Лекция

Сетевые протоколы и службы

Введение

Сетевые протоколы представляют собой наборы логических правил, согласно которым работает сеть.

Сетевые *службы,* например, службы разрешения имен или размещения адресов, выполняют специфические функции.

Существует множество типов сетевых протоколов, работающих на разных уровнях модели OSI. В источниках по сетям персональных компьютеров термином "протокол" часто обозначаются протоколы *сетевого* и *транспортного* уровней, т,е, уровней 3 и 4 модели OSI.

В этой лекции рассматриваются следующие протоколы сетевого и транспортного уровней.

1. NetBEUI;
2. IPX/SPX;
3. TCP/IP

Эти три стандартных стека протоколов поддерживаются большинством операционных систем персональных компьютеров. Каждый из них имеет определенные достоинства и недостатки, зависящие от условий работы локальной сети

В этой лекции рассматриваются также сетевые службы

1. DNS (Domain Name System);
2. WINS (Windows Internet Name Service);
3. DHCP (Dynamic Host Configuration Protocol).

Все они работают с протоколом TCP/IP, расширяя его функциональные возможности.

1. Протокол NetBIOS/NetBEUI

Служба NetBIOS (Network Basic Input/Output System) была разработана компанией IBM и применялась компанией Microsoft в ранних реализациях локальных сетей. В то время термин NetBIOS означал как API (Application Programming Interface), так и стек протоколов транспортного и сетевого уровней.

Стек протоколов — это группа из двух или более протоколов, которые работают совместно, примем каждый на своем уровне модели OSI.

В дальнейшем компоненты NetBIOS были разделены на, собственно, NetBIOS (API) и протокол NetBEUI (NetBIOS Extended User Interface), включающий протоколы сетевого и транспортного уровней.

NetBIOS не определяет кадры или формат передаваемых по сети данных; NetBEUI - определяет.

1.1. NetBIOS: API

NetBIOS работает с протоколами NetBEUI, IPX/SPX или TCP/IP. С помощью NetBIOS приложения могут взаимодействовать с распространенными интерфейсами программирования, разделяя информацию с приложениями, использующими другие протоколы более низких уровней.

Работая на более высоких уровнях (на уровне приложений модели DoD и на сеансовом уровне модели OSI), NetBIOS поддерживает два режима коммуникации: сеансовый и дейтаграммиый.

В сеансовом режиме NetBIOS позволяет устанавливать сеанс соединения между компьютерами с обнаружением ошибок и восстановлением данных.

В дейтаграммном режиме индивидуальные сообщения посылаются отдельно. Соединение при этом не устанавливается, поэтому обнаружение и исправление ошибок возлагается на приложение.

NetBIOS содержит также службу имен (т.е. имен NetBIOS), с помощью которых в сети можно идентифицировать компьютеры и приложения.

1.2. NetBEUI

NetBEUI — самый простой из трех рассматриваемых стеков протоколов. Простота делает его самым быстродействующим из этой тройки, однако эта же простота существенно ограничивает его функциональные возможности.

В протоколе NetBEUI нет средств логической адресации на сетевом уровне, поэтому его нельзя маршрутизировать из одной сети или подсети в другую.

Однако внутри одной локальной сети он работает неплохо.

Его легко конфигурировать.

NetBEUI можно использовать с маршрутизируемым протоколом, например с TCP/IP. При этом сохраняется преимущество высокого быстродействия NetBEUI внутри локальной сети и появляется возможность коммуникации с компьютерами за пределами локальной сети с помощью TCP/IP.

2. Протокол IPX/SPX

Протоколы IPX и SPX, работая совместно, обеспечивают маршрутизацию сетевых сообщений. Компания Novell разработала протокол IPX/SPX для серверов и клиентов NetWare, однако его можно использовать и в других операционных системах, например в Windows. Протокол IPX/SPX разрабатывался на основе протоколов XNS (Xerox Network System).

IPX/SPX, или NWLink, необходим для соединения клиентов Microsoft с серверами NetWare 4.x или более ранних версий. Некоторые серверы NetWare 5.x могут работать только с протоколом TCP/IP.

Производительность и легкость конфигурирования IPX/SPX лучше, чем у TCP/IP.

Иногда IPX/SPX используют во внутренней локальной сети специально для повышения безопасности: внешние компьютеры, получающие доступ к локальной сети посредством Internet, работают только по протоколу TCP/IP, поэтому они не могут получить доступ к компьютерам локальной сети, работающим с IPX/SPX.

Компания Microsoft поставляет совместимый с IPX/SPX стек протоколов NWLink, реализованный во всех современных операционных системах Windows. Однако по умолчанию NWLink не устанавливается.

Протокол IPX работает на сетевом уровне модели OSI, он относится к протоколам, работающим в режиме без установки соединения.

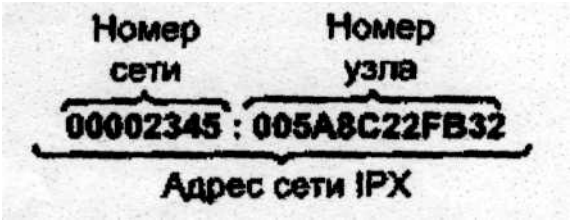
Протокол SPX работает на транспортном уровне модели OSI, он обеспечивает распознавание и сборку пакетов, а также другие службы режима с установкой соединения.

