов). Внутридоменным маршрутизаторам достаточно иметь информацию только о других маршрутизаторах в пределах своего домена, поэтому их алгоритмы маршрутизации могут быть упрощенными и, соответственно, уменьшен трафик корректировки маршрутизации.

Стандарт OSI предлагает протокол IS-IS внутридоменной маршрутизации промежуточных систем (Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol) и протокол ES-IS, вырабатывающий так называемые сообщения конфигурации (см. ниже). Выше уже были определены понятия конечной системы (end system - ES), промежуточной системы (intermediate system-IS), области (Area) и домена (Domain). Дополнительно определим понятия:
- маршрутизация уровня 1 (Level 1 routing) - маршрутизация в пределах области уровня 1;
- маршрутизация уровня 2 (Level 2 routing) - маршрутизация между областями уровня 1.

Рис. 2.11 иллюстрирует значение этих терминов.

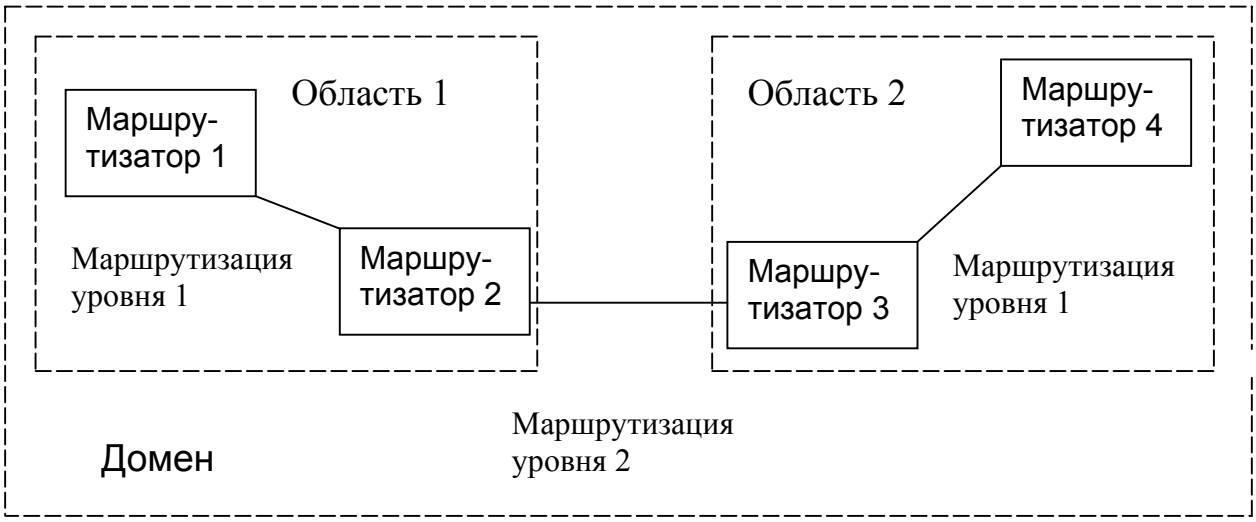


Рис. 2.11. Иерархия объединенных сетей OSI

Протокол ES-IS

Благодаря протоколу ES-IS выполняется процесс, называемый *конфигурацией* (configuration), с помощью которого системы ES и IS узнают о существовании друг друга и тем самым подготавливают следующий этап - собственно маршрутизацию. Протокол ES-IS различает три разных типа подсетей:

- *Двухточечные подсети* (Point-to-point subnetworks) обеспечивают непосредственное соединение между двумя системами. Большинство последовательных каналов глобальной сети являются двухточечными сетями.
- *Широковещательные подсети* (Broadcast subnetworks) направляют отдельное физическое сообщение во все узлы данной подсети. Примерами широковещательных подсетей являются Ethernet и IEEE 802.3.
- *Подсети с общей топологией* (General-topology subnetworks) поддерживают произвольное число систем. Однако в отличие от широковещательных подсетей, величина затрат на передачу по какому-нибудь маршруту непосредственно связана с размерами данной подсети в подсети с общей топологией. Примером подсети с общей топологией является X.25.

Сообщения конфигурации двух типов передаются через определенные интервалы времени. Приветственные сообщения ES (ESH) генерируются и отправляются в каждую систему IS данной подсети. Приветственные сообщения IS (ISH) генерируются и отправляются всем системам ES данной подсети. Эти

приветственные сообщения, в основном, предназначены для переноса адресов подсетей и адресов сетевого уровня тех систем, которые генерируют их.

При возможности ES-IS пытается отправить сообщения конфигурации одновременно в несколько систем. В широковещательных подсетях приветственные сообщения ES-IS отправляются во все IS с помощью специальной многопунктовой адресации. Промежуточные системы IS отправляют приветственные сообщения по специальному адресу многопунктовой адресации, определенному для всех конечных систем. При работе в подсети с общей топологией протокол ES-IS обычно не передает сообщения конфигурации из-за больших затрат на передачу с многопунктовой адресацией.

Протокол IS-IS

Протокол IS-IS является протоколом маршрутизации с указанием состояния канала. Для этого, используя лавинную адресацию, он передает по сети информацию о состоянии канала для построения полной, согласованной картины топологии сети.

Для упрощения построения и работы маршрутизатора протокол IS-IS различает системы IS уровней 1 и 2. Системы IS уровня 1 могут сообщаться с другими системами IS уровня 1, находящимися в той же области. Системы IS уровня 2 могут сообщаться с системами IS других областей. Таким образом, системы IS уровня 1 формируют области уровня 1, а системы IS уровня 2 осуществляют маршрутизацию между областями уровня 1.

Системы IS уровня 2 формируют стержень внутридоменной маршрутизации. Другими словами, системы IS уровня 2 могут попасть в другие системы IS уровня 2 только через системы IS уровня 2. Наличие такого стержня упрощает схему, так как в этом случае системам IS уровня 1 нужно уметь только попадать в ближайшую систему IS уровня 2.

Сообщение между системами ES

Каждая конечная система ES принадлежит конкретной области. Системы ES обнаруживают ближайшую систему IS путем прослушивания пакетов ISH. Если какая-нибудь система ES захочет отправить пакет в другую систему ES, она направляет пакет в одну из систем IS сети, к которой она непосредственно подключена. Маршрутизатор просматривает адрес пункта назначения и продвигает пакет по наилучшему маршруту. Если система ES пункта назначения находится в той же подсети, то местная система IS узнает об этом в результате прослуши-

вания ESH и соответствующим образом продвинет пакет. В этом случае система IS может также обеспечить отправку сообщения о переадресации (redirect - RD) в источник пакета, чтобы сообщить о доступности более прямого пути.

Если адресом пункта назначения является какая-нибудь система ES другой подсети той же области, то система IS узнает о точном маршруте и соответствующим образом продвинет пакет. Если адресом пункта назначения является какая-нибудь система ES другой области, то система IS уровня 1 отправляет этот пакет в в ближайшую систему IS уровня 2. Продвижение пакета через системы IS уровня 2 продолжается до тех пор, пока он не достигнет системы IS уровня 2 в области пункта назначения. В пределах области пункта назначения системы IS продвигают пакет по наилучшему маршруту, пока не будет достигнута система ES пункта назначения.

Каждая система IS генерирует корректировку, определяющую системы ES и IS, с которыми она соединена, а также связанные с ней показатели. Эта корректировка отправляется во все соседние системы IS, которые продвигают ее своим соседям, и т.д. (лавинная адресация). Номера последовательностей прекращают лавинную адресацию и отличают старые корректировки от новых. Так как каждая система IS получает корректировки о состоянии канала от всех других систем IS, то каждая система IS может построить полную базу данных всей топологии сети. При изменении топологии отправляются новые корректировки.

Показатели (метрики) протокола IS-IS

Протокол IS-IS использует один обязательный, устанавливаемый по умолчанию показатель с максимальным значением пути 1024. Этот показатель является произвольным и обычно назначается администратором сети. Отдельный канал может иметь максимальное значение 64. Эти значения используются для поиска кратчайшего пути (наилучшего маршрута).

Протокол IS-IS также определяет три дополнительных показателя:

- величину задержки в канале (delay);
- коммуникационные затраты (expense) ;
- коэффициент ошибок канала (error).

Используя эти показатели, протокол IS-IS определяет интегральный показатель качества обслуживания (quality-of-service - QOS) и может вычислять маршруты через объединенную сеть.

Глава 3
Стандартные сетевые протоколы

3.1. Классификация протоколов

Протокол – это набор правил и технических процедур, регулирующих порядок выполнения некоторой связи между компьютерами в компьютерной сети. Протоколы работают на разных уровнях модели OSI. Один протокол может решать задачи нескольких смежных уровней модели OSI. Все протоколы можно разделить на три группы, в зависимости от услуг, которые они предоставляют смежным уровням или прикладным процессам семиуровневой модели OSI (рис. 3.1):

- Прикладные услуги.
- Транспортные услуги.
- Сетевые услуги.

Каждый протокол имеет определенное назначение, решает конкретные задачи и характеризуется такими показателями, как сложность, быстродействие, качество решения и надежность.

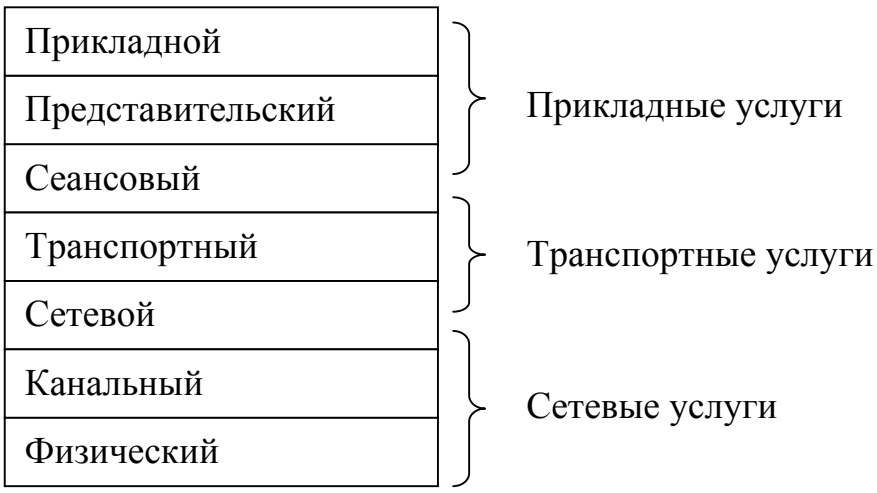


Рис. 3.1. Уровни модели OSI и типы протоколов

Протоколы, взаимодействующие между собой, объединяются в *стеки*. Процесс привязки определяет очередность выполнения протоколов стека операционной системой. Чтобы протокол мог взаимодействовать с платой сетевого

адаптера, он также должен быть привязан к ней. Детали процесса привязки рассмотрены в главе 5.

На компьютере-отправителе протоколы стека выполняются сверху вниз, т.е. от протоколов верхних уровней модели OSI к протоколам нижних уровней:

- данные сообщения разбиваются на небольшие блоки, называемые пакетами;
- к пакетам добавляется информация, позволяющая компьютеру- получателю определить, что данные предназначены именно ему и то, как их следует обрабатывать;
- пакеты подготавливаются для передачи в сетевой кабель через плату сетевого адаптера.

На компьютере-получателе протоколы стека выполняются снизу вверх:

- пакеты принимаются из сетевого кабеля и через плату сетевого адаптера поступают в компьютер;
- их пакетов удаляется служебная информация;
- данные из пакетов копируются в буфер и объединяются в нужном порядке;
- сообщение, сформированное в буфере, передается приложению.

Наибольшей популярностью пользуются следующие стандартные стеки протоколов:

- набор протоколов ISO/OSI (в качестве эталонной модели);
- IBM System Network Architecture (SNA);
- Digital DECnet;
- Novell NetWare;
- Apple Apple Talk;
- набор протоколов Internet, TCP/IP.

Протоколы, обеспечивающие прикладные услуги

Протоколы, обеспечивающие прикладные услуги, соответствуют трем верхним уровням модели OSI (табл. 3.1). Они контролируют взаимодействие приложений в сетевой среде.

