Лекция 8
Тема. Определение имен узлов

План лекции

Определение имен узлов

Протокол определения адреса ARP.

Проблема заключается в том, что IP адреса имеют какое-либо значение только в семействе протоколов TCP/IP. Канальные уровни имеют собственную схему адресации (в основном 48-битные адреса); сетевые уровни, в свою очередь, используют эти канальные уровни. Компьютеры использующие разные сетевые протоколы могут находиться на одном и том же физическом кабеле. Драйвер сетевой платы никогда не смотрит на IP адрес назначения в IP датаграмме. Другими словами возникает необходимость установить соответствие между двумя различными формами адресов: 32-битными IP адресами и каким-либо типом адресов канального уровня. Мы рассмотрим два протокола: протокол определения адреса (ARP - address resolution protocol) и обратный протокол определения адреса (RARP - reverse address resolution protocol). ARP предоставляет динамическое сопоставление IP адресов и соответствующих аппаратных адресов. Мы используем термин динамическое, так как это происходит автоматически и обычно не зависит от используемых прикладных программ или воли системного администратора.

RARP, в основном, используется системами без жестких дисков (бездисковые рабочие станции или X терминалы), однако здесь требуется ручная конфигурация с участием системного администратора.

Пример. Если мы введем команду % **ftp bsdi** будет выполнена следующая последовательность действий:

1. Приложение, FTP клиент, вызывает функцию DNS, чтобы конвертировать имя хоста (bsdi) в 32-битный IP адрес.
2. FTP клиент требует установить TCP соединение с указанным IP адресом.
3. TCP посылает запрос на установление соединения удаленному хосту, посылая IP датаграммы по указанному IP адресу.
4. Если хост назначения подключен к сети (Ethernet, Token ring, или к другому концу канала точка-точка), IP датаграмма может быть послана непосредственно хосту. Если хост назначения находится в удаленной сети, IP маршрутизатор определяет Internet адрес непосредственно подключенного маршрутизатора следующей пересылки, чтобы послать туда IP датаграмму. В обоих случаях IP датаграмма посылается либо хосту, либо маршрутизатору, подключенные непосредственно к данной сети.
5. Если используется Ethernet, посылающий хост должен конвертировать 32-битный адрес в 48-битный Ethernet адрес. Или другими словами, осуществить преобразование из логического Internet адреса в соответствующий физический аппаратный адрес. Этим занимается ARP. ARP работает в широковещательных сетях, где много хостов или маршрутизаторов подключено к одной и той же сети.
6. ARP посылает фрейм Ethernet, который называется ARP запрос (ARP request), каждому хосту в сети. Подобный метод рассылки называется широковещательным запросом (broadcast). ARP запрос содержит IP адрес хоста назначения (имя которого bsdi) и запрос "если Вы владелец этого IP адреса, пожалуйста сообщите мне Ваш аппаратный адрес".
7. Хост назначения на ARP уровне получает этот широковещательный запрос, определяет, что отправитель спрашивает именно его IP адрес, и отвечает на него ARP откликом (ARP reply). Этот отклик содержит IP адрес и соответствующий аппаратный адрес.
8. ARP отклик принимается, и IP датаграмма, из-за которой начался обмен ARP запрос - ARP отклик, может быть послана.
9. IP датаграмма отправляется на хост назначения.

Фундаментальная концепция, заложенная в ARP, заключается в следующем. Сетевой интерфейс имеет аппаратный адрес Фреймы, которыми обмениваются на аппаратном уровне, должны адресоваться к корректному интерфейсу. Однако TCP/IP использует собственную схему адресации: 32-битные IP адреса. Знание IP адреса хоста не позволяет ядру послать датаграмму этому хосту. Драйвер Ethernet должен знать аппаратный адрес пункта назначения, чтобы послать туда данные. В задачу ARP входит обеспечение динамического соответствия между 32-битными IP адресами и аппаратными адресами, используемыми различными сетевыми технологиями.

Эффективность функционирования ARP во многом зависит от ARP кэша (ARP cache), который присутствует на каждом хосте. В кэше содержатся Internet адреса и соответствующие им аппаратные адреса. Стандартное время жизни каждой записи в кэше составляет 20 минут с момента создания записи.

