ПРАКТИЧЕСКАЯ РАБОТА №1

ИЗУЧЕНИЕ IP-АДРЕСАЦИИ В ЛОКАЛЬНЫХ СЕТЯХ. МАСКИ.

ПОДСЕТИ И НАДСЕТИ.

Цель работы: научиться переводить числа из десятично-точечной

нотации в двоичную и обратно, разбивать адресное пространство на

подсети, строить надсети.

Краткие теоретические сведения:

Для обмена данными в частной TCP/IP-сети или через Интернет,

каждый сетевой узел должен обладать уникальным 32-битным IP-

адресом. IP-адреса делятся на общие и частные. Первые уникальны в

глобальном масштабе и используются для адресации в Интернете. Вторые

ограничены диапазонами, которые обычно используются в частной сети,

но не видны из Интернета.

Общие IP-адреса

Каждый IP-адрес в Интернете уникален. Для обеспечения такой

уникальности адресов сетей в Интернете организация IANA (Internet

Assigned Numbers Authority) разделила незанятую часть пространства IP-

адресов и делегировала полномочия по их распределению региональным

реестрам, среди которых Asia-Pacific Network Information Center (APNIC),

American Registry for Internet Numbers (ARIN) и Reseaux IP Europeens

(RIPE NCC). Региональные регистраторы выделяют блоки адресов

небольшому количеству крупных поставщиков интернет-услуг (ISP),

которые затем выдают более мелкие блоки своим клиентам и менее

крупным провайдерам. Как правило, интернет-провайдер выдает по

одному общему IP-адресу на каждый напрямую подключенный к

провайдеру компьютер. Этот IP-адрес может назначаться динамически в

момент подключения компьютера к ISP или статически закрепляться за

выделенной линией или модемным подключением.

Частные IP-адреса

Часть IP-адресов никогда не используется в Интернете. Они

называются частными и используются для организации адресации в сетях,

которые «не видны» в общей сети. Например, пользователю,

объединяющему компьютеры в домашнюю TCP/IP-сеть, не надо

назначать общие IP-адреса каждому узлу — он использует частные

адреса. Диапазоны частных адресов варьируются:

Начальный адрес Конечный адрес

10.0.0.0 10.255.255.255

172.16.0.0 172.31.255.255

192.168.0.0 192.168.255.255

5

Узлы с частными IP-адресами могут подключаться к Интернету

через прокси-сервер или компьютер с Windows, сконфигурированный в

качестве NAT-сервера (Network Address Translation). Windows также

поддерживает сервис общего доступа к Интернету (Internet Connection

Sharing, ICS), предоставляющий клиентам частной сети упрощенные

сервисы NAT.

Методы IР- адресации

IP-адреса могут назначаться вручную, динамически (DHCP-

сервером) или автоматически [например, с помощью APIPA (Automatic

Private IP Addressing)].

Ручная IР - адресация

Назначение IP-адресов вручную используется нечасто, но иногда

без него не обойтись. Например, ручное конфигурирование потребуется в

сети, состоящей из нескольких сегментов, при отсутствии DHCP-сервера,

или если IP-адрес DHCP-сервера также назначается вручную. Наконец,

важным сетевым серверам, например DNS- или WINS-серверу или

контроллеру домена, обычно назначают статические IP-адреса.

Статические IP- адреса можно выделить по механизму резервирования

DHCP-адресов, но большинство администраторов предпочитает не

перепоручать это дело DHCP-серверу и назначают их вручную. Во всех

остальных случаях ручное конфигурирование рекомендуется, только если

невозможно использовать DHCP. Администрирование назначенных

вручную IP-адресов отнимает много времени и чревато ошибками,

особенно в средних и крупных сетях.

Протокол DHCP

DHCP-сервер автоматически выделяет DHCP-клиентам IP-адреса

из заданных администратором диапазонов. DHCP-сервер можно

настроить на конфигурирование других параметров TCP/IP, например

адресов DNS- и WINS-серверов, основных шлюзов и т. п.

Автоматическое назначение частных IP-адресов

APIPA (Automatic Private IP Addressing) служит для

автоматического назначения адресов и применяется в простых

односегментных сетях без DHCP-сервера (см. главу 1).

Альтернативная конфигурация

Подобно APIPA, альтернативная конфигурация позволяет

назначить IP-адрес компьютерам, которым недоступен DHCP-сервер.

Однако в отсутствие такого сервера компьютер с альтернативной

конфигурацией не сможет использовать APIPA, даже если этот протокол

будет доступен в сети. Эта функция полезна, когда компьютер работает в

нескольких сетях, в одной из которых нет DHCP-сервера. Например,

портативный компьютер, используемый для работы в офисе и дома. В

обеих сетях используется один и тот же адаптер и локальное

подключение, настроенное на автоматическое получение IP-адреса. При

6

подключении к корпоративной сети параметры TCP/IP настраиваются

DHCP-сервером. Дома DHCP- сервера нет, поэтому используется

определенная альтернативная конфигурация: IP-адрес, маска подсети и

основной шлюз для домашней сети.

Структура IP-адреса

IP-адреса привычно представляется в форме четырех чисел,

разделенных точкой, например 192.168.100.22. Однако это лишь одна из

форм IP-адреса, которая называется десятично-точечной нотацией и

используется для удобства запоминания адреса. В компьютере

применяется двоичная нотация, в которой все числа представлены только

цифрами 1 и 0. Это «родная» форма IP-адреса. Логика IP-адресации

становится понятной при рассмотрении двоичной версии IP- адреса. Для

конфигурирования, управления и устранения неполадок IP-адресации

надо уметь работать с IP-адресами в двоичной форме, а также переводить

их из двоичного в десятичное представление и обратно.

Преобразование двоичного и десятичного представлений

В десятично-точечной нотации каждое 32-битное число IP-адреса

представляется в виде четырех десятичных групп, значение каждой из

которых лежит в диапазоне 0—255, например 192.168.0.225. Эти числа

представляют четыре 8-битных значения, составляющих 32-битный адрес.

В любой нотации каждая из четырех групп называется октет. Но только

двоичная форма позволяет наглядно увидеть значение каждого бита.

Например, IP-адрес 192.168.0.225 в двоичной форме выглядит так:

11000000 10101000 00000000 11100001. В IP-адресах октеты и биты

считаются слева направо. Первый октет соответствует первому слева, а

биты с 1 по 8 соответствуют первым восьми битам, начиная с самого

левого. Второй октет— это следующие восемь битов (9—16), затем идет

третий октет (биты 17—24), а замыкает последовательность четвертый

октет (биты 25—32). В десятично - точечной нотации октеты отделяются

точками, а в двоичной — пробелами.

В табл. 1.1 показаны экспоненциальное и десятичное

представление битов в двоичном октете. Обратите внимание: если

смотреть слева направо, то первый бит дает значение 128, а каждый

последующий бит — половину значения предыдущего. И, наоборот, в

направлении справа налево, начиная с восьмого бита (значение 1), цена

каждого последующего бита в два раза больше, чем предыдущего.

Обратите внимание, что вклад бита в общую сумму ненулевой,

только если он содержит 1. Например, если первый бит — 1, ему

соответствует десятичное значений 128. Если же его значение — 0, то и

десятичное значение равно нулю. Октету со всеми битами; равными 1,

соответствует десятичное значение 255. Если все биты содержат 0,

десятичное значение октета равно 0.

7

Таблица 1.1. Представление битов октета в двоичной нотации

Пример перевода из двоичной нотации в десятичную.

Пусть первый октет IP-адреса в двоичном представлении выглядит

так: 10101100

Первый, третий, пятый и шестой биты содержат 1, а остальные —

0. Для упрощения решения нарисуем таблицу перевода, в которой

отобразим возможные веса битов октета (см. табл. 1.2.):

Таблица 1.2. Пример перевода двоичного числа в десятичное

Сложим десятичные эквиваленты каждого бита и найдем

десятичную сумму октета: 1-й бит (128) + 3-й бит (32) + 5-й бит (8) + 6-й

бит (4) = сумма октета (172) Поскольку сумма составляет 172, первый

октет нашего IP-адреса в десятичной форме равен 172. Применив этот же

метод, можно преобразовать полный IP-адрес вида 10101100 00010001

00000111 00011011 в десятично-точечное представление: 172.17.7.27.

Пример перевода из десятичной нотации в двоичную

Перевод октета из десятичной формы в двоичную осуществляется

записью 1 или 0 в соответствующий бит октета слева направо, пока не

будет получено искомое десятичное число. Если запись 1 в очередной бит

приводит к тому, что полученная сумма превосходит десятичное число,

просто запишите в этот бит 0 и перейдите к следующему. Допустим, надо

перевести IP-адрес 172.31.230.218 в двоичный вид. Первым делом

запишите последовательность возможных весов битов в таблицу 1.3:

Таблица 1.3. Таблица для перевода чисел в двоичную систему счисления

Начнем с первого числа — 128. Поскольку 128 меньше 172,

запишем 1 в первый бит, а наша промежуточная сумма будет 128. Затем

посмотрим вес второго бита — 64. Так как 128 + 64 больше 172, второй

8

бит установим в 0. Затем перейдем к третьему биту, вес которого — 32.

128 и 32 в сумме дают меньше 172, поэтому запишем в этот бит 1.

Промежуточная сумма становится 128 + 0 + 32=160. Перейдем к

четвертому биту, его вес — 16. 160 и 16 в сумме дают больше 172,

поэтому пишем 0. Вес пятого бита — 8. Сумма 160 + 8 меньше 172,

пишем в пятый бит 1, а промежуточная сумма становится 128 + 0 + 32 + 0

+ 8 = 168. И, наконец, вес шестого бита — 4, сумма 168 и 4 равна 172, т. е.

искомому числу. Поэтому пишем 1 в шестой бит, а оставшиеся седьмой и

восьмой биты заполняем нулями.

Таким образом, первый октет в двоичной форме выглядит так:

10101100

Выполнив аналогичные операции с остальными октетами получим

двоичное представление адреса 172.31.230.218: 10101100 00011111

11100110 11011010

Перевод между системами счисления с помощью калькулятора.

При помощи Калькулятора эта операция выполняется намного

быстрее. Чтобы воспользоваться функцией перевода между системами

счисления, в меню Вид (View) выберите Инженерный (Scientific) и

установите переключатель в положение Dec или Bin (в зависимости от

того, из какой системы счисления необходимо перевести число).

Например, для перевода двоичного числа 11001100 в десятичное

представление, отметьте Bin, введите двоичное число. После ввода

двоичного числа просто установите Dec и получите число в десятичном

представлении.

Примечание: как и в десятичной нотации, калькулятор

отбрасывает крайние левые нули, октет 00001110 отображается как 1110.

Поэтому необходимо контролировать число отображаемых в окошке

калькулятора битов, чтобы не перепутать значения. Например, легко

спутать двоичное число 1100001 (десятичное 97) с 11000001 (десятичное

значение 193). Если число бит меньше 8, для представления октета IP-

адреса надо добавить необходимое количество нулей слева.

Идентификаторы сети и узла

Маршрутизаторы, переправляющие пакеты данных между TCP/IP-

сетями не обязаны знать, какому именно узлу предназначен тот или иной

IP-пакет. Вместо этого маршрутизатор считывает из IP-пакета только

адрес сети, в которой находится узел — приемник пакета, а затем на

основе своей таблицы маршрутизации определяет, каким образом

доставить пакет в сеть, в которой расположен адресат. Точное

местоположение узла определяется только после доставки пакета в

нужный сегмент сети.

Такой механизм маршрутизации возможен благодаря делению IP-

адреса на два компонента:

9

• идентификатор сети (network ID) — первая часть IP-адреса,

представляющая конкретную сеть в более крупной TCP/IP-сети (например

в Интернете);

• идентификатор узла (host ID) — вторая часть IP-адреса,

определяющая узел TCP/IP (рабочую станцию, сервер, маршрутизатор

или любое другое ТСР/IР- устройство). Например, разбиение IP-адреса

(131.107.16.200) на идентификаторы сети (это первые два октета —

131.107) и узла (последние два октета — 16.200).

Маска подсети

Еще один необходимый для нормальной работы TCP/IP параметр

— маска подсети (subnet mask), которая служит для определения, в какой

сети находится приемник пакета — локальной или внешней. Маска

подсети — это 32-битный адрес, представляющий собой

последовательность битов со значением 1, который используется для

выделения, или маскировки, идентификатора сети адреса назначения

пакета и отделения идентификаторов сети и узла. Каждому узлу сети

TCP/IP нужна маска подсети (если сеть не разбита.на подсети, т. е.

состоит из одной подсети) или маска по умолчанию (в случае разбиения

сети на подсети).

Например, такое 32-битное число представляет маску подсети по

умолчанию для узлов с адресами класса В (например 172.20.16.200):

11111111 11111111 00000000 00000000 (255.255.0.0)

Когда TCP/IP-узел с адресом 172.20.16.200 отправляет пакет по

адресу 172.21.17.201, он сначала выполняет побитовую операцию И по

отношению к локальным адресу и маске подсети. Поскольку эта

логическая операция в результате дает 0 во всех битах кроме тех, в

который в обоих операндах стояли 1, то 172.20.16.200 И 255.255.0.0 =

172.20.0.0.

Затем узел повторяет эту операцию, но вместо адреса отправителя

подставляет адрес получателя. В результате получается 172.21.0.0. Затем

TCP/IP сравнивает результаты этих операций. Если они совпадают,

получатель расположен в этой же подсети. Иначе приемник и получатель

расположены в разных подсетях.

Длина префикса сети в маске подсети

Поскольку биты идентификатора сети всегда идут последовательно

и начинаются с самого левого, самый простой способ показать маску

подсети — это указать количество битов идентификатора сети в виде

префикса сети. Таким образом, маска подсети выражается в виде IP-

адрес/префикс сети. Например, IP-адрес 131.107.16.200 и маску подсети

255.255.0.0 можно записать в виде 131.107.16.200/16. Число 16 после

слеша обозначает количество единичных битов в маске подсети. Точно

так же, /24 обозначает маску подсети 255.255.255.0 для адреса класса С,

например 206.73.118.23/24 (см. табл. 1.4).

10

Примечание: нотация с префиксом сети также известна как

бесклассовая междоменная маршрутизация (Classless Interdomain

Routing, CIDR).

Таблица 1.4. Маски по умолчанию различных классов сетей

Основной шлюз

Связь между TCP/IP-узлами разных сетей, как правило,

выполняется через маршрутизаторы. Маршрутизатор — это устройство с

несколькими интерфейсами, подключенными к разным сетям, а

маршрутизация — процесс приема IP-пакетов на одном интерфейсе и

пересылка их на другой интерфейс в направлении адресата. С точки

зрения узла сети TCP/IP, основной шлюз — это IP-адрес маршрутизатора,

сконфигурированного на пересылку IP-трафика в другие сети. Пытаясь

передать информацию другому узлу IP-сети, компьютер определяет тип

узла (локальный или удаленный) по маске подсети. Если узел-получатель

расположен в локальном сегменте сети, пакет направляется в локальную

сеть по методу широковещания. В противном случае компьютер

пересылает пакет в основной шлюз, определенный в параметрах TCP/IP.

Обязанность дальнейшей пересылки пакета в нужную сеть возлагается на

маршрутизатор, адрес которого указан в качестве основного шлюза.

Разбиение IP-сетей на подсети и создание надсетей

Маски подсети позволяют настраивать адресное пространство в

соответствии с требованиями к сети. Разбиение на подсети позволяет

организовать иерархическую структуру сетей, а надсети и CIDR

позволяют объединить разные сети в едином адресном пространстве.

Разбиение на подсети

Маски подсети помогают определить, как IP-адрес разбивается на

идентификаторы сети и узла. В адресах классов А, В и С применяются

стандартные маски подсети, занимающие соответственно первые 8, 16 и

24 бита 32-битового адреса. Подсетью называется логическая сеть,

определяемая маской подсети.

Стандартные маски годятся для сетей, которые не предполагается

разбивать. Например, в сети из 100 компьютеров, соединенных с

помощью карт гигабитного Ethernet, кабелей и коммутаторов, все узлы

могут обмениваться информацией по локальной сети. Сеть не нуждается в

маршрутизаторах для защиты от чрезмерного широковещания или для

11

связи с узлами, расположенными в отдельных физических сегментах. В

таком простом случае вполне достаточно идентификатора сети класса С.

Механизм разбиения на подсети

Разбиение на подсети (subnetting) — это логическое разделение

адресного пространства сети путем установки в «1» дополнительных

битов маски подсети. Такое расширение позволяет создавать многие

подсети в адресном пространстве сети.

Например, если маска подсети по умолчанию 255.255.0.0

используется для узлов сети класса В 131.107.0.0, IP-адреса 131.107.1.11 и

131.107.2.11 находятся в одной подсети и поддерживают взаимодействие

посредством широковещания. Но если расширить маску подсети до

255.255.255.0, то эти адреса окажутся в разных подсетях и для обмена

данными соответствующим узлам придется пересылать пакеты на

основной шлюз, который перенаправит дейтаграммы в нужную подсеть.

Внешние по отношению к сети узлы по-прежнему используют маску

подсети по умолчанию для взаимодействия с узлами внутри сети. Обе

версии показаны на рис. 1.1 и 1.2.

Единственный сегмент сети 131.107.0.0/16

Рис. 1.1. Неразбитое на подсети адресное пространство класса В

12

Рис. 1.2. Разбитое на подсети адресное пространство класса В

Показанное на рис. 1.1 исходное адресное пространство класса В,

состоящее из единственной подсети, может содержать максимум 65 534

узлов, а новая маска подсети (рис. 1.2) позволяет разделить адресное

пространство на 256 подсетей, в каждой из которых можно разместить до

254 узлов.

Преимущества разбиения на подсети

Разбиение на подсети часто используют для обеспечения

соответствия физической и логической топологии сети или для

ограничения широковещательного трафика. Другие несомненные

преимущества: более высокий уровень защиты (благодаря ограничению

неавторизованного трафика маршрутизаторами) и упрощение

администрирования (благодаря передаче управления подсетями другим

отделам или администраторам).

Соответствие физической топологии

Допустим, вам поручили спроектировать университетскую сеть,

состоящую из 200 узлов, распределенных в четырех зданиях — Voter Hall,

Twilight Hall, Monroe Hall и Sunderland Hall. В каждом здании

планируется разместить по 50 узлов. Если интернет-провайдер выделил

адрес 208.147.66.0 класса С, вам доступны адреса 208.147.66.1 -

208.147.66.254. Однако из-за размещения в четырех физически

отделенных зданиях, узлы не могут обмениваться данными по локальной

сети. Расширив маску подсети на 2 бита (т. е. позаимствовав их у

идентификатора узла), сеть разбивают на четыре логические подсети, а

для связи устанавливается маршрутизатор (рис. 1.3).

13

208.147.66.192/26

Рис. 1.3. Разбиение на подсети в соответствии с физической топологией

Ограничение широковещательного трафика

Широковещание — рассылка сообщений с одного компьютера на

все расположенные в локальном сегменте устройства. Широковещание

существенно нагружает ресурсы, поскольку занимает полосу пропускания

и требует участия всех сетевых адаптеров и процессоров логического

сегмента сети.

Маршрутизаторы блокируют широковещание и защищают сети от

излишнего трафика. Поскольку маршрутизаторы также определяют

логические ограничения подсетей, разбиение на подсети позволяет

косвенно ограничивать широковещательный трафик в сети.

Определение максимального количества узлов в сети

Зная сетевой адрес, определить максимальное количество узлов в

сети просто: надо возвести 2 в степень, равную количеству битов в

14

идентификаторе узла, отведенных для идентификации узлов, то есть

«нулей» - s, и вычесть 2. Например, в сетевом адресе 192.168.0.0/24 под

идентификатор узла отведено 8 бит, поэтому возможное максимальное

число узлов — 2 = 254.

Исключение идентификаторов узла, состоящих из одних нулей

или одних единиц

Значение 2х показывает общее количество комбинаций значений

битов двоичного числа х, включая комбинации из одних нулей и одних

единиц. Например, V дает 8, т. е. количество различных комбинаций из 3

битов (dec. означает десятичную систему счисления):

000 = 0 (дес.)

001 = 1 (дес.)

010 = 2 (дес.)

011 = 3 (дес.)

100 = 4 (дес.)

101 = 5 (дес.)

110 = 6 (дес.)

111 = 7 (дес.)

Однако узлам нельзя назначать адреса, состоящие из одних только

нулей или единиц, поскольку они зарезервированы для других целей.

Идентификатор узла, состоящий из одних нулей, на самом деле

определяет сеть без указания конкретного узла. Идентификатор узла из

одних единиц зарезервирован в протоколе IP для широковещания

(передачи сообщения всем узлам сети). При подсчете максимального

количества узлов в сети эти варианты надо исключить из рассмотрения (т.

е. вычесть из него 2).

Определение емкости подсети

При увеличении количества битов в маске подсети для создания

подсетей в адресном пространстве идентификатор узла укорачивается, и

создается новое адресное пространство для идентификатора подсети (рис.

1.4).

Чтобы определить количество доступных в адресном пространстве

подсетей, просто возведите 2 в степень у, где у — количество бит в

идентификаторе подсети. Например, если в адресном пространстве

172.16.0.0/16 выделить 8 бит на адрес подсети (т. е. привести к виду

172.16.0.0/24), количество доступных подсетей станет 2s, или 256. Из него

не надо вычитать 2, поскольку большинство современных

маршрутизаторов принимают идентификаторы подсетей только из единиц

или нулей.

15

Рис. 1.4. Адресное пространство с идентификатором подсети

Планируя адресное пространство и маски подсети убедитесь, что

отведенных на идентификатор подсети бит достаточно для размещения

всех подсетей, а также обеспечен резерв для расширения сети в будущем.

Помните, что любую физическую сеть надо рассматривать как подсеть.

Калькулятор позволяет быстро определить необходимое число бит

для идентификатора подсети. Вычтите 1 из требуемого количества

подсетей в десятичном формате, переведите результат в двоичный вид и

посчитайте количество бит в нем. Например, если нужна 31 подсеть,

введите 30 и установите переключатель Bin. Полученное число НПО

говорит, что под идентификатор подсети нужно зарезервировать 5 бит.

Количество узлов в подсети. Количество идентификаторов узлов

в подсети определяется так же, как и узлов в сети — оно равно 2х — 2,

где х — количество бит в идентификаторе узла. Например, в адресе

172.16.0.0/24 резервируется 8 бит под идентификатор узла, поэтому число

узлов в подсети равно 28 — 2, т. е. 254. Для вычисления количества узлов

во всей сети умножают полученный результат на количество подсетей.

В нашем примере адресное пространство 172.16.0.0/24 дает 254

сетей х=256, узлов = 65 024.

16

Конфигурируя адресное пространство и маски подсети в

соответствии с требованиями сети убедитесь, что отвели на

идентификатор узла достаточно бит с учетом возможного увеличения

количества узлов в подсети в будущем.

Калькулятор позволяет быстро определить необходимое

количество бит для идентификатора узла. Прибавьте 1 к требуемому

количеству узлов в подсети в десятичном формате, переведите результат в

двоичный вид и посчитайте количество бит в нем. Например, если нужно

33 узла в подсети, введите 34 и установите переключатель Bin. Результат

100010 говорит о том, что нужно зарезервировать 6 бит под

идентификатор подсети.

Примеры подсетей

В предыдущем примере мы расширили маску подсети для

адресного пространства 172.16.0.0/16 до 255.255.255.0, увеличив ее на

целый октет. На практике маску расширяют более мелкими порциями,

вплоть до отдельных битов.

Например, на рис. 1.5 показано адресное пространство 10.0.0.0/12.

Поскольку адрес относится к классу А, количество единичных битов в

маске подсети по умолчанию равно 8. Мы расширили его на 4 бита, т. е. 4

бита отдано под идентификатор подсети, а оставшиеся 20 служат

идентификатором узла. В такой сети диапазон адресов первой подсети

10.0.0.1-10.15.255.254.

На рис. 1.6 адресу класса В 172.20.0.0 назначена нестандартная

маска подсети 255.255.248.0, расширяющая маску по умолчанию на 5 бит.

В такой сети диапазон адресов первой подсети 172.20.0.1-172.20.7.254.

На рис. 1.7 показан адрес класса С 192.168.0.0/26. В этом примере 2

бита зарезервированы для идентификатора подсети и 6 — под

идентификатор узла. В такой сети диапазон адресов первой подсети

192.168.0.1—192.168.0.62.

Определение диапазонов адресов подсети

Десятично-точечная форма маски подсети позволяет определить

диапазоны IP-адресов в каждой подсети простым вычитанием из 256

числа в соответствующем октете маски.

Например, в сети класса С с адресом 207.209.68.0 с маской подсети

255.255.255.192 вычитание 192 из 256 даст 64. Таким образом, новый

диапазон начинается после каждого 64 адреса: 207.209.68.0-207.209.68.63,

207.209.68.64-207.209.68.127 и т.д. В сети класса В 131.107.0.0 с маской

подсети 255.255.240.0 вычитание 240 из 256 дает 16. Следовательно,

диапазоны адресов подсетей группируются по 16 в третьем октете, а

четвертый октет принимает значения из диапазона 0-255: 131.107.0.0-

131.107.15.255, 131.107.16.0-131.107.31.255 и т. д.

17

Кол-во подсетей

Кол-во узлов в подсети

Рис. 1.5. Разделение на подсети адресного пространства класса А

Кол-во подсетей

Кол-во узлов в подсети

Рис. 1.6. Разделение на подсети адресного пространства класса В

18

Кол-во подсетей

Кол-во узлов в подсети

Рис. 1.7. Разделение на подсети адресного пространства класса С

Узлам нельзя назначать идентификаторы из одних нулей или

единиц, так что исключаются первый и последний адрес каждого

диапазона.

Сложение маршрутов путем создания надсетей

Чтобы предотвратить истощение доступных идентификаторов

сетей старших классов, организации, ответственные за адресацию в

Интернете, предложили схему, называемую созданием надсетей

(supernetting), согласно которой несколько сетей (маршрутов) можно

объединить (или сложить) в единую более крупную сеть. Надсети

позволяют эффективнее управлять выделением участков адресного

пространства.

Допустим, организации нужно объединить в сеть 2000 узлов. Это

слишком много для одной сети класса С, которая поддерживает не более

254 узлов. Сеть класса В поддерживает 65 534 узла, но таких сетей

возможно всего 16 383 и количество свободных стремительно

сокращается. Интернет-провайдеру нет смысла (да и возможности)

выделять сети класса В клиентам, которые будет использовать только 3%

диапазона адресов.

19

Надсети позволяют Интернет-провайдеру выделить клиенту блок

адресов класса С, который будет рассматриваться как единая сеть,

представляющая собой нечто среднее между классами С и В. В нашем

примере блок из 8 идентификаторов сети класса С даст возможность

организации объединить в сеть до 2032 узлов.

Как работают надсети

Надсети отличаются от подсетей тем, что заимствуют биты

идентификатора сети и маскируют их как идентификатор узла. Допустим,

интернет-провайдер выделил блок из 8 адресов сети: 207.46.168.0—

207.46.175.0. Если определить на маршрутизаторах провайдера и всех

узлов сети маску подсети /21 (вместо /24 по умолчанию), все сети будут

казаться единственной сетью из-за того, что их идентификаторы

(урезанные до 21 бита) будут выглядеть одинаково (рис. 1.8).

Рис. 1.8. Надсеть на основе блока адресов класса С

Использование бесклассовой междоменной маршрутизации

CIDR — это эффективный метод поддержки надсетей с помощью

таблиц маршрутизации. Не будь CIDR, в таблицах маршрутизации

следовало бы размещать отдельные записи для каждой сети в надсети, а

так вся надсеть представляется одной записью (рис. 1.9).

Выделенные региональными регистраторами Интернета или

Интернет-провайдером блоки адресов надсети часто называют CIDR-

блоками, а термин CIDR часто используется для обозначения самих

надсетей.

CIDR не совместим с устаревшим протоколом RIP (Routing

Information Protocol) версии 1, который применялся в старых

маршрутизаторах, и требует, чтобы маршрутизатор использовал

бесклассовый протокол маршрутизации, такой как RIP версии 2 или OSPF

(Open Shortest Path First).

20

Рис. 1.9. Использование CIDR для упрощения создания надсетей

Будущее адресного пространства

Использование CIDR для выделения адресов дает новую жизнь

идентификаторам сети. CIDR-блок из предыдущего примера

(207.46.168.0, 255.255.248.0) можно рассматривать двояко:

• как блок 8 адресов сетей класса С;

• как адресное пространство, в котором зафиксирован 21 бит, a 11

битов доступны для изменения.

Во втором случае идентификаторы сети освобождаются от

классовой наследственности и становятся частью бесклассового

пространства IP-адресов. Каждый идентификатор сети независимо от

длины представляет адресное пространство, в котом биты

идентификатора сети зафиксированы, а биты узла можно менять. Биты

узла можно использовать в качестве идентификаторов узлов или в других

целях (допустим, для организации подсетей) и таким образом наилучшим

образом удовлетворить потребности организации в поддержке сетей.

Маски подсети переменной длины

Традиционно все узлы и маршрутизаторы организации используют

одну маску подсети. В этом случае сеть может разбиваться на подсети, в

которых максимальное количество идентификаторов узлов одинаковое.

Однако поддержка масок подсети переменной длины (variable-

length subnet mask, VLSM) позволяет маршрутизаторам обслуживать

21

разные маски. Чаще всего VLSM применяют для разбиения на подсети

самих подсетей. Допустим, большой организации принадлежит большое

адресное пространство 131.107.0.0/16. Внешние маршрутизаторы для

определения идентификатора сети используют первые 16 бит адреса и в

соответствии с этим осуществляют маршрутизацию. При получении

данных из Интернета маршрутизаторы организации используют маску

подсети /22 для перенаправления трафика в любой из 64 региональных

отделений организации. А маршрутизаторы региональных офисов в свою

очередь используют маску подсети /25 для маршрутизации трафика в 8

отделов в рамках отделения.

Как и CIDR, работа масок подсетей переменной длины основана на

бесклассовых протоколах маршрутизации, таких как RIP версии 2 и

OSPF. VRLM несовместим с более старыми протоколами маршрутизации

(например с RIP версии 1).

Использование VLSM для поддержки подсетей разного размера

VLSM также позволяет разбивать сеть на подсети разных размеров

на одном уровне иерархии и более эффективно использовать адресное

пространство.

Например, если одна подсеть должна объединять 100 компьютеров,

вторая - 50, а третья - 20, то не удастся обойтись традиционной маской по

умолчанию для единственного идентификатора сети класса С. Как видно

из табл. 1.5, никакая из масок подсети по умолчанию не обеспечивает

одновременно достаточное число подсетей и узлов в подсети.

Таблица 1.5. Параметры масок подсети класса С (статические)

В таких ситуациях проблему решает VLSM. При этом не надо

обращаться к Интернет-провайдеру за новым диапазоном адресов.

При разбиении на подсети различного размера нужно использовать

специальный шаблон с завершающими нулями; сеть класса С

поддерживает до семи подсетей. Завершающие нули нужны для

предотвращения пересечения адресных пространств подсетей.

Если идентификатор подсети с маской переменной длины

соответствует шаблону из табл. 1.6, подсети не пересекутся, и адреса

будут интерпретироваться однозначно.

22

Таблица 1.6. Идентификация подсети на основе VLSM

На рис. 1.10 показано, как с помощью VLSM построить 3 сети с

100, 50 и 20 узлами соответственно.

Рис. 1.10. VLSM дает дополнительную гибкость при разбиении на

подсети

Увеличение количества доступных узлов средствами VLSM.

Обратите внимание, что в табл. 1.6 седьмая (и последняя) подсеть имеет

такое же количество узлов, как и шестая, отличаются только

идентификаторы подсети, да и то всего одним битом (в идентификаторе 7-

й сети в отличие от остальных отсутствует завершающий нуль). Можно не

использовать все семь подсетей — достаточно определить состоящий из

одних единиц идентификатор подсети на любом уровне, который заменит

0

10

110

1110

11110

111110

1111111

23

все перечисленные в следующих строках таблицы подсети. Например,

определить идентификатор подсети 1111, который заменит подсети 5—7

(см. табл. 2-6). Благодаря этому вы получите еще одну подсеть с 14

узлами вместо 3 подсетей, вместе содержащих только 10 узлов. Это

позволит максимизировать количество узлов, которые вмещает сеть,

состоящая из 5 подсетей.

Если сеть класса С разбита на 3, 5, 6 или 7 подсетей, VLSM

позволяет максимизировать количество доступных узлов.

Задания к практической работе:

1. Перевод числа из десятичного представления в двоичное

вручную посредством табл. 1.3. Исходные данные представлены

в табл. 1.7.

Таблица 1.7. Исходные данные к заданию 1

Вар. Числа Вар. Числа Вар. Числа

1. 105; 224 11. 51; 133 21. 135; 95

2. 97; 251 12. 30; 201 22. 44; 146

3. 163; 35 13. 198; 177 23. 88; 137

4. 24; 132 14. 122; 55 24. 62; 128

5. 154; 67 15. 65; 146 25. 241; 89

6. 201; 22 16. 233; 48 26. 79; 117

7. 11;156 17. 26; 157 27. 150; 71

8. 34; 41 18. 80; 139 28. 15; 149

9. 33;187 19. 140; 77 29. 196; 18

10. 90; 62 20. 83; 111 30. 27; 166

2. Перевод числа из двоичного представления в десятичное

вручную посредством табл. 1.2. Исходные данные приведены в

табл. 1.8.

Таблица 1.8. Исходные данные к заданию 2

Вар. Число Вар. Число Вар. Число

1. 10110111 11. 00110110 21. 00111011

2. 00101001 12. 10110110 22. 10001101

3. 11101101 13. 10110101 23. 10001101

4. 10111101 14. 00011011 24. 11001100

5. 11110101 15. 11101011 25. 10110000

6. 10110111 16. 11011101 26. 10010001

7. 11111011 17. 10110110 27. 10101010

8. 11001100 18. 00011000 28. 10110110

9. 10100011 19. 11011010 29. 10110101

10. 10100101 20. 10101010 30. 11011111

24

3. Преобразование маски подсети из десятично-точечной формы в

форму с префиксом сети и обратно. Привести маску в двоичном

виде, в форме префикса. Исходные данные приведены в табл. 1.9.

Таблица 1.9. Исходные данные к заданию 3

Вар. Маска; префикс Вар. Маска; префикс

1. 255.255.255.128; /15 16. 128.0.0.0; /26

2. 255.255.252.0; /12 17. 248.0.0.0; /27

3. 255.255.192.0; /13 18. 255.255.128.0; /28

4. 255.255.248.0; /14 19. 255.255.248.0; /29

5. 255.224.0.0; /15 20. 255.255.255.240; /30

6. 255.240.0.0; /19 21. 255.255.255.248; /31

7. 255.255.255.252; /17 22. 255.192.0.0; /7

8. 255.255.255.192; /18 23. 252.0.0.0; /12

9. 255.254.0.0; /19 24. 255.0.0.0; /9

10. 240.0.0.0; /20 25. 254.0.0.0; /10

11. 192.0.0.0; /21 26. 255.248.0.0; /11

12. 255.224.0.0; /22 27. 255.255.255.0; /5

13. 255.252.0.0; /23 28. 255.255.224.0; /3

14. 255.255.252.0; /18 29. 224.0.0.0; /6

15. 255.128.0.0; /25 30. 255.255.224.0; /4

4. Вычисление масок подсети. Выделен адрес сети. Рассчитайте

маску сети и запишите еѐ в десятично-точечной нотации, а также

в виде префикса, для каждой задачи, отталкиваясь, в первом

случае, от требований к количеству подсетей, а во втором – к

количеству узлов. Укажите класс рассчитанной сети в каждой из

задач. Исходные данные для обеих задач приведены в табл. 1.10.

Таблица 1.10. Исходные данные к заданию 4

Вар.

Исходный префикс

сети

Требуемое

кол-во

подсетей

Требуемое

кол-во узлов

1 2 3 4

1. 10.54.2.0/24 20 1520

2. 192.168.0.0/25 7 418

3. 196.52.251.0/10 42 122

4. 207.83.100.0/24 9 541

5. 172.22.0.0/16 6 3200

6. 15.12.0.0/8 18 1932

7. 154.23.85.0/24 5 643

8. 10.23.2.0/16 20 2312

25

Окончание таблицы 1.10

1 2 3 4

9. 192.167.0.0/16 14 567

10. 192.168.0.0/24 5 164

11. 191.168.0.0/16 8 2389

12. 100.124.0.0/8 31 12453

13. 172.3.85.0/24 2 142 2 требуемое колво подсетей,142 требуемое число узлов,

14. 190.145.23.0/18 9 4856

15. 211.56.84.0/24 7 120

16. 175.96.41.0/17 56 780

17. 14.53.0.0/9 32 452

18. 164.8.0.0/16 4 2147

19. 174.123.0.0/16 6 8632

20. 153.145.0.0/16 9 452

21. 233.54.93.0/23 12 180

22. 14.59.73.0/17 4 270

23. 192.54.82.0/22 8 135

24. 14.152.0.0/18 5 9641

25. 18.23.4.0/19 3 1236

26. 238.124.0.0/19 11 578

27. 211.162.0.0/25 16 111

28. 198.63.2.0/16 25 368

29. 182.10.5.0/16 17 451

30. 14.2.8.0/10 16 97

5. Вычисление диапазонов адресов подсети. В этом упражнении

нужно вычислить диапазоны адресов подсети, определив

диапазоны первых трех подсетей сети. Для каждого адреса сети и

маски подсети (столбец А), вычтите из 256 значение

соответствующего октета маски подсети. Запишите полученное

значение в колонку В. Затем впишите в колонку С первые четыре

кратные единицы (начните с 0) этого значения. С помощью этих

значений заполните колонки D и Е, как показано в примере (см.

рис.1.11). Исходные данные представлены в табл. 1.11.

Рис. 1.11. Пример выполнения задания 5

26

Таблица 1.11. Исходные данные к заданию 5

Вар.

Исходный префикс

сети

Вар.

Исходный префикс

сети

1. 10.54.2.0/24 16. 175.96.41.0/17

2. 192.168.0.0/25 17. 14.53.0.0/9

3. 196.52.251.0/10 18. 164.8.0.0/16

4. 207.83.100.0/24 19. 174.123.0.0/16

5. 172.22.0.0/16 20. 153.145.0.0/16

6. 15.12.0.0/8 21. 233.54.93.0/23

7. 154.23.85.0/24 22. 14.59.73.0/17

8. 10.23.2.0/16 23. 192.54.82.0/22

9. 192.167.0.0/16 24. 14.152.0.0/18

10. 192.168.0.0/24 25. 18.23.4.0/19

11. 191.168.0.0/16 26. 238.124.0.0/19

12. 100.124.0.0/8 27. 211.162.0.0/25

13. 172.3.85.0/24 28. 198.63.2.0/16

14. 190.145.23.0/18 29. 182.10.5.0/16

15. 211.56.84.0/24 30. 14.2.8.0/10

6. Проверка двух адресов на принадлежность одной подсети. С

помощью логической функции ―И‖ калькулятора можно

определить, принадлежат ли два адреса одной и той же

логической подсети. Необходимо выполнить две операции ―И‖

над соответствующими октетами маски подсети и октетами

заданного IP-адреса. Если результаты совпадут, адреса

принадлежат одной логической подсети. Например, при маске

подсети 255.255.255.240 и IP-адресах 192.168.0.220 и

192.168.0.190, выражение ―240 И 220‖ дает 208, а ―240 И 190‖ -

176. Они отличаются, поэтому адреса принадлежат разным

логическим подсетям. Исходные данные представлены в табл.

1.12.

Таблица 1.12. Исходные данные к заданию 6

Вар. IP-адрес 1 IP-адрес 2 Маска

1 2 3 4

1. 10.54.2.24 10.54.2.48 255.255.255.128

2. 192.168.0.25 192.168.2.5 255.255.252.0

3. 196.52.251.10 196.52.240.97 255.255.192.0

4. 207.83.100.24 207.83.71.39 255.255.248.0

5. 172.22.0.16 172.1.3.8 255.224.0.0

6. 15.12.0.8 15.122.0.7 255.240.0.0

7. 154.23.85.24 154.23.54.1 255.255.255.252

8. 10.23.2.16 10.23.9.2 255.255.255.192

27

Окончание таблицы 1.12

1 2 3 4

9. 192.167.0.6 192.154.0.78 255.254.0.0

10. 192.168.0.4 192.52.14.87 240.0.0.0

11. 191.168.0.1 171.168.0.85 192.0.0.0

12. 100.124.0.8 100.128.98.4 255.224.0.0

13. 172.3.85.24 ip адрес 1 , 183.54.78.6 ip адрес 2, 255.252.0.0-маска

14. 190.145.23.18 190.145.17.5 255.255.252.0

15. 211.56.84.24 211.128.5.4 255.128.0.0

16. 175.96.41.17 39.214.5.7 128.0.0.0

17. 14.53.0.9 10.5.87.5 248.0.0.0

18. 164.8.0.16 164.8.8.8 255.255.128.0

19. 174.123.0.16 174.123.95.7 255.255.248.0

20. 153.145.0.16 153.145.0.31 255.255.255.240

21. 233.54.93.23 233.251.8.5 255.255.255.248

22. 14.59.73.17 14.59.73.45 255.192.0.0

23. 192.54.82.22 198.0.56.21 252.0.0.0

24. 14.152.0.18 15.124.5.6 255.0.0.0

25. 18.23.4.19 19.34.52.6 254.0.0.0

26. 238.124.0.19 238.124.98.4 255.248.0.0

27. 211.162.0.25 211.162.89.5 255.255.255.0

28. 198.63.2.16 198.63.9.1 255.255.224.0

29. 182.10.5.6 111.0.0.3 224.0.0.0

30. 14.2.8.10 14.2.22.39 255.255.224.0

Контрольные вопросы:

1. Понятие IP-адреса, маски, шлюза.

2. Что такое префикс сети?

3. Что такое префикс маски?

4. Классы сетей. Свойства.

5. Механизм построения надсети.

6. Механизм разбиения на подсети.

ПРАКТИЧЕСКАЯ РАБОТА №2

УТИЛИТЫ ДИАГНОСТИКИ КОМПЬЮТЕРНЫХ СЕТЕЙ

(WINDOWS)

Цель работы: изучить принцип действия и получить практические

навыки использования основных сетевых утилит MS Windows.

Краткие теоретические сведения:

В состав TCP/IP входят диагностические утилиты,

предназначенные для проверки конфигурации тестирования сетевого

соединения. Некоторые из них приведены в табл. 2.1.

Таблица 2.1. Утилиты диагностики сети MS Windows

Наименование Назначение

arp

Выводит для просмотра и изменения таблицу

трансляции адресов, используемую протоколом

разрешения адресов ARP (Address Resolution

Protocol - определяет локальный адрес по IP-

адресу)

hostname

Выводит имя локального хоста. Используется без

параметров.

ipconfig

Выводит значения для текущей конфигурации стека

TCP/IP: IP-адрес, маску подсети, адрес шлюза по

умолчанию, адреса WINS (Windows Internet Naming

Service) и DNS (Domain Name System)

nbtstat

Выводит статистику и текущую информацию по

NetBIOS, установленному поверх TCP/IP.

Используется для проверки состояния текущих

соединений NetBIOS.

netstat

Выводит статистику и текущую информацию по

соединению TCP/IP.

nslookup

Осуществляет проверку записей и доменных

псевдонимов хостов, доменных сервисов хостов, а

также информации операционной системы, путем

запросов к серверам DNS.

ping

Осуществляет проверку правильности

конфигурирования TCP/IP и проверку связи с

удаленным хостом.

29

Окончание таблицы 2.1

Наименование Назначение

route

Модифицирует таблицы маршрутизации IP.

Отображает содержимое таблицы, добавляет и

удаляет маршруты IP.

tracert

Осуществляет проверку маршрута к удаленному

компьютеру путем отправки эхо-пакетов

протокола ICMP (Internet Control Message

Protocol). Выводит маршрут прохождения пакетов

на удаленный компьютер.

Проверка правильности конфигурации TCP/IP

При устранении неисправностей и проблем в сети TCP/IP следует

сначала проверить правильность конфигурации TCP/IP. Для этого

используется утилита ipconfig.

Эта команда полезна на компьютерах, работающих с DHCP

(Dynamic Host Configuration Protocol), так как дает пользователям

возможность определить, какая конфигурация сети TCP/IP и какие

величины были установлены с помощью DHCP.

Тестирование связи с использованием утилиты ping

Утилита ping (Packet Internet Grouper) используется для проверки

конфигурирования TCP/IP и диагностики ошибок соединения. Она

определяет доступность и функционирование конкретного хоста.

Использование ping лучший способ проверки того, что между локальным

компьютером и сетевым хостом существует маршрут. Хостом называется

любое сетевое устройство (компьютер, маршрутизатор), обменивающееся

информацией с другими сетевыми устройствами по TCP/IP.

Команда ping проверяет соединение с удаленным хостом путем

посылки к этому хосту эхо-пакетов ICMP и прослушивания эхо-ответов.

Ping ожидает каждый посланный пакет и печатает количество переданных

и принятых пакетов. Каждый принятый пакет проверяется в соответствии

с переданным сообщением. Если связь между хостами плохая, из

сообщений ping станет ясно, сколько пакетов потеряно.

По умолчанию передается 4 эхо-пакета длиной 32 байта

(периодическая последовательность символов алфавита в верхнем

регистре). Ping позволяет изменить размер и количество пакетов, указать,

следует ли записывать маршрут, который она использует, какую величину

времени жизни (ttl) устанавливать, можно ли фрагментировать пакет и т.д.

При получении ответа в поле time указывается, за какое время (в

миллисекундах) посланный пакет доходит до удаленного хоста и

возвращается назад. Так как значение по умолчанию для ожидания

отклика равно 1 секунде, то все значения данного поля будут меньше 1000

30

миллисекунд. Если вы получаете сообщение «Request time out»

(Превышен интервал ожидания), то, возможно, если увеличить время

ожидания отклика, пакет дойдет до удаленного хоста. Это можно сделать с

помощью ключа -w.

Ping можно использовать для тестирования как имени хоста (DNS

или NetBIOS), так и его IP-адреса. Если ping с IP-адресом выполнилась

успешно, а с именем - неудачно, это значит, что проблема заключается в

распознавании соответствия адреса и имени, а не в сетевом соединении.

Изучение маршрута между сетевыми соединениями с помощью

утилиты tracert

Tracert - это утилита трассировки маршрута. Она использует поле

TTL (time-to-live, время жизни) пакета IP и сообщения об ошибках ICMP

для определения маршрута от одного хоста до другого.

Утилита tracert может быть более содержательной и удобной, чем

ping, особенно в тех случаях, когда удаленный хост недостижим. С

помощью нее можно определить район проблем со связью (у Internet-

провайдера, в опорной сети, в сети удаленного хоста) по тому, насколько

далеко будет отслежен маршрут. Если возникли проблемы, то утилита

выводит на экран звездочки (\*), либо сообщения типа «Destination net

unreachable», «Destination host unreachable», «Request time out», «Time

Exeeded ».

Утилита tracert работает следующим образом: посылается по 3

пробных эхо-пакета на каждый хост, через который проходит маршрут до

удаленного хоста. На экран при этом выводится время ожидания ответа на

каждый пакет (Его можно изменить с помощью параметра -w). Пакеты

посылаются с различными величинами времени жизни. Каждый

маршрутизатор, встречающийся по пути, перед перенаправлением пакета

уменьшает величину TTL на единицу. Таким образом, время жизни

является счетчиком точек промежуточной доставки (хопов). Когда время

жизни пакета достигнет нуля, предполагается, что маршрутизатор пошлет

в компьютер-источник сообщение ICMP ``Time Exeeded'' (Время истекло).

Маршрут определяется путем посылки первого эхо-пакета с TTL=1. Затем

TTL увеличивается на 1 в каждом последующем пакете до тех пор, пока

пакет не достигнет удаленного хоста, либо будет достигнута максимально

возможная величина TTL (по умолчанию 30, задается с помощью

параметра -h).

Маршрут определяется путем изучения сообщений ICMP, которые

присылаются обратно промежуточными маршрутизаторами.

Примечание: некоторые маршрутизаторы просто молча

уничтожают пакеты с истекшим TTL и не будут видны утилите tracert.

Утилита ARP

Основная задача протокола ARP - трансляция IP-адресов в

соответствующие локальные адреса. Для этого ARP-протокол использует

31

информацию из ARP-таблицы (ARP-кэша). Если необходимая запись в

таблице не найдена, то протокол ARP отправляет широковещательный

запрос ко всем компьютерам локальной подсети, пытаясь найти владельца

данного IP-адреса. В кэше могут содержаться два типа записей:

статические и динамические. Статические записи вводятся вручную и

хранятся в кэше постоянно. Динамические записи помещаются в кэш в

результате выполнения широковещательных запросов. Для них

существует понятие времени жизни. Если в течение определенного

времени (по умолчанию 2 мин.) запись не была востребована, то она

удаляется из кэша.

Утилита netstat.

Утилита netstat позволяет получить статическую информацию по

некоторым из протоколов стека (TCP, UDP, IP, ICMP), а также выводит

сведения о текущих сетевых соединениях. Особенно она полезна на

брандмауэрах, с ее помощью можно обнаружить нарушения безопасности

периметра сети.

Задания к практической работе:

1. Получение справочной информации по командам.

Выведите на экран справочную информацию по утилитам arp,

ipconfig, nbstat, netstat, nslookup, route, ping, tracert, hostname. Для

этого в командной строке введите имя утилиты без параметров

или с /?. Изучите и запишите ключи, используемые при запуске

утилит.

2. Получение имени хоста. Выведите на экран имя локального хоста

с помощью команды hostname.

3. Изучение утилиты ipconfig . Проверьте конфигурацию TCP/IP с

помощью утилиты ipconfig. Заполните таблицу:

IP-адрес

Маска подсети

Основной шлюз

Описание адаптера

Физический адрес

сетевого адаптера

Адрес DNS-сервера

Адрес WINS-сервера

4. Тестирование связи с помощью утилиты ping. Проверьте

правильность установки и конфигурирования TCP/IP на

локальном компьютере. Проверьте, правильно ли добавлен в сеть

32

локальный компьютер и не дублируется ли IP-адрес. С помощью

команды ping проверьте перечисленные ниже адреса и для

каждого из них отметьте время отклика. Попробуйте увеличить

время отклика.

127.0.0.1

172.22.222.1

Задайте различную длину посылаемых пакетов. Определите

доменное имя компьютера.

5. Определение пути IP-пакета. С помощью команды tracert

проверьте для перечисленных ниже адресов, через какие

промежуточные узлы идет сигнал. Отметьте их:

172.22.222.1

mail.ru

bstu.ru

6. Просмотр ARP-кэша. С помощью утилиты arp просмотрите ARP-

таблицу локального компьютера.

7. Получение информации о текущих сетевых соединениях и

протоколах стека TCP/IP. С помощью утилиты netstat выведите

перечень сетевых соединений и статистическую информацию для

протоколов UDP, TCP, ICMP, IP.

8. Net view. Выводит список доменов, компьютеров или общих

ресурсов на данном компьютере. Вызванная без параметров,

команда net view выводит список компьютеров в текущем домене.

Исследовать ресурсы домена bstu.ru с помощью команды net

view. Получить списки общих ресурсов компьютеров вашей

аудитории.

Контрольные вопросы:

1. Какие утилиты можно использовать для проверки правильности

конфигурирования TCP/IP?

2. Каким образом команда ping проверяет соединение с удаленным

хостом?

3. Что такое хост?

4. Что такое петля обратной связи?

5. Сколько промежуточных маршрутизаторов сможет пройти IP-

пакет, если его время жизни равно 30?

6. Как работает утилита tracert?