2.1. Протокол сетевого уровня IPX

Сетевой уровень модели OSI осуществляет логическую адресацию и маршрутизацию сообщений, т.е. обеспечивает передачу сообщений по нужному адресу. Это и есть главная задача протокола IPX.

Маршрутизируемый протокол должен иметь возможность идентифицировать сеть, в которой расположен принимающий компьютер. Для идентификации сети (подсети) в IPX используется шестнадца-теричный ***номер сети.*** Типичный номер сети в IPX выглядит так: 805609a0. Этот номер присваивается сети администратором.

Адрес IPX состоит из двух частей: номера сети и *номера узла* (рис.). Номер узла идентифицирует конкретное устройство на основе MAC-адреса сетевого адаптера.



В сетях, использующих одновременно TCP/IP и IPX/SPX, номер сети часто получают из IP-адреса путем простого преобразования десятичного числа (IP-адреса) в шестнадцатеричное. Например, десятичный IP-адрес 214.12.1.42 преобразуется в шестнадцатеричный D6C12A.

Протокол извещения об услугах

Для извещения клиентов об услугах различных сетевых служб (например, файловых серверов) в IPX используется протокол SAP (Service Advertising Protocol). Каждой сетевой службе присваивается идентификатор SAP (число, которое называется SAP ID). Широковещательные сообщения SAP передаются каждые 60 с. Маршрутизаторы и серверы поддерживают таблицы, отображающие SAP ID на службы, и динамически обновляют эти таблицы с каждым широковещательным сообщением SAP. Постоянное обновление таблиц значительно загружает сети.

2.2. Протокол транспортного уровня SPX

Протокол SPX работает на один уровень выше, чем IPX, т.е. на транспортном уровне. В отличие от IPX, работающего в режиме без установки соединения, SPX работает в режиме с установкой соединения. Это делает SPX более надежным, что весьма уместно на транспортном уровне, ответственном за подтверждения, обнаружение ошибок и другие аспекты обеспечения надежности.

Протокол IPX доставляет пакеты по назначению, а SPX следит за тем, чтобы пакеты прибыли полностью и в целостном состоянии. Протокол SPX поддерживает нумерацию пакетов и отслеживает количество переданных пакетов. Он гарантирует доставку пакетов путем контроля принятых данных.

3. Протокол TCP/IP

Стек протоколов TCP/IP — фундамент Internet. Он неуклонно превращается в наиболее распространенный протокол сетевого и транспортного уровней для сетей всех размеров и конфигураций.

Пакет протоколов TCP/IP

Протокол TCP/IP — это не только стек протоколов сетевого и транспортного уровней, но и полный набор протоколов, работающих на многих уровнях сетевой модели. Пакет протоколов TCP/IP включает также дополнительные компоненты, необязательные в процессах сетевой коммуникации, например, утилиты прикладного уровня, также входящие в состав пакета TCP/IP.

Многие протоколы, входящие в пакет, предназначены для сбора информации или для устранения неполадок.

3.1. Протокол сетевого уровня IP

На сетевом уровне выполняются задачи маршрутизации. В протоколах TCP/IP маршрутизация поддерживается путем применения IP-адресов, идентифицирующих сетевые устройства. Каждый компьютер, принтер (подключенный к сети), маршрутизатор или любое другое сетевое устройство имеет уникальный IP-адрес.

IP-адресация

Каждый IP-адрес состоит из двух частей. Вместе они идентифицируют сеть, в которой расположено устройство, и само устройство. Один раздел IP-адреса представляет *сеть,* а другой — *хост* (отдельный компьютер).

Разделы называются *октетами.*

В традиционной схеме IP части IP-адреса, представляющие сеть и хост, могут иметь разные размеры для различных *классов* IP-адреса.

Классы IP-адресов

Подобно любой обрабатываемой компьютером информации, IP-адрес состоит из двоичных цифр, т.е. битов. Обычно IP-адреса записываются в *десятичном* формате.

Длина октета равна восьми битам, т.е. октет — это последовательность из восьми нулей или единиц. Иногда четыре октета обозначаются как w.x.y.z. В такой записи первый октет (правый) называется z-октетом, следующий — y-октетом и т.д.

IP-адрес состоит из четырех октетов по восемь битов каждый, всего — 32 бита. Это значит, что максимальное количество различимых IP-адресов равно 2^{32}, или 4 294 967 296. Типичный IP-адрес выглядит так: 192.168.1.12.

Здесь рассматривается протокол IP версии IPv4, которая используется в Internet в настоящее время. Однако сейчас разрабатывается новый стандарт— IPv6. В новом стандарте IP-адрес будет 128-битовым, т.е. максимальное количество различимых адресов будет равно $\mathbf{2^{128}}$.

Одна часть IP-адреса идентифицирует сеть, а другая — компьютер (хост). Где же расположены эти части в IP-адресе? Ответ на этот вопрос неоднозначен. Традиционно это зависит от класса сети, являющегося одновременно классом IP-адреса.

IP-адреса предоставляет организация IANA (Internet Assigned Numbers Authority).