Содержимое ARP кэша можно увидеть с использованием команды arp.

Команда arp. Опции команды:

- -a - отобразить все записи ARP кэша.
- -d, удалить запись из ARP кэша.
- -s, добавить запись в ARP кэша. При использовании этой опции необходимо указать имя хоста и Ethernet адрес, IP адрес, соответствующий имени хоста, и Ethernet адрес добавляются в кэш. Подобная запись делается на постоянной основе (она не будет удалена из кэша по тайм-ауту), если только в конце командной строки не будет использовано ключевое слово temp.

ARP это основной протокол, который используется практически во всех реализациях TCP/IP. Обычно его функционирование не зависит от используемых приложений или воли системного администратора. ARP кэш является фундаментом этой работы. Мы использовали команду arp, чтобы просмотреть или модифицировать кэш. Каждая запись в кэше имеет таймер, который используется для удаления незавершенных или завершенных записей. Команда arp отображает модифицированные записи в ARP кэше.

Мы посмотрели обычное функционирование ARP и специализированные версии: уполномоченный агент ARP (когда маршрутизатор отвечает на ARP запросы для хостов, находящихся на другом интерфейсе маршрутизатора) и "беспричинный" ARP (посылающий ARP запросы для своего собственного IP адреса, обычно во время загрузки).

Программа Ping. Программа Ping предназначена для проверки доступности удаленного хоста. Программа посылает ICMP эхо запрос на хост и ожидает возврата ICMP эхо отклика.

Обычно, если Вы не можете послать Ping на хост, то не сможете получить доступ к этому хосту, используя Telnet или FTP. С другой стороны, если Вы не можете зайти на хост с помощью Telnet, Ping, как правило, начальная точка, с которой начинается идентификация проблемы. Помимо этого, с помощью Ping можно оценить время возврата пакета от хоста, что дает представление о том, "насколько далеко" находится хост. Номер последовательности начинается с 0 и увеличивается на единицу каждый раз когда посылается следующий эхо запрос, ping печатает номер последовательности каждого возвращенного пакета, позволяя нам увидеть, потерялся ли пакет, поменялась ли последовательность движения пакетов и был ли пакет продублирован. Так как IP является ненадежным сервисом доставки датаграмм, любое из трех вышеперечисленных условий может появиться при работе программы ping.

Программа ping является основным тестирующим средством, которое позволяет определить наличие соединения между системами, использующими TCP/IP. Она использует ICMP эхо запрос и эхо отклик и не использует транспортные уровни (TCP или UDP). Ping сервер обычно является частью реализации ядра ICMP.

Служба DNS. Система имен доменов (DNS - Domain Name System) это распределенная база данных, которая используется приложениями TCP/IP, для установления соответствия между именами хостов и IP адресами. DNS также используется для маршрутизации электронной почты. Мы используем термин распределенная, потому что на одном узле Internet не хранится вся необходимая информация. Каждый узел (университет, университетский городок, компания или отдел внутри компании) поддерживает собственную информационную базу данных и запускает программу сервер, которая может отправить запрос по Internet к другим системам. DNS предоставляет протокол, который позволяет клиентам и серверам общаться друг с другом.

Пространство имен DNS имеет иерархическую структуру.

Каждый узел (кружочки на рис. 9.2) имеет метку длиной до 63 символов. Корень дерева это специальный узел без метки. Метки могут содержать заглавные буквы или маленькие. Имя домена (domain name) для любого узла в дереве - это последовательность меток, которая начинается с узла выступающего в роли корня, при этом метки разделяются точками. Каждый узел дерева должен иметь уникальное имя домена, однако одинаковые метки могут быть использованы в различных точках дерева.

Имя домена, которое заканчивается точкой, называется абсолютным именем домена (absolute domain name) или полным именем домена (FQDN - fully qualified domain name). Например, sun.tuc.noao.edu.. Если имя домена не заканчивается на точку, подразумевается, что имя должно быть завершено.

