

Содержание

1. Введение	1
2. Основы TCP/IP	1
2.1. Модуль IP создает единую логическую сеть	1
2.2. Структура связей протокольных модулей	2
2.3. Терминология	3
2.4. Потоки данных	3
2.5. Работа с несколькими сетевыми интерфейсами	5
3. Ethernet	6
3.1. Аналогия с разговором	7
4. Протокол ARP	7
4.1. ARP-таблица для преобразования адресов	8
4.2. Порядок преобразования адресов	8
4.3. Запросы и ответы протокола ARP	9
4.4. Продолжение преобразования адресов	10
5. Межсетевой протокол IP	11
5.1. Прямая маршрутизация	11
5.2. Косвенная маршрутизация	12
5.3. Правила маршрутизации в модуле IP	14
5.4. IP-адрес	15
5.5. Выбор адреса	17
5.6. Подсети	17
5.7. Как назначать номера сетей и подсетей	18
5.8. Имена	19
5.9. IP-таблица маршрутов	20
5.10. Подробности прямой маршрутизации	21
5.11. Порядок прямой маршрутизации	22
5.12. Подробности косвенной маршрутизации	22
5.13. Порядок косвенной маршрутизации	23
6. Установка маршрутов	25
6.1. Фиксированные маршруты	25
6.2. Перенаправление маршрутов	26
6.3. Слежение за маршрутизацией	28
6.4. Протокол ARP с представителем	30
7. Протокол UDP	32
7.1. Порты	33
7.2. Контрольное суммирование	33
8. Протокол TCP	34
9. Протоколы прикладного уровня	35
9.1. Протокол TELNET	36
9.2. Протокол FTP	37
9.3. Протокол SMTP	37
9.4. r-команды	37
9.5. NFS	38
9.6. Протокол SNMP	38
9.7. X-Window	38
10. Взаимозависимость протоколов семейства TCP/IP	39
Приложение 1. Путеводитель по RFC	40
Приложение 2. Стандарты семейства протоколов TCP/IP	75

1. Введение

Семейство протоколов TCP/IP широко применяется во всем мире для объединения компьютеров в сеть Internet. Единая сеть Internet состоит из множества сетей различной физической природы, от локальных сетей типа Ethernet и Token Ring, до глобальных сетей типа NSFNET. Основное внимание в книге уделяется принципам организации межсетевого взаимодействия. Многие технические детали, исторические вопросы опущены. Более подробную информацию о протоколах TCP/IP можно найти в RFC (Requests For Comments) – специальных документах, выпускаемых Сетевым Информационным Центром (Network Information Center – NIC). Приложение 1 содержит путеводитель по RFC, а приложение 2 отражает положение дел в области стандартизации протоколов семейства TCP/IP на начало 1991 года.

В книге приводятся примеры, основанные на реализации TCP/IP в ОС UNIX. Однако основные положения применимы ко всем реализациям TCP/IP.

Надеюсь, что эта книга будет полезна тем, кто профессионально работает или собирается начать работать в среде TCP/IP: системным администраторам, системным программистам и менеджерам сети.

2. Основы TCP/IP

Термин "TCP/IP" обычно обозначает все, что связано с протоколами TCP и IP. Он охватывает целое семейство протоколов, прикладные программы и даже саму сеть. В состав семейства входят протоколы UDP, ARP, ICMP, TELNET, FTP и многие другие. TCP/IP – это технология межсетевого взаимодействия, технология internet. Сеть, которая использует технологию internet, называется "internet". Если речь идет о глобальной сети, объединяющей множество сетей с технологией internet, то ее называют Internet.

2.1. Модуль IP создает единую логическую сеть

Архитектура протоколов TCP/IP предназначена для объединенной сети, состоящей из соединенных друг с другом шлюзами отдельных разнородных пакетных подсетей, к которым подключаются разнородные машины. Каждая из подсетей работает в соответствии со своими специфическими требованиями и имеет свою природу средств связи. Однако предполагается, что каждая подсеть может принять пакет информации (данные с соответствующим сетевым заголовком) и доставить его по указанному адресу в этой конкретной под-

–□– 2□2 –□–

сети. Не требуется, чтобы подсеть гарантировала обязательную доставку пакетов и имела надежный сквозной протокол. Таким образом, две машины, подключенные к одной подсети могут обмениваться пакетами.

Когда необходимо передать пакет между машинами, подключенными к разным подсетям, то машина-отправитель посылает пакет в соответствующий шлюз (шлюз подключен к подсети также как обычный узел). Оттуда пакет направляется по определенному маршруту через систему шлюзов и подсетей, пока не достигнет шлюза, подключенного к той же подсети, что и машина-получатель; там пакет направляется к получателю. Объединенная сеть обеспечивает датаграммный сервис.

Проблема доставки пакетов в такой системе решается путем реализации во всех узлах и шлюзах межсетевого протокола IP. Межсетевой уровень является по существу базовым элементом во всей архитектуре протоколов, обеспечивая возможность стандартизации протоколов верхних уровней.

2.2. Структура связей протокольных модулей

Логическая структура сетевого программного обеспечения, реализующего протоколы семейства TCP/IP в каждом узле сети internet, изображена на рис.1. Прямоугольники обозначают обработку данных, а линии, соединяющие прямоугольники, – пути передачи данных. Горизонтальная линия внизу рисунка обозначает кабель сети Ethernet, которая используется в качестве

примера физической среды; "o" - это трансивер. Знак "*" - обозначает

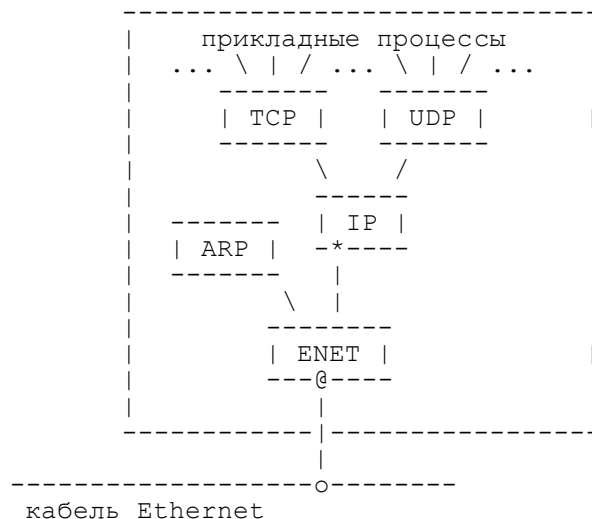


Рис.1. Структура протокольных модулей в узле сети TCP/IP

-□- 3□3 -□-

IP-адрес, а "@" - адрес узла в сети Ethernet (Ethernet-адрес). Понимание этой логической структуры является основой для понимания всей технологии internet. В дальнейшем мы будем часто ссылаться на эту схему.

2.3. Терминология

Введем ряд базовых терминов, которые мы будем использовать в дальнейшем.

Драйвер - это программа, непосредственно взаимодействующая с сетевым адаптером. Модуль - это программа, взаимодействующая с драйвером, сетевыми прикладными программами или другими модулями. Драйвер сетевого адаптера и, возможно, другие модули, специфичные для физической сети передачи данных, предоставляют сетевой интерфейс для протокольных модулей семейства TCP/IP.

Название блока данных, передаваемого по сети, зависит от того, на каком уровне стека протоколов он находится. Блок данных, с которым имеет дело сетевой интерфейс, называется кадром; если блок данных находится между сетевым интерфейсом и модулем IP, то он называется IP-пакетом; если он - между модулем IP и модулем UDP, то - UDP-датаграммой; если между модулем IP и модулем TCP, то - TCP-сегментом (или транспортным сообщением); наконец, если блок данных находится на уровне сетевых прикладных процессов, то он называется прикладным сообщением.

Эти определения, конечно, несовершенны и неполны. К тому же они меняются от публикации к публикации. Более подробные определения можно найти в RFC-1122, раздел 1.3.3.

2.4. Поток данных

Рассмотрим потоки данных, проходящие через стек протоколов, изображенный на рис.1. В случае использования протокола TCP (Transmission Control Protocol - протокол управления передачей), данные передаются между прикладным процессом и модулем TCP. Типичным прикладным процессом, использующим протокол TCP, является модуль FTP (File Transfer Protocol - протокол передачи файлов). Стек протоколов в этом случае будет FTP/TCP/IP/ENET. При использовании протокола UDP (User Datagram Protocol - протокол пользовательских датаграмм), данные передаются между прикладным процессом и модулем UDP. Например, SNMP (Simple Network Management

Protocol - простой протокол управления сетью) пользуется транспортными услугами UDP. Его стек протоколов выглядит так: SNMP/UDP/IP/ENET.

Модули TCP, UDP и драйвер Ethernet являются мультиплексорами $n \times 1$. Действуя как мультиплексоры, они переключают несколько входов на один выход. Они также являются демультимплексорами $1 \times n$. Как демультимплексоры, они переключают один вход на один из многих выходов в соответствии с полем типа в заголовке протокольного блока данных (рис.2).

Когда Ethernet-кадр попадает в драйвер сетевого интерфейса Ethernet, он может быть направлен либо в модуль ARP (Address Resolution Protocol - адресный протокол), либо в модуль IP (Internet Protocol - межсетевой протокол). На то, куда должен быть направлен Ethernet-кадр, указывает значение поля типа в заголовке кадра.

Если IP-пакет попадает в модуль IP, то содержащиеся в нем данные могут быть переданы либо модулю TCP, либо UDP, что определяется полем "протокол" в заголовке IP-пакета.