Размер выделяемого блока адресов завит от размера локальной сети. Большим компаниям нужны были большие блоки адресов, а маленьким — маленькие. *Классы адресов* назначались на основе размера локальной сети, т.е. количества хостов. В табл. показаны традиционные классы IP-адресов

Таблица . Классы IP-адресов

Класс адреса	Количество сетей	Количество хостов на сеть
A	126*	16 777 216
B	16 384	65 535
C	2 097 152	254
D (широковещательный)	–	–

В классе A адреса 127.x x.x зарезервированы для обратных адресов, *используемых в проверочных и диагностических целях.*

В блоке 127.x.x.x содержится 24 млн IP-адресов. В первые годы существования Internet это не было проблемой, потому что свободных IP-адресов было значительно больше, чем подключенных к Internet компьютеров.

Как видно из табл., доступны только 126 адресов класса A. К настоящему времени все они уже заняты. Они присвоены наиболее крупным корпорациям и учебным центрам, таким, как IBM, Hewlett Packard, Xerox, MIT (Massachusetts Institute of Technology), Columbia University, Digital Equipment Corporation, General Electric и Apple. В каждой из этих сетей индивидуальные номера можно присвоить 16 млн хостов.

В приведенной схеме адресации можно создать 2 млн сетей класса C, в каждой из которых может быть не более 254 адресов хостов. Адреса класса C выделяются провайдерам Internet.

Адреса класса B занимают промежуточное положение. Они присваивались главным образом большим компаниям, размер которых в то время был недостаточен для класса A. Компании Microsoft выделены адреса класса B.

Адреса класса D предназначены для *широковещательных сообщений,* т.е. для передачи одного сообщения одновременно многим получателям. Адрес класса D присваивается специальной группе компьютеров, в этом случае пакеты обрабатываются и распределяются широковещательными протоколами.

Деление IP-адресов на классы в зависимости от размера сети называется *классовой адресацией.*

Классовая адресация

. В табл. показано, как определить класс адреса, исходя из значения первого октета.

Таблица . Определение класса адреса

Класс IP-адреса	Старшие биты	Диапазон адресов первого октета	Количество битов в адресе сети
Класс A	0	0-127*	7
Класс B	10	128-191	14
Класс C	110	192-223	21
Класс D	1110	224-239	28

** В классе A адреса 127x.x.x зарезервированы для* обратных адресов, *используемых в проверочных и диагностических целях.*

Класс адреса идентифицируется *битами высших разрядов или* несколькими первыми битами самого левого октета (называемого w-октетом).

Как видно из табл., адрес принадлежит классу A, если первый бит равен O, классу B — если первые два бита равны 10, классу C — если первые три бита равны 110, классу D — если первые четыре бита равны 1110.

Рассмотрим такой IP-адрес: 11001111.00101100.010100001.11100111. Первые три бита равны 110, следовательно, адрес принадлежит классу C. Преобразуем двоичные числа этого адреса в десятичные. В десятичном формате адрес имеет вид 207.44.81.231.

Первый октет (207) попадает в диапазон 192-223, приведенный в табл. для класса C. Первый октет адресов класса C попадает в диапазон 192-223 потому, что первые три бита равны 110. Это значит, что класс IP-адреса можно определить по значению первого октета.

В предыдущем примере октеты 207.44.81 определяют сеть (подсеть), в которой расположен компьютер. Для всех компьютеров этой подсети первые три октета их IP-адресов равны 207.44.81. Число 231 определяет конкретный хост. Ни для какого другого устройства этой подсети последний октет IP-адреса не может быть равным 231. На рис. показано, как IP-адрес делится на номер сети и номер хоста.

Рис В адресе класса С первые три октета определяют номер сети

Как определить максимальное количество сетей и компьютеров, которые можно реализовать в каждом классе? Традиционно в адресе класса А первый октет служит адресом сети, а три остальных — адресом хоста. Первый (старший) бит первого октета в адресе класса А используется для идентификации класса, поэтому для идентификации сети остается только семь битов.

В адресе класса В для идентификации сети используются два первых октета, а для идентификации хоста — два вторых. Первые два октета содержат 16 бит, однако первые два из них используются для идентификации класса. Таким образом, для идентификации номера сети остается 14 бит.

В классе С первые три октета (24 бита) используются для идентификации сети, а последний — для идентификации хостов. Три первых бита идентифицируют класс, поэтому, чтобы получить количество битов, выделенных для идентификации сети, нужно из 24 вычесть 3. Другими словами для идентификации номера сети используется 21 бит.

В сети класса А для идентификации номера сети выделено 7 бит. Если все 7 бит "включены" (равны единице), то получается самое большое число, которому может быть равен номер сети, — 1111111. В десятичном формате это число равно 127. Учитывая, что адрес 0.0.0.0 зарезервирован для представления всех IP-адресов и что номер 127 занят для обратных адресов, делаем вывод, что в классе А может существовать 126 различимых сетей.

Этот результат можно получить проще, если возвести число 2 в степень *x,* где *x* — количество битов, выделенных для номера сети. Так как $2^7 = 128$, значит, в семи битах можно записать 128 различимых чисел. Но, поскольку два адреса уже заняты (адрес 0.0.0.0 и обратный адрес), мы опять получаем, что максимальное количество сетей класса А равно 126.

Аналогично вычисляется максимальное количество сетей классов В и С.

- **Класс В — 14** бит: $\mathbf{2^{14} = 16\ 384}$.
- **Класс С - 21** бит: $\mathbf{2^{21} = 2\ 097\ 252}$.

