Network Working Group
Request for Comments: 1180

T. Socolofsky
C. Kale
Spider Systems Limited

January 1991

Учебник по TCP/IP

A TCP/IP Tutorial

Статус документа

Данный документ представляет собой учебное пособие по стеку протоколов TCP/IP, включающее в себя вопросы пересылки дейтаграмм IP между отправителем и получателем через цепочку маршрутизаторов. Документ не является стандартом Internet. Разрешается свободное распространение документа.

Оглавление

1. Введение

В документе описаны основные аспекты стека протоколов TCP/IP. Здесь не рассматривается история разработки и развития протоколов, не приводится примеров практического использования и не дается сравнений с протоколами ISO/OSI. Опущено также множество технических деталей, связанных с рассматриваемыми протоколами. То, что приведено в документе, составляет лишь минимум информации, требующийся каждому профессионалу, работающему

в среде TCP/IP. К числу таких профессионалов относятся системные администраторы, системные программисты и администраторы сетей.

В документе используются примеры для среды UNIX TCP/IP, однако основное внимание обращено на вопросы, не зависящие от реализации TCP/IP. Документ не содержит определений новых протоколов, он предназначен для обучения. При возникновении вопросов по тем или иным протоколам, обращайтесь к соответствующим RFC.

В следующем параграфе приведен обзор TCP/IP, за которым следует детальное описание отдельных компонент.

2. Обзор TCP/IP

Термином TCP/IP зачастую обозначают все, что относится к сетевым протоколам TCP и IP - другие протоколы, приложения и даже сетевые среды. Примерами протоколов могут служить UDP, ARP и ICMP; примерами приложений - TELNET, FTP и rcp. Более точным является термин «технология internet». Сеть, использующая технологию internet также может обозначаться термином internet.

2.1 Базовая структура

Для понимания технологии нужно сначала разобраться с приведенной ниже логической структурой.

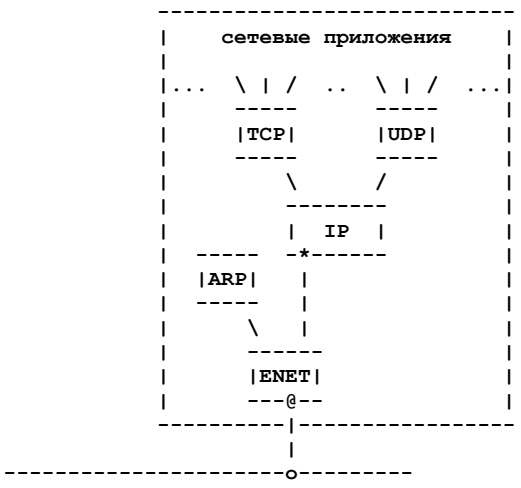

Рисунок 1. Узел сети TCP/IP.

Показанная на рисунке схема описывает логическую структуру многоуровневых протоколов в компьютере, подключённом к сети. Все компьютеры, поддерживающие подобную структуру, могут обмениваться информацией, используя сетевые протоколы. Приведённая на рисунке структура определяет поведение компьютера в сети. Каждый из прямоугольников на схеме показывает процесс обработки данных в компьютере, а линии обозначают потоки данных. Горизонтальная линия в нижней части рисунка представляет кабель Ethernet; "o" показывает трансивер (приемопередатчик, сетевой адаптер). Символ "*" представляет IP-адрес компьютера, а "@" - адрес Ethernet (MAC-адрес). Эта структура важна для понимания сетевых технологий, поэтому мы будем неоднократно ссылаться в документе на приведённую здесь схему.

2.2 Терминология

Название элементов данных, принимаемых из сети, зависит от уровня в стеке протоколов:

- для Ethernet модули данных называют кадрами (Ethernet frame)
- данные между драйвером адаптера Ethernet и модулем IP передаются в пакетах (IP packet);
- данные между модулями IP и UDP передаются с помощью дейтаграмм (UDP datagram);
- модули данных, передаваемые между IP и TCP, называют сегментами (TCP segment) или транспортными сообщениями
- данные на уровне сетевых приложений передаются с помощью сообщений.

Эти определения не являются общепринятыми и в публикациях вы можете встретить множество иных терминов или толкований приведённых здесь терминов. Основные определения терминов можно найти в RFC 1122 (параграф 1.3.3).

Драйвер представляет собой программу, взаимодействующую на аппаратном уровне с сетевым интерфейсом. Модулем будем называть программу, взаимодействующую с драйвером, прикладной программой или другим модулем.

Термины драйвер, модуль, кадр Ethernet, пакет IP, дейтаграмма UDP, сегмент TCP, сообщение прикладной программы будут постоянно встречаться вам на протяжении документа.

2.3 Поток данных

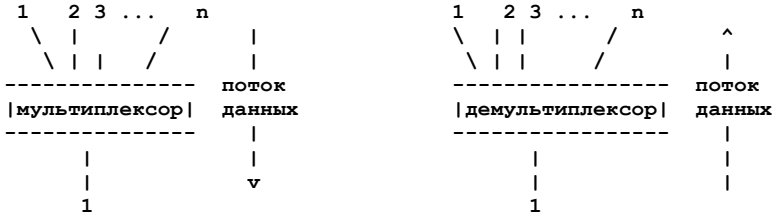

Рисунок 2. Мультиплексор и демультиплексор.

Рассмотрим поток данных, проходящий вниз через стек протоколов на рисунке 1. Для приложений, использующих протокол TCP (Transmission Control Protocol - протокол управления передачей), данные передаются между приложением и модулем TCP. Для приложений, использующих протокол UDP (User Datagram Protocol - протокол пользовательских дейтаграмм), обмен данными идёт между приложением и модулем UDP. FTP (File Transfer Protocol - протокол передачи файлов) является типичным примером использования протокола TCP. В данном случае стек протоколов будет иметь вид FTP/TCP/IP/ENET. Приложения SNMP (Simple Network Management Protocol - простой протокол сетевого управления) используют протокол UDP и стек будет иметь вид SNMP/UDP/IP/ENET.

В модулях TCP, UDP и драйвере Ethernet выполняется мультиплексирование STSn x 1CH (мультиплексор имеет один выход и множество входов). Существует также обратная операция - демультиплексирование STS1 x nCH (демультиплексор имеет один вход и множество выходов). Схематическое представление мультиплексоров и демультиплексоров приведено на рисунке 2.

Если кадр Ethernet попадает в драйвер Ethernet из сети, этот кадр передается модулю преобразования адресов ARP (Address Resolution Protocol) или модулю IP (Internet Protocol). Выбор одного из этих модулей (ARP или IP) определяется полем типа в заголовке кадра Ethernet.

Если пакет попадает в модуль IP, после этого он передаётся модулю TCP или UDP в соответствии со значением поля протокола в заголовке IP.