Если UDP-датаграмма попадает в модуль UDP, то на основании значения поля "порт" в заголовке датаграммы определяется прикладная программа, которой должно быть передано прикладное сообщение. Если TCP-сообщение попадает в модуль TCP, то выбор прикладной программы, которой должно быть передано сообщение, осуществляется на основе значения поля "порт" в заголовке TCP-сообщения.

Мультиплексирование данных в обратную сторону осуществляется довольно просто, так как из каждого модуля существует только один путь вниз. Каждый протокольный модуль добавляет к пакету свой заголовок, на основании которого машина, принявшая пакет, выполняет демультимплексирование.

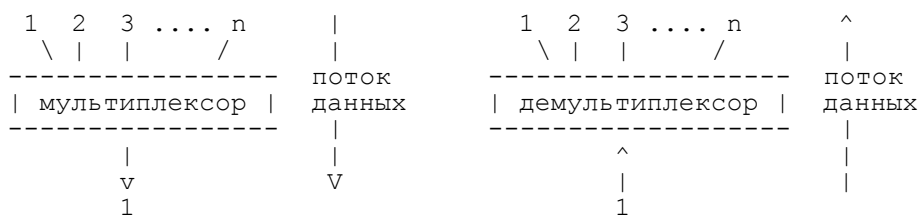


Рис.2. Мультиплексор $n \times 1$ и демультимплексор $1 \times n$

Данные от прикладного процесса проходят через модули TCP или UDP, после чего попадают в модуль IP и оттуда - на уровень сетевого интерфейса.

Хотя технология internet поддерживает много различных сред передачи данных, здесь мы будем предполагать использование Ethernet, так как именно эта среда чаще всего служит физической основой для IP-сети. Машина на рис.1 имеет одну точку соединения с Ethernet. Шестибайтный Ethernet-адрес является уникальным для каждого сетевого адаптера и распознается драйвером.

Машина имеет также четырехбайтный IP-адрес. Этот адрес обозначает точку доступа к сети на интерфейсе модуля IP с драйвером. IP-адрес должен быть уникальным в пределах всей сети Internet.

Работающая машина всегда знает свой IP-адрес и Ethernet-адрес.

2.5. Работа с несколькими сетевыми интерфейсами

Машина может быть подключена одновременно к нескольким средам передачи данных. На рис.3 показана машина с двумя сетевыми интерфейсами Ethernet. Заметим, что она имеет 2 Ethernet-адреса и 2 IP-адреса.

Из представленной схемы видно, что для машин с несколькими сетевыми интерфейсами модуль IP выполняет функции мультиплексора $n \times m$ и демultipлексора $m \times n$ (рис.4).

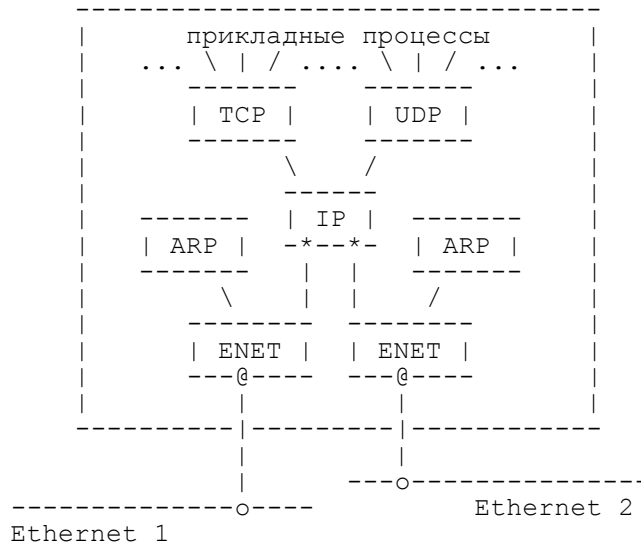


Рис.3. Узел сети TCP/IP с двумя сетевыми интерфейсами

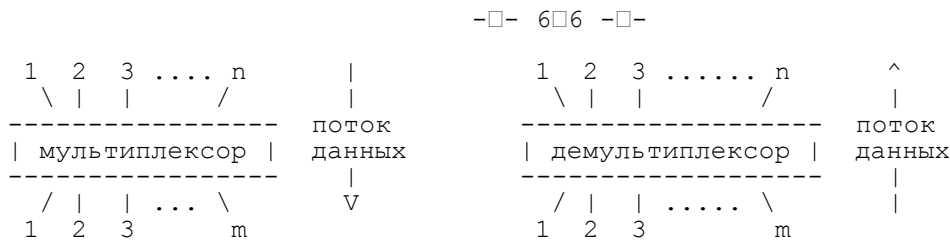


Рис.4. Мультиплексор $n \times m$ и демultipлексор $m \times n$

Таким образом, он осуществляет мультиплексирование входных и выходных данных в обоих направлениях. Модуль IP в данном случае сложнее, чем в первом примере, так как может передавать данные между сетями. Данные могут поступать через любой сетевой интерфейс и быть ретранслированы через любой другой сетевой интерфейс. Процесс передачи пакета в другую сеть называется ретрансляцией IP-пакета. Машина, выполняющая ретрансляцию, называется шлюзом. [1]

Как показано на рис.5, ретранслируемый пакет не передается модулям TCP или UDP. Некоторые шлюзы вообще могут не иметь модулей TCP и UDP.

3. Ethernet

В этом разделе мы кратко рассмотрим технологию Ethernet.

Кадр Ethernet содержит адрес назначения, адрес источника, поле типа и данные. Размер адреса в Ethernet - 6 байт. Каждый сетевой адаптер имеет свой Ethernet-адрес. Адаптер контролирует обмен информацией, про-



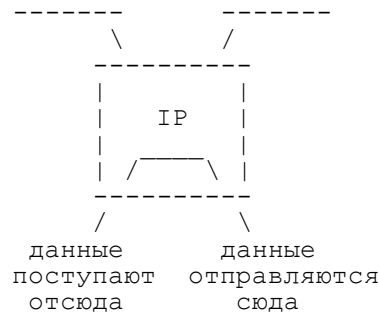


Рис.5. Пример межсетевой ретрансляции пакета модулем IP

[1] В документации по TCP/IP термины шлюз (gateway) и IP-маршрутизатор (IP-router) часто используются как синонимы. Мы сочли возможным использовать более распространенный термин "шлюз".

-□- 7□7 -□-

исходящий в сети, и принимает адресованные ему Ethernet-кадры, а также Ethernet-кадры с адресом "FF:FF:FF:FF:FF:FF" (в 16-ричной системе), который обозначает "всем", и используется при широковещательной передаче.

Ethernet реализует метод МДКН/ОС (множественный доступ с контролем несущей и обнаружением столкновений). Метод МДКН/ОС предполагает, что все устройства взаимодействуют в одной среде, в каждый момент времени может передавать только одно устройство, а принимать могут все одновременно. Если два устройства пытаются передавать одновременно, то происходит столкновение передач, и оба устройства после случайного (краткого) периода ожидания пытаются вновь выполнить передачу.

3.1. Аналогия с разговором

Хорошей аналогией взаимодействиям в среде Ethernet может служить разговор группы вежливых людей в небольшой темной комнате. При этом аналогией электрическим сигналам в коаксиальном кабеле служат звуковые волны в комнате.

Каждый человек слышит речь других людей (контроль несущей). Все люди в комнате имеют одинаковые возможности вести разговор (множественный доступ), но никто не говорит слишком долго, так как все вежливы. Если человек будет невежлив, то его попросят выйти (т.е. удалят из сети). Все молчат, пока кто-то говорит. Если два человека начинают говорить одновременно, то они сразу обнаруживают это, поскольку слышат друг друга (обнаружение столкновений). В этом случае они замолкают и ждут некоторое время, после чего один из них вновь начинает разговор. Другие люди слышат, что ведется разговор, и ждут, пока он кончится, а затем могут начать говорить сами. Каждый человек имеет собственное имя (аналог уникального Ethernet-адреса). Каждый раз, когда кто-нибудь начинает говорить, он называет по имени того, к кому обращается, и свое имя, например, "Слушай Петя, это Андрей, ... ля-ля-ля ...". Если кто-то хочет обратиться ко всем, то он говорит: "Слушайте все, это Андрей, ... ля-ля-ля ..." (широковещательная передача).

4. Протокол ARP

В этом разделе мы рассмотрим то, как при посылке IP-пакета определяется Ethernet-адрес назначения. Для отображения IP-адресов в Ethernet-адреса используется протокол ARP (Address Resolution Protocol - адресный

протокол). Отображение выполняется только для отправляемых IP-пакетов, так как только в момент отправки создаются заголовки IP и Ethernet.

4.1. ARP-таблица для преобразования адресов

Преобразование адресов выполняется путем поиска в таблице. Эта таблица, называемая ARP-таблицей, хранится в памяти и содержит строки для каждого узла сети. В двух столбцах содержатся IP- и Ethernet-адреса. Если требуется преобразовать IP-адрес в Ethernet-адрес, то ищется запись с соответствующим IP-адресом. Ниже приведен пример упрощенной ARP-таблицы.

IP-адрес	Ethernet-адрес
223.1.2.1	08:00:39:00:2F:C3
223.1.2.3	08:00:5A:21:A7:22
223.1.2.4	08:00:10:99:AC:54

Табл.1. Пример ARP-таблицы

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

ARP-таблица необходима потому, что IP-адреса и Ethernet-адреса выбираются независимо, и нет какого-либо алгоритма для преобразования одного в другой. IP-адрес выбирает менеджер сети с учетом положения машины в сети internet. Если машину перемещают в другую часть сети internet, то ее IP-адрес должен быть изменен. Ethernet-адрес выбирает производитель сетевого интерфейсного оборудования из выделенного для него по лицензии адресного пространства. Когда у машины заменяется плата сетевого адаптера, то меняется и ее Ethernet-адрес.