Автоматическое выделение адресов

Для коммуникации с помощью протокола TCP/IP компьютер (или другое сетевое устройство) должен иметь уникальный IP-адрес, т.е. *логический* адрес, обрабатываемый на сетевом уровне.

Присвоить IP-адрес компьютеру можно двумя способами.

- Вручную. Для этого адрес необходимо ввести при конфигурировании параметров TCP/IP.
- Автоматически. Для этого в сети должен быть компьютер, конфигурированный как сервер DHCP. Он присваивает другим компьютерам IP-адреса из списка допустимых адресов. Если компьютер не контактирует с сервером DHCP, то он может сам автоматически присвоить себе адрес с помощью процедуры автоматической IP-адресации API PA (Automatic Private IP Addressing), входящей в состав операционной системы.

Служба DHCP

Протокол DHCP (Dynamic Host Configuration Protocol — протокол динамической конфигурации хоста) выполняется на компьютере, конфигурированном как сервер DHCP. Сервер DHCP автоматически выделяет IP-адреса компьютерам, конфигурированным как клиенты DHCP (рис.).

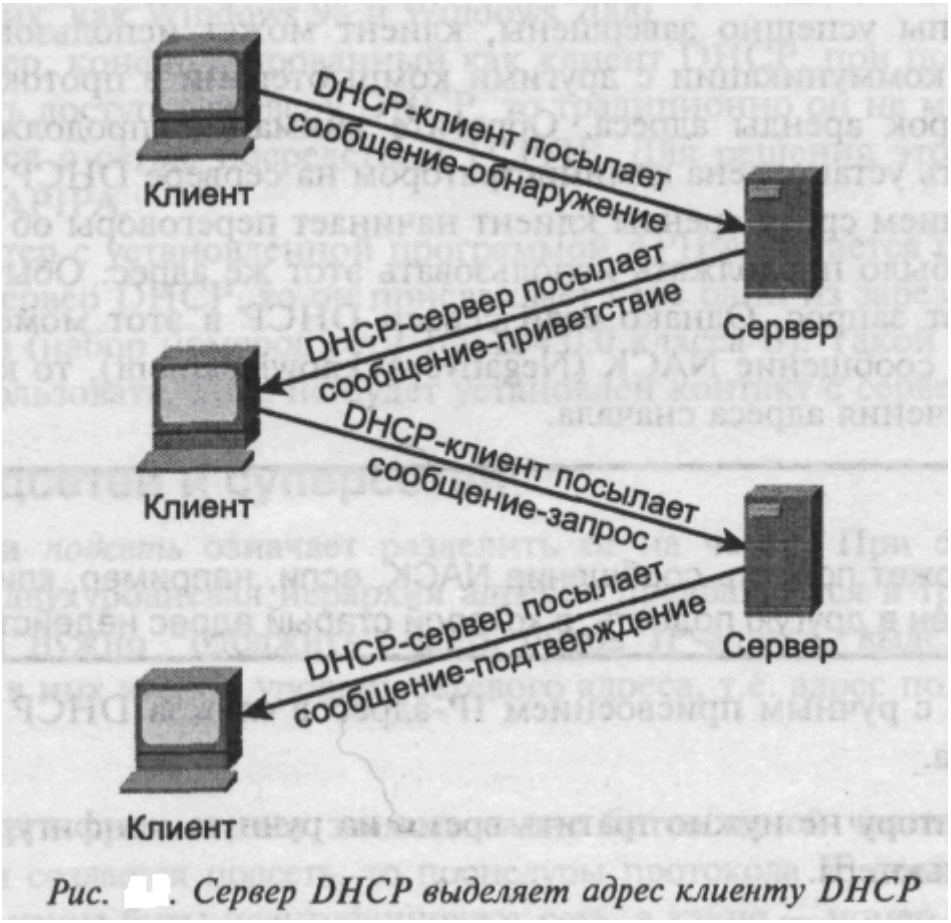


Рис. . Сервер DHCP выделяет адрес клиенту DHCP

Выделение IP-адреса сервером DHCP клиенту DHCP выполняется в несколько этапов.

1 . Компьютер, у которого свойства TCP/IP конфигурированы на получение IPадреса посредством DHCP, подключается к сети. При этом компьютер посылает в сеть (или подсеть) широковещательное сообщение, называемое *открытием DHCP.* (Широковещательные сообщения посылаются всем компьютерам с помощью специального широковещательного адреса.)
2. Если в сети присутствует сервер DHCP, то он получает широковещательное сообщение клиента и посылает в ответ сообщение, называемое *предложением DHCP.* В этом сообщении клиенту предлагается IP-адрес из хранящегося на сервере списка допустимых адресов. Предложенный адрес временно резервируется, пока сервер не получит ответ от клиента. Предложение DHCP также передается по широковещательному адресу, потому что не имеет своего IP-адреса, по которому можно было бы передать однонаправленное сообщение.
3. Если в сети есть несколько серверов DHCP, то клиент может получить несколько предложений. Когда поступает первое предложение, клиент посылает сообщение, называемое *запросом DHCP.* Это сообщение означает, что клиент одобрил первое полученное предложение. Запрос DHCP тоже является широковещательным сообщением, поэтому его получают все серверы DHCP. Те из них, чьи предложения DHCP не одобрены, узнают об этом из запроса DHCP и возвращают зарезервированные адреса обратно в список допустимых адресов.
4. Последний этап переговоров — *подтверждение DHCP* (или сообщение *ACK,* от *acknowledgment — подтвреждениё).* Сервер DHCP, предложение которого одобрено, получает посланный клиентом запрос DHCP. После этого сервер посылает клиенту подтверждение DHCP и присваивает ему IP-адрес, действительный в течение предопределенного периода времени — срока аренды адреса. Сервер может послать клиенту также дополнительную информацию по конфигурированию TCP/IP, например IP-адреса серверов DNS и WINS (рассматриваются далее в лекции).