Таблица 3.1. Протоколы, обеспечивающие прикладные услуги

Протокол	Назначение
APPC (Advanced Program-to-Program Communication)	Одноранговый SNA-протокол фирмы IBM, используемый в основном на AS/400
FTAM (File Transfer Access and Management)	Протокол OSI для доступа к файлам
X.400	Протокол CCITT для международного обмена эл. почтой
X.500	Протокол CCITT для служб файлов и каталогов на нескольких системах
SMTP (Simple Mail Transfer Protocol)	Протокол Internet для обмена электронной почтой
FTP (File Transfer Protocol)	Протокол Internet для пересылки файлов
SNMP (Simple Network Management Protocol)	Протокол Internet для мониторинга сети и сетевых компонентов
Microsoft SMBs (Server Message Blocks)	Блоки сообщений сервера и клиентские оболочки или редиректоры
NCP (Novell NetWare Core Protocol)	Блоки сообщений сервера и клиентские оболочки или редиректоры фирмы Novell
Apple Talk и Apple Share	Набор сетевых протоколов фирмы Apple
AFP (Apple Talk Filing Protocol)	Протокол удаленного доступа к файлам фирмы Apple
DAP (Data Access Protocol)	Протокол доступа к файлам сетей DECnet

Протоколы, обеспечивающие транспортные услуги

Протоколы, обеспечивающие транспортные услуги, поддерживают сеансы связи между компьютерами и надежный обмен данных между ними. Наиболее популярные транспортные протоколы приведены в табл. 3.2.

Протоколы, обеспечивающие сетевые услуги

Сетевые протоколы обеспечивают услуги связи. Для этого они управляют данными различного типа, связанными с адресацией и маршрутизацией пакетов, контролем ошибок и запросами на повторную передачу. Наиболее популярные сетевые протоколы приведены в табл. 3.3.

Таблица 3.2. Протоколы, обеспечивающие транспортные услуги

Протокол	Назначение
TCP (Transmission Control Protocol)	TCP/IP-протокол для гарантированной доставки данных, разбитых на последовательность фрагментов
SPX (Sequential Packet Exchange)	Часть комплекта протоколов IPX/SPX фирмы Novell для данных, разбитых на последовательность фрагментов
NWLink	Реализация протоколов IPX/SPX от фирмы Microsoft
NetBEUI (NetBIOS (Network Basic Input Output System) Extended User Interface)	Установление связи между компьютерами (NetBIOS) и предоставление транспортных услуг верхним уровням
ATP (Apple Talk Transaction Protocol), NBP (Name Binding Protocol)	Протоколы сеансов связи и транспортировки данных фирмы Apple

Таблица 3.3. Протоколы, обеспечивающие сетевые услуги

Протокол	Назначение
IP (Internet Protocol)	Протокол Internet сетевого уровня, контролирует адресацию узлов и маршрутизацию
IPX (Internet Packet Exchange)	Протокол для ЛВС, поддерживает маршрутизацию пакетов
NWLink	Реализация протоколов IPX/SPX от фирмы Microsoft
NetBEUI	Устанавливает связи между компьютерами (NetBIOS) и предоставляет транспортные услуги верхним уровням
DDP (Datagram Delivery Protocol)	Протокол доставки датаграмм

Протоколы Internet

Протоколы, обеспечивающие связность гетерогенных (разнородных) вычислительных сетей, лежат в основе сетевых операционных систем. В середине 1970-х гг. агентство по внедрению научно-исследовательских проектов при Министерстве обороны США (DARPA) финансировало исследования, проводимые Стэнфордским университетом и компаниями Bolt, Beranek и Newman (BBN) с

целью создания ряда протоколов связи для объединения гетерогенных систем на основе технологии сети с коммутацией пакетов. Практическая задача заключалась в обеспечении связи между научно-исследовательскими институтами в США.

Результатом этих исследований был комплект протоколов Internet, из которых наиболее известными являются Transmission Control Protocol (TCP) и Internet Protocol (IP). Протоколы Internet получили наибольшее распространение для создания как крупномасштабных, так и локальных сетей.

Комплект протоколов Internet состоит как из протоколов низших уровней (TCP и IP), так и протоколов верхних уровней (почта, эмуляция терминалов, передача файлов). В табл. 3.4 представлены наиболее важные протоколы Internet с указанием их соответствия уровням эталонной модели OSI.

Таблица 3.4. Протоколы Internet

<table>
<tr><td>Уровень модели OSI</td><td colspan="2">Комплект протоколов Internet</td></tr>
<tr><td>Прикладной</td><td rowspan="3">FTP, Telnet,
SMTP, SNMP</td><td>NFS</td></tr>
<tr><td>Представительский</td><td>XDR</td></tr>
<tr><td>Сеансовый</td><td>RCP</td></tr>
<tr><td>Транспортный</td><td colspan="2">TCP, UDP</td></tr>
<tr><td rowspan="2">Сетевой</td><td colspan="2">Протоколы маршрутизации IP, ICMP</td></tr>
<tr><td colspan="2">ARP, RARP</td></tr>
<tr><td>Канальный</td><td colspan="2" rowspan="2">Не специфицированы</td></tr>
<tr><td>Физический</td></tr>
</table>

3.2. Протоколы Internet сетевого уровня

В комплекте протоколов Internet сетевого уровня протокол IP является основным и выполняет следующие функции:

- маршрутизация пакетов в объединенных сетях;
- разбиение дейтаграмм на фрагменты (фрагментация) и обратная их сборка
- сообщения об ошибках.

Формат пакета IP представлен на рис. 3.2.

<table>
<tr><td>Номер версии</td><td>Длина заголовка</td><td>Тип услуги</td><td colspan="2">Общая длина</td></tr>
<tr><td colspan="3">Идентификатор дейтаграммы</td><td>Флаги</td><td>Смещение фрагмента</td></tr>
<tr><td colspan="2">Срок жизни</td><td>Протокол высшего уровня</td><td colspan="2">Контрольная сумма заголовка</td></tr>
<tr><td colspan="5">Адрес источника</td></tr>
<tr><td colspan="5">Адрес пункта назначения</td></tr>
<tr><td colspan="5">Опции + заполнитель</td></tr>
<tr><td colspan="5">Данные (размер переменный)</td></tr>
</table>

Рис.3.2. Формат пакета IP

Заголовок пакета IP содержит следующие поля:

- номер версии протокола IP;
- длина заголовка дейтаграммы в 4-байтовых словах;
- тип услуги (указывает способ обработки дейтаграммы, требуемый конкретным протоколом высшего уровня);
- общая длина пакета IP в байтах (включая данные и заголовок);
- идентификатор дейтаграммы - число, обозначающее текущую дейтаграмму (используется для соединения фрагментов дейтаграммы);
- поле флагов определяет возможность разбиения дейтаграммы на фрагменты, а также служит указателем последнего фрагмента;
- смещение фрагмента;
- поле срока жизни (счетчик, значение которого постепенно уменьшается до нуля; для предотвращения зацикливания пакетов дейтаграммы с нулевым значением этого поля отвергаются);
- протокол высшего уровня, принимающий входящие пакеты после завершения обработки IP;
- контрольная сумма заголовка (обеспечивает его целостность);
- адрес источника (адрес узла-отправителя);
- адрес пункта назначения (адрес узла-получателя);
- опции (указывают факультативные возможности IP, например, защиту данных).

Заполнители в поле опций обеспечивают выравнивание длины заголовка IP-пакета. Поле данных содержит информацию высших уровней. Его длина равна разности общей длины пакета и длины заголовка.

Адресация IP

Адрес IP имеет длину 32 бита и разделяется на две или три части. Первая часть представляет адрес сети, вторая (если администратор сети принял решение о разделении сети на подсети) - адрес подсети, и третья - адрес хоста. Длины полей адреса сети, подсети и хоста являются переменными величинами.

Адресация IP обеспечивает пять классов сетей: A, B, C, D и E. Самые крайние левые биты адреса обозначают класс сети. Адреса IP записываются в формате десятичного числа с проставленными точками, например, 34.0.0.1.

Для некоторых сред, например ЛВС IEEE 802, физические адреса и IP-адреса определяются динамически с помощью протоколов ARP и RARP. Протокол разрешения адреса ARP (Address Resolution Protocol) использует широковещательные сообщения для определения , физические адреса (уровень MAC), соответствующего конкретному IP-адресу. ARP достаточо универсален и может работать практически любым методом доступа к носителю.

Протокол разрешения обратного адреса RARP (Reverse Address Resolution Protocol) использует широковещательные сообщения для определения IP-адреса, связанного с конкретным физическим адресом. RARP особенно необходим для начальной загрузки узлов, которые не знают своего IP-адреса, потому что не имеют дисковой памяти. Дополнительные сведения о классах сетей и адресации в Internet приведены в главе 4.

Маршрутизация Internet организована в соответствии с иерархическим принципом. Устройства маршрутизации в Internet называются маршрутизаторами (gateway). Поскольку устройства маршрутизации обычно называются маршрутизаторами, а термин “маршрутизатор” используется в другом смысле, будем использовать традиционное название этих устройств и для Internet.

Некоторые IP-маршрутизаторы используются для перемещения информации через одну конкретную группу сетей, называемую *автономной системой* (autonomous system), находящихся под одним и тем же административным управлением.

Внутренние IP-маршрутизаторы (interior routers) работают в пределах автономных систем и используют различные протоколы внутренней маршрутизации (interior gateway protocol - IGP). Маршрутизаторы, перемещающие информацию между автономными системами, называются *внешними маршрутизаторами* (exterior routers).

Протоколы маршрутизации IP выполняют динамическую маршрутизацию - dynamic routing (см. главу 2). Маршрутизатор IP определяет перемещения дейтаграмм IP через сеть по одной пересылке за раз. В начале перемещения весь маршрут не известен. В каждом промежуточном пункте по таблице маршрутизации определяется следующий пункт, вне зависимости от того, достигнет или нет пакет конечного пункта назначения. Другими словами, IP не информирует узел-источник о нарушении маршрутизации. Эту задачу решает другой протокол Internet , а именно протокол управляющих сообщений ICMP(Internet Control Message Protocol).

Протокол ICMP

Протокол ICMP выполняет следующие задачи:

- сообщает узлу-источнику об отказах маршрутизации;
- проверяет способности узлов образовывать повторное эхо в объединенной сети (сообщения Echo и Reply ICMP);
- стимулирует более эффективную маршрутизацию (с помощью сообщений Redirect ICMP - переадресации ICMP);
- информирует узел-источник о том, что некоторая дейтаграмма превысила назначенное ей время существования в пределах данной сети (сообщение Time Exceeded ICMP - "время превышено");
- обеспечивает для новых узлов возможность нахождения маски подсети, используемой в объединенной сети в данный момент.

Протокол IS-IS также является официальным протоколом маршрутизации IP; он рассматривается в главе 2.

3.3. Протоколы Internet транспортного уровня

Транспортный уровень Internet реализуется TCP и протоколом дейтаграмм пользователя UDP (User Datagram Protocol). TCP обеспечивает транспортировку данных с установлением соединения, в то время как UDP работает без установления соединения.

Протокол управления передачей TCP

Протокол TCP (Transmission Control Protocol) обеспечивает полностью дублированные, с подтверждением и управлением потоком данных, услуги для протоколов высших уровней. Он перемещает данные в непрерывном неструктурированном потоке, в котором байты идентифицируются по номерам последовательностей. TCP может также поддерживать многочисленные одновременные диалоги высших уровней. Формат пакета TCP представлен на рис. 3.3.