Одна важная характеристика DNS, не показанная на рисунке 5.3, это передача ответственности внутри DNS. Не существует организации, которая бы управляла и обслуживала все дерево в целом и каждую метку в отдельности. Вместо этого, одна организация (NIC) обслуживает только часть дерева (домены верхнего уровня), а ответственность за определенные зоны передает другим организациям.

Зона (zone) это отдельно администрируемая часть дерева DNS. Например, домен второго уровня noao.edu это отдельная зона. Многие домены второго уровня поделены на меньшие зоны. Например, университет может поделить свою зону на подзоны по факультетам, а компания может поделить себя на зоны по принципу деления на филиалы или отделы.

С того момента, как выбрана организация или персона, которая несет ответственность за управление зоной, эта организация или персона должна организовать несколько серверов DNS (name servers) для этой зоны. Как только в зоне появляется новая система, администратор этой зоны помещает имя и IP адрес нового хоста в базу данных сервера DNS. В небольших университетах, например, один человек может делать это каждый раз при появлении новой системы, однако в больших университетах ответственность должна быть распределена (например, по департаментам), так как один человек не может осуществлять эту работу в целом.

Что произойдет, если сервер DNS не содержит необходимой информации? Он должен установить контакт с другим сервером DNS. (В этом заключается распределенная природа DNS.) Однако не каждый сервер DNS знает, как обратиться к другому серверу. Вместо этого каждый сервер DNS должен знать, как установить контакт с корневыми серверами DNS (root name servers). В апреле 1993 года существовало восемь корневых серверов, все первичные сервера должны знать IP адреса каждого корневого сервера. (Эти IP адреса находятся в конфигурационных файлах первичного сервера. Первичные сервера должны знать именно IP адреса корневых серверов, а не их DNS имена.) Корневой сервер, в свою очередь, знает имена и положения (IP адрес) каждого официального сервера DNS для всех доменов второго уровня. При этом возникает последовательный процесс: запрашивающий сервер должен установить контакт с корневым сервером. Корневой сервер сообщает запрашивающему серверу о необходимости обратиться к другому серверу и так далее.

DNS это одна из важнейших частей любого хоста, подключенного к Internet, эта система также широко используется в частных объединенных сетях. Основа организации - иерархическое дерево, которое формирует пространство имен DNS.

Приложения обращаются к разборщикам, чтобы конвертировать имя хоста в IP адрес и наоборот. Разборщики обращаются к локальным серверам DNS, а они, свою очередь, могут обратиться к одному из корневых серверов или к другим серверам, чтобы получить ответ на запрос.

Все DNS запросы и отклики имеют один и тот же формат. Эти сообщения содержат записи ресурсов (RR) вопросов и, возможно, ответов, RR полномочий и RR дополнительной информации. Мы рассмотрели множество примеров, в которых показана конфигурация разборщика и некоторые принципы организации DNS: указатели на имена доменов (чтобы уменьшить размер сообщений), кэширование, домен in-addr.arpa (поиск имени по заданному IP адресу) и возвращаемые дополнительные RR (для того чтобы запрашивающий не выдавал повторный запрос).

Протокол TCP

Протокол TCP (Transmission Control Protocol) предназначен для обеспечения надежных прямых соединений между парами процессов на хост-компьютерах, включенных в различные компьютерные коммуникационные сети, которые объединены в единую систему. Он находится на транспортном уровне между протоколами IP и собственно приложением.

Основные функции TCP

1. Базовая передача данных. Модуль TCP выполняет передачу непрерывных потоков данных в обоих направлениях. Этот поток TCP рассматривает как поток октетов. Он разделяет этот поток на части (дейтграммы) для отправки данных TCP вызывает модуль IP.
2. Обеспечение достоверности. Модуль TCP обеспечивает защиту от повреждения, потери, дублирования и нарушения очередности данных. Для этого все октеты нумеруются сквозным образом по возрастающей. Заголовок каждого сегмента содержит число октетов и порядковый номер первого октета. Затем каждый сегмент имеет значение контрольной суммы, по которой проверяется поврежденность данных (рис. 9.4).
3. Разделение каналов. Протокол TCP обеспечивает работу одновременно нескольких соединений. Поэтому в заголовке TCP-сегмента есть номера портов. Все распространенные сервисы Интернета имеют стандартные номера портов. Например, электронная почта – 25, FTP – 21 и т.д. Совокупность IP-адреса и номера порта называют ***сокетом***. Сокет уникальным образом идентифицирует прикладной процесс Интернета.
4. Управление соединениями. ***Соединение*** – это совокупность информации о состоянии потока данных, включающая сокеты, номера посланных, принятых и подтвержденных октетов, размеры окон.