Дейтаграммы UDP, приходящие в одноименный модуль, преобразуются в сообщения прикладного уровня и передаются программам, выбор которых определяется номером порта в заголовке UDP. Сегменты TCP в одноименном модуле преобразуются в сообщения прикладного уровня и передаются пользовательским программам в соответствии с номером порта в заголовке TCP.

Мультиплексирование исходящего потока является очень простой задачей, поскольку на каждом уровне существует только один путь передачи информации; каждый протокол просто добавляет в пакет свой заголовок, обеспечивающий демультиплексирование данных на приемной стороне.

Данные передаются от прикладных программ через TCP или UDP, преобразуются модулем IP и передаются в сеть с использованием драйвера сетевой платы на самом нижнем уровне.

Хотя в сетях может использоваться множество различных технологий, все примеры здесь построены на основе технологии Ethernet, которая на сегодняшний день является самой распространенной для передачи IP-трафика. Компьютер на рисунке 1 имеет одно соединение Ethernet. Шестибайтовый адрес Ethernet является уникальным для каждого адаптера Ethernet и задается на аппаратном уровне.

Компьютер также имеет 4-байтовый адрес IP. Этот адрес используется на интерфейсе нижнего уровня в модуле IP. Этот адрес должен быть уникальным в масштабах сети.

Работающий компьютер всегда знает свои адреса IP и Ethernet.

2.4 Два сетевых интерфейса

На рисунке 3 схематически изображён компьютер с двумя сетевыми интерфейсами.

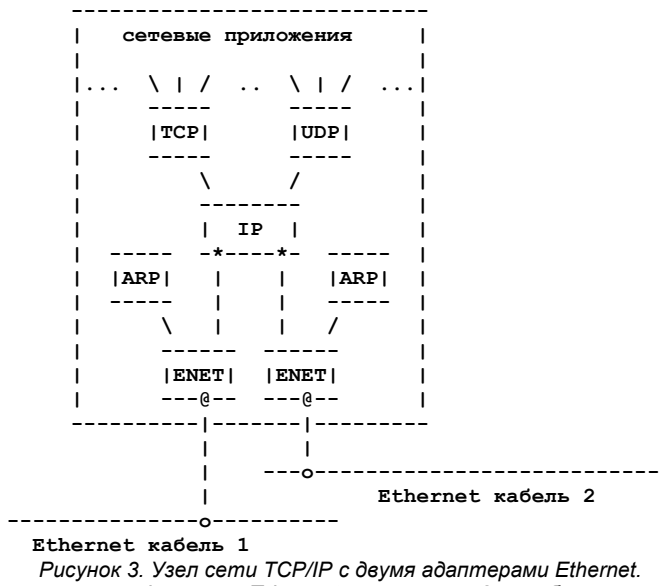

Рисунок 3. Узел сети TCP/IP с двумя адаптерами Ethernet.

Отметим, что компьютер с двумя интерфейсами Ethernet использует 2 или более адресов IP.

Из приведенной схемы очевидно, что в компьютерах с несколькими физическими интерфейсами модуль IP работает как мультиплексор-демультиплексор m x n.

Мультиплексирование выполняется при передаче данных в обоих направлениях. Модуль IP с несколькими физическими интерфейсами на самом деле более сложен, чем показано на рисунке 4, поскольку такой модуль может обеспечивать пересылку данных в другие сети (данные, принятые через один интерфейс, передаются через другой).

Процесс отправки пакета IP в другую сеть называется пересылкой (forwarding) пакетов IP. Компьютер, принимающий решение о пересылке пакетов IP, обычно называют машрутизатором (IP-router).

Как можно видеть на приведенном рисунке, в процессе пересылки пакетов IP модули TCP и UDP на маршрутизаторе IP участия не принимают. Некоторые из реализаций IP-маршрутизаторов просто не включают модулей TCP и UDP.

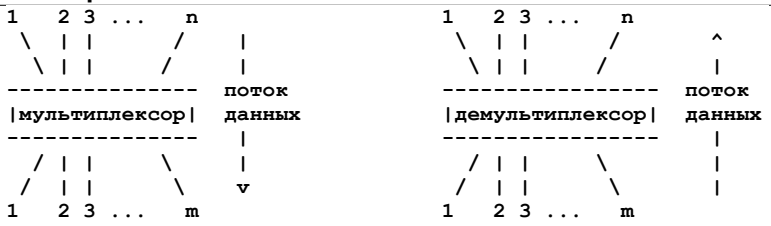

Рисунок 4. Мультиплексор и демультиплексор n x m.

2.5 IP создает одну логическую сеть

Модуль IP является краеугольным камнем технологии Internet. Каждый модуль или драйвер добавляет свой заголовок к пакету по мере прохождения информации от верхнего уровня к нижнему через стек протоколов. На приемной стороне каждый модуль или драйвер извлекает из пакета соответствующий заголовок. Заголовок IP содержит IP-адрес, позволяющий построить одну логическую сеть на базе множества физических сетей. Такое соединение множества сетей между собой и послужило основой для создания термина **Internet**. Множество соединенных между собой физический сетей, объединенное общими пространством уникальных адресов IP, называется internet.

2.6 Независимость от физического интерфейса

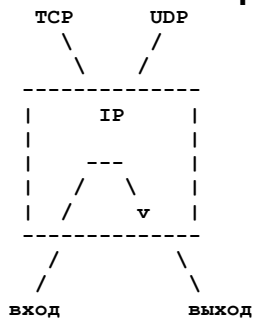

Рисунок 5. Пример пересылки пакета IP.

IP «прячет» сетевое оборудование нижележащих уровней от сетевых приложений. Если вы создали новую физическую сеть, ее можно соединить с другими сетями (internet), используя соответствующие драйверы и модули IP. Таким образом, сетевые приложения становятся независимыми от физических интерфейсов и не подвержены влиянию в результате замены сетевых устройств или подключения новой технологии.

2.7 Функциональная совместимость

Если два компьютера могут обмениваться данными через сеть, мы говорим, что они функционально совместимы. Если технология internet реализована на хорошем уровне, эта технология должна обеспечивать функциональная совместимость. Пользователи обычных (general-purpose - общего назначения) компьютеров могут воспользоваться преимуществами internet за счёт обеспечения функциональной совместимости с компьютерами других типов. В общем случае купленный вами компьютер может оказаться функционально несовместимым, т. е. он не сможет взаимодействовать с другими компьютерами по причине использования в нем экзотических узлов и протоколов.

2.8 Что дальше?

После рассмотрения первооснов мы ответим на следующие вопросы:

- Как определяется Ethernet-адрес получателя при отправке пакетов IP?
- Как протокол IP узнает об используемом на физическом уровне интерфейсе при передаче пакетов IP?
- Как клиентская программа на одном компьютере связывается с программой-сервером на другом?
- Почему используются оба протокола TCP и UDP, а не выбран какой-либо один?
- Какие сетевые приложения доступны?