<table>
<tr><td colspan="2">Порт источника</td><td colspan="2">Порт пункта назначения</td></tr>
<tr><td colspan="4">Номер последовательности</td></tr>
<tr><td colspan="4">Номер подтверждения</td></tr>
<tr><td>Смещение данных</td><td>Резерв</td><td>Флаги</td><td>Окно</td></tr>
<tr><td colspan="3">Контрольная сумма</td><td>Указатель срочности</td></tr>
<tr><td colspan="4">Опции + заполнитель</td></tr>
<tr><td colspan="4">Данные (переменная длина)</td></tr>
</table>

Рис. 3.3. Формат пакета TCP

Рассмотрим назначение полей:

- порт источника (source port) обозначает точку, в которой конкретный процесс высшего уровня источника принимает услуги TCP;
- порт пункта назначения (destination port) обозначает порт процесса высшего уровня пункта назначения для услуг TCP;
- номер последовательности (sequence number) обозначает номер, первого байта данных в текущем сообщении (в некоторых случаях -номер исходной последовательности, который должен использоваться в предстоящей передаче);
- номер подтверждения (acknowledgement number), т.е. номер следующей ожидаемой отправителем последовательности байта данных, которую отправитель пакета ожидает для приема;
- смещение данных (data offset) - число 32-битовых слов в заголовке TCP;
- резерв (reserved) - зарезервировано для использования разработчиками протокола в будущем;
- флаги (flags) - содержит различную управляющую информацию;
- окно (window) - обозначает размер окна приема отправителя; (буферный объем, доступный для поступающих данных);

- контрольная сумма (checksum) указывает, был ли заголовок поврежден при пересылке;
- указатель срочности (urgent pointer) - указывает на первый байт срочных данных в пакете;
- опции (options) - обозначает различные факультативные возможности TCP.

Протокол дейтаграмм пользователя UDP

Протокол UDP используется в тех случаях, когда мощные средства обеспечения надежности протокола TCP не требуются. Реализация UDP намного проще, чем TCP. Заголовок UDP имеет четыре поля:

- порт источника (source port) - те же функции, что и в заголовке TCP;
- порт пункта назначения (destination port) - те же функции, что и в заголовке TCP;
- длина (length) - длина заголовка UDP и данных;
- контрольная сумма UDP (checksum UDP) - обеспечивает проверку целостности пакета (факультативная возможность).

3.4. Основы TCP/IP – связь протоколов Internet сетевого и транспортного уровней

Термин "TCP/IP" обозначает технологию межсетевого взаимодействия (технологию internet) на основе семейства протоколов TCP и IP. В это семейство входят протоколы UDP, ARP, ICMP, TELNET, FTP и многие другие. Сеть, использующая технологию internet, называется "internet". Глобальная сеть, объединяющая множество сетей с технологией internet, называется Internet.

IP-маршрутизация

Семейство протоколов TCP/IP предназначено для сети, состоящей из разнородных пакетных подсетей, объединенных посредством IP-маршрутизаторов. Каждая подсеть состоит из разнородных машин, имеет свою среду передачи и работает в соответствии со своими специфическими требованиями. Две машины, подключенные к одной подсети могут обмениваться пакетами: приняв пакет информации с соответствующим сетевым заголовком, подсеть доставляет

его по указанному адресу, не гарантируя обязательную доставку пакетов (не требуется, чтобы подсеть имела надежный сквозной протокол).

Если необходимо передать пакет между машинами, подключенными к разным подсетям, машина-отправитель посылает пакет в соответствующий IP-маршрутизатор, который подключается к подсети как обычный узел. Далее пакет направляется по определенному маршруту через систему маршрутизаторов и подсетей, пока не достигнет маршрутизатора, подключенного к той же подсети, в которой находится машина-получатель. Использование во всех узлах и маршрутизаторах межсетевого протокола IP решает проблему доставки пакетов. Таким образом обеспечивается дейтаграммный сервис на межсетевом уровне Internet. Этот уровень обеспечивает возможность стандартизации протоколов верхних уровней и является основой архитектуры TCP/IP.

Взаимодействие модулей, реализующих протоколы TCP/IP

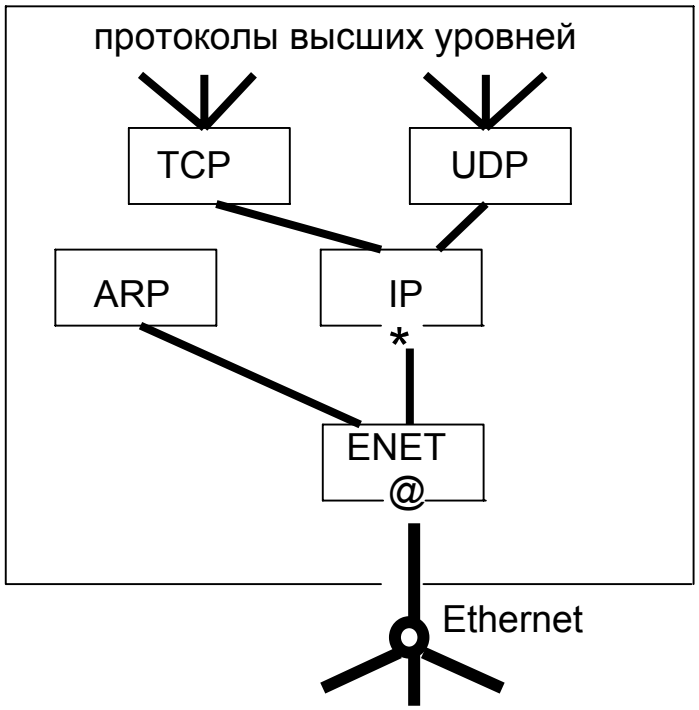


Рис. 3.4. Связь протоколов в узле TCP/IP

Рассмотрим структуру взаимодействия модулей, реализующих стек протоколов TCP/IP в каждом узле сети internet, изображенную на рис. 3.4. Изучение этой структуры поможет лучше понять технологию internet. На этом рисунке прямоугольники обозначают процессы обработки блоков данных (работу модулей протоколов или драйверов):

- TCP (Transmission Control Protocol) - протокол управления передачей;
- UDP (User Datagram Protocol) - протокол пользовательских дейтаграмм;
- ARP (Address Resolution Protocol) - адресный протокол;
- ENET - драйвер платы сетевого адаптера.

Предполагается, что физической средой передачи является Ethernet. Хотя технология internet поддерживает много различных сред передачи данных, среда Ethernet чаще всего служит физической основой для IP-сети. Линии, соединяющие прямоугольники, показывают пути передачи данных, а узел и расходящиеся линии внизу рисунка - концентратор сети Ethernet. Знак * обозначает IP-адрес, а @ - адрес узла сети Ethernet, или Ethernet-адрес.

Название блока данных, передаваемого по сети, зависит от того, на каком уровне стека протоколов он находится. Название программ обработки данных также зависит от того, с какими программами они взаимодействуют. Поэтому будем придерживаться в дальнейшем следующей терминологии:

- драйвер - программа, непосредственно взаимодействующая с сетевым адаптером;
- модуль - программа, взаимодействующая с драйвером, сетевыми прикладными программами или другими модулями. Драйвер сетевого адаптера обеспечивает сетевой интерфейс для модулей протоколов семейства TCP/IP;
- кадр - блок данных, с которым имеет дело сетевой интерфейс;
- IP-пакет - блок данных, поступающий из сетевого интерфейса в модуль IP;
- UDP-дейтаграмма - блок данных, поступающий из модуля IP в модуль UDP;
- TCP-сегмент (или транспортное сообщение) – блок данных, поступающий из модуля IP в модуль TCP;
- прикладное сообщение - блок данных на уровне сетевых прикладных процессов.

Рассмотрим прохождения блоков данных через стек протоколов, изображенный на рис. 3.4. Если используется протокол TCP, данные передаются между уровнем прикладных услуг и модулем TCP. Если на уровне прикладных услуг используется протокол передачи FTP, стек протоколов будет иметь вид FTP/TCP/IP/ENET. При использовании протокола UDP, данные передаются между уровнем прикладных услуг и модулем UDP. Если транспортными услугами UDP пользуется, например, "простой протокол управления сетью" SNMP (Simple Network Management Protocol), стек протоколов имеет вид: SNMP/ UDP/ IP/ ENET.

Модули протоколов TCP, UDP и драйвер Ethernet работают как мультиплексоры при продвижении блоков данных от нескольких протоколов верхнего уровня на один выход. При обработке поступающих блоков данных каждый такой модуль работает как демультиплексор: он направляет поток данных с одного входа на один из своих выходов в соответствии с полем типа в заголовке блока данных:

- данные Ethernet-кадра, поступившего на вход драйвера сетевого интерфейса Ethernet, могут быть направлены либо в модуль ARP , либо в модуль IP в соответствии с полем типа в заголовке Ethernet-кадра;
- данные IP-пакета, принятого модулем IP, могут быть переданы либо модулю TCP, либо UDP, что определяется полем "протокол" в заголовке IP-пакета;
- данные UDP-дейтаграммы, попавшей в модуль UDP, на основании значения поля "порт" в заголовке дейтаграммы передаются прикладной программе;
- TCP-сообщение, попавшее в модуль TCP, на основании значения поля "порт" в заголовке TCP-сообщения, передается соответствующей прикладной программе.

Продвижение данных от верхних уровней к нижним уровням модели OSI осуществляется просто, так как из каждого модуля существует только один путь вниз: данные от прикладного процесса проходят через модули TCP или UDP, после чего попадают в модуль IP и оттуда - на уровень сетевого интерфейса, причем каждый протокольный модуль добавляет к пакету свой заголовок, на основании которого машина, принявшая пакет, выполняет демультиплексирование.

Обратимся к примеру на рис. 3.4. Каждая машина имеет уникальный в пределах всей сети Internet четырехбайтный IP-адрес, обозначающий точку доступа к сети на интерфейсе модуля IP с драйвером. Каждая машина имеет также одну точку подключения к Ethernet: уникальный шестибайтный Ethernet-адрес каждого сетевого адаптера распознается драйвером, причем работающая машина всегда знает свой IP-адрес и Ethernet-адрес.

Работа с несколькими сетевыми интерфейсами

Одна машина может быть подключена одновременно к нескольким сегментам сети (средам передачи данных). Например, машина на рис.3.5 имеет два сетевых интерфейса Ethernet и, следовательно, 2 Ethernet-адреса. Из рис.3.5 также видно, что эта машина имеет также 2 IP-адреса. Из этого рисунка видно, что в рассматриваемом случае модуль IP выполняет более сложную функцию - сложнее, чем в первом примере, так как осуществляет мультиплексирование входных и выходных данных в обоих направлениях и может передавать (ретранслировать) данные между сетями. Такая функция, называется маршрутизацией. Данные, поступившие через один сетевой интерфейс, могут быть ретранслированы через другой сетевой интерфейс. Из рис.3.5 видно, что ретранслируемый пакет не поступает в модули TCP или UDP. Если модули TCP и UDP отсутствуют, то мы имеем дело с *машиной-маршрутизатором*. Если модули TCP и UDP имеются, то это *рабочая станция*.

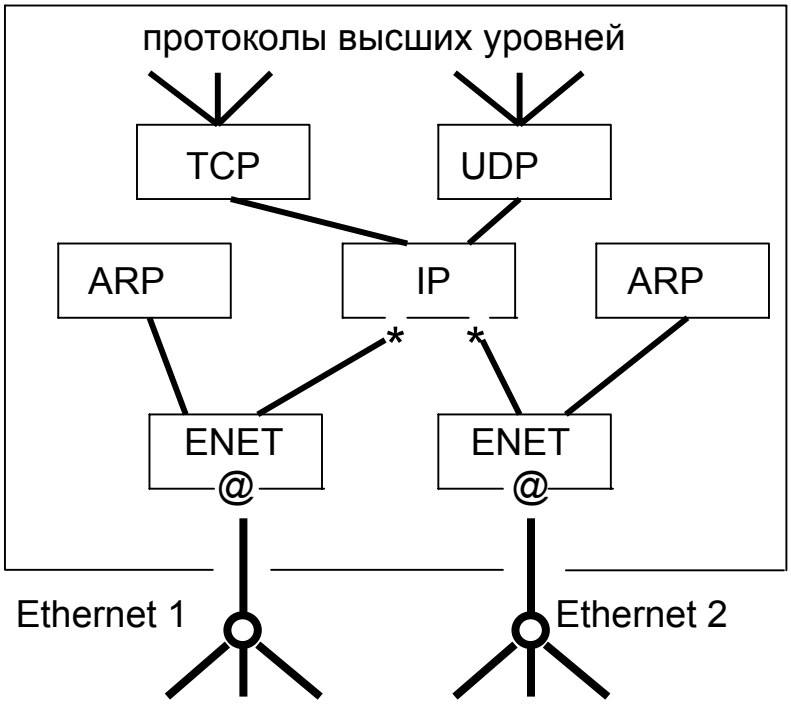


Рис. 3.5. Связь протоколов в узле с двумя интерфейсами

Протокол ARP (Address Resolution Protocol - адресный протокол)

При посылке IP-пакета Ethernet-адрес назначения определяется протоколом с помощью ARP-таблицы. Рассмотрим пример упрощенной ARP-таблицы (табл.3.5). В двух столбцах этой таблицы содержатся IP- и Ethernet-адреса. Если требуется преобразовать IP-адрес в Ethernet-адрес, то ищется запись с соответствующим IP-адресом. Преобразование выполняется только для отправляемых IP-пакетов, так как только в момент отправки создаются заголовки IP и Ethernet.