4.2. Порядок преобразования адресов

В ходе обычной работы сетевая программа, такая как TELNET, отправляет прикладное сообщение, пользуясь транспортными услугами TCP. Модуль TCP посылает соответствующее транспортное сообщение через модуль IP. В результате составляется IP-пакет, который должен быть передан драйверу Ethernet. IP-адрес места назначения известен прикладной программе, модулю TCP и модулю IP. Необходимо на его основе найти Ethernet-адрес

места назначения. Для определения искомого Ethernet-адреса используется ARP-таблица.

4.3. Запросы и ответы протокола ARP

Как же заполняется ARP-таблица? Она заполняется автоматически модулем ARP, по мере необходимости. Когда с помощью существующей ARP-таблицы не удастся преобразовать IP-адрес, то происходит следующее:

- 1) По сети передается широковещательный ARP-запрос.
- 2) Исходящий IP-пакет ставится в очередь.

Каждый сетевой адаптер принимает широковещательные передачи. Все драйверы Ethernet проверяют поле типа в принятом Ethernet-кадре и передают ARP-пакеты модулю ARP. ARP-запрос можно интерпретировать так: "Если ваш IP-адрес совпадает с указанным, то сообщите мне ваш Ethernet-адрес". Пакет ARP-запроса выглядит примерно так:

IP-адрес отправителя	223.1.2.1
----------------------	-----------

Ethernet-адрес отправителя	08:00:39:00:2F:C3	
Искомый IP-адрес	223.1.2.2	
Искомый Ethernet-адрес	<пусто>	

Табл.2. Пример ARP-запроса

Каждый модуль ARP проверяет поле искомого IP-адреса в полученном ARP-пакете и, если адрес совпадает с его собственным IP-адресом, то посылает ответ прямо по Ethernet-адресу отправителя запроса. ARP-ответ можно интерпретировать так: "Да, это мой IP-адрес, ему соответствует такой-то Ethernet-адрес". Пакет с ARP-ответом выглядит примерно так:

IP-адрес отправителя	223.1.2.2	
Ethernet-адрес отправителя	08:00:28:00:38:A9	
Искомый IP-адрес	223.1.2.1	
Искомый Ethernet-адрес	08:00:39:00:2F:C3	

Табл.3. Пример ARP-ответа

-□- 1□10□0 -□-

Этот ответ получает машина, сделавшая 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.

Обновленная таблица выглядит следующим образом:

IP-адрес	Ethernet-адрес	
223.1.2.1	08:00:39:00:2F:C3	
223.1.2.2	08:00:28:00:38:A9	
223.1.2.3	08:00:5A:21:A7:22	
223.1.2.4	08:00:10:99:AC:54	

Табл.4. ARP-таблица после обработки ответа

4.4. Продолжение преобразования адресов

Новая запись в ARP-таблице появляется автоматически, спустя несколько миллисекунд после того, как она потребовалась. Как вы помните, ранее на шаге 2 исходящий IP-пакет был поставлен в очередь. Теперь с использованием обновленной ARP-таблицы выполняется преобразование IP-адреса в Ethernet-адрес, после чего Ethernet-кадр передается по сети. Полностью порядок преобразования адресов выглядит так:

- 1) По сети передается широковещательный ARP-запрос.
- 2) Исходящий IP-пакет ставится в очередь.
- 3) Возвращается ARP-ответ, содержащий информацию о соответствии IP- и Ethernet-адресов. Эта информация заносится в ARP-таблицу.
- 4) Для преобразования IP-адреса в Ethernet-адрес у IP-пакета, поставленного в очередь, используется ARP-таблица.
- 5) Ethernet-кадр передается по сети Ethernet.

Короче говоря, если с помощью ARP-таблицы не удастся сразу осуществить преобразование адресов, то IP-пакет ставится в очередь, а необходи-

мая для преобразования информация получается с помощью запросов и ответов протокола ARP, после чего IP-пакет передается по назначению.

-□- 1□11□1 -□-

Если в сети нет машины с искомым IP-адресом, то ARP-ответа не будет и не будет записи в ARP-таблице. Протокол IP будет уничтожать IP-пакеты, направляемые по этому адресу. Протоколы верхнего уровня не могут отличить случай повреждения сети Ethernet от случая отсутствия машины с искомым IP-адресом.

Некоторые реализации IP и ARP не ставят в очередь IP-пакеты на то время, пока они ждут ARP-ответов. Вместо этого IP-пакет просто уничтожается, а его восстановление возлагается на модуль TCP или прикладной процесс, работающий через UDP. Такое восстановление выполняется с помощью таймаутов и повторных передач. Повторная передача сообщения проходит успешно, так как первая попытка уже вызвала заполнение ARP-таблицы.

Следует отметить, что каждая машина имеет отдельную ARP-таблицу для каждого своего сетевого интерфейса.

5. Межсетевой протокол IP

Модуль IP является базовым элементом технологии internet, а центральной частью IP является его таблица маршрутов. Протокол IP использует эту таблицу при принятии всех решений о маршрутизации IP-пакетов. Содержание таблицы маршрутов определяется администратором сети. Ошибки при установке маршрутов могут заблокировать передачи.

Чтобы понять технику межсетевого взаимодействия, нужно понять то, как используется таблица маршрутов. Это понимание необходимо для успешного администрирования и сопровождения IP-сетей.

5.1. Прямая маршрутизация

На рис.6 показана небольшая IP-сеть, состоящая из 3 машин: A, B и C. Каждая машина имеет такой же стек протоколов TCP/IP как на рис.1. Каждый сетевой адаптер этих машин имеет свой Ethernet-адрес. Менеджер сети должен присвоить машинам уникальные IP-адреса.

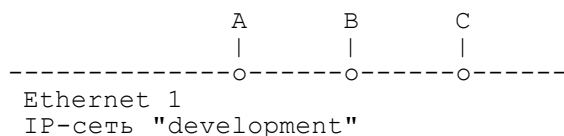


Рис.6. Простая IP-сеть

-□- 1□12□2 -□-

Когда A посылает IP-пакет B, то 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.

	адрес	отправитель	получатель
	IP-заголовок	A	B
	Ethernet-заголовок	A	B

Если машина А посылает машине Е IP-пакет, то IP-адрес и Ethernet-адрес отправителя соответствуют адресам А. IP-адрес места назначения является адресом Е, но поскольку модуль IP в А посылает IP-пакет через D, Ethernet-адрес места назначения является адресом D.

-□- 1□14□4 -□-

	адрес	отправитель	получатель	
	IP-заголовок	А	Е	
	Ethernet-заголовок	А	D	

Табл.6. Адреса в Ethernet-кадре, содержащем IP-пакет от А к Е (до шлюза D)

Модуль IP в машине D получает IP-пакет и проверяет IP-адрес места назначения. Определив, что это не его IP-адрес, шлюз D посылает этот IP-пакет прямо к Е.

	адрес	отправитель	получатель	
	IP-заголовок	А	Е	
	Ethernet-заголовок	D	Е	

Табл.7. Адреса в Ethernet-кадре, содержащем IP-пакет от А к Е (после шлюз D)

Итак, при 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 данном примере сеть internet является очень простой. Реальные сети могут быть гораздо сложнее, так как могут содержать несколько шлюзов и несколько типов физических сред передачи. В приведенном примере несколько сетей Ethernet объединяются шлюзом для того, чтобы локализовать широковещательный трафик в каждой сети.

5.3. Правила маршрутизации в модуле IP

Выше мы показали, что происходит при передаче сообщений, а теперь рассмотрим правила или алгоритм маршрутизации.

Для отправляемых IP-пакетов, поступающих от модулей верхнего уровня, модуль IP должен определить способ доставки – прямой или косвенный – и выбрать сетевой интерфейс. Этот выбор делается на основании результатов поиска в таблице маршрутов.

-□- 1□15□5 -□-

Для принимаемых IP-пакетов, поступающих от сетевых драйверов, модуль

IP должен решить, нужно ли ретранслировать IP-пакет по другой сети или передать его на верхний уровень. Если модуль IP решит, что IP-пакет должен быть ретранслирован, то дальнейшая работа с ним осуществляется также, как с отправляемыми IP-пакетами.

Входящий IP-пакет никогда не ретранслируется через тот же сетевой интерфейс, через который он был принят.

Решение о маршрутизации принимается до того, как IP-пакет передается сетевому драйверу, и до того, как происходит обращение к ARP-таблице.

5.4. IP-адрес

Менеджер сети присваивает IP-адреса машинам в соответствии с тем, к каким IP-сетям они подключены. Старшие биты 4-х байтного IP-адреса определяют номер IP-сети. Оставшаяся часть IP-адреса - номер узла (хост-номер). Для машины из табл.1 с IP-адресом 223.1.2.1 сетевой номер равен 223.1.2, а хост-номер - 1. Напомним, что IP-адрес узла идентифицирует точку доступа модуля IP к сетевому интерфейсу, а не всю машину.

Существуют 5 классов IP-адресов, отличающиеся количеством бит в сетевом номере и хост-номере. Класс адреса определяется значением его первого октета.

В табл.8 приведено соответствие классов адресов значениям первого октета и указано количество возможных IP-адресов каждого класса.