Отметим, что каждая метка указывает здесь место для соответствующего бита.

Source Port (порт отправителя) 16 бит номер порта отправителя

Destination Port (порт получателя) 16 бит номер порта получателя

Sequence Number (номер очереди) 32 бита. Номер очереди для первого октета данных в данном сегменте (за исключением тех случаев, когда присутствует флаг синхронизации SYN). Если же флаг SYN присутствует, то номер очереди является инициализационным (ISN), а номер первого октета данных - ISN+1.

Acknowledgment Number (номер подтверждения) 32 бита. Если установлен контрольный бит ACK, то это поле содержит следующий номер очереди, который отправитель данной датаграммы желает получить в обратном направлении. Номера подтверждения посылаются постоянно, как только соединение будет установлено.

Data Offset (смещение данных) 4 бита. Количество 32-битных слов в TCP заголовке. Указывает на начало поля данных. TCP заголовок всегда кончается на 32-битной границе слова, даже если он содержит опции.

Reserved 6 бит. Это резервное поле, должно быть заполнено нулями.

Control Bits (контрольные биты) 6 бит. Биты этого поля слева направо

Window (окно) 16 бит. Количество октетов данных, начиная с октета, чей номер указан в поле подтверждения. Количество октетов, получения которых ждет отправитель настоящего сегмента.

Checksum (контрольная сумма) 16 бит. Поле контрольной суммы - это 16-битное дополнение суммы всех 16- битных слов заголовка и текста. Если сегмент содержит в заголовке и тексте нечетное количество октетов, подлежащих учету в контрольной сумме, последний октет будет дополнен нулями справа с тем, чтобы образовать для предоставления контрольной сумме 16-битное слово. Возникший при таком выравнивании октет не передается вместе с сегментом по сети. Перед вычислением контрольной суммы поле этой суммы заполняется нулями.

Urgent Pointer (срочный указатель) 16 бит. Это поле сообщает текущее значение срочного указателя. Последний является положительной величиной - смещением относительно номера очереди данного сегмента. Срочный указатель сообщает номер очереди для октета, следующего за срочными данными. Это поле интерпретируется только в том случае, когда в сегменте выставлен контрольный бит URG.

Options (опции) длина переменная. Опции могут располагаться в конце TCP заголовка, а их длина кратна 8 бит. Все опции учитываются при расчете контрольной суммы.

Padding (выравнивание) длина переменная Выравнивание TCP заголовка осуществляется с тем, чтобы убедиться в том, что TCP заголовок заканчивается, а поле данных сегмента начинается на 32-битной границе. Выравнивание выполняется нулями.

Процедура установления соединения

Взаимодействие партнеров с использованием протокола TCP строится в три этапа:

- установление логического соединения;
- обмен данными;
- закрытие соединения.

Этап установления соединения реализуется как **"трехшаговое рукопожатие"** (three-way handshake). 1) TCP-модуль A, играя роль клиента, посылает TCP-модулю B пакет с установленным флагом SYN и начальным значением номера в последовательности данных. 2) TCP-модуль B, будучи готов со своей стороны установить соединение, отвечает TCP-пакетом, подтверждающим правильный прием запроса и информирующим о готовности установить соединение 3) На третьем шаге TCP-модуль A подтверждает правильность приема TCP-пакета от B.

Протокол TCP взаимодействует с одной стороны с пользователем или прикладной программой, а с другой - с протоколом более низкого уровня, таким как протокол Internet.