Таблица 3.5. Пример ARP-таблицы

IP-адрес	Ethernet-адрес
199.2.3.1	08:00:5B:22:51:E5
199.2.3.3	08:00:7C:42:C9:44
199.2.3.4	08:00:32:BB:CE:76

Все байты 4-байтного IP-адреса записываются десятичными числами, разделенными точками. При записи 6-байтного Ethernet-адреса каждый байт указывается в 16-ричной системе и отделяется двоеточием.

IP-адреса и Ethernet-адреса для какой-либо машины выбираются независимо. Ethernet-адрес выбирает производитель платы сетевого адаптера из выделенного для него по лицензии диапазона адресов. Если заменяется плата сетевого адаптера, то изменяется и Ethernet-адрес. IP-адрес выбирает менеджер сети с учетом положения машины в сети Internet. При перемещении машины в другую сеть Internet ее IP-адрес должен быть изменен. Поэтому невозможно сформулировать правило преобразования IP-адреса в Ethernet-адрес, кроме как на основе таблицы.

Рассмотрим последовательность преобразования адреса. Предположим, что прикладная программа отправляет сообщение в IP-адрес места назначения, пользуясь транспортными услугами TCP. Модуль TCP формирует транспортное сообщение через модуль IP. В результате создается IP-пакет, поступающий в драйвер Ethernet, причем IP-адрес получателя известен прикладной программе, модулю TCP и модулю IP. Для определения Ethernet-адреса, по которому должен быть отправлен пакет, используется ARP-таблица. Следует отметить, что каждая машина имеет отдельную ARP-таблицу для каждого своего сетевого интерфейса.

Заполнение ARP-таблицы

ARP-таблица заполняется автоматически, по мере необходимости. Если существующая ARP-таблица не содержит искомый IP-адрес, то модуль ARP генерирует широковещательный ARP-запрос и соответствующий исходящий IP-пакет ставится в очередь. Каждый сетевой адаптер принимает все широковещательные пакеты, а все драйверы Ethernet проверяют поле типа в принятом Ethernet-кадре и передают ARP-пакеты модулю ARP. Пакет ARP-запроса выглядит как показано в табл.3.6.

Таблица 3.6. Пример ARP-запроса

IP-адрес отправителя Ethernet-адрес отправителя	199.2.3.1 08:00:5B:22:51:E5
Искомый IP-адрес Искомый Ethernet-адрес	199.2.3.2 <пусто>

Поскольку искомый Ethernet-адрес отсутствует, ARP-запрос означает: "Сообщите мне ваш Ethernet-адрес, если ваш IP-адрес совпадает с искомым ". Пример пакета с ARP-ответом показан в табл.3.7.

Таблица 3.7. Пример ARP-ответа

IP-адрес отправителя Ethernet-адрес отправителя	199.2.3.2 08:00:4A:22:5A:CB
Искомый IP-адрес Искомый Ethernet-адрес	199.2.3.1 08:00:5B:22:51:E5

Этот пакет поступает в машину, сделавшую ARP-запрос. Драйвер этой машины проверяет поле типа в Ethernet-кадре и передает ARP-пакет модулю ARP. Модуль ARP анализирует ARP-пакет и добавляет запись в свою ARP-таблицу (табл.3.8).

Таблица 3.8. Обновленная ARP-таблица

IP-адрес	Ethernet-адрес
199.2.3.1	08:00:5B:22:51:E5
199.2.3.2	08:00:4A:22:5A:CB
199.2.3.3	08:00:7C:42:C9:44
199.2.3.3	08:00:32:BB:CE:76

На автоматическое обновление ARP-таблицы затрачивается несколько миллисекунд. Полностью порядок преобразования адресов выглядит так:

1. Для преобразования IP-адреса в Ethernet-адрес передаваемого IP-пакета используется ARP-таблица. Если ARP-таблица содержит преобразуемый IP-адрес, то переход к п.6.
2. По сети передается широковещательный ARP-запрос.
3. Исходящий IP-пакет ставится в очередь.
4. Если получен ARP-ответ, содержащий информацию о соответствии IP- и Ethernet-адресов, то эта информация заносится в ARP-таблицу. Если ARP-ответ не получен, т.е. машина с искомым IP-адресом не найдена, протокол IP уничтожает IP-пакеты, направляемые по этому адресу.
5. Для преобразования IP-адреса в Ethernet-адрес IP-пакета, стоящего в очереди, используется ARP-таблица.
6. Ethernet-кадр передается через сеть Ethernet по назначению.

3.5. Протоколы Internet высших уровней

Комплект протоколов Internet включает в себя большое число протоколов высших уровней:

- Протокол передачи файлов FTP (File Transfer Protocol) - обеспечивает способ перемещения файлов между компьютерными системами.
- Протокол Telnet - обеспечивает виртуальную терминальную эмуляцию.
- Протокол управления простой сетью SNMP (Simle Network Management Protocol) используется для сообщения об аномальных ситуациях в сети и установления значений допустимых порогов в сети.
- Протокол X Windows позволяет терминалу с интеллектом связываться с отдаленными компьютерами таким образом, как если бы они были непосредственно подключенными мониторами.

- Комбинация протоколов сетевой файловой системы NFS (Network File System), представление внешней информации XDR (eXternal Data Representation) и вызов процедуры обращений к отдаленной сети RPC (Remote Procedure Call) обеспечивают прозрачный доступ к ресурсам отдаленной сети.
- Простой протокол передачи почты SMTP (Simple Mail Transfer Protocol) обеспечивает механизм передачи электронной почты.

Эти и другие протоколы высших уровней используют базовые сетевые услуги TCP/IP и других протоколов Internet низших уровней.

Глава 4
Классы сетей и маршрутизация в Internet

4.1. IP-адресация и классы сетей

Старшие биты 4-байтного IP-адреса определяют номер IP-сети, а оставшиеся биты − номер узла . Как было сказано ранее, IP-адрес узла идентифицирует точку доступа модуля IP к сетевому интерфейсу, а не всю машину. IP-адреса машинам дает администратор сети в соответствии с тем, к каким IP-сетям они подключены.

IP-адреса разделяются на 5 классов, отличающихся количеством бит в цифровом адресе сети и цифровом адресе узла. Значение первого байта адреса определяет класс адреса (рис. 4.1). Соответствие классов адресов значениям первого байта и количество возможных IP-адресов каждого класса приведено в табл.4.1.

Рис. 4.1. Структура IP-адреса

Таблица 4.1. Характеристики классов адресов

Класс	Диапазон значений первого байта	Возможное количество сетей	Возможное количество узлов
A	1 - 126	126	16777214
B	128-191	16382	65534
C	192-223	2097150	254
D	224-239	-	2^{28}
E	240-247	-	2^{27}

Адреса класса A предназначены для больших сетей общего пользования. Возможное число сетей класса A равно 126, так как они используют всего 7 битов для поля адреса сети, однако они допускают большое количество цифровых адресов для узлов.

Адреса класса B используются в сетях среднего размера, например, сетях университетов и крупных компаний. Сети этого класса используют 14 битов для поля адреса сети и 16 битов для поля адреса узла. Тем самым обеспечивается хороший компромисс между адресным пространством сети и узла.

Адреса класса C используются в сетях с небольшим числом компьютеров. Для этих сетей выделяют 22 бита для поля адреса сети и только 8 битов для поля узла, поэтому число узлов, приходящихся на сеть, может стать ограничивающим фактором.

Адреса класса D используются при обращениях к группам машин. В адресах класса D четыре бита наивысшего порядка устанавливаются на значения 1,1,1 и 0.

Адреса класса E зарезервированы на будущее.

4.2. Выделение подсетей

Для обеспечения дополнительной гибкости администрирования сеть IP может быть разделена на более мелкие единицы, называемые подсетями (subnets), с каждой из которых можно работать как с обычной сетью TCP/IP. Как правило, подсеть соответствует одной физической сети, например, одной сети Ethernet или Token Ring. Таким образом единая IP-сеть организации может строиться как объединение подсетей.

Рассмотрим выделение подсетей на примере сетей класса B (рис.4.2). Предположим, что адрес сети представлен в виде десятичного числа с точками

Рис. 4.2. Выделение подсетей

128.10.0.0 (наличие одних нулей в поле узла обозначает всю сеть).

Если администратор сети решил использовать восемь битов для организации подсети, то третий байт адреса IP класса B используется как номер этой подсети. В рассматриваемом примере адрес 128.10.1.0 относится к сети 128.10, подсети 1; адрес 128.10.2.0 относится к сети 128.10, подсети 2, и т.д.

Число битов, занимаемых адресом подсети, выбирает администратор. Для задания этого числа протокол IP предусматривает использование маски подсети. Она используется сетевым программным обеспечением для выделения номера подсети из IP-адресов. Маска подсети содержит единицы во всех битах, кроме тех, которые определяют поле узла. Биты, определяющие номер узла, в маске подсети должны быть равны 0.

Если в IP-адресе класса B третий байт используется для задания номера подсети, то на его основе можно иметь 256 подсетей, в каждой из которых может быть до 254 узлов. Маска подсети в такой системе равна 255.255.255.0. Если же, например, требуется большее число подсетей с числом узлов не более 60 в каждой, то следует использовать маску 255.255.255.192. Поскольку 192 в двоичной системе 11000000, это позволяет иметь 1024 подсети и до 62 узлов в каждой, поскольку номера узлов 0 и "все единицы", как будет сказано ниже, используются особым образом.

Еще пример: при использовании 8 битов для организации подсети в сети класса A с адресом 34.0.0.0 маска подсети имеет вид 255.255.0.0. При использовании же 16 битов для организации подсети маска подсети принимает вид 255.255.255.0. Это позволяет иметь 256 подсетей, в каждой из которых может быть до 254 узлов. Обычно маска подсети указывается в файле стартовой конфигурации сетевого программного обеспечения. Протоколы TCP/IP позволяют также запрашивать эту информацию по сети.