По сравнению с ручным присвоением IP-адресов служба DHCP имеет существенные преимущества.

- Администратору не нужно тратить время на ручную конфигурацию свойств каждого компьютера.
- Работа сети становится более упорядоченной, потому что все необходимые адреса постоянно присвоены и администратору не нужно следить за тем, какие адреса заняты или свободны и не истекает ли у кого-нибудь из них срок аренды.

Услугами DHCP можно воспользоваться и в том случае, если компьютеры должны иметь один и тот же IP-адрес (часто это нужно для серверов). Такие компьютеры следует конфигурировать на использование

зарезервированного адреса. Клиенту, заказавшему резервирование, сервер DHCP всегда присваивает один и тот же адрес. Резервирование осуществляется на основе MAC-адреса клиента (физического адреса).

Протокол DHCP не ориентирован на определенную операционную систему. Его можно использовать с Windows, UNIX, NetWare и другими операционными системами. Однако конкретные реализации поставщиков DHCP могут несколько отличаться друг от друга. Например, серверы DHCP для Windows интегрированы с Active Directory.

Создание подсетей и суперсетей

Создать в сети *подсеть* означает разделить ее на части. При создании подстети описанная выше двухуровневая иерархия адресов превращается в трехуровневую. Для создания подсети нужно "одолжить" часть битов IP-адреса, выделенных под адрес хоста, и записать в них второй уровень сетевого адреса, т.е. адрес подсети.

Маска подсети

Создание подсети предполагает заимствование битов одной части адреса для другой части адреса. Если создается подсеть, то процедуры протокола IP должны иметь возможность определить, какие биты идентифицируют сеть, а какие — номер хоста. Это делается с помощью *маски подсети* — 32-битового числа, введенного сетевым администратором при конфигурировании свойств TCP/IP. В маске подсети биты, определяющие номер сети, установлены в 1, а биты, определяющие номер хоста — в 0. Часть адреса, отведенная для номера сети, называется "замаскированной" битами, содержащими единицы.

По умолчанию в классе A для номера сети отведены биты первого октета, в классе B — биты двух первых октетов, а в классе C — биты трех первых октетов. В этом случае *маска подсети по умолчанию* выглядит как показано в табл.

Таблица . . Маски подсети по умолчанию

Класс адреса	Двоичная маска подсети
Класс A	11111111 00000000 00000000 00000000
Класс B	11111111 11111111 00000000 00000000
Класс C	11111111 11111111 11111111 00000000

Маска подсети, приведенная в табл., применима к сетям, *не имеющим подсетей.*

Однако что получится, если разделить сеть на подсети? Допустим, сети присвоен адрес класса B, например 181.25.0.0. Как известно, сеть класса B может содержать до 65 535 хостов. Но в сети с таким огромным количеством компьютером управление сообщениями станет невозможным.

Чтобы решить эту проблему, разделим сеть на шесть подсетей. Для этого придется позаимствовать несколько битов из адреса хоста для адреса подсети. Нужно также создать правильную маску подсети, с помощью которой процедуры IP смогут определить адрес подсети.

Вычисление маски подсети

Маска подсети, отличная от маски по умолчанию, называется маской *переменной длины* или *пользовательской* маской. Для вычисления маски подсети в нашем примере с шестью отдельными подсетями сначала нужно определить, сколько битов должно быть позаимствовано из числа битов, отведенных под номера хостов.

Двоичная система счисления имеет основание 2, поэтому максимальное количество создаваемых подсетей равно целой степени двойки. Мы должны найти, какая степень двойки (за вычетом 2) равна 6 или ближайшему большему числу.

Возведем двойку во вторую степень: $2^2 = 4$. Как видите, это меньше шести. Поэтому возведем двойку в следующую, третью степень: $2^3 = 8$. Теперь необходимо вычесть 2, чтобы удовлетворить правило, согласно которому номер сети не может состоять только из нулей или только из единиц. Вычтя 2, мы получили 6 оставшихся допустимых номеров сети. Следовательно, для номера сети нужно позаимствовать три бита из части адреса, отведенной под номера хостов. Для этого мы должны в маске подсети по умолчанию превратить три первых нуля в единицы. Теперь маска подсети выглядит следующим образом: 11111111.11111111.11100000.00000000. В десятичном формате она выглядит так: 255.255.224.0.

Сколько компьютеров можно разместить в каждой подсети? Посмотрим на оставшиеся нули в маске подсети. Осталось 13 нулей. Другими словами, в этих битах можно разместить $2^{13} = 8\ 192$ адресов. Однако номер хоста не может состоять из одних нулей или единиц, поэтому из полученного количества два номера недопустимы. Значит, в каждой из шести подсетей нашего примера можно разместить 8 190 компьютеров.

В табл. приведено максимальное количество подсетей и хостов для каждой маски подсети.

Таблица. Допустимое количество подсетей и хостов

Десятичное значение первого изменяемого октета	Количество подсетей	Количество хостов класса A	Количество хостов класса B	Количество хостов класс C
.192	2	4 194 302	16382	62
.224	6	2 097 150	8190	30
.240	14	1 048 574	4094	14
.248	30	524 286	2046	6
.252	62	262 142	1022	2
.254	126	131 070	510	—
.255	254	65 534	254	—

Использование логического И

Когда сообщение передается в сеть с помощью TCP/IP, процедуры протокола IP должны определить, расположены ли передающий и принимающий компьютеры в одной и той же подсети. Если да, то сообщение передается как широковещательное. Если нет, то сообщение передается по адресу *шлюза по умолчанию,* т.е. на интерфейс маршрутизатора (на выходе из сети функции шлюза выполняет маршрутизатор).

Для определения принадлежности передающего и принимающего компьютеров используется поразрядное применение логической операции И к IP-адресу и маске подсети каждого компьютера. Для каждого разряда операция И дает следующие результаты:

- 1 И 1 = 1;
- 1 И 0 = 0;
- 0 И 0 = 0.

Приведем пример. Допустим, передающий компьютер имеет IP-адрес 192.168.1.1 и маску подсети 255.255.255.0, а принимающий компьютер— IP-адрес 192.168.3.1 и маску подсети 255.255.255.0.

Сначала пересчитаем адрес и маску подсети передающего компьютера:

```
192.168.1.1     =  11000000.10101000.00000001.00000001
255.255.255.0   =  11111111.11111111.11111111.00000000
Результат       =  11000000.10101000.00000001.00000000
```

Затем выполним эти же вычисления для принимающего компьютера:

```
192.168.3.1     =  11000000.10101000.00000011.00000001
255.255.255.0   =  11111111.11111111.11111111.00000000
Результат       =  11000000.10101000.00000011.00000000
```

Результаты отличаются друг от друга, поэтому можно сделать вывод, что передающий и принимающий компьютеры находятся в разных подсетях. Следовательно, сообщение передается на маршрутизатор (шлюз по умолчанию) и направляется в нужную подсеть.

Если эти результаты совпадут, то принимающий компьютер находится в этой же подсети и сообщение будет передано с использованием широковещательного протокола ARP, который рассматривается далее в лекции.

Преимущества создания подсетей

Разделение больших сетей на две или более подсетей предоставляет ряд преимуществ.

- Уменьшается загрузка сети. Подсети соединены друг с другом с помощью маршрутизаторов, которые по умолчанию не пропускают широковещательные сообщения. Это приводит к существенной экономии пропускной способности сетевых каналов. Компьютеры, расположенные в разных местах, организуются в подсети, поэтому ими легче управлять;
- Повышается безопасность, поскольку различные части сети изолируются друг от друга и передаваемые между ними сообщения можно фильтровать;
- Более эффективно используются выделенные адреса, так как уменьшается количество утраченных адресов.

3.2. Протоколы транспортного уровня: TCP и UDP

Транспортный уровень отвечает за обеспечение надежной коммуникации одного компьютера с другим. Эта задача реализуется такими механизмами, как, например, подтверждение получения

данных принимающим компьютером без потерь или повреждений.

Протоколы транспортного уровня предназначены также для идентификации сообщений, прибывающих на один и тот же компьютер. Различные приложения, установленные на одном компьютере, могут передавать и принимать сообщения одновременно. Чтобы разделять эти сообщения, в протоколах транспортного уровня используются *порты*.

Пакет TCP/IP содержит не один, а два протокола транспортного уровня

- **TCP (Transmission Control Protocol — протокол управления передачей)** Протокол, ориентированный на установку соединения;
- **UDP (User Datagram Protocol — протокол пользовательских дейтаграмм)** Протокол, *не* ориентированный на установку соединения.

Протокол TCP используется, когда более важна надежность передачи данных, а UDP — когда более важной характеристикой является производительность (скорость) коммуникации.

Протокол транспортного уровня TCP

Прежде чем начать передавать данные, TCP устанавливает между двумя сообщающимися компьютерами сеанс соединения. Для этого используются сообщения уведомления и ответа. Затем выполняются процедуры обнаружения и исправления ошибок и данные разбиваются на пакеты.

В каждый пакет добавляется информация о нумерации пакетов, чтобы на принимающем конце их можно было собрать в правильной последовательности. Нумерация пакетов позволяет принимающему компьютеру обнаружить недостающие пакеты. Благодаря этим процедурам протокол TCP надежнее, чем UDP, однако выполнение дополнительных операции существенно снижает производительность.

Протокол транспортного уровня UDP

Протокол U DP не ориентирован на установление соединения. Не выполняется также нумерация пакетов данных, поэтому он более пригоден для передачи небольших сообщений, которые можно разместить в одном пакете. Протокол UDP не отслеживает также, что было передано и что получено. Однако в UDP выполняется проверка контрольной суммы, чем гарантируется правильность данных, поступивших на принимающий компьютер. Как и в TCP, в UDP, чтобы различить сообщения для или от разных приложений одного и того же компьютера, используются номера портов.

Протокол UDP не занимается нумерацией пакетов или обнаружением ошибок, поэтому его производительность высокая. Заголовок пакета UDP проще заголовка TCP. Протокол UDP используется в протоколах RIP (Routing Information Protocol), TFTP (Trivial File Transfer Protocol) и др.

Дейтаграмма - это самодостаточный независимый модуль данных, содержащий информацию, достаточную для маршрутизации этих данных от передающего к принимающему компьютеру без предварительного или текущего обмена между передающим и принимающим компьютерами и сетью.

В некоторых случаях термины "дейтаграмма" и "пакет" взаимозаменяемы. Пакет представляет собой модуль данных, являющийся частью группы нумерованных модулей, на которые разбито сообщение. Пакеты могут проходить по сети к адресату различными маршрутами. В принимающем компьютере сообщение собирается из поступивших пакетов. В то же время термин **"дейтаграмма" означает более простые, ненумерованные модули данных, передаваемые с помощью UDP.**

Порты и сокеты

Для идентификации передающего и принимающего компьютеров в TCP/IP используются логические адреса, состоящие из двух частей, т.е. IP-адреса. Но что получается, если два сетевых приложения, выполняющихся на одном компьютере, посылают или принимают сообщения одновременно? Например, если одно поступающее сообщение предназначено для программы электронной почты, а другое — для Web-броузера? Протоколы должны различать эти сообщения. Для этого используются порты TCP и UDP.

Номера портов

IP-адрес принимающего компьютера состоит из двух частей: адреса сети и адреса хоста. Номера портов можно представить себе как специфическую маршрутную информацию внутри адреса. Это дополнение к IP-адресу.

Порт — это *точка логического соединения*. В транспортных протоколах TCP и UDP порты используются для идентификации конкретного приложения, передающего или принимающего сообщение.

Наиболее распространенные приложения Internet используют предопределенные номера портов. Стандартизация облегчает процесс коммуникации. В табл. приведен список предопределенных номеров портов, используемых наиболее распространенными приложениями.

Таблица . Предопределенные порты TCP и UDP

Предопределенный порт	Протокол	Приложение
80	TCP	HTTP
21	TCP/UDP	FTP
23	TCP/UDP	Telnet
25	TCP/UDP	SMTP
110	TCP/UDP	POP3
119	TCP/UDP	NNTP
137	TCP/UDP	Служба имен NetBIOS
161	TCP/UDP	SNMP
194	TCP/UDP	IRC
389	TCP/UDP	LDAP
396	TCP/UDP	NetWare в IP
458	TCP/UDP	QuickTime компании Apple
500	TCP/UDP	ISAKMP

Существует 65 536 допустимых номеров портов. Порты от 0 до 1 024 зарезервированы для предопределенных служб, подобных приведенным в табл.

Понятие сокета

Сокет определяется как "конечная точка соединения". Чтобы коммуникация состоялась, должен быть создан сокет.

Разные типы сокетов используют разные методы адресации. Наибольшее распространение получила идентификация сокета по IP-адресу и номеру порта. В терминологии UNIX это называется адресацией AF_INET. В другом методе адресации — AF_UNIX — для идентификации сокетов используются пути.

В TCP/IP стандартным интерфейсом являются сокеты BSD (Berkeley Sockets). Распространенный вариант сокетных интерфейсов — *Windows Sockets, или Winsock.* Реализация Winsock предоставляет программный интерфейс для приложений Internet, выполняющихся в операционных системах Windows. Программа Winsock загружается как динамически подключаемая библиотека DLL.

3.3. Разрешение имен

IP-адреса и номера портов используются в TCP/IP для идентификации сети, компьютеров и конкретных сетевых приложений, передающих или принимающих сообщения. Однако многие предпочитают использовать для идентификации не номера, а имена.

Компьютеры могут работать только с числами. Из-за этой несовместимости нужны специальные службы, преобразующие имена в IP-адреса. С помощью таких служб можно, обращаясь, например, к Web-узлу компании Xerox, вводить в поле адреса броузера **www.xerox.com, а не 208.134.240.50.** Адрес, состоящий из имен, значительно легче запомнить. Чтобы найти в Internet нужный Web-сервер и получить запрошенную страницу, браузер автоматически преобразует имя хоста в IP-адрес.

Что такое имя

В процессах сетевой коммуникации используются различные типы имен. В Internet имена хостов (компьютеров) организованы в иерархическую структуру внутри *доменов.* В США наиболее распространены следующие домены верхнего уровня:

- com — первоначально предназначался для коммерческих организаций;
- net — первоначально предназначался для сетей провайдеров услуг Internet;
- org — первоначально предназначался для бесприбыльных организаций;
- edu — для учебных организаций;
- gov — для правительственных организаций;
- mil — для военных организаций;

- int — для международных организаций.

За пределами США для идентификации доменов используются следующие коды стран:

- uk — Великобритания;
- au — Австралия;
- ca — Канада;
- ru — Россия;
- ua — Украина.

Коммерческие и некоммерческие организации, другие учреждения, а также отдельные лица регистрируют *имена доменов второго уровня* внутри доменов верхнего уровня (например, ibm.com).

Внутри домена второго уровня индивидуальные компьютеры идентифицируются именем хоста, именем домена второго уровня и именем домена верхнего уровня.

В сетях Microsoft каждому компьютеру присваивается также имя NetBIOS. Это 16-символьное имя, присвоенное администратором, используется для идентификации ресурсов локальной сети.

Чтобы коммуникация в протоколе TCP/IP стала возможной, имена обоих этих типов должны быть преобразованы в IP-адреса.

Преобразование имен в числа

Для преобразования имен в IP-адреса можно воспользоваться одним из следующих средств:

- файлы HOSTS и LMHOSTS;
- службы DNS или DDNS;
- серверы WINS.

Далее рассматриваются все три метода.

Файлы HOSTS и LMHOSTS

В первое время после появления Internet для сопоставления имен хостов (компьютеров) с IP-адресами с целью коммуникации в протоколе TCP/IP использовался файл HOSTS. Это текстовый файл, расположенный на локальном жестком диске. В нем содержится список всех имен хостов и соответствующих IP-адресов. В листинге приведен пример файла HOSTS.

Листинг . Локальный файл HOSTS отображает IP-адреса на имена хостов

```
102.54.94.97   rhino.acme.com   # source server
38.25.63.10    x.acme.com       # x client host
127.0.0.1      localhost
```

Файлы HOSTS были удовлетворительным решением проблемы, пока количество компьютеров в Internet было небольшим. Когда пользователь запрашивал другой компьютер по имени хоста, операционная система находила в файле HOSTS соответствующий IP-адрес.

Однако при добавлении в сеть новых хостов файл HOSTS нужно было вручную модифицировать и копировать на все компьютеры сети. По мере роста Internet эта задача усложнялась.

Службы DNS и DDNS

Служба имен доменов (Domain Name System — DNS) разработана с целью устранения недостатков файлов HOSTS. На серверах DNS хранятся базы данных, в которых IP-адреса отображаются на имена хостов, а свойства TCP/IP клиентских компьютеров конфигурируются с адресом сервера DNS.

В системе Internet существует иерархия серверов DNS. Различные серверы поддерживают информацию DNS для своих зон, или областей ответственности. Если на сервере DNS нет запрошенного клиентским компьютером IP-адреса или имени хоста, он передает запрос на другой сервер DNS, пока запрошенная информация не будет найдена.

Для коммуникации в Internet служба DNS не является абсолютно необходимой, однако без нее вместо имен хостов пришлось бы использовать IP-адреса. Например, если в свойствах TCP/IP клиентского компьютера не сконфигурирован адрес сервера DNS, то обратиться к Web-узлу можно, набрав IP-адрес в поле ввода URL, однако если в поле URL ввести имя хоста, то браузер не отобразит запрошенную страницу. Адрес сервера DNS можно ввести вручную или получить от сервера DHCP (если компьютер конфигурирован как клиент DHCP).

По сравнению с локальными файлами HOSTS служба DNS обладает существенными преимуществами. База данных хранится на центральном сервере, поэтому обновлять ее нужно только в одном месте, а не на

всех клиентских компьютерах. Однако база данных на сервере по-прежнему должна обновляться вручную Эту проблему решает применение *динамической DNS (DDNS),* в которой базы данных DNS обновляются автоматически. С помощью этой расширенной версии DNS клиентские компьютеры могут регистрировать и обновлять на сервере DNS свои записи с информацией о ресурсах в момент внесения в них изменений.

Серверы WINS

Еще один метод отображения имен хостов на IP-адреса состоит в применении серверов WINS. В базе данных на сервере WINS хранятся имена NetBIOS, используемые для идентификации компьютеров и служб в сетях Microsoft. Серверы Windows могут работать как серверы WINS.

В отличие от FQDN (имен хостов в DNS), имена NetBIOS не иерархические, а однородные. Если некоторый сервер имеет имя FQDN, например, exeter.taceam.net, то его имя NetBIOS можно записать просто как Exeter. Протокол TCP не понимает имена NetBIOS, для коммуникации с серверами ему нужны IP-адреса. Как и статический файл HOSTS, сервер WINS преобразует имя в IP-адрес.

В отличие от первоначального DNS, в сервере WINS применяется автоматическое обновление базы данных. Когда клиент WINS подключается к сети, он объявляет об этом серверу WINS и передает ему свое имя и IP-адрес. На основе этой информации сервер WINS поддерживает свою базу данных.

В одной и той же сети службы DHCP, DNS и WINS могут работать совместно и одновременно. В новых операционных системах эти три службы объединены и эффективно взаимодействуют.

Обзор методов разрешения имен

В табл. приведена краткая сводка характеристик этих методов.

Таблица . Методы разрешения имен

Метод разрешения имен	Типы отображаемых имен	Характеристики
Файл HOSTS	Имена хостов на IP-адреса	Текстовый файл: нуждается в ручном обновлении на каждом компьютере
Файл LMHOSTS	Имена NetBIOS на IP-адреса	Текстовый файл: нуждается в ручном обновлении на каждом компьютере
DNS	Имена хостов на IP-адреса	Централизованная база данных, управляемая сервером DNS; нуждается в ручном обновлении
DDNS	Имена хостов на IP-адреса	Централизованная база данных, управляемая сервером DNS; обновляется динамически
WINS	Имена NetBIOS на IP-адреса	Централизованная база данных, управляемая сервером WINS; обновляется динамически