Ответы на все эти вопросы будут приведены ниже, после краткого рассмотрения технологии Ethernet.

3. Ethernet

В этом параграфе приведено краткое описание технологии Ethernet.

Кадры Ethernet содержат адреса получателя (destination) и отправителя (source), поле типа и данные.

Размер адреса Ethernet составляет 6 байтов. Каждое устройство имеет свой адрес Ethernet и слушает кадры Ethernet, просматривая в них поле с адресом получателя для определения «своих» кадров. Все устройства также воспринимают кадры Ethernet с широковещательным (broadcast) адресом FF-FF-FF-FF-FF-FF (шестнадцатиричный формат).

Ethernet использует технологию CSMA/CD (Carrier Sense and Multiple Access with Collision Detection - множественный доступ с детектированием несущей и обнаружением конфликтов). При использовании CSMA/CD все устройства подключены к общей среде (сегменту, кабелю), но в каждый момент времени передавать данные в общую (разделяемую) среду разрешено только одному устройству. Переданные в среду кадры слышат все устройства. При попытке двух (или более) устройств одновременно передавать данные в среду возникает конфликт (или коллизия - collision), который обнаруживается с помощью CSMA/CD. При возникновении конфликта все станции должны прервать передачу и могут повторять попытку только по истечении некоторого времени (определяется случайным образом).

3.1 Аналогия

Хорошей аналогией Ethernet является разговор группы людей в тёмной комнате. Средой передачи является воздух в комнате, обеспечивающий возможность распространения голоса.

Каждый человек в комнате может слышать, когда кто-то начинает говорить (детектирование несущей). Каждый из находящихся в комнате людей может начать говорить (множественный доступ к среде). Если кто-то из находящихся в комнате слишком разговорчив, его могут попросить покинуть помещение (отключить от сети неисправное устройство).

Один человек в комнате может говорить, остальные слушают. Однако существует вероятность того, что несколько человек начнут говорить одновременно (конфликт). В этом случае никто не может услышать нормальной речи и все говорящие должны замолчать. По истечении некоторого времени кто-либо может начать говорить снова. Таким образом, во избежание конфликтов каждый желающий что-либо сказать должен сначала убедиться, что не говорит кто-то другой.

Каждый человек в комнате имеет свое имя (уникальный адрес Ethernet), позволяющее точно адресовать сообщения. Когда кто-либо начинает говорить, он сначала адресует свои слова кому-то из слушателей (привет, имярек, ...). Если говорящий хочет обратиться ко всем, он передает свое сообщение в широковещательном режиме (эй, люди, ...).

4. ARP

Как при передаче пакета IP определить Ethernet-адрес получателя?

Протокол преобразования адресов ARP (Address Resolution Protocol) позволяет определить адрес Ethernet на основе IP-адреса. Преобразование осуществляется только для исходящих пакетов при создании полей заголовков IP и Ethernet .

4.1 Таблица ARP для преобразования адресов

Преобразование адресов выполняется путем просмотра таблицы ARP, хранящейся в памяти компьютера и содержащей строку с парой адресов (IP и Ethernet) для каждого компьютера. При трансляции IP -> Ethernet в таблице находится строка, содержащая нужный IP-адрес и из второго поля найденной строки берется искомый адрес Ethernet. Ниже показан пример небольшой таблицы ARP.

Адрес	
IP	**Ethernet**
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

Для записи адресов IP используют 4 десятичных поля (по одному для каждого байта), разделенных точками. Адреса Ethernet записываются в шестнадцатеричном формате с разделением байтов пробелом или знаком «-».

Таблица ARP необходима для работы, поскольку адреса IP и Ethernet никак не связаны между собой (возможны произвольные комбинации этих адресов). Адреса IP задаются администраторами сетей из выделенного для сети пространства, а адреса Ethernet задаются производителями оборудования при его производстве. В случае перемещения компьютера в другую подсеть IP-адрес этого компьютера изменяется, а для смены адреса Ethernet в компьютере нужно заменить сетевой адаптер.

4.2 Типичный вариант преобразования адресов

При нормальной работе сетевых приложений (таких, как TELNET) программа передает сообщение модулю TCP, этот модуль шлет соответствующий TCP-сегмент IP-модулю. IP-адрес получателя известен прикладной программе, модулю TCP и модулю IP. Модуль IP завершает подготовку пакета и может передать его драйверу Ethernet, но сначала он должен определить Ethernet-адрес получателя этого пакета.

Для поиска адресов Ethernet используются таблицы ARP.

4.3 Пара ARP Request/Response (запрос - отклик)

Откуда же появляются записи об адресах в таблице ARP?

В процессе работы специальная программа (демон) заполняет таблицу по мере появления информации об адресах.

Если нужного адреса нет в таблице ARP, выполняются следующие операции:

1. Передается запрос ARP с широковещательным адресом Ethernet.
2. Исходящий пакет IP помещается в очередь.

Интерфейс Ethernet каждого из компьютеров получает широковещательный кадр Ethernet с запросом ARP. После этого каждый из интерфейсов Ethernet проверяет значение поля Type (тип) в полученном кадре и передает пакет ARP модулю ARP. Пакет запроса ARP говорит: "Если ваш IP-адрес соответствует IP-адресу получателя пакета, скажите мне свой Ethernet-адрес." Пакет запроса ARP выглядит следующим образом:

IP-адрес отправителя 223.1.2.1 Ethernet-адрес отправителя 08-00-39-00-2F-C3
IP-адрес получателя 223.1.2.2 Ethernet-адрес получателя <пустое поле>

Каждый модуль ARP проверяет локальные адреса IP и адрес получателя на предмет их совпадения. Если адреса совпадают, модуль шлет отклик на запрос, содержащий искомый адрес, по Ethernet-адресу отправителя запроса ARP. Пакет с откликом ARP говорит: "Да, искомый адрес IP принадлежит мне и я сообщаю свой адрес Ethernet ." Пакет отклика ARP меняет значения адресов отправителя/получателя в соответствии с направлением передачи отклика:

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

Отклик приходит обратно на компьютер, пославший запрос. Драйвер Ethernet просматривает поле Type в кадре Ethernet и передает пакет модулю ARP. Модуль ARP проверяет пакет и добавляет адреса IP и Ethernet в таблицу ARP.