Интерфейс между прикладным процессом и протоколом TCP мы поясняем с приемлемой детализацией. Этот интерфейс состоит из набора вызовов, которые похожи на вызовы операционной системы, предоставляемые прикладному процессу для управления файлами. Например, в этом случае имеются вызовы для открытия и закрытия соединений, для отправки и получения данных на установленных соединениях. Предполагается также, что протокол TCP сможет асинхронно взаимодействовать с прикладными программами. Хотя разработчикам TCP протокола и предоставлена значительная свобода в создании интерфейсов, которые соответствуют свойствам конкретной операционной системы, все же от любой

приемлемой реализации требуются некие обязательные минимальные функции интерфейса между протоколом TCP и пользователем.

Протокол UDP

Протокол дэйтаграмм пользователя UDP (User Datagram Protocol) является протоколом транспортного уровня и базируется на возможностях, предоставляемых межсетевым протоколом IP. Основная задача UDP - обеспечение "быстрой" передачи данных в сети.

Его основные характеристики перечислены ниже:

- реализует взаимодействие в режиме без установления логического (виртуального) соединения;
- организует поблочный (дэйтаграммный, пакетный) тип передачи данных;
- для идентификации партнеров по взаимодействию на транспортном уровне использует 16-битовые "номера портов";
- не гарантирует надежной передачи данных (возможна как потеря UDP-пакетов, так и их дублирование);
- не имеет средств уведомления источника UDP-пакета о правильности/ошибочности в его приеме адресатом;
- не обеспечивает правильный порядок доставки UDP-пакетов от источника к приемнику;
- может гарантировать целостность данных в UDP-пакете за счет использования контрольной суммы;
- очень простой (особенно, по сравнению с протоколом TCP).

Следует отметить, что, по сути дела, протокол транспортного уровня UDP играет роль интерфейса для прикладных программ к средствам протокола межсетевого уровня IP. На рис. 9.5 приведен формат заголовка UDP-пакета.

0 15	31
Порт источника	Порт приемника
Длина	Контрольная сумма
Октеты данных	

Рис. 9.5 Формат заголовка для дейтграммы протокола UDP

Порт источника и порт приемника - 16-битовые поля, содержащие номера портов, соответственно, источника и адресата UDP-пакета.

Длина - 16-битовое поле, содержащее длину (в байтах) всего UDP-пакета, включая заголовок и данные.

Контрольная сумма -16-битовое поле, содержащее Internet-контрольную сумму, подсчитанную для UDP-заголовка.

Протоколы прикладного уровня

1.Протокол передачи файлов (File Transfer Protocol - FTP) представляет собой алгоритм передачи файлов из одной компьютерной системы в другую, использует TCP и дополнительно обеспечивает пользовательскую аутентификацию с помощью ID пользователя и пароля. FTP представляет собой интерактивный сервис, который служит для соединения с удаленным компьютером, копирование файлов и закрытие линии связи после завершения передачи. Для исполнения этих операций используются следующие базовые команды:

- Open (Открыть) – подключиться к удаленному компьютеру;
- Get (Получить) – получить файл из компьютера после установления соединения;
- Bye (Закрыть) – отключить соединение и завершить программу FTP.

FTP – это самый распространенный протокол передачи файла между компьютерами. Он позволяет передавать как текстовые, так и двоичные файлы.

2.Сервис Telnet – один из первых протоколов Интернета. Его можно использовать как удаленный терминал хоста Интернета. Во время связи с хост-компьютером Интернета компьютер работает так, как будто его клавиатура и дисплей подключены непосредственно к удаленному компьютеру. Пользователь может запускать программы на компьютере, находящемся на противоположной стороне земного шара, с той же легкостью, словно он сидит за ним.

3. Упрощенный протокол пересылки электронной почты. Данный протокол (Simple Mail Transfer Protocol - SMTP) представляет собой основанный на TCP протокол клиент/сервер, который используется для обмена электронной почтой. Сервер помещает входящие сообщения в почтовые ящики, а клиенты извлекают их, используя либо почтовый протокол (Post Office Protocol - POP), либо протокол интерактивного доступа к электронной почте (Internet Message Access Protocol – IMAP).