	0	8	16	24	31
Класс А	0	номер сети		номер узла	
Класс В	10	номер сети		номер узла	
Класс С	110	номер сети		номер узла	
Класс D	1110	групповой адрес			
Класс Е	11110	зарезервировано			

Рис.8. Структура IP-адресов

-□- 1□16□6 -□-

	Класс	Диапазон значений первого октета	Возможное кол-во сетей	Возможное кол-во 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

Табл.8. Характеристики классов адресов

Адреса класса А предназначены для использования в больших сетях общего пользования. Они допускают большое количество номеров узлов. Адреса класса В используются в сетях среднего размера, например, сетях университетов и крупных компаний. Адреса класса С используются в сетях с небольшим числом компьютеров. Адреса класса D используются при обращениях к группам машин, а адреса класса Е зарезервированы на будущее.

Некоторые IP-адреса являются выделенными и трактуются по-особому.

	все нули			Данный узел		

	номер сети		все нули		Данная IP-сеть	

	все нули		номер узла		Узел в данной (локальной) IP-сети	

	все единицы			Все узлы в данной (локальной) IP-сети		

	номер сети		все единицы		Все узлы в указанной IP-сети	

	127		что-нибудь (часто 1)			"Петля"

Рис.9. Выделенные IP-адреса

Как показано на рис.9, в выделенных IP-адресах все нули соответствуют либо данному узлу, либо данной IP-сети, а IP-адреса, состоящие из всех единиц, используются при широковещательных передачах. Для ссылок на всю IP-сеть в целом используется IP-адрес с нулевым номером узла. Особый смысл имеет IP-адрес, первый октет которого равен 127. Он используется для тестирования программ и взаимодействия процессов в пределах одной машины. Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется как бы "петля". Данные не передаются по сети, а возвращаются

-□- 1□17□7 -□-

модулям верхнего уровня, как только что принятые. Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127.

5.5. Выбор адреса

Прежде чем вы начнете использовать сеть с TCP/IP, вы должны получить один или несколько официальных сетевых номеров. Выделением номеров (как и многими другими вопросами) занимается DDN Network Information Center (NIC) [2]. Выделение номеров производится бесплатно и занимает около недели. Вы можете получить сетевой номер вне зависимости от того, для чего предназначена ваша сеть. Даже если ваша сеть не имеет связи с объединенной сетью Internet, получение уникального номера желательно, так как в этом случае есть гарантия, что в будущем при включении в Internet или при подключении к сети другой организации не возникнет конфликта адресов.

Одно из важнейших решений, которое необходимо принять при установке сети, заключается в выборе способа присвоения IP-адресов вашим машинам. Этот выбор должен учитывать перспективу роста сети. Иначе в дальнейшем вам придется менять адреса. Когда к сети подключено несколько сотен машин, изменение адресов становится почти невозможным.

Организации, имеющие небольшие сети с числом узлов до 126, должны запрашивать сетевые номера класса С. Организации с большим числом машин могут получить несколько номеров класса С или номер класса В. Удобным средством структуризации сетей в рамках одной организации являются подсети.

5.6. Подсети

Адресное пространство сети internet может быть разделено на непере-секающиеся подпространства - "подсети", с каждой из которых можно работать как с обычной сетью TCP/IP. Таким образом единая IP-сеть организации может строиться как объединение подсетей. Как правило, подсеть соответствует одной физической сети, например, одной сети Ethernet.

Конечно, использование подсетей необязательно. Можно просто назначить для каждой физической сети свой сетевой номер, например, номер

[2] SRI International, Room EJ210, 333 Ravenswood Avenue, Menlo Park, California 94025, USA. Тел. 1-800-235-3155. E-mail: NIC@NIC.DDN.MIL

-□- 1□18□8 -□-

класса С. Однако такое решение имеет два недостатка. Первый, и менее существенный, заключается в пустой трате сетевых номеров. Более серьезный недостаток состоит в том, что если ваша организация имеет несколько сетевых номеров, то машины вне ее должны поддерживать записи о маршрутах доступа к каждой из этих IP-сетей. Таким образом, структура IP-сети организации становится видимой для всего мира. При каких-либо изменениях в IP-сети информация о них должна быть учтена в каждой из машин, поддерживающих маршруты доступа к данной IP-сети.

Подсети позволяют избежать этих недостатков. Ваша организация должна получить один сетевой номер, например, номер класса В. Стандарты TCP/IP определяют структуру IP-адресов. Для IP-адресов класса В первые два октета являются номером сети. Оставшаяся часть IP-адреса может использоваться как угодно. Например, вы можете решить, что третий октет будет определять номер подсети, а четвертый октет - номер узла в ней. Вы должны описать конфигурацию подсетей в файлах, определяющих маршрутизацию IP-пакетов. Это описание является локальным для вашей организации и не видно вне ее. Все машины вне вашей организации видят одну большую IP-сеть. Следовательно, они должны поддерживать только маршруты доступа к шлюзам, соединяющим вашу IP-сеть с остальным миром. Изменения, происходящие в IP-сети организации, не видны вне ее. Вы легко можете добавить новую подсеть, новый шлюз и т.п.

5.7. Как назначать номера сетей и подсетей

После того, как решено использовать подсети или множество IP-сетей, вы должны решить, как назначать им номера. Обычно это довольно просто. Каждой физической сети, например, Ethernet или Token Ring, назначается отдельный номер подсети или номер сети. В некоторых случаях имеет смысл назначать одной физической сети несколько подсетевых номеров. Например, предположим, что имеется сеть Ethernet, охватывающая три здания. Ясно, что при увеличении числа машин, подключенных к этой сети, придется ее разделить на несколько отдельных сетей Ethernet. Для того, чтобы избежать необходимости менять IP-адреса, когда это произойдет, можно заранее выделить для этой сети три подсетевых номера - по одному на здание. (Это полезно и в том случае, когда не планируется физическое деление сети. Просто такая адресация позволяет сразу определить, где находится та или иная машина.) Однако прежде, чем выделять три различных подсетевых номера

-□- 1□19□9 -□-

одной физической сети, тщательно проверьте, что все ваши программы способны работать в такой среде.

Вы также должны выбрать "маску подсети". Она используется сетевым программным обеспечением для выделения номера подсети из IP-адресов. Биты IP-адреса, определяющие номер IP-сети, в маске подсети должны быть равны 1, а биты, определяющие номер узла, в маске подсети должны быть равны 0. Как уже отмечалось, стандарты TCP/IP определяют количество октетов, задающих номер сети. Часто в IP-адресах класса В третий октет используется для задания номера подсети. Это позволяет иметь 256 подсетей, в каждой из которых может быть до 254 узлов. Маска подсети в такой

системе равна 255.255.255.0. Но, если в вашей сети должно быть больше подсетей, а в каждой подсети не будет при этом более 60 узлов, то можно использовать маску 255.255.255.192. Это позволяет иметь 1024 подсети и до 62 узлов в каждой. (Напомним, что номера узлов 0 и "все единицы" используются особым образом.)

Обычно маска подсети указывается в файле стартовой конфигурации сетевого программного обеспечения. Протоколы TCP/IP позволяют также записывать эту информацию по сети.

5.8. Имена

Людям удобнее называть машины по именам, а не числами. Например, у машины по имени alpha может быть IP-адрес 223.1.2.1. В маленьких сетях информация о соответствии имен IP-адресам хранится в файлах "hosts" на каждом узле. Конечно, название файла зависит от конкретной реализации. В больших сетях эта информация хранится на сервере и доступна по сети. Несколько строк из файла "hosts" могут выглядеть примерно так:

223.1.2.1	alpha
223.1.2.2	beta
223.1.2.3	gamma
223.1.2.4	delta
223.1.3.2	epsilon
223.1.4.2	iota

В первом столбце - IP-адрес, во втором - название машины.

В большинстве случаев файлы "hosts" могут быть одинаковы на всех узлах. Заметим, что о узле delta в этом файле есть всего одна запись, хотя он имеет три IP-адреса (рис.11). Узел delta доступен по любому из

-□- 2□20□0 -□-

этих IP-адресов. Какой из них используется, не имеет значения. Когда узел delta получает IP-пакет и проверяет IP-адрес места назначения, то он опознает любой из трех своих IP-адресов.

IP-сети также могут иметь имена. Если у вас есть три IP-сети, то файл "networks" может выглядеть примерно так:

223.1.2	development
223.1.3	accounting
223.1.4	factory

В первой колонке - сетевой номер, во второй - имя сети.

В данном примере alpha является узлом номер 1 в сети development, beta является узлом номер 2 в сети development и т.д.

Показанный выше файл hosts удовлетворяет потребности пользователей, но для управления сетью internet удобнее иметь названия всех сетевых интерфейсов. Менеджер сети, возможно, заменит строку, относящуюся к delta:

223.1.2.4	devnetrouter	delta
223.1.3.1	accnetrouter	
223.1.4.1	facnetrouter	

Эти три строки файла hosts задают каждому IP-адресу узла delta символичные имена. Фактически, первый IP-адрес имеет два имени: "devnetrouter" и "delta", которые являются синонимами. На практике имя "delta" используется как общепотребительное имя машины, а остальные три имени - для администрирования сети.

Файлы hosts и networks используются командами администрирования и прикладными программами. Они не нужны собственно для работы сети internet, но облегчают ее использование.

5.9. IP-таблица маршрутов

Как модуль IP узнает, какой именно сетевой интерфейс нужно использовать для отправления IP-пакета? Модуль IP осуществляет поиск в таблице маршрутов. Ключом поиска служит номер IP-сети, выделенный из IP-адреса места назначения IP-пакета.

-□- 2□21□1 -□-

Таблица маршрутов содержит по одной строке для каждого маршрута. Основными столбцами таблицы маршрутов являются номер сети, флаг прямой или косвенной маршрутизации, IP-адрес шлюза и номер сетевого интерфейса. Эта таблица используется модулем IP при обработке каждого отправляемого IP-пакета.

В большинстве систем таблица маршрутов может быть изменена с помощью команды "route". Содержание таблицы маршрутов определяется менеджером сети, поскольку менеджер сети присваивает машинам IP-адреса.

5.10. Подробности прямой маршрутизации

Рассмотрим более подробно, как происходит маршрутизация в одной физической сети.

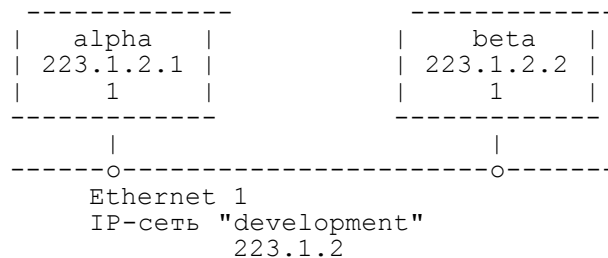


Рис.10. Одна физическая сеть

Таблица маршрутов в узле alpha выглядит так:

сеть	флаг вида маршрутизации	шлюз	номер интерфейса
development	прямая	<пусто>	1

Табл.9. Пример таблицы маршрутов

В данном простом примере все узлы сети имеют одинаковые таблицы маршрутов.

Для сравнения ниже представлена та же таблица, но вместо названия сети указан ее номер.

-□- 2□22□2 -□-

сеть	флаг вида маршрутизации	шлюз	номер интерфейса

223.1.2	прямая	<пусто>	1	
---------	--------	---------	---	--

Табл.10. Пример таблицы маршрутов с номерами сетей

5.11. Порядок прямой маршрутизации

Узел alpha посылает IP-пакет узлу beta. Этот пакет находится в модуле IP узла alpha, и IP-адрес места назначения равен IP-адресу beta (223.1.2.2). Модуль IP с помощью маски подсети выделяет номер сети из IP-адреса и ищет соответствующую ему строку в таблице маршрутов. В данном случае подходит первая 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 beta.

Если прикладная программа пытается послать данные по IP-адресу, который не принадлежит сети development, то модуль IP не сможет найти соответствующую запись в таблице маршрутов. В этом случае модуль IP отбрасывает IP-пакет. Некоторые реализации протокола возвращают сообщение об ошибке "Сеть не доступна".

5.12. Подробности косвенной маршрутизации

Теперь рассмотрим более сложный порядок маршрутизации в IP-сети, изображенной на рис.11.

Таблица маршрутов в узле alpha выглядит так:

сеть	флаг вида маршрутизации	шлюз	номер интерфейса
development	прямая	<пусто>	1
accounting	косвенная	devnetrouter	1
factory	косвенная	devnetrouter	1

Табл.11. Таблица маршрутов в узле alpha

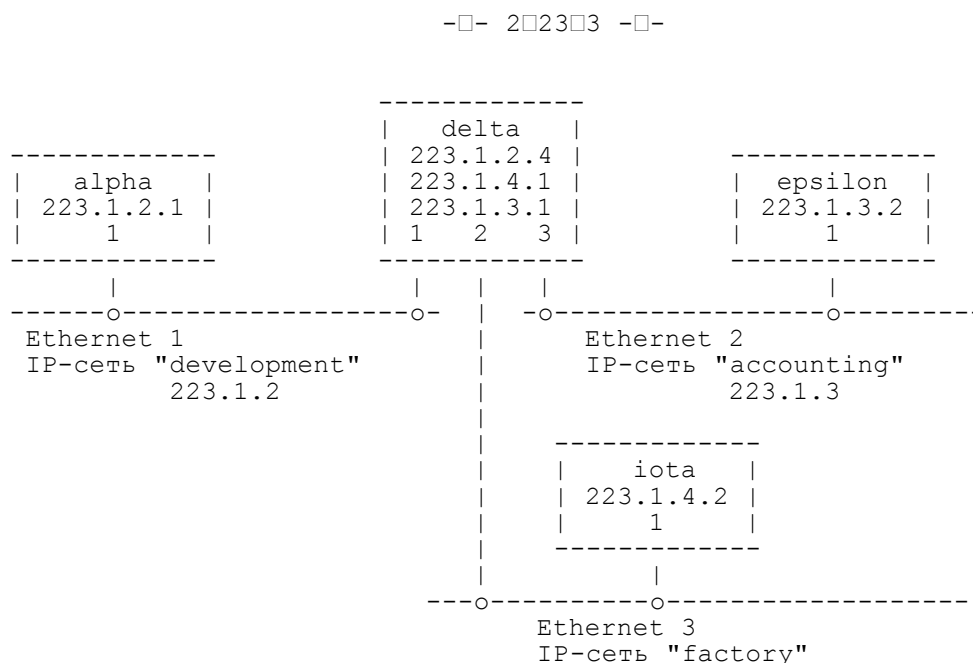


Рис.11. Подробная схема трех сетей

Та же таблица с IP-адресами вместо названий.

сеть	флаг вида маршрутизации	шлюз	номер интерфейса
223.1.2	прямая	<пусто>	1
223.1.3	косвенная	223.1.2.4	1
223.1.4	косвенная	223.1.2.4	1

Табл.12. Таблица маршрутов в узле alpha (с номерами)

В столбце "шлюз" таблицы маршрутов узла alpha указывается IP-адрес точки соединения узла delta с сетью development.

5.13. Порядок косвенной маршрутизации

Узел alpha посылает IP-пакет узлу epsilon. Этот пакет находится в модуле IP узла alpha, и IP-адрес места назначения равен IP-адресу узла epsilon (223.1.3.2). Модуль IP выделяет сетевой номер из IP-адреса (223.1.3) и ищет соответствующую ему строку в таблице маршрутов. Соответствие находится во второй строке.

Запись в этой строке указывает на то, что машины требуемой сети доступны через шлюз devnetrouter. Модуль IP в узле alpha осуществляет поиск в ARP-таблице, с помощью которого определяет Ethernet-адрес, соответствующий IP-адресу devnetrouter. Затем IP-пакет, содержащий IP-адрес места

-□- 2□24□4 -□-

назначения epsilon, посылается через интерфейс 1 шлюзу devnetrouter.

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

Модуль IP в узле delta выделяет сетевой номер из IP-адреса места назначения IP-пакета (223.1.3) и ищет соответствующую запись в таблице маршрутов. Таблица маршрутов в узле delta выглядит так:

сеть	флаг вида маршрутизации	шлюз	номер интерфейса
development	прямая	<пусто>	1
accounting	прямая	<пусто>	3
factory	прямая	<пусто>	2

Табл.13. Таблица маршрутов в узле delta

Та же таблица с IP-адресами вместо названий.

сеть	флаг вида маршрутизации	шлюз	номер интерфейса
223.1.2	прямая	<пусто>	1
223.1.3	прямая	<пусто>	3
223.1.4	прямая	<пусто>	2

Табл.14. Таблица маршрутов в узле delta (с номерами)

Соответствие находится во второй строке. Теперь модуль IP напрямую посылает IP-пакет узлу epsilon через интерфейс номер 3. Пакет содержит IP- и Ethernet-адреса места назначения равные epsilon.

Узел epsilon принимает IP-пакет, и его модуль IP проверяет IP-адрес места назначения. Он соответствует IP-адресу epsilon, поэтому содержащееся в IP-пакете сообщение передается протокольному модулю верхнего уровня.

-□- 2□25□5 -□-

6. Установка маршрутов

До сих пор мы рассматривали то, как используется таблица маршрутов для маршрутизации IP-пакетов. Но откуда берется информация в самой таблице маршрутов? В данном разделе мы рассмотрим методы, позволяющие поддерживать корректность таблиц маршрутов.

6.1. Фиксированные маршруты

Простейший способ проведения маршрутизации состоит в установке маршрутов при запуске системы с помощью специальных команд. Этот метод можно применять в относительно маленьких IP-сетях, в особенности, если их конфигурации не часто меняются.

На практике большинство машин автоматически формирует таблицы маршрутов. Например, UNIX добавляет записи о IP-сетях, к которым есть непосредственный доступ. Стартовый файл может содержать команды

```
ifconfig ie0 128.6.4.4 netmask 255.255.255.0
ifconfig ie1 128.6.5.35 netmask 255.255.255.0
```

Они показывают, что существуют два сетевых интерфейса, и устанавливают их IP-адреса. Система может автоматически создать две записи в таблице маршрутов:

сеть	флаг вида маршрутизации	шлюз	интерфейс
128.6.4	прямая	<пусто>	ie0
128.6.5	прямая	<пусто>	ie1

Табл.15. Автоматически создаваемые записи

Эти записи определяют, что IP-пакеты для локальных подсетей 128.6.4 и 128.6.5 должны посылаться через указанные интерфейсы.

В стартовом файле могут быть команды, определяющие маршруты доступа к другим IP-сетям. Например,

```
route add 128.6.2.0 128.6.4.1 1
route add 128.6.6.0 128.6.5.35 0
```

Эти команды показывают, что в таблицу маршрутов должны быть добавлены две записи. Первый адрес в командах является IP-адресом сети, второй адрес

-□- 2□26□6 -□-

указывает шлюз, который должен использоваться для доступа к данной IP-

сети, а третий параметр является метрикой. Метрика показывает, на каком "расстоянии" находится описываемая IP-сеть. В данном случае метрика - это количество шлюзов на пути между двумя IP-сетями. Маршруты с метрикой 1 и более определяют первый шлюз на пути к IP-сети. Маршруты с метрикой 0 показывают, что никакой шлюз не нужен - данный маршрут задает дополнительный сетевой номер локальной IP-сети.

Таким образом, команды, приведенные в примере, говорят о том, что для доступа к IP-сети 128.6.2 должен использоваться шлюз 128.6.4.1, а IP-сеть 128.6.6 - это просто дополнительный номер для физической сети, подключенной к интерфейсу 128.6.5.35.

сеть	флаг вида маршрутизации	шлюз	интерфейс
128.6.2	косвенная	128.6.4.1	ie0
128.6.6	прямая	<пусто>	ie1

Табл.16. Записи, добавляемые в таблицу маршрутов

Можно определить маршрут по умолчанию, который используется в тех случаях, когда IP-адрес места назначения не встречается в таблице маршрутов явно. Обычно маршрут по умолчанию указывает IP-адрес шлюза, который имеет достаточно информации для маршрутизации IP-пакетов со всеми возможными адресами назначения.

Если ваша IP-сеть имеет всего один шлюз, тогда все, что нужно сделать, - это установить единственную запись в таблице маршрутов, указав этот шлюз как маршрут по умолчанию. После этого можно не заботиться о формировании маршрутов в других узлах. (Конечно, сам шлюз требует больше внимания.)

Следующие разделы посвящены IP-сетям, где есть несколько шлюзов.

6.2. Перенаправление маршрутов

Большинство экспертов по межсетевому взаимодействию рекомендуют оставлять решение проблем маршрутизации шлюзам. Плохо иметь на каждой машине большую таблицу маршрутов. Дело в том, что при каких-либо изменениях в IP-сети приходится менять информацию во всех машинах. Например, при отключении какого-нибудь канала связи для восстановления нормальной

-□- 2□27□7 -□-

работы нужно ждать, пока кто-то заметит это изменение в конфигурации IP-сети и внесет исправления во все таблицы маршрутов.

Простейший способ поддержания адекватности маршрутов заключается в том, что изменение таблицы маршрутов каждой машины выполняется по командам только одного шлюза. Этот шлюз должен быть установлен как маршрут по умолчанию. (В ОС UNIX это делается командой "route add default 128.6.4.27 1", где 128.6.4.27 является IP-адресом шлюза.) Как было описано выше, каждая машина посылает IP-пакет шлюзу по умолчанию в том случае, когда не находит лучшего маршрута. Однако, когда в IP-сети есть несколько шлюзов, этот метод работает не так хорошо. Кроме того, если таблица маршрутов имеет только одну запись о маршруте по умолчанию, то как использовать другие шлюзы, если это более выгодно? Ответ состоит в том, что большинство шлюзов способны выполнять "перенаправление" в тех случаях, когда они получают IP-пакеты, для которых существуют более выгодные маршруты. "Перенаправление" является специальным типом сообщения протокола ICMP (Internet Control Message Protocol - протокол межсетевых управляющих сообщений). Сообщение о перенаправлении содержит информацию, которую можно интерпретировать так: "В будущем для IP-адреса XXXX используйте шлюз YYYY, а не меня". Корректные реализации TCP/IP должны использовать сообщения о перенаправлении для добавления записей в таблицу маршрутов. Предположим, таблица маршрутов в начале выглядит следующим образом:

адрес назначения	флаг вида маршрутизации	шлюз	интерфейс
127.0.0	прямая	<пусто>	lo0
128.6.4	прямая	<пусто>	re0
default	косвенная	128.6.4.27	re0

Табл.17. Таблица маршрутов в начале работы

Эта таблица содержит запись о локальной IP-сети 128.6.4 и маршрут по умолчанию, указывающий шлюз 128.6.4.27. Допустим, что существует шлюз 128.6.4.30, который является лучшим путем доступа к IP-сети 128.6.7. Как им воспользоваться? Предположим, что нужно посылать IP-пакеты по IP-адресу 128.6.7.23. Первый IP-пакет пойдет на шлюз по умолчанию, так как это единственный подходящий маршрут, описанный в таблице. Однако шлюз 128.6.4.27 знает, что существует лучший маршрут, проходящий через шлюз

-□- 2□28□8 -□-

128.6.4.30. (Как он узнает об этом, мы сейчас не рассматриваем. Существует довольно простой метод определения лучшего маршрута.) В этом случае шлюз 128.6.4.27 возвращает сообщение перенаправления, где указывает, что IP-пакеты для узла 128.6.7.23 должны посылаться через шлюз 128.6.4.30. Модуль IP на машине-отправителе должен добавить запись в таблицу маршрутов:

адрес назначения	флаг вида маршрутизации	шлюз	интерфейс
128.6.7.23	косвенная	128.6.4.30	re0

Табл.18. Новая запись в таблице маршрутов

Все последующие IP-пакеты для узла 128.6.7.23 будут посланы прямо через указанный шлюз.

До сих пор мы рассматривали способы добавления маршрутов в IP-таблицу, но не способы их исключения. Что случится, если шлюз будет выключен? Хотелось бы иметь способ возврата к маршруту по умолчанию после того, как какой-либо маршрут разрушен. Однако, если шлюз вышел из строя или был выключен, то он уже не может послать сообщение перенаправления. Поэтому должен существовать метод определения работоспособности шлюзов, с которыми ваша машина связана непосредственно. Лучший способ обнаружения неработающих шлюзов основан на выявлении "плохих" маршрутов. Модуль TCP поддерживает различные таймеры, которые помогают ему определить разрыв соединения. Когда случается сбой, то можно пометить маршрут как "плохой" и вернуться к маршруту по умолчанию. Аналогичный метод может использоваться при обработке ошибок шлюза по умолчанию. Если два шлюза отмечены как шлюзы по умолчанию, то машина может использовать их по очереди, переключаясь между ними при возникновении сбоев.

6.3. Слежение за маршрутизацией

Заметим, что сообщения перенаправления не могут использоваться самими шлюзами. Перенаправление - это просто способ оповещения обычного узла о том, что нужно использовать другой шлюз. Сами шлюзы должны иметь полную картину о положении дел в сети internet и уметь вычислять оптимальные маршруты доступа к каждой подсети. Обычно они поддерживают эту картину, обмениваясь информацией между собой. Для этой цели существуют

несколько специальных протоколов маршрутизации. Один из способов, с помощью которого узлы могут определять действующие шлюзы, состоит в слежении за обменом сообщениями между ними. Для большинства протоколов маршрутизации существует программное обеспечение, позволяющее обычным узлам осуществлять такое слежение. При этом на узлах поддерживается полная картина положения дел в сети internet точно также, как это делается в шлюзах. Динамическая корректировка таблицы маршрутов позволяет посылать IP-пакеты по оптимальным маршрутам.

Таким образом, слежение за маршрутизацией в некотором смысле "решает" проблему поддержания корректности таблиц маршрутов. Однако существуют несколько причин, по которым этот метод применять не рекомендуется. Наиболее серьезной проблемой является то, что протоколы маршрутизации пока еще подвергаются частым пересмотрам и изменениям. Появляются новые протоколы маршрутизации. Эти изменения должны учитываться в программном обеспечении всех машин.

Несколько более специальная проблема связана с бездисковыми рабочими станциями. По своей природе бездисковые машины сильно зависят от сети и от файл-серверов, с которых они осуществляют загрузку программ, и где располагается их область своппинга. Исполнение программ, следящих за ширококестельными передачами в сети, на бездисковых машинах связано с большими трудностями. Протоколы маршрутизации построены в основном на ширококестельных передачах. Например, все сетевые шлюзы могут ширококестельно передавать содержание своих таблиц маршрутов через каждые 30 секунд. Программы, которые следят за такими передачами, должны быть загружены на бездисковые станции через сеть. На достаточно занятой машине программы, которые не используются в течение нескольких секунд, обычно отправляются в область своппинга. Поэтому программы, следящие за маршрутизацией, большую часть времени находятся в своппинге. Когда они вновь активизируются, должна производиться подкачка из своппинга. Как только посылается ширококестельное сообщение, все машины активизируют программы, следящие за маршрутизацией. Это приводит к тому, что многие бездисковые станции будут выполнять подкачку из своппинга в одно и тоже время. Поэтому в сети возникнет временная перегрузка. Таким образом, исполнение программ, прослушивающих ширококестельные передачи, на бездисковых рабочих станциях очень нежелательно.

6.4. Протокол ARP с представителем

Протокол ARP с представителем является альтернативным методом, позволяющим шлюзам принимать все необходимые решения о маршрутизации. Он применяется в сетях с ширококестельной передачей, где для отображения IP-адресов в сетевые адреса используется протокол ARP или ему подобный. Здесь мы вновь будем предполагать, что имеем дело с сетью Ethernet.

Во многом метод, реализуемый протоколом ARP с представителем, аналогичен использованию маршрутов по умолчанию и сообщений перенаправления. Но протокол ARP с представителем не затрагивает таблиц маршрутов, все делается на уровне адресов Ethernet. Протокол ARP с представителем может использоваться либо для маршрутизации IP-пакетов ко всем сетям, либо только в локальной сети, либо в какой-то комбинации подсетей. Проще всего продемонстрировать его использование при работе со всеми адресами.

Чтобы использовать протокол, нужно настроить узел так, как будто все машины в мире подключены непосредственно к вашей локальной сети Ethernet. В ОС UNIX это делается командой "route add default 128.6.4.2 0", где 128.6.4.2 - IP-адрес вашего узла. Как уже отмечалось, метрика 0 говорит о том, что все IP-пакеты, которым подходит данный маршрут, должны посылаться напрямую по локальной сети.

Когда нужно послать IP-пакет узлу в локальной сети Ethernet, ваша машина должна определить Ethernet-адрес этого узла. Для этого она использует ARP-таблицу. Если в ARP-таблице уже есть запись, соответству-

ющая IP-адресу места назначения, то из нее просто берется Ethernet-адрес, и кадр, содержащий IP-пакет, отправляется. Если такой записи нет, то посылается широковещательный ARP-запрос. Узел с искомым IP-адресом назначения принимает его и в ARP-ответе сообщает свой Ethernet-адрес. Эти действия соответствуют обычному протоколу ARP, описанному выше.

Протокол ARP с представителем основан на том, что шлюзы работают как представители удаленных узлов. Предположим, в подсети 128.6.5 имеется узел 128.6.5.2 (узел А на рис.12). Он желает послать IP-пакет узлу 128.6.4.194, который подключен к другой сети Ethernet (узел В в подсети 128.6.4). Существует шлюз с IP-адресом 128.6.5.1, соединяющий две подсети (шлюз R).

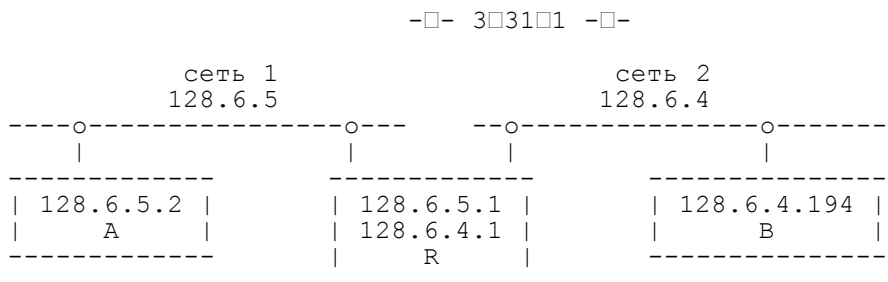


Рис.12. Сеть, использующая протокол ARP с представителем

Если в ARP-таблице узла А нет маршрута доступа к узлу В, то узел А посылает ARP-запрос узлу В. Фактически машина А спрашивает: "Если кто-нибудь знает Ethernet-адрес узла 128.6.4.194, сообщите мне его". Узел В не может ответить на запрос самостоятельно. Он подключен к другой сети Ethernet и никогда даже не увидит этот ARP-запрос. Однако шлюз R может работать от его имени. Шлюз R отвечает: "Я здесь, IP-адресу 128.6.4.194 соответствует Ethernet-адрес 2:7:1:0:ЕВ:СD", где 2:7:1:0:ЕВ:СD в действительности является Ethernet-адресом шлюза. Это создает иллюзию, что узел 128.6.4.194 подключен непосредственно к той же локальной сети Ethernet, что и узел А, и имеет Ethernet-адрес 2:7:1:0:ЕВ:СD. Когда узел А захочет послать новый IP-пакет узлу В, он использует указанный Ethernet-адрес. Кадр, содержащий IP-пакет, попадет к шлюзу R, а он переправит его по назначению.

Заметим, что полученный эффект такой же, как если бы в таблице маршрутов была запись

адрес назначения	флаг вида маршрутизации	шлюз	интерфейс
128.6.4.194	косвенная	128.6.5.1	re0

за исключением того, что маршрутизация выполняется на уровне модуля ARP, а не модуля IP.

Обычно рекомендуется использовать таблицу маршрутов, так как архитектура протоколов TCP/IP предусматривает выполнение маршрутизации на межсетевом уровне. Однако иногда протокол ARP с представителем очень полезен. Он может помочь в следующих случаях:

- 1) в IP-сети есть узел, который не умеет работать с подсетями;

- 2) в IP-сети есть узел, который не может соответствующим образом реагировать на сообщения перенаправления;
- 3) нежелательно выбирать какой-либо шлюз как маршрут по умолчанию;
- 4) программное обеспечение не способно восстанавливаться при сбоях на маршрутах.

Иногда протокол ARP с представителем выбирают из-за удобства. Дело в том, что он упрощает работу по начальной установке таблицы маршрутов. Даже в простейших IP-сетях требуется устанавливать маршрут по умолчанию, то есть использовать команду типа "route add default ...", как в ОС UNIX. При изменении IP-адреса шлюза эту команду приходится менять во всех узлах. Если же использовать протокол ARP с представителем, т.е. в команде установки маршрута по умолчанию указать метрику 0, то при замене IP-адреса шлюза команду начальной установки менять не придется, так как протокол ARP с представителем не требует явного задания IP-адресов шлюзов. Любой шлюз может ответить на ARP-запрос.

Для того, чтобы избавить пользователей от обязательной начальной установки маршрутов, некоторые реализации TCP/IP используют протокол ARP с представителем по умолчанию в тех случаях, когда не находят подходящих записей в таблице маршрутов.

7. Протокол UDP

Протокол UDP (User Datagram Protocol - протокол пользовательских датаграмм) является одним из двух основных протоколов, расположенных непосредственно над IP. Он предоставляет прикладным процессам транспортные услуги, которые не многим отличаются от услуг, предоставляемых протоколом IP. Протокол UDP обеспечивает ненадежную доставку датаграмм и не поддерживает соединений из конца в конец. К заголовку IP-пакета он добавляет два поля, одно из которых, поле "порт", обеспечивает мультиплексирование информации между разными прикладными процессами, а другое поле - "контрольная сумма" - позволяет поддерживать целостность данных.

Примерами сетевых приложений, использующих UDP, являются NFS (Network File System - сетевая файловая система) и SNMP (Simple Network Management Protocol - простой протокол управления сетью).

7.1. Порты

Взаимодействие между прикладными процессами и модулем UDP осуществляется через UDP-порты. Порты нумеруются начиная с нуля. Прикладной процесс, предоставляющий некоторые услуги другим прикладным процессам (сервер), ожидает поступления сообщений в порт, специально выделенный для этих услуг. Сообщения должны содержать запросы на предоставление услуг. Они отправляются процессами-клиентами.

Например, сервер SNMP всегда ожидает поступлений сообщений в порт 161. Если клиент SNMP желает получить услугу, он посылает запрос в UDP-порт 161 на машину, где работает сервер. В каждом узле может быть только один сервер SNMP, так как существует только один UDP-порт 161. Данный номер порта является общеизвестным, то есть фиксированным номером, официально выделенным для услуг SNMP. Общеизвестные номера определяются стандартами Internet.

Данные, отправляемые прикладным процессом через модуль UDP, достигают места назначения как единое целое. Например, если процесс-отправитель производит 5 записей в UDP-порт, то процесс-получатель должен будет сделать 5 чтений. Размер каждого записанного сообщения будет совпадать с размером каждого прочитанного. Протокол UDP сохраняет границы сообщений, определяемые прикладным процессом. Он никогда не объединяет

несколько сообщений в одно и не делит одно сообщение на части.

7.2. Контрольное суммирование

Когда модуль UDP получает датаграмму от модуля IP, он проверяет контрольную сумму, содержащуюся в ее заголовке. Если контрольная сумма равна нулю, то это означает, что отправитель датаграммы ее не подсчитывал, и, следовательно, ее нужно игнорировать. Если два модуля UDP взаимодействуют только через одну сеть Ethernet, то от контрольного суммирования можно отказаться, так как средства Ethernet обеспечивают достаточную степень надежности обнаружения ошибок передачи. Это снижает накладные расходы, связанные с работой UDP. Однако рекомендуется всегда выполнять контрольное суммирование, так как возможно в какой-то момент изменения в таблице маршрутов приведут к тому, что датаграммы будут посылаться через менее надежную среду.

-□- 3□34□4 -□-

Если контрольная сумма правильная (или равна нулю), то проверяется порт назначения, указанный в заголовке датаграммы. Если к этому порту подключен прикладной процесс, то прикладное сообщение, содержащееся в датаграмме, становится в очередь для прочтения. В остальных случаях датаграмма отбрасывается. Если датаграммы поступают быстрее, чем их успевает обрабатывать прикладной процесс, то при переполнении очереди сообщений поступающие датаграммы отбрасываются модулем UDP.

8. Протокол TCP

Протокол TCP предоставляет транспортные услуги, отличающиеся от услуг UDP. Вместо ненадежной доставки датаграмм без установления соединений, он обеспечивает гарантированную доставку с установлением соединений в виде байтовых потоков.

Протокол TCP используется в тех случаях, когда требуется надежная доставка сообщений. Он освобождает прикладные процессы от необходимости использовать таймауты и повторные передачи для обеспечения надежности. Наиболее типичными прикладными процессами, использующими TCP, являются FTP (File Transfer Protocol - протокол передачи файлов) и TELNET. Кроме того, TCP используют система X-Window, rcp (remote copy - удаленное копирование) и другие "r-команды". Большие возможности TCP даются не бесплатно. Реализация TCP требует большой производительности процессора и большой пропускной способности сети. Внутренняя структура модуля TCP гораздо сложнее структуры модуля UDP.

Прикладные процессы взаимодействуют с модулем TCP через порты. Для отдельных приложений выделяются общеизвестные номера портов. Например, сервер TELNET использует порт номер 23. Клиент TELNET может получать услуги от сервера, если установит соединение с TCP-портом 23 на его машине.

Когда прикладной процесс начинает использовать TCP, то модуль TCP на машине клиента и модуль TCP на машине сервера начинают общаться. Эти два оконечных модуля TCP поддерживают информацию о состоянии соединения, называемого виртуальным каналом. Этот виртуальный канал потребляет ресурсы обоих оконечных модулей TCP. Канал является дуплексным; данные могут одновременно передаваться в обоих направлениях. Один прикладной процесс пишет данные в TCP-порт, они проходят по сети, и другой приклад-

-□- 3□35□5 -□-

ной процесс читает их из своего TCP-порта.

Протокол TCP разбивает поток байт на пакеты; он не сохраняет границ между записями. Например, если один прикладной процесс делает 5 записей в TCP-порт, то прикладной процесс на другом конце виртуального канала может выполнить 10 чтений для того, чтобы получить все данные. Но этот же процесс может получить все данные сразу, сделав только одну операцию чтения. Не существует зависимости между числом и размером записываемых сообщений с одной стороны и числом и размером считываемых сообщений с другой стороны.

Протокол TCP требует, чтобы все отправленные данные были подтверждены принявшей их стороной. Он использует таймауты и повторные передачи для обеспечения надежной доставки. Отправителю разрешается передавать некоторое количество данных, не дожидаясь подтверждения приема ранее отправленных данных. Таким образом, между отправленными и подтвержденными данными существует окно уже отправленных, но еще неподтвержденных данных. Количество байт, которые можно передавать без подтверждения, называется размером окна. Как правило, размер окна устанавливается в стартовых файлах сетевого программного обеспечения. Так как TCP-канал является дуплексным, то подтверждения для данных, идущих в одном направлении, могут передаваться вместе с данными, идущими в противоположном направлении. Приемники на обеих сторонах виртуального канала выполняют управление потоком передаваемых данных для того, чтобы не допускать переполнения буферов.

9. Протоколы прикладного уровня

Почему существуют два транспортных протокола TCP и UDP, а не один из них? Дело в том, что они предоставляют разные услуги прикладным процессам. Большинство прикладных программ пользуются только одним из них. Вы, как программист, выбираете тот протокол, который наилучшим образом соответствует вашим потребностям. Если вам нужна надежная доставка, то лучшим может быть TCP. Если вам нужна доставка датаграмм, то лучше может быть UDP. Если вам нужна эффективная доставка по длинному и ненадежному каналу передачи данных, то лучше может подойти протокол TCP. Если нужна эффективность на быстрых сетях с короткими соединениями, то лучшим может быть протокол UDP. Если ваши потребности не попадают ни в одну из этих категорий, то выбор транспортного протокола не ясен. Однако прикладные

-□- 3□36□6 -□-

программы могут устранять недостатки выбранного протокола. Например, если вы выбрали UDP, а вам необходима надежность, то прикладная программа должна обеспечить надежность. Если вы выбрали TCP, а вам нужно передавать записи, то прикладная программа должна вставлять маркеры в поток байтов так, чтобы можно было различить записи.

Какие же прикладные программы доступны в сетях с TCP/IP?

Общее их количество велико и продолжает постоянно увеличиваться. Некоторые приложения существуют с самого начала развития internet. Например, TELNET и FTP. Другие появились недавно: X-Window, SNMP.

Протоколы прикладного уровня ориентированы на конкретные прикладные задачи. Они определяют как процедуры по организации взаимодействия определенного типа между прикладными процессами, так и форму представления информации при таком взаимодействии. В этом разделе мы коротко опишем некоторые из прикладных протоколов.

9.1. Протокол TELNET

Протокол TELNET позволяет обслуживающей машине рассматривать все удаленные терминалы как стандартные "сетевые виртуальные терминалы" строчного типа, работающие в коде ASCII, а также обеспечивает возможность согласования более сложных функций (например, локальный или удаленный эхо-контроль, страничный режим, высота и ширина экрана и т.д.) TELNET работает на базе протокола TCP. На прикладном уровне над TELNET находится либо программа поддержки реального терминала (на стороне пользователя), либо прикладной процесс в обслуживающей машине, к которому осу-

ществляется доступ с терминала.

Работа с TELNET походит на набор телефонного номера. Пользователь набирает на клавиатуре что-то вроде

```
telnet delta
```

и получает на экране приглашение на вход в машину delta.

Протокол TELNET существует уже давно. Он хорошо опробован и широко распространен. Создано множество реализаций для самых разных операционных систем. Вполне допустимо, чтобы процесс-клиент работал, скажем, под управлением ОС VAX/VMS, а процесс-сервер под ОС UNIX System V.

-□- 3□37□7 -□-

9.2. Протокол FTP

Протокол FTP (File Transfer Protocol - протокол передачи файлов) распространен также широко как TELNET. Он является одним из старейших протоколов семейства TCP/IP. Также как TELNET он пользуется транспортными услугами TCP. Существует множество реализаций для различных операционных систем, которые хорошо взаимодействуют между собой. Пользователь FTP может вызывать несколько команд, которые позволяют ему посмотреть каталог удаленной машины, перейти из одного каталога в другой, а также скопировать один или несколько файлов.

9.3. Протокол SMTP

Протокол SMTP (Simple Mail Transfer Protocol - простой протокол передачи почты) поддерживает передачу сообщений (электронной почты) между произвольными узлами сети internet. Имея механизмы промежуточного хранения почты и механизмы повышения надежности доставки, протокол SMTP допускает использование различных транспортных служб. Он может работать даже в сетях, не использующих протоколы семейства TCP/IP. Протокол SMTP обеспечивает как группирование сообщений в адрес одного получателя, так и размещение нескольких копий сообщения для передачи в разные адреса. Над модулем SMTP располагается почтовая служба конкретных вычислительных систем.

9.4. r-команды

Существует целая серия "r-команд" (от remote - удаленный), которые впервые появились в ОС UNIX. Они являются аналогами обычных команд UNIX, но предназначены для работы с удаленными машинами. Например, команда rcp является аналогом команды cp и предназначена для копирования файлов между машинами. Для передачи файла на узел delta достаточно ввести

```
rcp file.c delta:
```

Для выполнения команды "cc file.c" на машине delta можно использовать команду rsh:

```
rsh delta cc file.c
```

Для организации входа в удаленную систему предназначена команда rlogin:

```
rlogin delta
```

-□- 3□38□8 -□-

Команды r-серии используются главным образом в системах, работающих под управлением ОС UNIX. Существуют также реализации для MS-DOS. Команды избавляют пользователя от необходимости набирать пароли при входе

в удаленную систему и существенно облегчают работу.

9.5. NFS

Сетевая файловая система NFS (Network File System) впервые была разработана компанией Sun Microsystems Inc. NFS использует транспортные услуги UDP и позволяет монтировать в единое целое файловые системы нескольких машин с ОС UNIX. Бездисковые рабочие станции получают доступ к дискам файл-сервера так, как-будто это их локальные диски.

NFS значительно увеличивает нагрузку на сеть. Если в сети используются медленные линии связи, то от NFS мало толку. Однако, если пропускная способность сети позволяет NFS нормально работать, то пользователи получают большие преимущества. Поскольку сервер и клиент NFS реализуются в ядре ОС, все обычные несетевые программы получают возможность работать с удаленными файлами, расположенными на подмонтированных NFS-дисках, точно также как с локальными файлами.

9.6. Протокол SNMP

Протокол SNMP (Simple Network Management Protocol - простой протокол управления сетью) работает на базе UDP и предназначен для использования сетевыми управляющими станциями. Он позволяет управляющим станциям собирать информацию о положении дел в сети internet. Протокол определяет формат данных, их обработка и интерпретация остаются на усмотрение управляющих станций или менеджера сети.

9.7. X-Window

Система X-Window использует протокол X-Window, который работает на базе TCP, для многооконного отображения графики и текста на растровых дисплеях рабочих станций. X-Window - это гораздо больше, чем просто утилита для рисования окон; это целая философия человеко-машинного взаимодействия.

-□- 3□39□9 -□-

10. Взаимозависимость протоколов семейства TCP/IP

Ниже на рисунке представлена схема взаимосвязей между протоколами семейства TCP/IP.

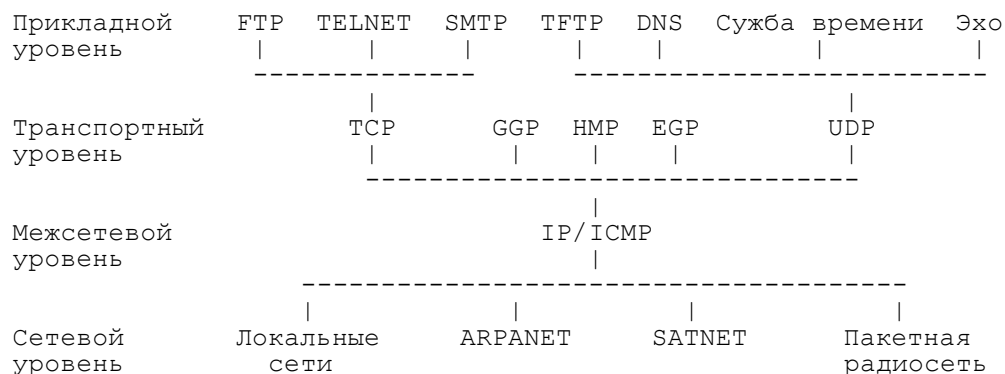


Рис.13. Структура взаимосвязей протоколов семейства TCP/IP

Подробное описание протоколов можно найти в RFC, тематический каталог которых приведен в Приложении 1, а состояние стандартов отражено в Приложении 2.