Пример обновленной таблицы показан ниже:

Адрес IP	*Адрес Ethernet*
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

4.4 Продолжение трансляции адресов

Новая запись помещается в таблицу в течение нескольких миллисекунд после возникновения потребности в соответствующем адресе. Как было указано в п. 2 выше, исходящий пакет IP был помещен в очередь. На следующем этапе выполняется требуемое преобразование адресов (IP - Ethernet) и кадр передается через сеть Ethernet. Следовательно, добавление в процесс трансляции новых операций 3, 4, 5 завершает сценарий преобразования адресов:

1. Передается запрос ARP с широковещательным адресом Ethernet.
2. Исходящий пакет IP помещается в очередь.
3. Приходит отклик ARP и в таблицу вносится новая пара адресов IP - Ethernet.
4. Для помещенного в очередь пакета IP определяется адрес Ethernet из обновленной таблицы ARP.
5. Кадр Ethernet передается в сеть Ethernet.

При отсутствии в таблице ARP нужной записи один пакет IP просто помещается в очередь. Нужные для преобразования данные быстро вносятся в таблицу ARP с использованием пары пакетов request/response и пакет IP из очереди передается в сеть.

Каждый компьютер поддерживает свою таблицу ARP для каждого из имеющихся в нем интерфейсов Ethernet. Если искомый компьютер отсутствует в локальной сети, для запрошенного адреса не приходит отклика ARP, а в таблице ARP не будет нужной записи. Пакеты IP, передаваемые по этому адресу, будут отбрасываться модулем IP.

Некоторые реализации IP и ARP не используют очереди для пакетов IP в процессе ожидания отклика ARP. Вместо размещения в очереди пакет просто отбрасывается и потом должен быть восстановлен модулем TCP или сетевым приложением UDP. Такое восстановление выполняется по истечении заданного времени (тайм-аут) путем повторной передачи пакета. Повторный пакет передается успешно, поскольку в таблице ARP уже присутствует нужная запись.

5. Протокол IP

Модуль IP является центральным узлом технологии internet и сущность IP заключается в таблицах маршрутизации. Модуль IP использует хранящиеся в памяти таблицы маршрутов для принятия решений о пересылке пакетов IP. Содержимое таблицы маршрутизации определяется опциями, установленными администратором сети. Ошибки в таблицах маршрутизации могут сделать работу сети просто невозможной.

Понимание процессов маршрутизации является основой, обеспечивающей успех работы сетевого администратора.

Чтобы лучше понять таблицы маршрутизации рассмотрим сначала сам процесс маршрутизации, изучим адресацию IP и тогда приступим к детальному рассмотрению таблиц.

5.1 Прямая маршрутизация

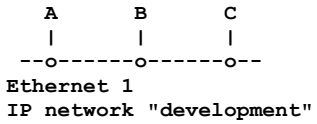

Рисунок 6. Сеть IP.

На рисунке 6 показана небольшая сеть с 3 компьютерами - A, B и C. Каждый из компьютеров имеет стек TCP/IP, показанный на рисунке 1. Адаптер Ethernet в каждом из компьютеров имеет уникальный адрес Ethernet. Для каждого компьютера администратор сети выделил адрес IP, который связан с интерфейсом Ethernet, установленным в компьютере.

Когда A передает IP-пакет компьютеру B, в заголовке пакета IP содержится IP-адрес компьютера A (адрес отправителя) и адрес Ethernet компьютера A. В заголовке IP содержится также IP-адрес компьютера B (адрес получателя) и адрес Ethernet компьютера B.

Таблица 1. Адреса в кадре Ethernet для пакета IP от A к B.

Адрес	*Отправитель*	*Получатель*
Заголовок IP	A	B
Заголовок Ethernet	A	B

В этом простом примере IP почти ничего не добавляет к сервису, обеспечиваемому Ethernet. Однако IP увеличивает расход системных ресурсов - требуется дополнительное процессорное время и дополнительная полоса канала для генерации, передачи и разборки заголовков IP.

Когда IP-модуль компьютера B принимает пакет IP от компьютера A, он проверяет IP-адрес получателя (на предмет соответствия своему адресу) и передает дейтаграмму вышележащему уровню (если адреса совпадают).

Обмен пакетами между A и B использует прямую маршрутизацию (direct routing).

5.2 Непрямая маршрутизация

На рисунке 7 представлен более реальный пример internet - сеть содержит три сети Ethernet (три сети IP), соединенных IP-маршрутизатором (компьютер D). Каждая из сетей IP содержит по 4 компьютера и каждый из этих компьютеров имеет свои адреса IP и Ethernet.

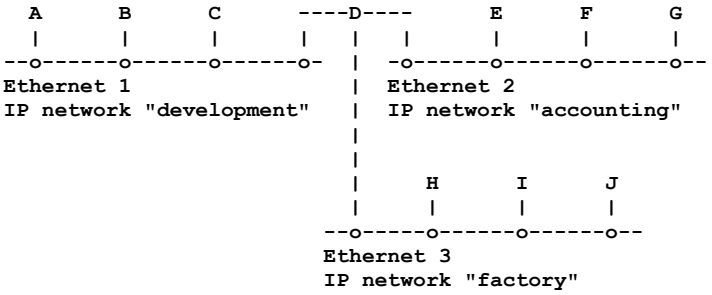

Рисунок 7. Три сети IP в одной сети internet.

На всех компьютерах, за исключением D, используется стек TCP/IP, показанный на рисунке 1. Компьютер D является IP-маршрутизатором - он подключен ко всем 3 сетям и, следовательно, имеет три адреса IP и 3 адреса Ethernet. В компьютере D используется стек TCP/IP, схематически изображенный на рисунке 3 (вместо двух адаптеров используется три). В компьютере D используются 3 модуля ARP и 3 драйвера Ethernet, но модуль IP по-прежнему один.

Администратор сети присваивает уникальный номер (IP-адрес) каждому из адаптеров Ethernet. Адреса IP не показаны на рисунке 7 - сети обозначены именами.

Когда компьютер A передает пакет IP компьютеру B, процесс не отличается от описанной выше прямой маршрутизации. Любой обмен пакетами между компьютерами одной сети IP осуществляется с использованием прямой маршрутизации.

При взаимодействии компьютеров D и A также используется прямая маршрутизация, аналогично протекает и процесс взаимодействия между компьютерами D и E, D и H (каждая из этих пар находится в одной сети IP).

Однако при обмене пакетами между компьютером A, расположенным по одну сторону маршрутизатора, и компьютером, расположенным по другую сторону маршрутизатора IP, прямая маршрутизация уже не будет работать. Компьютер A должен использовать маршрутизатор D для пересылки пакетов IP в другую IP-сеть. Такой процесс называется непрямой маршрутизацией (indirect routing).

Маршрутизация пакетов IP осуществляется IP-модулями и прозрачна для TCP, UDP и сетевых приложений.

Если A передает пакет IP компьютеру E, в качестве адресов отправителя указаны адреса IP и Ethernet компьютера A. Получателем пакетов IP является компьютер E и его адрес IP указывается в заголовке пакета, но, поскольку IP-модуль компьютера A посылает пакет маршрутизатору D для дальнейшей пересылки, в качестве Ethernet-адреса получателя указывается адрес D.