Некоторые IP-адреса выделены для специального назначения. Признаки выделенных адресов показаны на на рис.4.3. В выделенных IP-адресах все нули соответствуют либо данному узлу, либо данной IP-сети. IP-адреса, состоящие из всех единиц, используются при широковещательных передачах. Для ссылок на всю IP-сеть в целом используется IP-адрес с нулевым цифровым адресом узла. В IP-сетях запрещается присваивать машинам IP-адреса, начинающиеся со 127. IP-адреса, первый байт которых равен 127, используются для тестирования программ и взаимодействия процессов в пределах одной машины. Когда про-

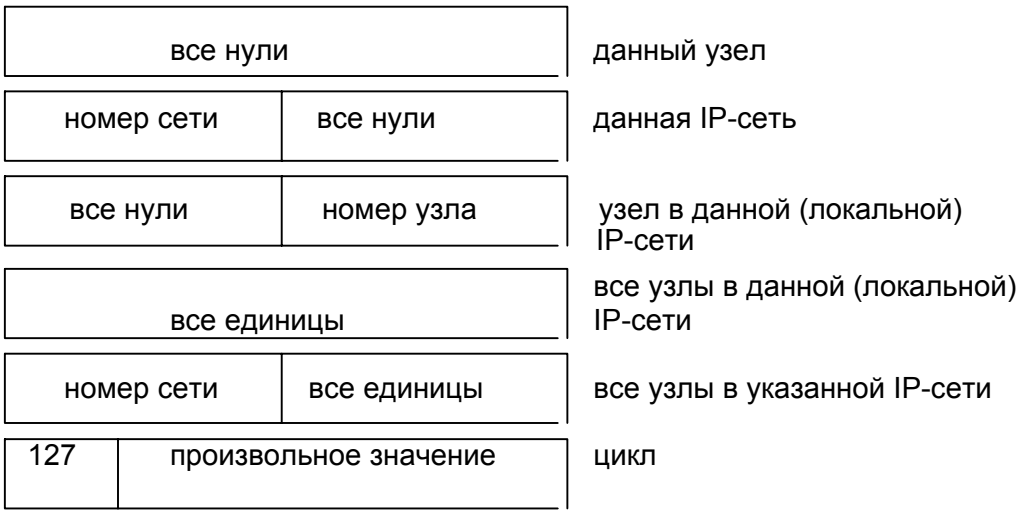


Рис. 4.3. Выделенные IP-адреса

грамма посылает данные по IP-адресу 127.0.0.1, то данные не передаются по сети, а возвращаются модулям верхнего уровня, как только что принятые.

4.3. Рекомендации по выбору IP-адресов

Для эксплуатации сети TCP/IP необходимо получить один или несколько официальных IP-адресов. IP-адреса предоставляются бесплатно, причем все оформление занимает около недели. Рекомендуется получить уникальный сетевой цифровой адрес вне зависимости от того, для чего предназначена регистрируемая сеть. Получение зарегистрированного цифрового адреса желательно, даже если создаваемая сеть не имеет связи с сетью Internet, для того, чтобы была гарантия, что в будущем - при включении в Internet или при подключении к сети другой организации - не возникнет конфликта адресов.

Выбор способа назначения IP-адресов сетевым машинам

Особое внимание при создании сети следует уделить выбору способа присвоения IP-адресов сетевым машинам с учетом перспектив развития, а именно выбору класса адреса для создаваемой сети и выделению подсетей.

Следует учитывать, что когда к сети подключено несколько сотен машин, изменение адресов становится почти невозможным.

Для небольших сетей с числом узлов до 254 используются цифровые адреса сетей класса С. Организации с большим числом машин могут использовать два варианта:

- получить несколько цифровых адресов класса С
- получить один цифровой адрес класса В и использовать в рамках одной организации подсети.

Первый вариант предполагает, что для каждой физической сети организации выделяется свой цифровой адрес класса С. Главный недостаток такого решения состоит в том, что структура IP-сети организации становится видимой для всего мира и машины вне рассматриваемой организации должны поддерживать записи о маршрутах доступа к каждой из IP-сетей класса С, обслуживающих данную организацию. Информация об изменениях IP-сети должна быть учтена в каждой из машин, поддерживающих маршруты доступа к данной IP-сети. Менее существенный недостаток использования нескольких адресов класса С для одной организации в рассматриваемом случае заключается в пустой трате сетевых цифровых адресов.

Второй вариант - использование одной сети класса В для всей организации и выделение на ее основе подсетей - предпочтительное решение. Как уже было сказано выше, для IP-адресов класса В первые два байта являются номером сети. Использование оставшейся части IP-адреса, а именно конфигурация подсетей, описывается в файлах, определяющих маршрутизацию IP-пакетов. Это описание является локальным для рассматриваемой организации и не видно вне ее. Все машины вне организации видят одну большую IP-сеть. Следовательно, они должны поддерживать только маршруты доступа к маршрутизаторам, соединяющим объединенную IP-сеть организации с остальным миром, а изменения, происходящие в IP-сети организации, не видны вне ее. Благодаря этому облегчается сетевое администрирование, появляется возможность безболезненно модифицировать и развивать сеть организации (добавлять новые подсети, маршрутизаторы и т.п.).

Например, предположим, что имеется сеть Ethernet, охватывающая три здания, причем в перспективе ожидается увеличение числа машин, подключен-

ных к этой сети и разделение ее на подсети. В этом случае имеет смысл назначить одной физической сети несколько цифровых адресов подсетей - по одному на здание. Такая адресация облегчит администрирование, поскольку позволит сразу определить, где находится та или иная машина, и отпадет необходимость менять IP-адреса, когда произойдет разделение сети.

4.4. IP-маршрутизация

Понимание работы межсетевого протокола IP необходимо для успешного администрирования и сопровождения IP-сетей. Модуль IP и его таблица маршрутизации являются основным элементом межсетевого протокола IP. При статической маршрутизации содержание таблицы определяется администратором сети. Протокол IP использует эту таблицу при принятии решений о маршрутизации IP-пакетов. Ошибки при установке маршрутизации могут заблокировать межсетевое взаимодействие.

Рассмотрим, как используется таблица маршрутизации на примере *прямой маршрутизации*. На рисунке 4.4 показана простая IP-сеть, состоящая из 3-х машин: A, B и C. Администратор сети присваивает машинам IP-адреса. Каждый сетевой адаптер этих машин имеет свой уникальный Ethernet-адрес. Стек протоколов TCP/IP каждой машины такой же как на рис. 3.4.

Предположим, что машина A посылает IP-пакет машине B. В этом случае, как показано в табл.4.2, заголовок IP-пакета содержит в поле отправителя IP-адрес узла A, а заголовок Ethernet-кадра содержит в поле отправителя Ethernet-адрес A. Кроме этого, IP-заголовок содержит в поле получателя IP-адрес узла B, а Ethernet-заголовок содержит в поле получателя Ethernet-адрес B.

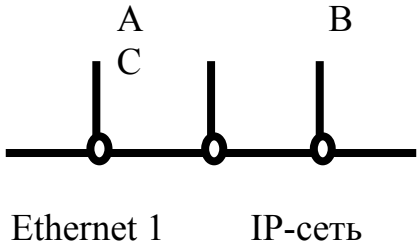


Рис. 4.4. Простая IP-сеть

Таблица 4.2. Адреса в Ethernet-кадре, передающем IP-пакет от A к B

Адрес	Отправитель	Получатель
IP-заголовок	A	B
Ethernet-заголовок	A	B

В этом примере расход ресурсов на создание, передачу и обработку IP-заголовка протоколом IP не связан с выполнением полезной функции межсете-

вого взаимодействия, поскольку модуль IP машины B, получив IP-пакет от машины A, сопоставляет IP-адрес места назначения со своим и в случае их совпадения передает дейтаграмму протоколу верхнего уровня. В этом и состоит прямая маршрутизация при взаимодействии машины A с машиной B.

Рассмотрим пример сети Internet, состоящей из трех сетей Ethernet, объединенных IP-маршрутизатором D (рис. 4.5). Каждая Ethernet-сеть включает четыре машины, имеющие свои собственные IP- и Ethernet-адреса.

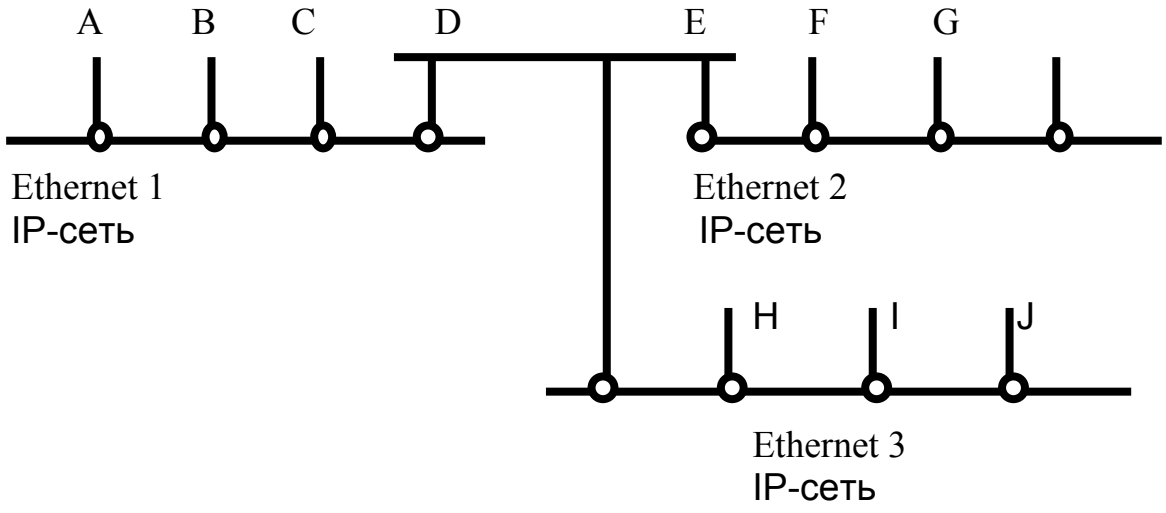


Рис. 4.5. IP-сеть, объединяющая три подсети

Все машины, за исключением D , имеют стек протоколов, показанный на рисунке 3.4. Маршрутизатор D соединяет все три сети и, следовательно, имеет три IP-адреса и три Ethernet-адреса. Поэтому маршрутизатор D имеет стек протоколов TCP/IP, аналогичный тому, что показан на рис.3.5, но вместо двух модулей ARP и двух драйверов, он содержит три модуля ARP и три драйвера Ethernet при одном модуле IP. Каждая сеть Ethernet имеет уникальный цифровой адрес, называемый цифровым IP-адресом сети. Цифровые IP-адреса создает администратор сети. Однако, на рис.4.5 вместо цифровых IP-адресов сетей показаны только их имена.

При всех взаимодействиях между машинами, подключенными к одной IP-сети, используется прямая маршрутизация, рассмотренная в предыдущем примере. Например, если машина A посылает IP-пакет машине B, передача осуществляется в пределах одной сети и, следовательно, используется прямая маршрутизация.

Если маршрутизатор D взаимодействует с машиной A, E или H, то это прямое взаимодействие. Если же, например, машина A взаимодействует с машинами, включенными в другую IP-сеть, то взаимодействие уже не прямое, а косвенное, поскольку машина A должна использовать маршрутизатор D для ретрансляции IP-пакетов в другую IP-сеть. Маршрутизация IP-пакетов, выполняемая модулями IP, обслуживает единообразно протоколы вышерасположенных уровней модели OSI и поэтому может быть названа прозрачной для модулей TCP, UDP и прикладных процессов.

Предположим, что машина A отправляет машине IP-пакет E (табл.4.3). В этом случае:

- IP- и Ethernet-адрес отправителя это соответствующие адреса A;
- IP-адрес места назначения является адресом E, но поскольку модуль IP в A посылает IP-пакет через D, Ethernet-адрес места назначения является адресом D.

Таблица 4.3. Адреса в Ethernet-кадре, содержащем IP-пакет от A к E (до маршрутизатора D)

Адрес	Отправитель	Получатель
IP-заголовок	A	E
Ethernet-заголовок	A	D

Далее, модуль IP маршрутизатора D получает IP-пакет и, определив, что IP-адрес места назначения не совпадает с его IP-адресом, направляет этот IP-пакет непосредственно к E (табл.4.4).

Таблица 4.4. Адреса в Ethernet-кадре, содержащем IP-пакет от A к E (после маршрутизатора D)

Адрес	Отправитель	Получатель
IP-заголовок	A	E
Ethernet-заголовок	D	E

Таким образом, в случае прямой маршрутизации IP- и Ethernet-адреса отправителя соответствуют адресам того узла, который послал IP-пакет, а IP- и Ethernet-адреса места назначения соответствуют адресам получателя. При косвенной маршрутизации IP- и Ethernet-адреса не образуют таких пар.

В рассмотренном примере сеть Internet была очень простой: несколько сетей Ethernet объединены маршрутизатором для того, чтобы локализовать широковещательный трафик в каждой сети. Реальные сети, как правило, сложнее,

содержат несколько маршрутизаторов, шлюзов и несколько типов физических сред передачи.

Правила (алгоритм) маршрутизации в модуле IP:

- Для отправляемых IP-пакетов, поступающих от модулей верхнего уровня, модуль IP определяет способ доставки - прямой или косвенный - и на основании результатов поиска в таблице маршрутизации выбирает сетевой интерфейс.
- Для принимаемых IP-пакетов, поступающих от сетевых драйверов, модуль IP решает, нужно ли ретранслировать IP-пакет по другой сети или необходимо передать его на верхний уровень. Принимаемый IP-пакет никогда не ретранслируется через тот же сетевой интерфейс, через который он был принят. Ретранслируемые пакеты обрабатываются далее также, как и отправляемые IP-пакеты.
- Решение о маршрутизации принимается до того, как IP-пакет передается сетевому драйверу, и до того, как происходит обращение к ARP-таблице.

4.5. Использование имен для узлов и сетей

Для удобства администрирования удобно присвоить машинам имена, наряду с цифровыми IP-адресами. Например, машине с IP-адресом 199.2.3.1 дается имя kappa. В больших сетях эта информация о соответствии цифровых IP-адресов именам хранится на сервере и доступна по сети. В небольших сетях такая информация может храниться в соответствующем файле (назовем его "hosts") на каждом узле. Например, несколько строк из файла "hosts" могут выглядеть так:

```
199.2.3.1       kappa
199.2.3.2       lambda
199.2.3.3       rho
199.2.3.4       sigma
199.2.4.2       tau
199.2.5.2       omega
```

В первом столбце указан IP-адрес, во втором - название машины. Как правило, файлы "hosts" могут быть одинаковы на всех узлах. Заметим, узел sigma представлен в файле "hosts" всего одной записью, хотя он имеет три IP-адреса (рис.4.6). Узел sigma доступен по любому из указанных на рис.4.6 IP-адресов, причем какой из них используется, не имеет значения. Когда узел sigma полу-

чает IP-пакет и проверяет IP-адрес места назначения, то он опознает любой из трех своих IP-адресов.

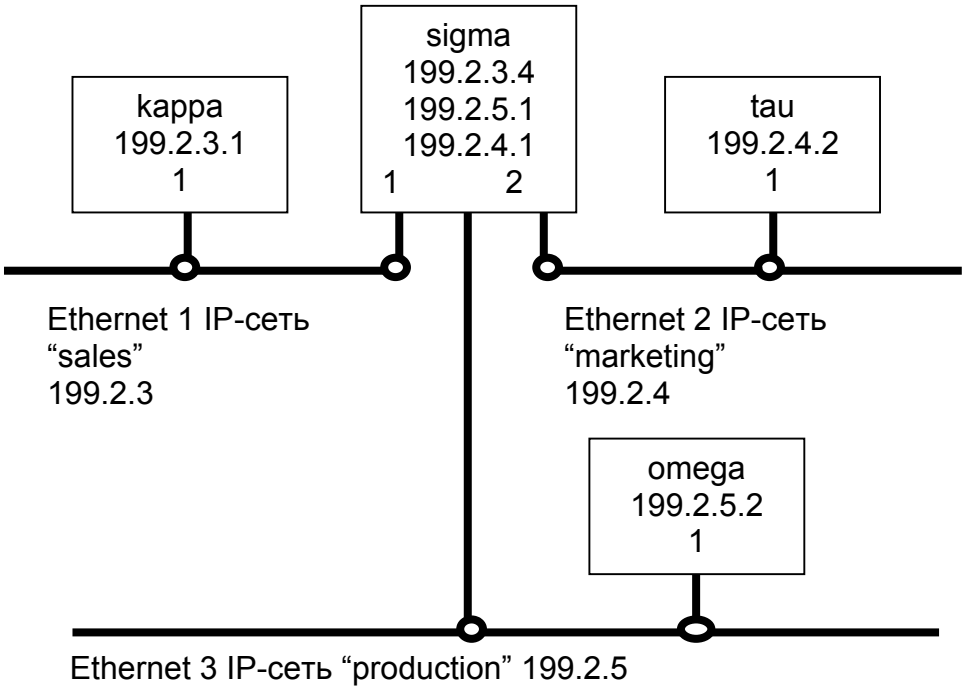


Рис. 4.6. IP-сеть, состоящая из трех подсетей

Для удобства администрирования удобно также присвоить имена сетям, наряду с цифровыми IP-адресами. Например, файл "networks", фиксирующий соответствие цифровых IP-адресов именам трех сетей, может иметь следующий вид:

```
199.2.3         sales
199.2.4         marketing
199.2.5         production
```

В первом столбце указан цифровой адрес сети, во втором - имя сети. В данном примере kappa является узлом номер 1 в сети sales, lambda является узлом номер 2 в сети sales и т.д.

Файл hosts, приведенный выше, достаточен для пользователей, но для администрирования объединенной сети удобнее иметь названия всех сетевых интерфейсов. Например, в файле hosts можно заменить строку, относящуюся к узлу sigma, тремя строками, дающими каждому IP-адресу узла sigma символьное имя:

```
199.2.3.4       salesnetrouter      sigma
199.2.4.1       marknetrouter
199.2.5.1       prodnetrouter
```

В приведенной записи первый IP-адрес имеет два имени: "salesnetrouter" и "sigma", которые являются синонимами, причем имя "sigma" предназначено для пользователей как общеупотребительное имя машины, а остальные три имени - для администрирования сети.

Файлы hosts и networks не нужны непосредственно для работы объединенной сети, но облегчают ее администрирование и могут использоваться в прикладных программах.

IP-таблица маршрутизации. Для выбора сетевого интерфейса, через который отправляется IP-пакет, модуль IP осуществляет поиск в таблице маршрутизации. Ключом поиска служит номер IP-сети, выделенный из IP-адреса получателя IP-пакета.

Таблица маршрутизации содержит одну строку для каждого маршрута. Основными столбцами таблицы маршрутизации являются цифровой адрес сети, флаг прямой или косвенной маршрутизации, IP-адрес маршрутизатора и цифровой адрес сетевого интерфейса. Эта таблица используется модулем IP при обработке каждого отправляемого IP-пакета. Содержание таблицы маршрутизации определяется администратором сети, который присваивает машинам IP-адреса. Как правило, система позволяет изменить таблицу маршрутизации с помощью команды "route".

Прямая маршрутизация. Рассмотрим маршрутизацию в одной физической сети, показанной на рис.4.7. В данном простом примере оба узла сети, kappa и lambda, имеют одинаковые таблицы маршрутизации (табл.4.5).

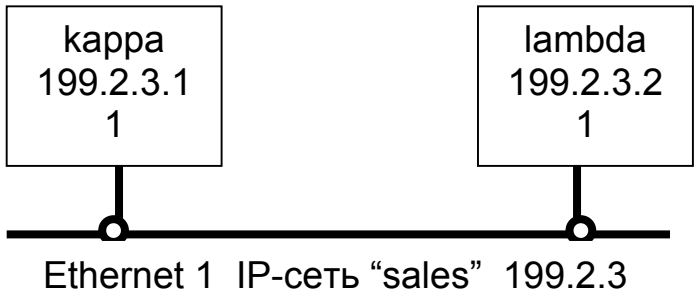


Рис. 4.7. Одна физическая сеть

Таблица 4.5. Пример таблицы маршрутизации

Сеть	Флаг вида маршрутизации	Маршрутизатор	Цифровой адрес интерфейса
sales	прямая	<пусто>	1

Для сравнения ниже представлена та же таблица, в которой вместо названия сети указан ее цифровой адрес (табл. 4.6).

Таблица 4.6. Пример таблицы маршрутизации с цифровыми адресами сетей

Сеть	Флаг вида маршрутизации	Маршрутизатор	Цифровой адрес интерфейса
199.2.3	прямая	<пусто>	1

Прямая маршрутизация выполняется в следующей последовательности. Если узел kappa посылает IP-пакет узлу lambda, то IP-адрес места назначения равен IP-адресу lambda (199.2.3.2). Модуль IP узла kappa с помощью маски подсети выделяет номер сети из IP-адреса и ищет соответствующую ему строку в таблице маршрутизации. В рассматриваемом случае подходит первая и единственная строка. Остальная информация в найденной строке указывает на то, что машины этой сети доступны напрямую через интерфейс номер 1. С помощью ARP-таблицы выполняется преобразование IP-адреса в соответствующий Ethernet-адрес, и через интерфейс 1 Ethernet-кадр посылается узлу lambda.

Если прикладная программа попытается послать данные по IP-адресу, который не принадлежит сети sales, то модуль IP не сможет найти соответствующую запись в таблице маршрутизации. В этом случае модуль IP отбрасывает IP-пакет и может выдать сообщение об ошибке "Сеть не доступна".

Косвенная маршрутизация. Рассмотрим более сложный порядок маршрутизации в IP-сети, изображенной на рис. 4.6. Рассмотрим табл. 4.7 маршрутизации в узле kappa.

Таблица 4.7. Таблица маршрутизации в узле kappa

Сеть	Флаг вида маршрутизации	Маршрутизатор	Номер интерфейса
sales	прямая	<пусто>	1
marketing	косвенная	salesnetrouter	1
production	косвенная	salesnetrouter	1

В табл. 4.8 маршрутизации для узла kappa даны IP-адреса вместо названий узлов.

Таблица 4.8. Таблица маршрутизации в узле kappa (с цифровыми адресами)

Сеть	Флаг вида маршрутизации	Маршрутизатор	Номер интерфейса
199.2.3	прямая	<пусто>	1
199.2.4	косвенная	199.2.3.4	1
199.2.5	косвенная	199.2.3.4	1

В столбце "маршрутизатор" таблицы маршрутизации узла kappa указывается IP-адрес точки соединения узла sigma с сетью sales.

Косвенная маршрутизация выполняется в следующей последовательности. Пусть узел kappa посылает IP-пакет узлу tau. Этот пакет находится в модуле IP узла kappa, и IP-адрес места назначения равен IP-адресу узла tau (199.2.4.2). Модуль IP выделяет номер сети (199.2.4) из IP-адреса узла и ищет соответствующую ему строку в таблице маршрутизации. Соответствие находится во второй строке.

Запись в этой строке указывает на то, что машины требуемой сети доступны через маршрутизатор salesnetrouter. Модуль IP в узле kappa осуществляет поиск в ARP-таблице, с помощью которого определяет Ethernet-адрес, соответствующий IP-адресу salesnetrouter. Затем IP-пакет, содержащий IP-адрес места назначения tau, посылается через интерфейс 1 маршрутизатору salesnetrouter.

IP-пакет принимается сетевым интерфейсом в узле sigma и передается модулю IP. Проверяется IP-адрес места назначения, и, поскольку он не соответствует ни одному из собственных IP-адресов sigma, маршрутизатор решает ретранслировать IP-пакет.

Модуль IP в узле sigma выделяет номер из IP-адреса места назначения IP-пакета (199.2.4) и ищет соответствующую запись в таблице маршрутизации (табл.4.9 или табл.4.10) для этого узла.

Таблица 4.9. Таблица маршрутизации в узле sigma

Сеть	Флаг вида маршрутизации	Маршрутизатор	Номер интерфейса
sales	прямая	<пусто>	1
marketing	прямая	<пусто>	2
production	прямая	<пусто>	3

Таблица 4.10. Таблица маршрутизации в узле sigma (с номерами)

Сеть	Флаг вида маршрутизации	Маршрутизатор	Номер интерфейса
199.2.3	прямая	<пусто>	1
199.2.4	прямая	<пусто>	2
199.2.5	прямая	<пусто>	3

Соответствие находится во второй строке. Теперь модуль IP напрямую посылает IP-пакет узлу tau через интерфейс 2. Пакет содержит IP- и Ethernet-адреса места назначения равные tau. Узел tau принимает IP-пакет, и его модуль IP проверяет IP-адрес места назначения. Он соответствует IP-адресу tau, поэтому содержащееся в IP-пакете сообщение передается протокольному модулю верхнего уровня.

Глава 5
Сетевое программное обеспечение

5.1. Сетевое ПО и операционные системы

Сетевое программное обеспечение (ПО) служит для управления ресурсами всей компьютерной сети. Основные функции сетевого ПО:

- связывает все компьютеры и периферийные устройства в сети;
- координирует работу всех компьютеров и периферийных устройств в сети;
- обеспечивает защищенный доступ к данным и устройствам в сети.

В состав сетевой ОС входят два основных компонента:

- сетевое ПО компьютеров-клиентов;
- сетевое ПО компьютеров-серверов.

Сетевое ПО компьютеров-клиентов

При автономной работе компьютера запрос на выполнение некоторых действий передается через локальную шину на процессор компьютера. При работе в сетевой среде запрос, относящийся к удаленному серверу, из локальной шины должен быть направлен в сеть и отослан на удаленный сервер.

ПО клиента включает в себя так называемый *редиректор* (redirector), который также может называться *оболочкой* (shell) или *запросчиком* (requester). Переадресация запросов выполняется редиректором. Редиректор – это небольшая программа сетевой ОС, которая выполняет следующие действия:

- перехватывает запросы в компьютере;
- определяет куда следует направить запрос: на локальную шину или в сеть для пересылки на удаленный сервер.

Редиректор может посылать запрос как к компьютерам, так и к сетевым периферийным устройствам. Например, редиректор может перехватывать задания на печать, адресуемые в LPT1 или COM1, и тем самым направлять их на соответствующий сетевой принтер.

Сетевое ПО компьютеров-серверов

Серверное ПО обеспечивает совместное использование ресурсов и координирует различные уровни доступа. Оно дает возможность всем сетевым компьютерам совместно использовать данные сервера и его периферийные устройства.

Существует два варианта реализации сетевого ПО:

- сетевое ПО является дополнением к существующей ОС;
- сетевое ПО интегрировано в сетевую ОС.

По первому принципу организована сетевая ОС Novell NetWare для локальных ЛВС или Microsoft LAN Manager. Это ПО позволяет включать в сеть рабочие станции с такими локальными ОС, как MS-DOS, UNIX и OS/2.

Например, ОС NetWare является мультизадачной ОС реального времени для работы в ЛВС с централизованным управлением. ОС NetWare состоит из:

- ядра, размещаемого на файл-сервере;
- сетевых утилит, размещаемых на файл-сервере;
- сетевых оболочек рабочих станций.

Сетевая оболочка ОС NetWare загружается в оперативную память рабочей станции как резидентная программа и выполняет функции запросчика (requester), т.е. редиректора запросов. ОС NetWare использует два основных протокола:

- IPX (Internetwork Packet eXchange) - базовый протокол сетевого уровня. Обеспечивает обмен блоками данных без предварительного установления соединения и без последующего подтверждения правильности доставки;
- SPX (Sequnced Packet eXchange) - устанавливает связь между рабочими станциями перед началом обмена и гарантирует доставку пакетов, давая подтверждение на каждый правильно доставленный пакет или запрос на повторную передачу при обнаружении ошибки.

В современных сетевых операционных системах автономная и сетевая ОС скомбинированы в одну ОС, которая поддерживает функционирование, как автономного компьютера, так и целой сети. Например, по этому принципу построена сетевая ОС Windows NT Server. Серверное ПО обеспечивает совместное использование ресурсов и координирует различные уровни доступа. Оно

дает возможность всем сетевым компьютерам совместно использовать данные сервера и его периферийные устройства. Администратор сети, через сервер, управляет и пользователями и сетью. Он может:

- добавлять в список пользователей сети новых пользователей;
- предоставлять привилегии отдельным пользователям сети или снять эти привилегии;
- удалять определенных пользователей из списка пользователей.

5.2. Сети с компонентами от разных производителей

ОС сервера, ОС клиента и редиректор должны быть совместимы. Например, на сервере может быть установлена ОС Microsoft Windows NT Server, а на клиентах Novell NetWare, Apple Macintosh и Microsoft Windows 95. Проблема совместимости при взаимодействии нескольких ОС может решаться как со стороны клиента, так и со стороны сервера.

Решение со стороны клиента заключается в том, что для каждой ОС (каждого ресурса) устанавливается соответствующий редиректор. Например, на рис. 5.1 показана рабочая станция Windows NT, использующая редиректор Microsoft для доступа к серверу Novell NetWare, установленный поверх Windows NT, а также редиректор Microsoft для доступа к сетевой среде Windows NT.

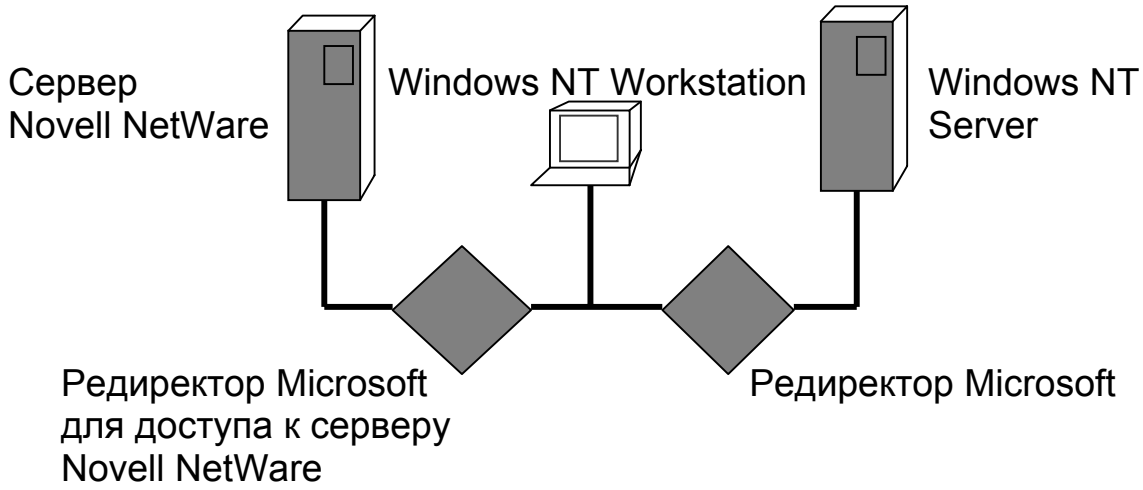


Рис. 5.1. Windows NT Workstation использует два редиректора

Решение со стороны сервера заключается в установке на нем соответствующей дополнительной услуги. Например, для включения компьютеров Apple Macintosh в среду Windows NT на сервере Windows NT Server устанавливается служба Services for Macintosh. При этом как пользователи Windows NT, так и пользователи Apple Macintosh продолжают пользоваться привычными интерфейсами для работы с файловыми системами.

Основные поставщики сетевого ПО - фирмы Microsoft, Novell и Apple - учитывают важность совместимости выпускаемых ими программных продуктов. Для этого выпускаются утилиты, которые:

- позволяют серверам распознавать клиентов остальных поставщиков;
- позволяют ОС клиентов связываться с серверами от других производителей.

Фирмы Microsoft, Novell и Apple поставляют также утилиты, позволяющие клиентам под управлением MS-DOS получать доступ к своим серверам. На одной машине могут быть установлены все три вида утилит.

Сети Microsoft

Фирма Microsoft производит следующие ОС: Windows NT, Windows 95 и Windows для рабочих групп. В каждой из этих ОС встроен редиректор. При установке ОС автоматически загружаются необходимые драйверы, редактируются файлы настройки и запускается редиректор, распознающий сетевую среду Microsoft.

При работе в среде Novell NetWare клиент Windows NT Workstation должен использовать протокол NWLink и службу Client Service for NetWare (CSNW). По существу, CSNW это редиректор ("запросчик") Microsoft для доступа к NetWare.

Для подключения сервера Windows NT Server к сети NetWare необходим протокол NWLink и служба Gateway Service for NetWare (GSNW). Напомним, что NWLink - это реализация протокола IPX/SPX фирмой Microsoft.

Для того, чтобы подключить клиента под управлением Windows 95 к сети NetWare используется протокол IPX/SPX и Microsoft Client for NetWare Networks. Усовершенствованное ПО клиента NetWare, называемое Microsoft Service for NetWare Directory Services (NDS), поддерживает Novell NetWare 4.x Directory Services.

Сети Novell

Фирма Novell поставляет запросчики для следующих ОС компьютеров-клиентов: MS-DOS, OS/2, Windows NT.

Клиенты NetWare с символьным интерфейсом под управлением MS-DOS могут подключаться :

- к серверам Novell NetWare;
- к компьютерам под управлением Windows NT Server.

Клиенты Windows NT с запросчиком Novell NetWare и редиректором Windows NT могут подключаться :

- к серверам Novell NetWare;
- к компьютерам под управлением Windows NT Workstation и Windows NT Server.

Фирма Apple

AppleShare - сетевая ОС фирмы Apple - обеспечивает совместное использование файлов, печать файлов и поставляется с клиентским ПО для работы в среде ОС AppleShare.

Персональные компьютеры-клиенты под управлением MS-DOS при установленной плате LocalTalk и при наличии драйвера LocalTalk могут использовать большинство протоколов AppleTalk.

При установленной службе Services for Macintosh сервер Windows NT становится доступным для клиентов Macintosh. Services for Macintosh поддерживает протоколы AppleTalk 2.0 и 2.1, LocalTalk, EtherTalk, TokenTalk и FDDITalk, а также принтеры LaserWriter версии 5.2 и выше.

5.3. Установка сетевой ОС

Установка драйверов

Драйверы (driver) – это программное обеспечение, позволяющее компьютеру работать с различными устройствами. Даже если некоторое устройство и подключено к компьютеру, операционная система не сможет с ним взаимодействовать до тех пор, пока не будет установлен и правильно сконфигурирован

драйвер этого устройства (если ОС не поддерживает спецификацию Plug and Play).

Большинство драйверов предоставляется производителями ОС. Если же драйвера для какого-то устройства нет, его следует искать на диске, входящем в комплект поставки оборудования. Время от времени производители вносят в драйверы дополнения или изменения. Эти изменения, в частности, распространяются через электронную доску объявлений или с помощью оперативных служб, таких, как The Microsoft Network (MSN) или CompuServe. Процесс обновления драйверов аналогичен процессу их установки.

Сетевые драйверы обеспечивают связь между платами сетевого адаптера и редиректорами (редиректор – это часть сетевого ПО, которая принимает запросы ввода/вывода, относящиеся к удаленным файлам, и переадресовывает их по сети на другой компьютер).

Драйверы платы сетевого адаптера располагаются на подуровне «Управление доступом к среде» (канальный уровень модели OSI). Подуровень «Управление доступом к среде» отвечает за совместный доступ плат сетевого адаптера к физическому уровню. Другими словами, драйвер платы сетевого адаптера обеспечивает прямую связь между компьютером и самой платой. Это, в свою очередь, связывает компьютер с сетью.

Ввод в действие и управление драйверами подразумевает их установку, настройку, обновление и удаление. Обычно платы сетевого адаптера имеют несколько параметров, от правильной установки которых зависит корректная работа самого адаптера. Раньше настройка параметров обычно осуществлялась перестановкой перемычек или DIP-переключателей. Большинство современных плат сетевого адаптера не имеют ни того, ни другого. Они конфигурируются программно – при установке драйверов или после нее.

Установка ОС

Установку сетевой ОС рассмотрим на примере Windows NT Server. Программа установки - это приложение, которое выполняет всю работу по установке сетевой ОС. В процессе установки она запрашивает следующие данные об идентификации сервера в сети:

- имя сегмента сети (например, имя домена или рабочей группы);
- имя сервера.

Первый сервер, устанавливаемый в домене, должен выступать главным контроллером домена (Primary Domain Controller, PDC). PDC не только содержит копию информации о домене и проверяет права пользователей, но может также выступать как сервер файлов, печати и приложений. Каждый домен обязательно включает один (и только один) PDC.

Некоторые серверы Windows NT, устанавливаемые после PDC, могут быть установлены как резервные контроллеры доменов (Backup Domain Controller, BDC). BDC - это компьютер, который хранит резервные копии средств безопасности домена и базу данных домена, а также проводит аутентификацию входов в сеть. Наличие в домене BDC необязательно, однако рекомендуется иметь как минимум один BDC. Резервный контроллер доменов также может функционировать как сервер файлов, печати и приложений.

Другие серверы устанавливаются как простые серверы, выполняющие роль серверов файлов, печати и приложений.

Во время установки Windows NT Server необходимо выбрать или сконфигурировать плату сетевого адаптера и выбрать протокол из списка протоколов, например, TCP/IP. Для Windows NT TCP/IP это стандартный, маршрутизируемый протокол сетей масштаба предприятия. Он имеет архитектуру, облегчающую обмен данными в гетерогенных средах и обеспечивает доступ в глобальную сеть Интернет.

Для установки Microsoft TCP/IP необходимы следующие три параметра конфигурации: IP-адрес, маска подсети, IP-маршрутизатор по умолчанию. Напомним, что IP-адрес это логический 32-битный адрес, который состоит из двух частей: идентификатора (ID) сети и ID узла. Каждый компьютер, на котором установлен протокол TCP/IP, должен иметь уникальный IP- адрес.

Маска подсети (subnet mask) используется для выделения частей IP-адреса. IP-маршрутизатор по умолчанию используется для пересылки IP-пакетов в удаленные сети. Если IP-маршрутизатор по умолчанию не указан, возможности связи ограничены локальной сетью.

Если доступен сервер с динамическим протоколом конфигурирования узла (хоста) (Dynamic Host Configuration Protocol, DHCP), то TCP/IP может быть сконфигурирован автоматически. Для этого во время установки надо активизировать флажок Enable Automatic DHCP Configuration.

Чтобы быть уверенным, что сервер и установленное на нем оборудование будет работать с устанавливаемой сетевой ОС, необходимо просмотреть список совместимого оборудования (Hardware Compatibility List). В этом списке указывается какое оборудование было протестировано с конкретной сетевой ОС.

Чтобы сеть могла выполнять различные сетевые задачи, необходима установка специальных приложений сетевых ОС, называемых сетевыми службами (service). Программа установки сетевой ОС гарантирует, что по умолчанию будет установлен минимально необходимый набор сетевых служб. Однако в процессе эксплуатации сети иногда возникает необходимость установить новые службы и функции. Их установка напоминает установку и удаление драйверов. Для этих целей применяются специальные утилиты с графическим интерфейсом.

При включении компьютера некоторые службы запускаются автоматически, другие - должны быть запущены вручную (manual). Чтобы службы были доступны всем сетевым компонентам, каждый из них должен быть *связан* с одним или несколькими сетевыми компонентами, предшествующими ему или следующими за ним. Например, протокол TCP/IP может быть *привязан* к драйверу адаптера 3Com Etherlink III Adapter, а этот драйвер, в свою очередь, привязан к плате этого адаптера.

5.4. Администрирование компьютерной сети

После того как сеть создана и начала функционировать ею необходимо управлять. Сетевое администрирование распространяется на пять основных областей:

1) управление пользователями - создание и поддержка учетных записей пользователей, управление доступом пользователей к ресурсам;
2) управление ресурсами - установка и поддержка сетевых ресурсов;
3) управление конфигурацией - планирование конфигурации сети, ее расширение, а также ведение необходимой документации;

4) управление производительностью - мониторинг и контроль за сетевыми операциями для поддержания и улучшения производительности системы;
5) поддержка - предупреждение, выявление и решение проблем сети.

Управление учетными записями пользователей

Каждому, кто работает в сети, необходимо выделить учетную запись пользователя. Учетные записи создаются и для индивидуальных пользователей, и для групп. Когда администратор впервые регистрируется в системе, автоматически создаются учетные записи Administrator и Guest (Гость).

При создании учетной записи пользователя необходимо вводить различные параметры (такие, как имя пользователя, пароль, привилегии работы в системе и права доступа к ресурсам), а также указать группы, в которые входит пользователь. Для управления рабочей средой пользователей служат <профили>.

При планировании сети особое внимание следует уделять различным группам. Создание групп в значительной мере упрощает работу администратора. Существуют группы четырех типов: локальные, глобальные, специальные (для Windows NT Server) и встроенные. Многие сетевые ОС во время установки сети автоматически создают встроенные локальные и глобальные группы.

Все сети имеют утилиты, которые помогают администраторам добавлять новые учетные записи. В Microsoft Windows NT Server утилита для создания учетных записей называется User Manager for Domains, она находится в группе программ Administrative Tools.

Учетная запись содержит информацию, которая определяет пользователя в системе безопасности сети, в том числе: имя и пароль пользователя; права пользователя на доступ к ресурсам системы; группы к которым относится учетная запись. Эти данные необходимы администратору для создания новой учетной записи.

Большинство сетей позволяет администраторам присваивать пользователям некоторые дополнительные параметры, в том числе: время регистрации - чтобы ограничить время, в течение которого пользователь может входить в сеть; домашний каталог - чтобы предоставить пользователю место для хране-

ния его личных файлов; продолжительность действия учетной записи - чтобы ограничить пребывание некоторых пользователей в сети.

Администратору предоставляется и другая возможность - построить для некоторых пользователей сетевое окружение. Это необходимо, например, для поддержки пользователей не овладевших компьютерами и сетями в такой степени, чтобы работать самостоятельно. Администратор может создать профили (profiles) для управления средой пользователей. К среде относятся сетевые подключения и доступные программы, а также: подключения к принтерам; настройки Program Manager; значки; настройки мыши; цвета экрана; хранители экрана. К параметрам профилей, кроме того, иногда относятся специальные условия входа в систему и информация о том, где пользователь может хранить свои файлы.

При установке сетевой операционной системы автоматически создается учетная запись пользователя, имеющего полный контроль над всеми сетевыми функциями. В сетевой среде Microsoft этот пользователь носит имя <Администратор> (Administrator). В среде Novell он называется <Супервизор> (Supervisor).

Другой стандартный пользователь, создаваемый программой установки, называется <Гость> (Guest). Эта учетная запись предназначена для людей, которые не являются полноправными пользователями сети, однако нуждаются во временном доступе к ней.

Управление группами пользователей

Для того, чтобы не выполнять однотипные операции над большим количеством учетных записей их объединяют в группы. Группа (group) - это учетная запись, которая содержит другие учетные записи. Введение групп упрощает администрирование. Группы предоставляют администраторам возможность оперировать большим числом пользователей как одним сетевым пользователем.

Группы помогают осуществлять следующие действия:

- предоставлять доступ к ресурсам (таким, как файлы, каталоги и принтеры). Права (permissions), предоставленные группе, автоматически предоставляются ее членам;

- предоставлять привилегии (rights) для выполнения системных задач (таких, как резервное копирование, восстановление файлов, изменение системного времени). Привилегии уполномочивают пользователя на выполнение некоторых действий, относящихся к системе в целом, и этим отличаются от прав;
- упрощать связь за счет уменьшения количества подготавливаемых и передаваемых сообщений.

Microsoft Windows NT Server использует группы четырех типов:

- Локальные (local) группы. Группы этого типа реализуются в базе данных учетных записей отдельного компьютера. Локальные группы состоят из учетных записей пользователей, которые имеют права и привилегии на локальном компьютере, и учетных записей глобальных групп.
- Глобальные (global) группы. Группы этого типа используются в границах всего домена. Глобальные группы регистрируются на главном контроллере домена (PDC) и могут содержать только тех пользователей, чьи учетные записи находятся в базе данных этого домена.
- Специальные (special) группы. Эти группы обычно используются Windows NT Server для внутрисистемных нужд.
- Встроенные (built-in) группы. Некоторые функции групп этого типа общие для всех сетей. К ним относится большинство задач администрирования и обслуживания. Чтобы выполнять некоторые стандартные операции, администраторы должны создавать учетные записи пользователей и группы с соответствующими привилегиями, поэтому многие поставщики сетей избавляют администраторов от этих хлопот, предлагая им встроенные локальные или глобальные группы. Встроенные группы делятся на три категории:
    1. Администраторы - пользователи этих групп имеют максимально возможные привилегии.
    2. Операторы - пользователи этих групп имеют ограниченные административные возможности для выполнения специфических задач;
    3. Другие - пользователи этих групп выполняют ограниченные задачи.

Например, Microsoft Windows NT Server предлагает следующие встроенные группы: Administrators, Users, Guests, Server Operators, Print Operators, Backup Operators, Account Operators, Replicator.

Один из способов предоставить одинаковые права большому количеству пользователей - присвоить эти права группе, а затем добавить в группу пользователей. Аналогично добавляются пользователи во встроенную группу. Например, если необходимо, чтобы какой-то пользователь выполнял в сети административные задачи его делают членом группы Administrators.

Литература

1. Компьютерные сети: Учебный курс/Пер. с англ.-М.: ТОО «Channel Trading Ltd», 1997.-696 с.
2. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Издательство "Питер", 2000. – 672 с.
3. Ларионов А.М. и др. Вычислительные комплексы, системы и сети: Учебник для вузов.- Л.: Энергоатомиздат, 1987. – 288 с.
4. Морозов В.К., Долганов А.В. Основы теории информационных сетей: Учебник. – М.: Высш. школа, 1987. – 271 с.
5. Перспективы развития вычислительной техники: Справ. пособие/Под ред. Ю.М. Смирнова: В 11 кн. Кн.10. Системы телеобработки и вычислительные сети. – М.: Высш. школа, 1989. – 144 с.
6. Самойленко С.И. Сети ЭВМ. – М.: Наука, 1986. – 160 с.
7. Советов Б.Я., Яковлев С.А. Построение сетей интегрального обслуживания. – Л.: Машиностроение, 1990. – 332 с.
8. Янбых Г.Ф., Столяров Б.А. Оптимизация информационно-вычислительных сетей. – М.: Радио и связь, 1987. – 232 с.
9. Англо-русский словарь по сетям и сетевым технологиям / Сост. С.Б.Орлов – М.: ”Солон”, 1997. – 301 с.
10. Анни П. Тонкие клиенты. – Jet Info, 2000, #6(85).
11. Ладыженский Г. Архитектура Интранет : новый подход к управлению информацией.- http://www.citforum.ru/ofis/ofis96/106.
12. Протоколы Internet .- http://www.citforum.ru/nets/ito/18.shtml.
13. Кульгин М. Технологии корпоративных сетей: Энциклопедия. – СПб.: Издательство “Питер”, 2000. - 704 с.
14. Олифер В.Г., Олифер Н.А. Новые технологии и оборудование IP-сетей. – СПб.: БХВ – Санкт-Петербург, 2000. – 512 с.
15. Гук М. Аппаратные средства локальных сетей: Энциклопедия.- СПб.: Издательство "Питер", 2000. – 576 с.
16. Ногл М. TCP/IP. Иллюстрированный учебник.- М.: ДМК Пресс, 2001.- 480 с.
17. Новиков Ю.В., Кондратенко С.В. Локальные сети: архитектура, алгоритмы, проектирование.- М.: Издательство ЭКОМ, 2000.- 312 с.
18. Уолрэнд Дж. Телекоммуникационные и компьютерные сети: Вводный курс/ Пер. с англ.- М.: Постмаркет, 2001.- 480с.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

ОГЛАВЛЕНИЕ

Георгий Иванович Анкудинов
Алексей Ильич Стрижаченко

Сети ЭВМ
и
телекоммуникации
Архитектура и протоколы

Учебное пособие

Редактор Т.В.Шабанова

Сводный темплан 2001 г.
Лицензия ЛР №020308 от 14.02.97

Подписано в печать **Формат 60 x 84 1/16**

Б. кн.-журн. **П.л. 5,75** **Б.л. 2,875** **РТП РИО СЗТУ.**

Тираж 200 **Заказ**

Северо-Западный государственный заочный технический университет

РИО СЗТУ, член Издательско-полиграфической ассоциации вузов Санкт-Петербурга

191186, Санкт-Петербург, ул.Миллионная, 5