Таблица 2. Адреса в кадре Ethernet для пакета IP от A к E (до D).

Адрес	***Отправитель***	***Получатель***
Заголовок IP	A	E
Заголовок Ethernet	A	D

Модуль IP компьютера D получает пакет IP и проверяет IP-адрес получателя, после чего может сказать: "Это не мой адрес," - и переслать пакет IP компьютеру E, используя прямую маршрутизацию.

Таблица 3. Адреса в кадре Ethernet для пакета IP от A к E (после D).

Адрес	***Отправитель***	***Получатель***
Заголовок IP	A	E
Заголовок Ethernet	D	E

В заключении отметим, что для прямой маршрутизации используются явно указанные пары адресов IP - Ethernet отправителя и получателя, а в случае непрямой маршрутизации такие пары использоваться не могут.

Приведенный пример internet очень прост. В реальных сетях ситуация осложняется множеством факторов и приходится использовать многочисленные маршрутизаторы IP и различные типы физических сетей. Приведенный выше пример показывает, как администратор может разбить большую сеть Ethernet на несколько сетей для снижения размеров областей распространения широковещательного трафика Ethernet.

5.3 Правила маршрутизации модуля IP

В этом кратком обзоре рассмотрен лишь сам процесс маршрутизации без описания механизмов последней. Мы поговорим лишь о правилах (алгоритмах), используемых модулем IP.

Для исходящих пакетов IP, полученных от вышележащего уровня, модуль IP должен решить следует посылать пакет с использованием прямой или непрямой маршрутизации, после чего пакет передается сетевому интерфейсу. Для принятия решения о способе маршрутизации используется таблица маршрутизации.

Для входящих пакетов IP, принимаемых от сетевого интерфейса, модуль IP должен решить следует ли передавать пакет вышележащему уровню. Если пакет должен быть передан на другой сетевой интерфейс, он трактуется уже как исходящий пакет (см. выше). Принятый от сетевого интерфейса пакет никогда не пересылается в тот же самый интерфейс.

5.4 IP-адрес

Администратор сети распределяет IP-адреса для компьютеров в соответствии с физической сетью, к которой подключен каждый компьютер. Часть 4-байтового адреса IP задает номер IP-сети, остальные биты адреса определяют номер данного компьютера в сети IP (номер хоста или host number). Для компьютера, приведенного в таблице 1, адрес IP имеет значение 223.1.2.1, номер сети - 223.1.2, а номер компьютера в сети - 1.

Сетевая часть адреса (номер сети) задается старшими битами адреса, а номер хоста - младшими битами. Все примеры адресов IP в данном документе даются для сетей класса C, это означает, что три старших байта (24 старших бита) определяют номер сети, а восемь младших битов - номер хоста. Такая адресация позволяет организовать 2,097,152 сетей класса C, каждая из которых может содержать по 254[1] хоста.

Адреса IP распределяются сетевым информационным центром - NIC (Network Information Center). Все сети, подключенные ко всемирной сети Internet, должны использовать сетевые номера в соответствии с полученными от NIC значениями. Если вы организуете частную сеть и не планируете напрямую подключать эту сеть к Internet, вы все равно должны получить адреса в NIC. Вы можете выбрать адреса для частной сети и без обращения в NIC, но при этом наверняка возникнут проблемы в случае соединения с другими сетями[2].

5.5 Имена

Большинство людей предпочитают использовать для обозначения компьютеров имена, а не числовые адреса. В приведенном ниже списке компьютер alpha использует адрес 223.1.2.1. Для небольшой сети преобразование имен в адреса может осуществляться путем просмотра файла, содержащего пары имя - адрес и хранящегося на каждом компьютере сети (обычно этот файл называется hosts). Для крупных сетей используются специальные серверы, на которых хранятся данные для преобразования имен компьютеров в адреса. К таким серверам обеспечивается доступ из любой точки сети. Информация об адресах и именах хранится в форме, подобной показанному ниже списку:

```
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 первой колонке указаны IP-адреса, а вторая колонка содержит имена компьютеров.

Во многих случаях возможна установка идентичных файлов hosts на все компьютеры сети. Вы могли заметить, что в приведенном примере компьютеру delta соответствует только одна строка в списке, хотя этот компьютер имеет 3 адреса IP. Для доступа к компьютеру Delta может использоваться любой из связанных с этим компьютером адресов IP. Когда компьютер delta принимает пакет IP и просматривает адрес получателя, он воспринимает пакеты для всех своих адресов IP.

Символьные имена используются и для сетей IP. Если у вас имеется 3 сети IP, в файле networks будут содержаться записи типа:

```
223.1.2     development
223.1.3     accounting
223.1.4     factory
```

В первой колонке указаны номера сетей IP, во второй - их имена.

Из приведенных примеров вы можете сделать вывод, что компьютер alpha имеет номер 1 в сети development, а компьютер beta - номер 2 в той же сети. Вы может использовать для этих компьютеров обозначения development.1 (alpha), development.2 (Beta) и т. д..

Описанные файлы удобны для пользователя, но администратор может предпочесть иной вариант записи для компьютера delta:

```
223.1.2.4     devnetrouter    delta
223.1.3.1     facnetrouter
223.1.4.1     accnetrouter
```

В трех строках файла указаны IP-адреса и имена. Фактически, первая строка задает два имени - delta и devnetrouter, которые являются синонимами. Имя delta служит для общего пользования, а остальные 3 имени применяются только для администрирования таблицы маршрутизации IP.

Эти файлы используются программами для нахождения соответствий между именами и адресами компьютеров. Сеть может работать с безымянными компьютерами, но имена удобней для пользователей.

5.6 Таблица IP-маршрутизации

Как модуль IP узнает, какой из сетевых интерфейсов использовать для передачи пакета IP? Модуль IP просматривает таблицу маршрутизации, используя в качестве ключей поиска номера сетей, определенные из IP-адресов получателей.

Таблица маршрутизации содержит по одной строке для каждого маршрута. Первая колонка таблицы указывает номер сети IP, вторая - флаг direct/indirect (прямая/непрямая маршрутизация), третья - IP-адрес маршрутизатора и последняя - номер сетевого интерфейса. По этой таблице можно определить интерфейс, через который следует передавать пакеты с каждым из адресов IP.

[1] Адреса 0 и 255 используются для служебных целей. *Прим. перев.*

[2] Для частных сетей выделены специальные группы адресов (см. RFC 1918), в частности - 192.168.0.0. *Прим. перев.*

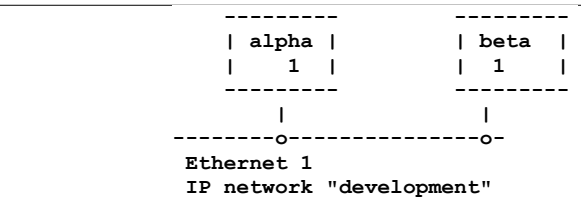

Рисунок 8. Фрагмент IP-сети.

На большинстве компьютеров таблицу маршрутизации можно просматривать и менять с помощью команды route. Содержимое таблицы маршрутизации задается администратором сети, поскольку он отвечает за распределение адресов IP между компьютерами.

5.7 Детали прямой маршрутизации

Для понимания процесса прямой маршрутизации вернемся к ранее рассмотренному примеру.

Таблица маршрутизации для компьютера alpha будет иметь вид:

Таблица 4. Пример простой таблицы маршрутизации с именами.

Сеть	*Флаг direct/indirect*	*маршрутизатор*	*Номер интерфейса*
development	direct		1

Получить эту таблицу на экране в большинстве UNIX-систем можно с помощью команды netstat -r. В простой сети таблицы маршрутизации на всех компьютерах будут идентичны.

Ниже приведен вариант той же таблицы с адресами сетей взамен их имен.

Таблица 5. Пример простой таблицы маршрутизации с адресами.

Сеть	*Флаг direct/indirect*	*маршрутизатор*	*Номер интерфейса*
223.1.2	direct		1

5.8 Сценарий прямой маршрутизации

Alpha передает пакет IP компьютеру beta. Пакет IP находится в модуле IP компьютера alpha и в нем указан IP-адрес получателя - beta или 223.1.2.2. Модуль IP выделяет сетевую часть адреса IP и просматривает первую колонку таблицы маршрутизации в поисках нужной записи. В нашем случае эта запись является единственной.

Остальные колонки этой записи показывают, что компьютеры этой сети доступны с использованием прямой маршрутизации через интерфейс 1. Таблица ARP позволяет определить адрес Ethernet для компьютера beta и пакет передается этому компьютеру через интерфейс 1.

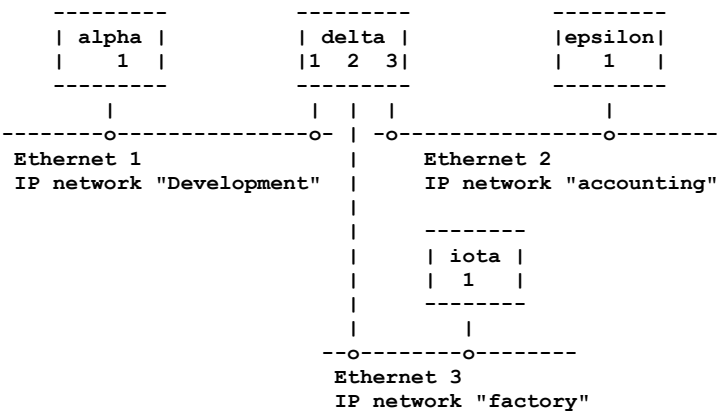

Рисунок 9. Непрямая маршрутизация для трех сетей IP.

Если приложение пытается передать данные по другому адресу IP, не входящему в сеть development, модуль IP не сможет найти соответствующей записи в таблице маршрутизации. В таком случае модуль IP просто отбрасывает пакеты, а приложение получает сообщение об ошибке типа Network not reachable (сеть недоступна).

5.9 Детали непрямой маршрутизации

Вернемся к более сложному примеру маршрутизации, который мы уже рассматривали ранее.

Таблица маршрутизации в компьютере alpha будет иметь вид:

Таблица 6. Таблица маршрутизации Alpha.

Сеть	*Флаг direct/indirect*	*Маршрутизатор*	*Номер интерфейса*
development	direct		1
accounting	indirect	devnetrouter	1
factory	indirect	devnetrouter	1

Ниже приведён вариант той же таблицы с адресами сетей взамен их имен.

Таблица 7. Таблица маршрутизации Alpha.

Сеть	*Флаг direct/indirect*	*Маршрутизатор*	*Номер интерфейса*
223.1.2	direct		1
223.1.3	indirect	devnetrouter	1
223.1.4	indirect	devnetrouter	1

В качестве адреса маршрутизатора в таблице маршрутизации компьютера Alpha указан IP-адрес интерфейса delta, подключенного к сети development.

5.10 Сценарий непрямой маршрутизации

Alpha передает пакет IP компьютеру epsilon. Пакет IP находится в модуле IP компьютера alpha и содержит адрес получателя epsilon (223.1.3.2). Модуль IP выделяет сетевую часть адреса IP (223.1.3) и просматривает первую колонку таблицы маршрутизации в поисках соответствия. Искомая запись находится во второй строке таблицы.

Эта запись показывает, что доступ к компьютерам сети 223.1.3 может осуществляться через IP-маршрутизатор devnetrouter. Модуль IP в компьютере Alpha выполняет трансляцию адресов с помощью таблицы ARP для IP-адреса маршрутизатора devnetrouter и шлет пакет IP этому маршрутизатору через свой сетевой интерфейс 1. Пакет IP по-прежнему в качестве адреса получателя содержит адрес epsilon.

Пакет IP приходит на сетевой интерфейс компьютера delta и передается модулю IP. Модуль проверяет адрес получателя и, не найдя совпадения ни с одним из своих интерфейсов, решает переслать пакет IP.

Модуль IP в компьютере Delta выделяет сетевую часть адреса получателя (223.1.3) и просматривает свою таблицу маршрутизации:

Таблица 8. Таблица маршрутизации Delta.

Сеть	***Флаг direct/indirect***	***Маршрутизатор***	***Номер интерфейса***
development	direct		1
accounting	direct		3
factory	direct		2

Ниже приведен вариант этой таблицы с адресами сетей взамен их имен.

Таблица 9. Таблица маршрутизации Delta.

Сеть	***Флаг direct/indirect***	***Маршрутизатор***	***Номер интерфейса***
223.1.2	direct		1
223.1.3	direct		3
223.1.4	direct		2

Нужная запись находится во второй строке таблицы. Модуль IP пересылает пакет IP компьютеру epsilon напрямую через интерфейс 3. Пакет содержит адреса IP и Ethernet компьютера epsilon.

Пакет IP приходит в компьютер epsilon и передается модулю IP, который проверяет IP-адрес и, найдя соответствие со своим адресом, передает пакет на вышележащий уровень.

5.11 Маршрутизация в больших сетях

При передаче пакетов IP через большую сеть они могут пройти через множество маршрутизаторов на пути к получателю. Путь пакета не может быть задан отправителем и определяется всякий раз путем просмотра таблиц маршрутизации вдоль пути доставки пакета. Каждый из компьютеров (маршрутизаторов) задает только адрес следующего маршрутизатора (next hop), которому пакет передается для дальнейшей пересылки.

5.12 Управление маршрутами

Поддержка корректных таблиц маршрутизации на всех компьютерах большой сети является сложной задачей, поскольку конфигурация сети постоянно изменяется администраторами с учетом требований пользователей. Ошибки в таблицах маршрутизации могут полностью нарушить работу сети, а поиск таких ошибок является достаточно сложной задачей.

Сохранение простоты в сетевой конфигурации играет важную роль при создании надежной сети. Например, наиболее простым способом распределения IP-адресов для сетей Ethernet является выделение единого номера сети IP для всей сети Ethernet.

При настройке маршрутизации могут оказать помощь некоторые сетевые протоколы и приложения. Протокол ICMP (Internet Control Message Protocol) позволяет находить проблемы в маршрутизации. Для небольших сетей таблицы маршрутизации создаются администраторами вручную на каждом компьютере. Для больших сетей используются специальные протоколы маршрутизации, обеспечивающие распространение таблиц маршрутизации по сети.

При переносе компьютера из одной IP-сети в другую, IP-адрес этого компьютера должен быть изменен. При удалении компьютера из сети его IP-адрес перестает быть корректным. Такие изменения в сети требуют постоянного обновления файлов hosts. В сетях даже средних размеров решение этой задачи может потребовать значительных ресурсов. Система доменных имен и службы DNS (Domain Name System) помогают решить эту проблему.

6. Протокол UDP

UDP является одним из 2 основных протоколов, используемых поверх IP. Этот протокол предоставляет свой сервис пользовательским сетевым программам. Примерами сетевых приложений на базе протокола UDP являются NFS (Network File System - сетевая файловая система) и SNMP (Simple Network Management Protocol - простой протокол управления сетью).

UDP обеспечивает сервис по доставке дейтаграмм (datagram) без организации соединений (connectionless) и гарантий доставки. Протокол UDP не организует сквозных (end-to-end) соединений с удаленными модулями UDP, он просто передаёт дейтаграммы в сеть и принимает их из сети.

UDP добавляет две новых услуги к тем, что предоставляются протоколом IP. Первой является мультиплексирование информации между приложениями по номерам портов, а второй - поддержка проверки целостности пакетов с помощью контрольных сумм.

6.1 Порты

Как клиентская программа на одном компьютере может связаться с сервером на другом компьютере?

Связь между приложениями и UDP осуществляется через порты UDP, задаваемые номерами (с нуля). Программа, которая предлагает свои услуги (сервер), ждет сообщений, адресованных в порт, выделенный для этого сервиса. Серверы терпеливо ждут любых запросов от клиентов.

Например, сервер SNMP (их называют агентами SNMP) всегда слушает порт 161. На компьютере может использоваться только один агент SNMP, поскольку для такого сервиса выделяется единственный порт UDP. Номер этого порта известен другим приложениям (well known) и клиенты SNMP, желающие прибегнуть к услугам агента, адресуют свои запросы в порт 161 по протоколу UDP на интересующем их компьютере.

Когда приложение передает данные с помощью UDP, эти данные приходят удаленному адресату в едином блоке. Например, если программа делает 5 записей в порт UDP, программа на удаленной стороне будет 5 раз выполнять операцию чтения из порта UDP. Размеры записываемых в порт блоков данных также совпадают с размерами читаемых из порта блоков.

UDP сохраняет границы сообщений, заданные приложениями, протокол никогда не объединяет сообщения и не делит их на части.

6.2 Контрольная сумма

Когда принимаемый пакет IP содержит в поле типа значение UDP, такой пакет передается модулю UDP. Когда UDP-модуль получает дейтаграмму UDP от модуля IP, он проверяет контрольную сумму UDP. Если поле контрольной суммы имеет нулевое значение, это говорит о том, что контрольная сумма не была задана отправителем и не должна приниматься во внимание. Модуль UDP на компьютере отправителя может указывать контрольную сумму или опускать её. Если между двумя модулями UDP находятся только сети Ethernet, контрольная сумма может и не потребоваться. Однако, рекомендуется использовать контрольные суммы, поскольку данные могут передаваться и через менее надёжные среды.

Если контрольная сумма имеет корректное значение или равна 0, проверяется порт назначения и (при наличии связи между этим портом и приложением) пакет помещается в очередь приложения для последующего прочтения. В противном случае дейтаграмма UDP просто отбрасывается. Если дейтаграммы UDP приходят из сети быстрее, чем приложение может их читать и очередь переполняется, протокол UDP также будет отбрасывать дейтаграммы, не помещающиеся в очередь.

7. Протокол TCP

Протокол TCP обеспечивает дополнительный сервис. TCP обеспечивает поддержку потоков данных на базе организованных соединений в отличие от передачи дейтаграмм без организации соединений, используемой протоколом UDP. Кроме того, протокол TCP обеспечивает гарантированную доставку.

TCP используется сетевыми приложениями, которым нужна гарантия доставки. Основными приложениями, использующими TCP, являются FTP (File Transfer Protocol - протокол передачи файлов) и TELNET. К популярным приложениям TCP относятся также X-Window, rcp (remote copy - удаленное копирование) и команды r-серии. Расширение возможностей TCP не проходит бесследно и для поддержки протокола требуются дополнительные ресурсы процессора и полоса канала связи. Устройство модуля TCP значительно сложнее по сравнению с модулем UDP.

Подобно UDP сетевые приложения подключаются к портам TCP. Хорошо известные номера портов выделены для популярных приложений. Например, сервер TELNET использует порт 23. Клиенты TELNET могут найти сервер, просто подключившись к порту 23 на нужном компьютере по протоколу TCP.

Когда приложение начинает сеанс работы с TCP, модули TCP на клиентском компьютере и сервере организуют между собой сеанс связи (сессию). Информация о состоянии соединения между конечными точками определяет виртуальное устройство (virtual circuit). Такое виртуальное устройство потребляет ресурсы на обеих сторонах соединения TCP. Виртуальное устройство является полнодуплексным - данные могут одновременно передаваться через него в обоих направлениях. Приложение записывает данные в порт TCP, эти данные передаются через сеть и прочитываются приложением на другом конце соединения.

TCP превращает поток байтов в пакеты, не сохраняя границ пользовательских сообщений. Например, если приложение делает 5 записей в порт TCP, приложение на другой стороне может прочесть из за 10 приемов или может случиться так, что все данные будут прочитаны за одно обращение к порту. Здесь нет корреляции между числом и размером записей в порт и числом обращений для прочтения данных на удаленной стороне соединения.

TCP является протоколом со скользящим окном (sliding window) и поддержкой тайм-аутов и повторных передач. Удаленная сторона должна подтвердить прием отправленных ей данных (подтверждения могут прицепляться к передаваемым данным). Управление потоком данных на обеих сторонах соединения предотвращает переполнение буферов.

Как и все протоколы со скользящим окном, TCP задает размер этого окна. Размер окна определяется количеством данных, которые можно передать до приема подтверждения. Для TCP размер окна задается не числом сегментов TCP, а числом передаваемых байтов.

8. Сетевые приложения

Зачем нужны два протокола TCP и UDP?

На самом деле эти протоколы обеспечивают разные наборы услуг. Многие приложения способны работать только с определенными типами сервиса. Если вы разрабатываете сетевые приложения, выбирайте протокол, который способен лучше решить поставленные задачи. Если требуется поддержка надежной доставки потока данных, лучше использовать протокол TCP. Если же вам требуется поддержка дейтаграмм, протокол UDP будет предпочтительней. Если нужна эффективная передача данных на значительное расстояние, TCP обеспечит лучшее решение, а для передачи данных в скоростных сетях с малыми задержками предпочтительней использовать протокол UDP. Если ваши задачи не подходят под перечисленные категории, выбор протокола становится более сложной задачей. Однако

приложение может взять на себя часть функций, которые не реализованы протоколом. Если вам требуется обеспечить надежную доставку данных на основе UDP, вопросы надежности должны быть решены в прикладной программе. Если вы выбрали протокол TCP и потребовалось организовать сервис на базе записей, приложение может помещать в поток информации маркеры, обозначающие границы каждой записи.

Какие сетевые приложения доступны?

Список сетевых приложений очень велик и число их постоянно возрастает. Некоторые приложения появились вместе с технологией internet (например, TELNET и FTP), а другие - сравнительно недавно (скажем, X-Windows или SNMP). Ниже приведен краткий обзор популярных сетевых приложений.

8.1 TELNET

TELNET обеспечивает возможность удаленного входа в систему по протоколу TCP. Организация удаленного входа в систему чем-то напоминает соединение с удаленным абонентом через телефонную сеть. Введя команду **telnet delta**, пользователь получит на экране приглашение на ввод имени пользователя компьютера delta.

TELNET работает хорошо - эта программа разработана давно и проверена годами практического использования. Доступ с помощью TELNET обычно не зависит от используемой на клиентской машине операционной системы. Например клиент TELNET может использоваться на компьютере VAX/VMS, а сервер может работать в среде UNIX System V.

8.2 FTP

Протокол передачи файлов FTP столь же стар, как TELNET, и также использует TCP. Работа с FTP похожа на сеансы TELNET и отличается, прежде всего, используемым набором команд, который специализирован именно для файловых операций (копирование файлов с одного компьютера на другой).

8.3 rsh

Remote shell (rsh или remsh) является одной из серии «удаленных» команд UNIX (например, команда копирования UNIX - cp имеет «удаленный» аналог - rcp, команда who - имеет аналог rwho и т. д.).

Команды r* работают прежде всего между системами UNIX и предназначены для организации взаимодействия между доверяющими друг другу (trusted) хостами. Использование таких команд несколько снижает уровень безопасности, но обеспечивает массу удобств для пользователей.

Для выполнения команды cc file.c (трансляция программы C) на удаленном компьютере достаточно ввести команду **rsh delta cc file.c**. Для копирования файла file.c на компьютер delta просто наберите **rcp file.c delta**; для входа в систему delta служит команда **rlogin delta** и т. п.

8.4 NFS

Сетевая файловая система NFS, разработанная Sun Microsystems Inc, использует протокол UDP, и очень удобна для монтирования файловых систем UNIX на множестве компьютеров. Бездисковые станции могут работать с дисками сервера (как будто это диск данной станции). База данных, хранящаяся на компьютере alpha, может использоваться с компьютера beta, если файловая система смонтирована на этом компьютере.

Использование NFS существенно повышает сетевой трафик и на медленных каналах NFS может не обеспечивать требуемой производительности, но преимущества использования дисков через сеть все равно достаточно велики. Клиенты NFS реализованы в ядре UNIX, что позволяет всем приложениям использовать разделы NFS как локальные диски компьютера.

8.5 SNMP

Протокол SNMP (Simple Network Management Protocol - простой протокол управления сетью) использует UDP и предназначен для организации управления сетью с центральной консоли. Очевидно, что при наличии достаточного объема информации администратору проще обнаружить и решить возникающие в сети проблемы. Консоль администратора использует протокол SNMP для сбора данных от других устройств сети. SNMP определяет форматы данных, оставляя вопросы их интерпретации на усмотрение управляющей станции и администратора сети.

8.6 X-Window

Оконная система X Window использует одноименный протокол поверх TCP для организации графической оконной среды на рабочих станциях. X Window представляет собой много больше, нежели просто систему вывода окон на экран - это целая философия организации пользовательских интерфейсов.

9. Дополнительная информация

В это краткое руководство включена лишь мизерная часть информации о технологии internet. В данном параграфе перечислены вопросы, понимание которых позволит перейти на более высокий уровень тем читателям, которые этого пожелают.

- Команды администрирования: arp, route, netstat
- ARP: постоянные объекты, публикуемые объекты, устаревшие объекты, подстановки (spoofing)
- таблица маршрутизации IP: хост, используемый по умолчанию шлюз, подсети
- IP: время жизни пакетов, фрагментация, ICMP
- RIP, петли в маршрутизации
- система доменных имен DNS

10. Литература

[1] Comer, D., "Internetworking with TCP/IP Principles, Protocols, and Architecture", Prentice Hall, Englewood Cliffs, New Jersey, U.S.A., 1988.

[2] Feinler, E., et al, DDN Protocol Handbook, Volume 2 and 3, DDN Network Information Center, SRI International, 333 Ravenswood Avenue, Room EJ291, Menlow Park, California, U.S.A., 1985.

[3] Spider Systems, Ltd., "Packets and Protocols", Spider Systems Ltd., Stanwell Street, Edinburgh, U.K. EH6 5NG, 1990.

11. Связи с другими RFC

Этот документ является учебным; он **не заменяет** и **не обновляет** другие RFC.

12. Вопросы безопасности

При использовании стека протоколов TCP/IP возникают вопросы, связанные с безопасностью. Некоторые считают эти вопросы важнейшими, другие просто игнорируют их - все зависит от требований системы и подхода администратора.

В этом документе вопросы безопасности не рассматриваются, но если вы хотите узнать больше о безопасности протоколов, начните с подстановок адресов (ARP-spoofing) и прочтите параграф ”Вопросы безопасности” в RFC 1122.

13. Адреса авторов

Theodore John Socolofsky

Spider Systems Limited
Spider Park
Stanwell Street
Edinburgh EH6 5NG
United Kingdom

Телефон:
Из Англии 031-554-9424
Из США 011-44-31-554-9424

Fax:
Из Англии 031-554-0649
Из США 011-44-31-554-0649

EMail: TEDS@SPIDER.CO.UK

Claudia Jeanne Kale

12 Gosford Place
Edinburgh EH6 4BJ
United Kingdom

Телефон:
Из Англии 031-554-7432
Из США 011-44-31-554-7432

EMail: CLAUDIAK@SPIDER.CO.UK

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru