**Билет 1. Архитектура КС, их характеристики**.

***Архитектура*** - спецификации связи, разработанные для определения функций сети и установления стандартов различных моделей вычислительных систем, предназначенных для обмена и обработки данных.

    Для стандартизации сетей Международная организация стандартов (OSI) предложила семиуровневую сетевую архитектуру. К сожалению, конкретные реализации сетей не используют все уровни международного стандарта. Однако этот стандарт дает общее представление о взаимодействии отдельных подсистем сети.

**Семиуровневая сетевая архитектура**

* Физический уровень (Physical Layer).
* Уровень управления линией передачи данных (Data Link).
* Сетевой уровень (Network Layer).
* Транспортный уровень (Transport Layer).
* Сеансовый уровень (Session Layer).
* Уровень представления (Presentation Layer).
* Уровень приложений (Application Layer).

***Физический уровень (Physical Layer)*** обеспечивает виртуальную линию связи для передачи данных между узлами сети. На этом уровне выполняется преобразование данных, поступающих от следующего, более высокого уровня (уровень управления передачей данных), в сигналы, передающиеся по кабелю.

    В глобальных сетях на этом уровне могут использоваться модемы и интерфейс RS-232-C. Характерные скорости передачи здесь определяются линиями связи и для телефонных линий (особенно отечественных) обычно не превышают 2400 бод.

    В локальных сетях для преобразования данных применяются сетевые адаптеры, обеспечивающие скоростную передачу данных в цифровой форме. Скорость передачи данных может достигать десятков и сотен мегабит в секунду.

***Уровень управления линией передачи данных (Data Link)***обеспечивает виртуальную линию связи более высокого уровня, способную безошибочно передавать данные в асинхронном режиме. При этом данные обычно передаются блоками, содержащими дополнительную управляющую информацию. Такие блоки называют кадрами.

    При возникновении ошибок автоматически выполняется повторная посылка кадра. Кроме того, на уровне управления линией передачи данных обычно обеспечивается правильная последовательность передаваемых и принимаемых кадров. Последнее означает, что если один компьютер передает другому несколько блоков данных, то принимающий компьютер получит эти блоки данных именно в той последовательности, в какой они были переданы.

***Сетевой уровень (Network Layer)*** предполагает, что с каждым узлом сети связан некий процесс. Процессы, работающие на узлах сети, взаимодействуют друг с другом и обеспечивают выбор маршрута передачи данных в сети (маршрутизацию), а также управление потоком данных в сети. В частности, на этом уровне должна выполняться буферизация данных.

***Транспортный уровень (Transport Layer)*** может выполнять разделение передаваемых сообщений на пакеты на передающем конце и сборку на приемном конце. На этом уровне может выполняться согласование сетевых уровней различных несовместимых между собой сетей через специальные шлюзы. Например, такое согласование потребуется для объединения локальных сетей в глобальные.

***Сеансовый уровень (Session Layer)*** обеспечивает интерфейс с транспортным уровнем. На этом уровне выполняется управление взаимодействием между рабочими станциями, которые участвуют в сеансе связи. В частности, на этом уровне выполняется управление доступом на основе прав доступа.

***Уровень представления (Presentation Layer)*** описывает шифрование данных, их сжатие и кодовое преобразование. Например, если в состав сети входят рабочие станции с разным внутренним представлением данных (ASCII для IBM PC и EBCDIC для IBM-370), необходимо выполнить преобразование.

***Уровень приложений (Application Layer)*** отвечает за поддержку прикладного программного обеспечения конечного пользователя.

Архитектуры КС

*O*DNA (DECNet) - Основным коммутационным элементом сетевой архитектуры DNA является узел. Все узлы равноправны, т.е. каждый узел, может выступать в качестве любого функционального элемента ИВС. Узлы закреплены за различными областями (подобластями). Каждая область имеет своего администратора и средства маршрутизации.   Каждый узел имеет свой уникальный адрес. Структура адреса DECnet-4: 1 байт -номер области, 1 байт - номер подобласти, 6 байт - Ethernet адрес.

* SNA
* DARPA (TCP/IP, Internet)
* Novell Netware
* SMB
* AppleTalk
* XNS
* IPv6

Характеристики:

1.Иерархия протоколов

2.Соответствие модели ISO

3.Адресация

* Узлов
  + Индивидуальная
  + Групповая
  + Широковещательная
  + Приложений

4.Связь с канальным уровнем

* Разрешение адресов
* Фрагментация
  + Поузловая
  + На источнике

5.Сетевые протоколы

6.Маршрутизация

* По типу маршрута
  + Индивидуальная
  + Групповая
* По адаптивности к изменениям в сети
  + Статическая
  + Динамическая
  + Предопределенная («от источника»)
* По месту проведения маршрутных вычислений
  + Централизованная
  + Децентрализованная
  + Гибридная
* По числу возможных маршрутов
  + Однопутевые
  + Многопутевые
* По характеру используемой информации
  + Глобальные
  + Локальные
  + Смешанные

7.Транспортные механизмы

* Дейтаграммные транспортные протоколы
* Потоковые транспортные протоколы
* Многопоточные транспортные протоколы

8.Именование ресурсов

* Одноуровневое
* Двухуровневое
* Иерархическое

9.Прикладные протоколы

* Протоколы удаленного терминала
* Протоколы передачи файлов
* Протоколы электронной почты
* …

10.Управление

11.Защита информации

**Билет 2. Архитектура TCP/IP. Иерархия протоколов.**

ARP Address Resolution Protocol - Отвечает за получение MAC адреса хоста, размещенного в текущей сети, по его IP адресу. Использует broadcast. Для передачи данных по сети хост должен знать MAC адрес хоста, которому передаются данные. Для получения МАС адреса по известному IP адресу служит протокол ARP.

ICMP Internet Control Message Protocol - Посылка сообщений об ошибках, обнаруженных в процессе передачи пакетов.

Служит для общения маршрутизатора с хостом, отправляющим или посылающим данные контрольными сообщениями и сообщениями об ошибках.

Использует для передачи IP и является его составной частью.

IGMP Internet Group Management Protocol - Информирует маршрутизаторы о наличии в данной сети multicast группы.

Информирует маршрутизаторы о наличии в данной сети multicast группы. Информация рассылается по маршрутизаторам, поддерживающим рассылку таких сообщений.

IP Internet Protocol - Обеспечивает маршрутизацию пакетов.

IP не устанавливает непосредственное соединение между хостами, а используют адреса, помещенные в заголовок IP пакета, для передачи их получателям. Выбор пути передачи называется маршрутизацией.

IP используют поля в заголовке для фрагментации и восстановления датаграмм Internet, когда это необходимо для их передачи через сети с малым размером пакетов.

Не требует подтверждения получения данных. Это означает, что отправитель и получатель не информируются о пропаже пакета или неправильной последовательности получения пакетов.

TCP Transmission Control Protocol - Обеспечивает соединение между двумя хостами, с гарантируемой доставкой пакетов.

Транспортный протокол с гарантированной доставкой пакетов. Данные представляются как поток байтов и могут передаваться в обоих направлениях. Некоторое количество октетов могут упаковываться в сегменты для передачи через системы Internet. Достоверность передачи информации достигается присваиванием каждому сегменту уникального номера в последовательности. Для проверки доставки сегмента используются подтверждения (ACK), которые возвращаются для каждого посланного сегмента. Если подтверждение не получено, то данные посылаюся еще раз через некоторый интервал времени (timeout). Если сегмент пришел поврежденным, то хост уничтожает этот пакет и не посылает подтверждение получения.

UDP User Datagram Protocol - Обеспечивает соединение между двумя хостами, при котором не гарантируется доставка пакетов.

Обеспечивает соединение с негарантированной доставкой пакетов. Используется в приложениях, не требующих подтверждения получения пакетов (NetBIOS name service, NetBIOS datagram service, SNMP)

Протоколы в зависимости от уровней:

7 SMTP, POP-3, HTTP, FTP, DNS DNS, TFTP, SNMP

6 -

5 -

4 TCP UDP

3 ICMP, IGMP IP | ARP

2 Ethernet, Token Ring, FDDI, …

1 –

**Билет 3. IP-адресация. Классы IP-сетей.**

Каждый узел TCP/IP идентифицируется логическим IP-адресом. Эти адреса уникальны для каждого из узлов, общающихся по протоколу TCP/IP. Каждый 32-битный IP-адрес идентифицирует местонахождение узла в сети точно так же, как обычный адрес обозначает дом на улице города.

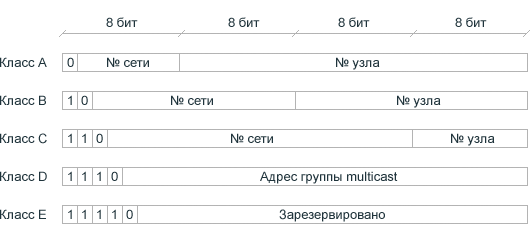
Максимум 2^32 узлов.

IP-адрес имеет две части — идентификатор сети и идентификатор узла.

Идентификатор сети, также называемый адресом сети, обозначает один сетевой сегмент в более крупной объединенной сети (сети сетей), использующей протокол TCP/IP. IP-адреса всех систем, подключенных к одной сети, имеют один и тот же идентификатор сети. Этот идентификатор также используется для уникального обозначения каждой сети в более крупной объединенной сети.

Идентификатор узла, также называемый адресом узла, определяет узел TCP/IP (рабочую станцию, сервер, маршрутизатор или другое TCP/IP-устройство) в пределах каждой сети. Идентификатор узла уникальным образом обозначает систему в том сегменте сети, к которой она подключена.

IP-адреса записываются в точечно-десятичной нотации. 32-битный IP-адрес делится на четыре 8-битных октета.



Вверху – классы IP сетей.

Класс - Наименьший адрес - Наибольший адрес

A - 01.0.0 - 126.0.0.0 сети большого размера 126 сетей. 2^24-2 узлов в сети

B - 128.0.0.0 - 191.255.0.0 сети среднего размера 2^14 сетей. 2^16-2 узлов в сети

C - 192.0.1.0. - 223.255.255.0 небольшие сети 2^21 сетей. 2^8-2 узлов

D - 224.0.0.0 - 239.255.255.255 групповые адреса 2^28 адресов

E - 240.0.0.0 - 247.255.255.255 для экспериментов 2^28 адресов

Зарезервированные адреса:

1. 0.0.0.0 – в таблицах маршрутизации – маршрут по умолчанию. При адресации – данная сеть.
2. Узел данной IP-сети. Примеры: 0.9.3.12 0.0.1.2 0.0.0.25
3. Конкретная IP-сеть. Пример 12.0.0.0. Используется в таблицах маршрутизации
4. Все узлы данной IP-сети. Пример 12.255.255.255
5. Все узлы данной локальной сети 255.255.255.255
6. Петля обратной связи Обычно 127.0.0.1.
7. IANA зарезервировала диапазон в RFC1918 10.0.0.0-12.255.255.255/172.16.0.0-172.31.255.255/192.168.0.0-192.168.255.255. 2.
   1. 224.0.0.1-все узлы данной подсети
   2. 224.0.0.2-все маршрутизаторы данной подсети
   3. 224.0.0.5-все OSPF маршрутизаторы
   4. 224.0.0.6-все назначенные OSPF- маршутизаторы
   5. 224.0.0.9-все RIP-2 маршрутзаторы
   6. 224.0.0.10-все IGRP маршрутизаторы

**Билет 4. Структуризация IP-сетей. Понятие маски сети. Организация подсетей, префикс сети.**

В терминологии сетей TCP/IP маской подсети или маской сети называется битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Например, узел с IP-адресом 12.34.56.78 и маской подсети 255.255.255.0 находится в сети 12.34.56.0/24 с длиной префикса 24 бита.

Другой вариант определения — это определение подсети IP-адресов. Например, с помощью маски подсети можно сказать, что один диапазон IP-адресов будет в одной подсети, а другой диапазон соответственно в другой подсети.

Маска назначается по следующей схеме 2^8 − n (для сетей класса C), где n — количество компьютеров в подсети + 2, округленное до ближайшей большей степени двойки (эта формула справедлива для n ≤ 254, для n > 254 будет другая формула).

Пример: В некой сети класса C есть 30 компьютеров, маска для такой сети вычисляется следующим образом:

2^8 - 32 = 224 (0E0h) < = > 255.255.255.224 (0xFFFFFFE0)

Префикс сети - число единиц в маске:

O Маска: 255.255.255.240 O Префикс: 28 O Записывается: 195.19.212.96/28

Префиксы для стандартных сетей:

O A – 8 O B – 16 O C – 24

*O*Надсети – для объединения нескольких сетей

*O*Пример 1:

*O*Сети 195.19.212.0 и 195.19.213.0

*O*Общая надсеть на 512 адресов: 195.19.212.0/255.255.254.0

*O*Пример 2:

*O*Сети 192.168.1.0 и 192.168.2.0

*O*Общая надсеть на 1024!! адреса: 192.168.0.0/255.255.252.0

**Структуризация сетей IP**

*O*A – IP-адрес узла

*O*M – маска подсети

*O*Адрес подсети, широковещательный адрес подсети, размер подсети?

*O*Адрес подсети:

*O*Net = A & M

*O*Широковещательный адрес:

*O*Broad = A v (!M)

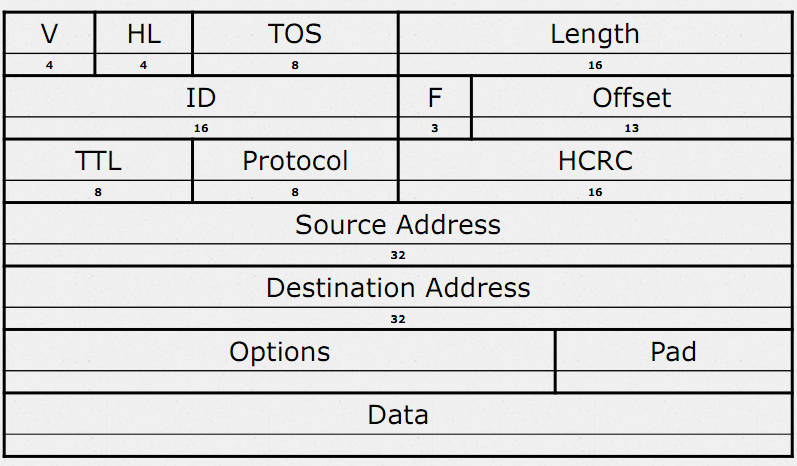
*O*Максимальное число узлов в сети:

*O*K = !M - 1

**Билет 5. Архитектура сетей ТСP/IP. Протокол IP.**

**+БИЛЕТ 2!!** Протокол IP (Internet Protocol) входит в состав стека протоколов TCP/IP и является основным протоколом сетевого уровня, использующимся в Интернет. IP - это не ориентированный на установление соединения и ненадежный протокол передачи. Термин "не ориентированный на установление соединения" означает, что сеанс для обмена данными не устанавливается. Термин "ненадежный" означает, что доставка не гарантируется. IP всегда предпринимает все усилия, чтобы доставить пакет. IP-пакет может быть потерян, доставлен вне очереди, дублирован или задержан. Протокол IP не пытается исправить ошибки этих типов. Подтверждение получения пакетов и повторное обращение за потерянными пакетами входят в круг обязанностей протокола более высокого уровня, например TCP.

Формат пакета:



Version - Версия протокола. Текущая версия 4 (0100).

Header Length - Количество 32 битных слов в заголовке пакета. Минимальная размер заголовка 20 байт, то есть в Header Length = 0x5. Наличие информации в поле Options может увеличить размер заголовка максимум на 4 байта. Если это поле заполнено не полностью, то заполненные биты покрываются 32 битными словами и незаполненная часть заполняется нулями.

Type of Service - Желаемое качество обслуживания пакета при его доставке.

Total Length - Общая длина IP пакета.

Identifier - Идентификатор пакета. Если пакет фрагментирован, то все фрагменты имеют одинаковый идентификатор. Это необходимо для восстановления исходного пакета.

Fragmentation Flags - Флаги фрагментации. В настоящее время используется только два бита. Один показывает, фрагментирован пакет или нет, второй говорит о наличии фрагментов, следующих за текущим.

Fragment Offset - Позиция фрагмента внутри пакета. Если пакет не фрагментирован то 0x0.

Time to Live - Время в секундах, в течении которого пакет может находиться в сети. Маршрутизаторы уменьшают значение этого поля на то время, которое пакет находится на нем (обычно от 1 до 16). По истечении TTL пакет уничтожается. В NT 4.0 по умолчанию равно 128.

Protocol - Тип транспортного протокола, используемого при передаче (TCP или UDP)

Header Checksum - Контрольная сумма.

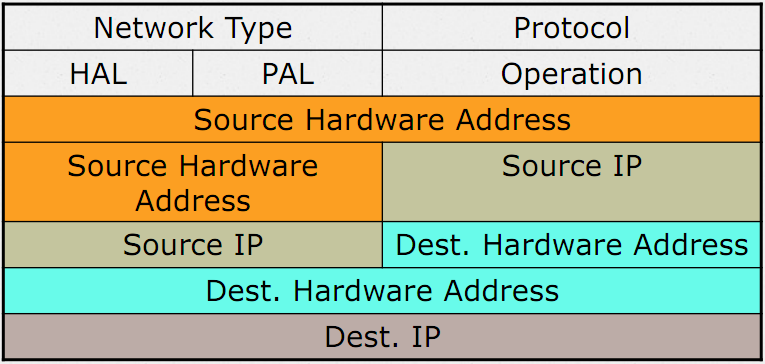
Source Address - Идентификатор отправителя пакета.

Destination Address - Идентификатор получателя пакета.

Options and Padding - Переменное число 32 битных слов (максимум 4 байта) используются для дополнительной информации о пакете.

**Билет 6. Связь c канальным уровнем в ТСP/IP. Протокол разрешения адреса ARP.**

Формат пакета ARP:



Для передачи данных по сети хост должен знать MAC адрес хоста, которому передаются данные. Для получения МАС адреса по известному IP адресу служит протокол ARP.

Sender’s Hardware Address - MAC адрес отправителя

Sender’s IP Address - IP адрес отправителя

Target’s Hardware Address - MAC адрес получателя

Target’s IP Address - IP адрес получателя

Network Type – тип канального протокола

* Для Ethernet – 1

OProtocol - протокол сетевого уровня

* IP – 2048

HAL - длина канального адреса

PAL - длина сетевого адреса

Operation - тип операции

* 1 – запрос
* 2 – ответ

Разрешение локального IP адреса

ARP запрос возникает всегда, когда хост пытается связаться с другим хостом. Если IP определит, что IP адрес получателя находится в локальной сети, то сначала хост проверяет наличие этого адреса в собственном кэше ARP.

В случае неудачи ARP строит запрос и посылает его широковещательным сообщением всем хостам подсети. Запрос имеет следующую структуру: “Хост с IP адресом a.b.c.d, сообщите мне Ваш MAC адрес”. Запрос также содержит информацию об IP и MAC адресе отправителя.

Каждый хост в подсети получает запрос и проверяет на соответствие свой IP адрес. Если он не совпадает с указанным в запросе, то запрос игнорируется.

Если IP адрес, указанный в запросе, совпадает с IP адресом хоста, то хост посылает ARP ответ непосредственно отправителю, используя его MAC адрес. И заносит информацию об IP/MAC адресе отправителя в ARP кэш.

Разрешение удаленного IP адреса

В этом случае инициируется ARP запрос маршрутизатору, который пересылает датаграммы в сеть назначения.

Если IP определит, что IP адрес получателя не находится в локадьной сети, то сначала проверяется наличие этого адреса в таблице маршрутизации. В случае неудачи происходит попытка найти MAC адрес default gateway в ARP кэше.

Если MAC адрем default gateway не найден, то формируется ARP запрос на определение его MAC адреса. После получения ответа хост посылает информацию через default gateway в сеть назначения.

Когда запрос приходит в сеть назначения, то маршрутизатор определяет MAC адрес получателя (см. разрешение локального IP адреса) и посылает ICMP ответ маршрутизатору хоста отправителя.

**Билет 7. Групповая доставка в TCP/IP. Протокол IGMP.**

Internet Group Management Protocol (IGMP)

* Протокол сетевого уровня
* Предназначен для решения задач:
* Управления
* Нотификации об ошибках и проблемах
* Тестирования и мониторинга
* Описан в стандарте RFC792
* Инкапсулируется в IP

GMP (англ. Internet Group Management Protocol — протокол управления группами Интернета) — протокол управления групповой (multicast) передачей данных в сетях, основанных на протоколе IP. IGMP используется маршрутизаторами и IP-узлами для организации сетевых устройств в группы. Информирует маршрутизаторы о наличии в данной сети multicast группы. Информация рассылается по маршрутизаторам, поддерживающим рассылку таких сообщений.

Version - Версия IGMP (0x1)

Type - Тип IGMP сообщения. Тип 0x1 называется Host Membership Query и используется маршрутизатором для проверки хоста на принадлежность к данной группе. Тип 0x2 называется Host Membership Report и используется хостом для определения своего членства в группе.

Unused - 0 для отправителя. Получателем игнорируется.

Checksum - Контрольная сумма заголовка.

Group Address - Групповой адрес.

Предназначен для информирования маршрутизатора о членстве в группах

O Стандарт: версия 1 – RFC 1112

O Маршрутизатор посылает запрос

O Отчет посылается с задержкой

O Хост должен посылать отчет о каждой группе

O Если на запрос не присылаются отчеты => членов

групп больше нет

IGMP V3 Позволяет:

O В одном пакете запросить несколько групп

O В одном пакете отчитаться о нескольких группах

**Билет 8. Управляющий протокол ICMP. Сигнализирующие сообщения.**

*O*Протокол сетевого уровня

*O*Предназначен для решения задач:

*O*Управления

*O*Нотификации об ошибках и проблемах

*O*Тестирования и мониторинга

*O*Описан в стандарте RFC792

*O*Инкапсулируется в IP

Internet Control Message Protocol (ICMP)

Type - Восьмибитовое поле, содержащее тип ICMP пакета (Echo request, Echo reply …).

Code - Восьмибитовое поле, содержащее номер функции соответствубщего типа сообщения. Если тип имеет только одну функцию, то значение поля равно 0.

Checksum - Контрольная сумма.

Type Specific Data - Дополнительные данные, свои для каждого типа пакета.

Служит для общения маршрутизатора с хостом, отправляющим или посылающим данные контрольными сообщениями и сообщениями об ошибках.

Использует для передачи IP и является его составной частью.

Если TCP/IP хост посылает датаграммы другому хосту так часто, что маршрутизаторы не успевают их пересылать, то маршрутизаторы могут посылать сообщения, указывающие хосту замедлить темп передачи датаграмм. Но, если Windows NT компьютер используется как маршрутизатор и не можит отправлять датаграммы с такой частотой, с которой они приходят, то он отбрасывает часть датаграмм не посылая ICMP Source Quench сообщения.

Типы сообщений и их коды:

Получатель недостижим (3). Код 0-12 – причина недоступности. Последнее поле содержит первую часть

пакета, который невозможно доставить.

0 - сеть недостижима 1 - узел недостижим 2 - протокол недостижим 3 - порт недостижим 4 - требуется фрагментация 5 - ошибка маршрутизации от источника 6 – сеть назначения неизвестна 7 - узел назначения неизвестен 8 – отправитель изолирован 9 – взаимодействие с сетью назначения административно

запрещено 10 - взаимодействие с узлом назначения административно запрещено 11 – Сеть недостижима из-за класса обслуживания 12 – Узел недостижим из-за класса обслуживания.

Превышено время (11).

Код 0-1 – причина превышения времени Последнее поле содержит первую часть пакета, фрагмент которого не дошел.

Коды:

0 – превышено значение счетчика времени

жизни 1 – Превышено время ожидания фрагмента при сборке.

Ошибка параметра (12) Указатель –номер байта в IP-пакете, в котором обнаружена ошибка Последнее поле содержит первую часть пакета, в которым обнаружена ошибка.

Изменение маршрута (5).

Запрос эха (8)

**Билет 9. Управляющий протокол ICMP. Управляющие и тестовые сообщения.**

Internet Control Message Protocol (ICMP)

Type - Восьмибитовое поле, содержащее тип ICMP пакета (Echo request, Echo reply …).

Code - Восьмибитовое поле, содержащее номер функции соответствубщего типа сообщения. Если тип имеет только одну функцию, то значение поля равно 0.

Checksum - Контрольная сумма.

Type Specific Data - Дополнительные данные, свои для каждого типа пакета.

Служит для общения маршрутизатора с хостом, отправляющим или посылающим данные контрольными сообщениями и сообщениями об ошибках.

Использует для передачи IP и является его составной частью.

Если TCP/IP хост посылает датаграммы другому хосту так часто, что маршрутизаторы не успевают их пересылать, то маршрутизаторы могут посылать сообщения, указывающие хосту замедлить темп передачи датаграмм. Но, если Windows NT компьютер используется как маршрутизатор и не можит отправлять датаграммы с такой частотой, с которой они приходят, то он отбрасывает часть датаграмм не посылая ICMP Source Quench сообщения.

Подавление источника (4)

Последнее поле содержит первую часть пакета, поток которого необходимо замедлить.

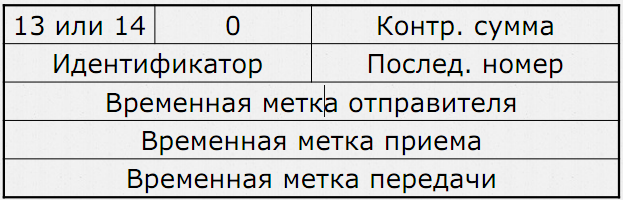
Изменение маршрута (5)

Последнее поле содержит первую часть пакета, маршрутизируемого нерационально

Код (0-3) – тип переназначения маршрута

O 0 – переназначения маршрута для сети 1 – переназначения маршрута для узла 2 – переназначения маршрута для типа сервиса и сети 3 – переназначения маршрута для типа сервиса и узла.

Запрос временной метки (13) и ответ на запрос временной метки (14).



O Идентификатор – номер потока сообщений Последовательный номер – номер пакета в потоке O Временная метка отправителя заполняется источником O Временная метка приема фиксируется при получении запроса приемником O Временная метка передачи заполняется приемником

Запрос маски адреса (17) и ответ на запрос маски адреса (18)

O Идентификатор – номер потока сообщений O Последовательный номер – номер пакета в потоке O Маска – записанная маска адреса приемника

**Билет 10. Адресация приложений. Понятие портов. Транспортный протокол UDP.**

В протоколах TCP и UDP (семейства TCP/IP) порт — идентифицируемый номером системный ресурс, выделяемый приложению, выполняемому на некотором сетевом хосте, для связи с приложениями, выполняемыми на других сетевых хостах (в том числе c другими приложениями на этом же хосте).

Порты TCP не пересекаются с портами UDP. То есть, порт 1234 протокола TCP не будет мешать обмену по UDP через порт 1234.

Ряд номеров портов стандартизован (см. Список портов TCP и UDP). Список поддерживается некоммерческой организацией IANA.

Список зарезервированных портов:

DISCARD: Discard port (RFC 863) FTP: 21 для команд, 20 для данных SSH: 22 (remote access) telnet: 23 (remote access) SMTP: 25 DNS: 53 (UDP) DHCP: 67, 68/UDP HTTP: 80, 8080 POP3: 110 IMAP: 143

UDP

UDP (англ. User Datagram Protocol — протокол пользовательских дейтаграмм) — это транспортный протокол для передачи данных в сетях IP без установления соединения. Он является одним из самых простых протоколов транспортного уровня модели OSI. Его IP-идентификатор — 0x11.

В отличие от TCP, UDP не подтверждает доставку данных, не заботится о корректном порядке доставки и не делает повторов. Поэтому аббревиатуру UDP иногда расшифровывают как Unreliable Datagram Protocol (протокол ненадёжных датаграмм). Зато отсутствие соединения, дополнительного трафика и возможность широковещательных рассылок делают его удобным для применений, где малы потери, в массовых рассылках локальной подсети, в медиапротоколах и т.п.

Формат пакета:



UDP-заголовок не содержит информации об адресе отправителя и получателя, поэтому даже при совпадении порта получателя нельзя с точностью сказать, что сообщение пришло в нужное место. Для проверки того, что UDP-сообщение достигло пункта своего назначения, используется дополнительный псевдозаголовок

Псевдозаголовок не включается в UDP-сообщение. Он используется для расчета контрольной суммы перед отправлением сообщения и при его получении.

Перед расчетом контрольной суммы UDP-сообщение дополняется в конце нулевыми битами до длины, кратной 16 битам (псевдозаголовок и добавочные нулевые биты не отправляются вместе с сообщением). Поле контрольной суммы в UDP-заголовке во время расчета контрольной суммы отправляемого сообщения принимается нулевым.

Для расчета контрольной суммы псевдозаголовок и UDP-сообщение разбивается на слова (1 слово = 2 байта (октета) = 16 бит). Затем рассчитывается поразрядное дополнение до единицы суммы всех слов с поразрядным дополнением. Результат записывается в соответствующее поле в UDP-заголовке.

**Билет 11. Транспортный протокол TCP. Формат пакета TCP.**

Transmission Control Protocol (TCP) (протокол управления передачей) — один из основных сетевых протоколов Интернета, предназначенный для управления передачей данных в сетях и подсетях TCP/IP.

Выполняет функции протокола транспортного уровня модели OSI.

TCP — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в достоверности получаемых данных, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета (см. также T/TCP). В отличие от UDP гарантирует целостность передаваемых данных и уведомление отправителя о результатах передачи.

Формат пакета:



Порт источника

Порт источника идентифицирует приложение клиента, с которого отправлены пакеты. По возвращении данные передаются клиенту на основании номера порта источника.

Порт назначения

Порт назначения идентифицирует порт, на который отправлен пакет.

TCP-порты

Существует набор служб (использующих для передачи данных TCP), за которыми закреплены определенные порты.

Номер последовательности

Номер последовательности выполняет две задачи:

1. Если установлен флаг SYN, то это начальное значение номера последовательности — ISN (Initial Sequence Number), и первый байт данных, которые будут переданы в следующем пакете, будет иметь номер последовательности, равный ISN + 1. 2. В противном случае, если SYN не установлен, первый байт данных, передаваемый в данном пакете, имеет этот номер последовательности.

Номер подтверждения

Если установлен флаг ACK, то это поле содержит номер последовательности, ожидаемый получателем в следующий раз. Помечает этот сегмент как подтверждение получения.

Смещение данных

Это поле определяет размер заголовка пакета TCP в 4-байтных (4-октетных) словах. Минимальный размер составляет 5 слов, а максимальный — 15, что составляет 20 и 60 байт соответственно. Смещение считается от начала заголовка TCP.

Флаги (управляющие биты)

Это поле содержит 6 битовых флагов:

URG — Поле «Указатель важности» задействовано

ACK — Поле «Номер подтверждения» задействовано

PSH — инструктирует получателя протолкнуть данные, накопившиеся в приемном буфере, в приложение пользователя

RST — Оборвать соединения, сбросить буфер

SYN — Синхронизация номеров последовательности

FIN— флаг, будучи установлен, указывает на завершение соединения.

Окно

В этом поле содержится число, определяющее в байтах размер данных, которые получатель готов принять.

**Билет 12. Транспортный протокол TCP. Алгоритм функционирования.**

Протокол TCP. Установление

соединения

O A => B:

O SYN, NseqA

O B=> A:

O ACK, NackB=NseqA+1

O SYN, NseqB

O A => B:

O ACK, NackA=NseqB+1

Процесс начала сеанса TCP - обозначаемое как "рукопожатие" (handshake), состоит из 3 шагов.

1. Клиент, который намеревается установить соединение, посылает серверу сегмент с номером последовательности и флагом SYN.

* Сервер получает сегмент, запоминает номер последовательности и пытается создать сокет (буферы и управляющие структуры памяти) для обслуживания нового клиента.
  + В случае успеха сервер посылает клиенту сегмент с номером последовательности и флагами SYN и ACK, и переходит в состояние SYN-RECEIVED.
  + В случае неудачи сервер посылает клиенту сегмент с флагом RST.

2. Если клиент получает сегмент с флагом SYN, то он запоминает номер последовательности и посылает сегмент с флагом ACK.

* Если он одновременно получает и флаг ACK (что обычно и происходит), то он переходит в состояние ESTABLISHED.
* Если клиент получает сегмент с флагом RST, то он прекращает попытки соединиться.
* Если клиент не получает ответа в течение 10 секунд, то он повторяет процесс соединения заново.

3. Если сервер в состоянии SYN-RECEIVED получает сегмент с флагом ACK, то он переходит в состояние ESTABLISHED.

* В противном случае после тайм-аута он закрывает сокет и переходит в состояние CLOSED.

Процесс называется "трехэтапным согласованием" ("three way handshake"), так как несмотря на то что возможен процесс установления соединения с использованием 4 сегментов (SYN в сторону сервера, ACK в сторону клиента, SYN в сторону клиента, ACK в сторону сервера), на практике для экономии времени используется 3 сегмента.

Протокол TCP. Разрыв

соединения

O A => B:

O FIN

O B=> A:

O ACK

O B=> A:

O FIN

O A => B:

O ACK

Завершение соединения можно рассмотреть в три этапа:

1. Посылка серверу от клиента флагов FIN и ACK на завершение соединения.
2. Сервер посылает клиенту флаги ответа ACK , FIN, что соединение закрыто.
3. После получения этих флагов клиент закрывает соединение и в подтверждение отправляет серверу ACK , что соединение закрыто.

**Билет 13. Протокол SCTP.**

SCTP (англ. Stream Control Transmission Protocol — «протокол передачи с управлением потоком»), протокол транспортного уровня в компьютерных сетях, появившийся в 2000 году в IETF. RFC 4960 описывает этот протокол, а RFC 3286 содержит техническое вступление к нему.

Как и любой другой протокол передачи данных транспортного уровня, SCTP работает аналогично TCP или UDP [1]. Но на самом деле SCTP имеет в арсенале широкий спектр приятных нововведений, таких как многопоточность, защита от SYN-flood атак, синхронное соединение между двумя хостами по двум и более независимым физическим каналам (multi-homing).

O Основная идея: Объединить достоинства UDP, TCP; Исправить недостатки UDP, TCP

O Пакет состоит из Заголовка; Нескольких субпакетов (chunk)

O Каждый субпакет: Имеет свой заголовок; Данные

Формат пакета::



SCTP пакеты имеют более простую структуру, чем пакеты TCP. Каждый пакет состоит из двух основных разделов:

Общий заголовок, который занимает первые 12 байт (выделены синим цветом)

Блоки данных, которые занимают оставшуюся часть пакета.

O Тип субпакета 0..255: 0 – несет полезные данные; >0 – служебная информация

O Флаги – служебные разряды, определяются типом субпакета

O Длина субпакета – 0..65535 – общая длина субпакета с заголовками

Сравнение с TCP и UDP.



**Билет 14. Маршрутизация в TCP/IP. Маршрутизаторы и шлюзы. Процесс доставки пакетов в сети.**

Общими словами маршрутизацию можно описать как процесс передачи пакетов между соединенными сетями. В TCP/IP-сетях маршрутизация является частью протокола IP (Internet Protocol) и используется в сочетании с другими службами сетевых протоколов для обеспечения передачи данных между узлами, расположенными в разных сегментах более крупной TCP/IP-сети.

IP — это своего рода «почтовая система» протокола TCP/IP, выполняющая сортировку и доставку IP-данных. Каждый входящий или исходящий пакет называется IP-датаграммой. Датаграмма IP содержит два IP-адреса: адрес источника (отправляющего узла) и адрес назначения (принимающего узла). В отличие от аппаратных адресов, IP-адреса в датаграмме в процессе передачи ее по TCP/IP-сети остаются постоянными.

Маршрутизация является основной функцией IP. Обмен IP-датаграммами и их обработка на каждом узле выполняются протоколом IP, работающим на межсетевом уровне.

Над этим уровнем транспортные службы узла-источника передают данные уровню IP в виде TCP-сегментов или UDP-сообщений.

Маршрутиза́тор - сетевое устройство, пересылающее пакеты данных между различными сегментами сети и принимающее решения на основании информации о топологии сети и определённых правил, заданных администратором.

Маршрутизаторы делятся на программные и аппаратные. Маршрутизатор работает на более высоком «сетевом» уровне 3 сетевой модели OSI, нежели коммутатор и сетевой мост.

Аппаратные маршрутизаторы

Особенности: O Поддержка различных канальных сред

O Наличие нескольких сетевых интерфейсов O Высокая производительность O Высокая надежность O Хорошая защищенность

Дополнительные функции: O Фильтрация O Трансляция адресов O Сбор статистики

Обычно высокая стоимость

Производители: O CISCO O Intel

Программно-аппаратные

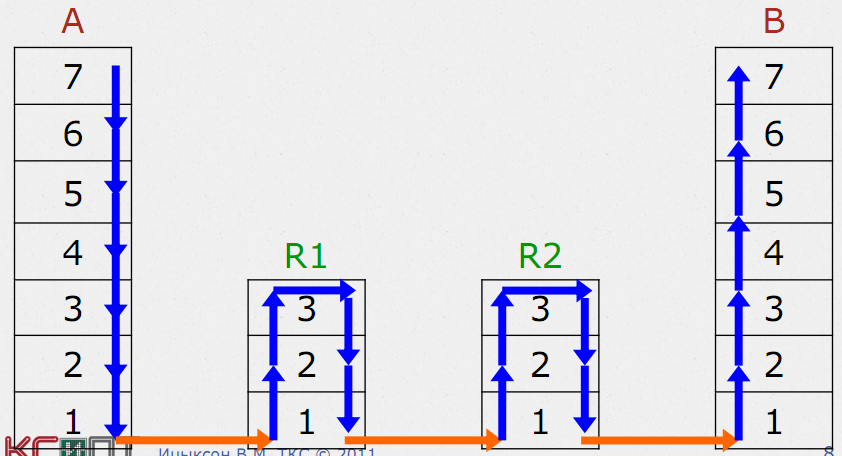
маршрутизаторы

Реализуются функциями ОС общего

назначения

Характеризуются O Невысокой производительностью O Невысокой стоимостью O Могут совмещать функции с обычными функциями ОС

Процесс маршрутизации:



**Билет 15. Статическая маршрутизация. Таблицы маршрутизации.**

Стати́ческая маршрутиза́ция - вид маршрутизации, при котором маршруты указываются в явном виде при конфигурации маршрутизатора. Вся маршрутизация при этом происходит без участия каких-либо протоколов маршрутизации.

При задании статического маршрута указывается:

Адрес сети (на которую маршрутизируется трафик), маска сети

Адрес шлюза (узла), который отвечает за дальнейшую маршрутизацию (или подключен к маршрутизируемой сети напрямую)

(опционально) метрика (иногда именуется также "ценой") маршрута.



Достоинства: Лёгкость отладки и конфигурирования в малых сетях. Отсутствие дополнительных накладных расходов (из-за отсутствия протоколов маршрутизации)

Мгновенная готовность (не требуется интервал для конфигурирования/подстройки). Низкая нагрузка на процессор маршрутизатора. Предсказуемость в каждый момент времени

Недостатки: Очень плохое масштабирование. Низкая устойчивость к повреждениям линий связи. Отсутствие динамического балансирования нагрузки. Необходимость в ведении отдельной документации к маршрутам.

В реальных условиях статическая маршрутизация используется в условиях наличия шлюза по умолчанию (узла, обладающего связностью с остальными узлами) и 1-2 сетями.

**Билет 16. Маршрутизация. Виды маршрутизации. Алгоритм выбора маршрута в РС.**

Общими словами маршрутизацию можно описать как процесс передачи пакетов между соединенными сетями. В TCP/IP-сетях маршрутизация является частью протокола IP (Internet Protocol) и используется в сочетании с другими службами сетевых протоколов для обеспечения передачи данных между узлами, расположенными в разных сегментах более крупной TCP/IP-сети.

IP — это своего рода «почтовая система» протокола TCP/IP, выполняющая сортировку и доставку IP-данных. Каждый входящий или исходящий пакет называется IP-датаграммой. Датаграмма IP содержит два IP-адреса: адрес источника (отправляющего узла) и адрес назначения (принимающего узла). В отличие от аппаратных адресов, IP-адреса в датаграмме в процессе передачи ее по TCP/IP-сети остаются постоянными.

Маршрутизация является основной функцией IP. Обмен IP-датаграммами и их обработка на каждом узле выполняются протоколом IP, работающим на межсетевом уровне.

Над этим уровнем транспортные службы узла-источника передают данные уровню IP в виде TCP-сегментов или UDP-сообщений.

Виды маршрутизации

По адаптивности:O Статическая маршрутизация O Динамическая маршрутизация O Маршрутизация «от источника»

По месту маршрутных вычислений: O Централизованные O Децентрализованные

По требуемой информации: O Локальные O Глобальные O Смешанные

Стати́ческая маршрутиза́ция - вид маршрутизации, при котором маршруты указываются в явном виде при конфигурации маршрутизатора. Вся маршрутизация при этом происходит без участия каких-либо протоколов маршрутизации.

Динамическая маршрутизация позволяет учитывать:

Изменяющиеся параметры в сети: O Топология сети

O Появление новых узлов O Появление новых каналов

Каналы связи: O Выход из строя канала O Ввод в строй канала

Узлы сети: O Выход из строя маршрутизатора O Ввод в строй маршрутизатора

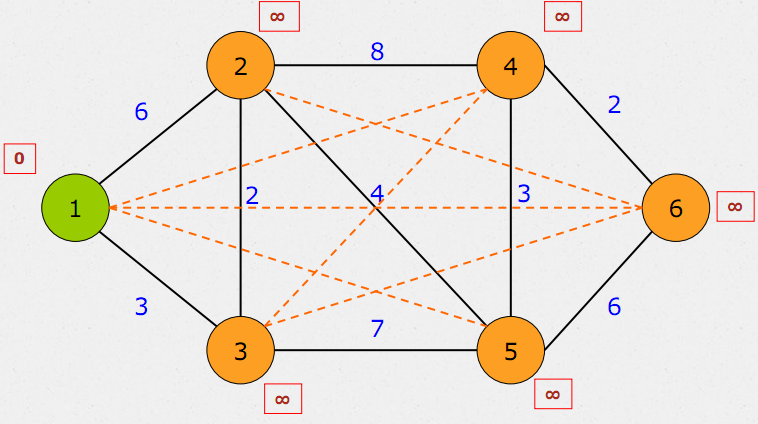
Изменение нагрузки в сети.

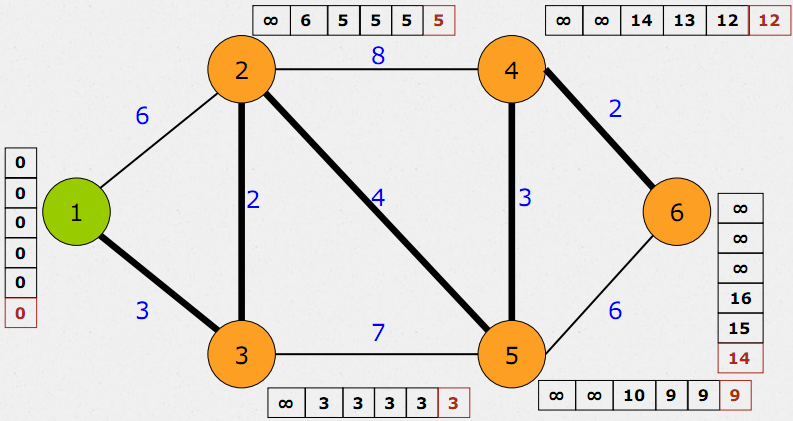
В данном виде маршрутизации маршрут меняется программно при помощи различных алгоритмов и протоколов маршрутизации.

Задача оптимальной маршрутизации – найти оптимальный путь для пакета в данный момент времени.

**Билет 17. Динамическая маршрутизация. Алгоритм Беллмана-Форда поиска кратчайшего пути**

Алгоритм ищет кратчайший путь от заданной вершины до всех остальных в графе с весами на ребрах. Идея в том, что мы храним в массиве текущее найденное расстояние до каждой из вершин и на каждом шаге алгоритма обновляем это расстояние, если нашли ребро которое улучшает результат. Таким образом, после каждого шага k будет получаться массив кратчайших расстояний с длиной пути не более k ребер. Вот и все, делаем число шагов, равное количеству вершин в графе.





Общее число операций: O N – число вершин O N-1 – число шагов

достоинства алгоритма: Хорошо распараллеливается

Просто реализуется; Не требует ресурсов памяти

Требуется информация только о соседней вершин

Часто заканчивается раньше N-1 итерации

Недостатки алгоритма:

В худшем случае количество операций - ~ N3

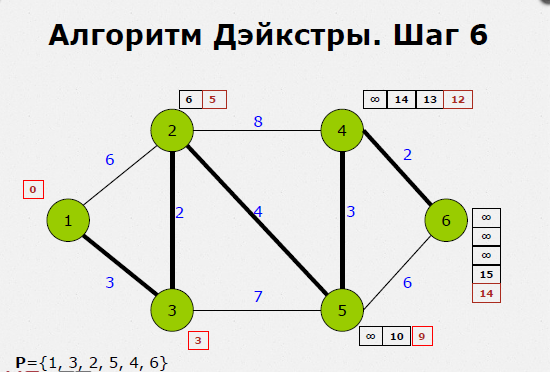
**Билет 18. Динамическая маршрутизация. Алгоритм Дэйкстры поиск кратчайшего пути.**

Алгори́тм Де́йкстры - алгоритм на графах, изобретённый нидерландским ученым Э. Дейкстрой в 1959 году. Находит кратчайшее расстояние от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса.

Каждой вершине из V сопоставим метку — минимальное известное расстояние от этой вершины до a. Алгоритм работает пошагово — на каждом шаге он «посещает» одну вершину и пытается уменьшать метки. Работа алгоритма завершается, когда все вершины посещены.

Инициализация. Метка самой вершины a полагается равной 0, метки остальных вершин — бесконечности. Это отражает то, что расстояния от a до других вершин пока неизвестны. Все вершины графа помечаются как непосещённые.

Шаг алгоритма. Если все вершины посещены, алгоритм завершается. В противном случае, из ещё не посещённых вершин выбирается вершина u, имеющая минимальную метку. Мы рассматриваем всевозможные маршруты, в которых u является предпоследним пунктом. Вершины, в которые ведут рёбра из u, назовем соседями этой вершины. Для каждого соседа вершины u, кроме отмеченных как посещённые, рассмотрим новую длину пути, равную сумме значений текущей метки u и длины ребра, соединяющего u с этим соседом. Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины.



Достоинства алгоритма: O Высокая скорость (~N2)

Недостатки алгоритма: O Плохо распараллеливается O Требуется иметь информацию о топологии всей сети O Требует существенных ресурсов памяти (~N2)

**Билет 19. Автономные системы. Внутренние и внешние протоколы маршрутизации. Характеристики протоколов маршрутизации**.

Автономная система (AS) в интернете — это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с Интернетом.

Протоколы маршрутизации внутри AS O Внутренние протоколы маршрутизации O IGP – Interior Gateway Protocol

Протоколы маршрутизации между AS O Внешние протоколы маршрутизации O EGP – Exterior Gateway Protocol

Типы

Автономные системы можно сгруппировать в 3 категории, в зависимости от их соединений и режима работы.

Многоинтерфейсная (multihomed) AS — это AS, которая имеет соединения с более чем одним Интернет-провайдером. Это позволяет данной AS оставаться подключенной к Интернету в случае выхода из строя соединения с одним из Интернет-провайдеров. Кроме того, этот тип AS не разрешает транзитный трафик от одного Интернет-провайдера к другому.

Ограниченная (stub) AS — это AS, имеющая единственное подключение к одной внешней автономной системе. Это расценивается как бесполезное использование номера AS, так как сеть размещается полностью под одним Интернет-провайдером и, следовательно, не нуждается в уникальной идентификации.

Транзитная (transit) AS — это AS, которая пропускает через себя транзитный трафик сетей, подключенных к ней. Таким образом, сеть A может использовать транзитную AS для связи с сетью B.

Протоколы маршрутизации:

Протокол маршрутизации — сетевой протокол, используемый маршрутизаторами для определения возможных маршрутов следования данных в составной компьютерной сети. Применение протокола маршрутизации позволяет избежать ручного ввода всех допустимых маршрутов, что, в свою очередь, снижает количество ошибок, обеспечивает согласованность действий всех маршрутизаторов в сети и облегчает труд администраторов.

Характеристики протоколов маршрутизации

O Название O Стандартизирующие документы O Алгоритм поиска маршрута O Метрика протокола O Сходимость O Избежание петель маршрутизации O Загрузка сети O Ресурсоѐмкость O Поддержка нескольких маршрутов на сеть O Аутентификации O Ограничения применения O Конфигурирование O Поддержка в маршрутизаторах O Достоинства O Недостатки

Метрики маршрутов

Метрика маршрута может зависеть от: O Числа промежуточных маршрутизаторов O Пропускной способности канала связи O Задержек в канале связи

O Надежности канала связи O Загрузки канала связи.

Сходимость протокола – способность протокола оперативно реагировать на изменения в сети и приводить маршрутизаторы к соответствующее состояние

Время сходимости – время за которое маршрутные таблицы переходят в состояние, адекватное изменившейся ситуации сети.

**Билет 20. Маршрутизация в сетях TCP/IP. Протокол маршрутизации RIP.**

Протокол маршрутной информации (англ. Routing Information Protocol) — один из самых простых протоколов маршрутизации. Применяется в небольших компьютерных сетях, позволяет маршрутизаторам динамически обновлять маршрутную информацию, получая ее от соседних маршрутизаторов.

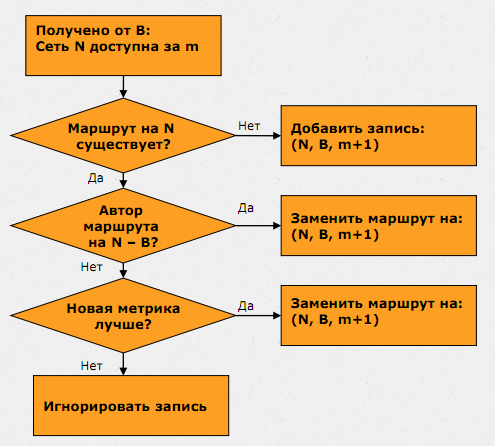
Алгоритм маршрутизации RIP (алгоритм Беллмана — Форда) был впервые разработан в 1969 году, как основной для сети ARPANET.

RIP. Метрика O Целое число из диапазона 0..15 O Измеряется числом промежуточных маршрутизаторов до сети назначения O Для непосредственно подсоединенных сетей – значение «0» O Значение «16» – сеть недоступна.

Не зависит от :O Задержек O Пропускной способности O Надежности O Загрузки

Каждый RIP-маршрутизатор по умолчанию вещает в сеть свою полную таблицу маршрутизации раз в 30 секунд, довольно сильно нагружая низкоскоростные линии связи. RIP работает на прикладном уровне стека TCP/IP, используя UDP порт 520.

Коррекция маршрута (таблицы):



Вместе с обновлением активируется таймер Tm

Состояния маршрутных записей

Tm ≤ 180c: маршрут в рабочем состоянии

O Используется для маршрутизации пакетов

O Рассылается соседним маршрутизаторам

180c < Tm ≤ 300c  маршрут устарел

O Используется для маршрутизации пакетов

O Не рассылается соседним маршрутизаторам

O Tm > 300c: маршрут не действителен

O Удаляется из таблицы маршрутизации

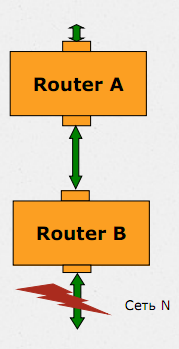
При приходе очередного обновления таймер сбрасывается и маршрут переводится в рабочее состояние

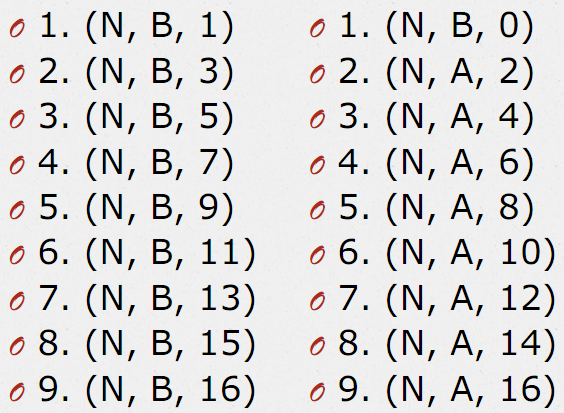
Петли маршрутизации

O В течение нескольких минут маршруты находятся в некорректном состоянии O Сходимость протокола – плохая O Необходимы способы преодоления петель Маршрутизации

**Билет 21. Маршрутизация в сетях TCP/IP. Методы борьбы с петлями маршрутизации в протоколе RIP. Протокол маршрутизации RIP2**

Пример появления петли маршрутизации:





Петли маршрутизации относятся к недостаткам RIP протокола.

В течение нескольких минут маршруты находятся в некорректном состоянии.

Сходимость протокола – плохая.

Необходимы способы преодоления петель маршрутизации.

Для того, чтобы это преодолеть, были разработаны различные методы борьбы с петлями:

Split Horizon - расщепление горизонта

O Обновления маршрута не посылаются на интерфейс, из которого этот маршрут получен O Блокируется обратная круговая передача маршрутной информации

Poison Reverse – обратный яд

O Обновления маршрута посылаются на интерфейс, из которого этот маршрут получен, но с метрикой 16 O Во время штатного работы такие обновления – игнорируются O В случае сбоя – сразу определяется

недоступность маршрута.

Triggered Updates – мгновенные обновления

O В случае изменения метрики маршрута маршрутизатор посылает обновления немедленно, не дожидаясь 30-секундного интервала O Позволяет улучшить сходимость протокола O Лавинообразно увеличивается трафик

Hold Down

O Кратковременное прекращение приема обновлений

маршрута после получения его обновления от автора с метрикой 16 O В течение 120 с маршрутизатор не принимает обновления этого маршрута, пережидая переходные процессы в сети.

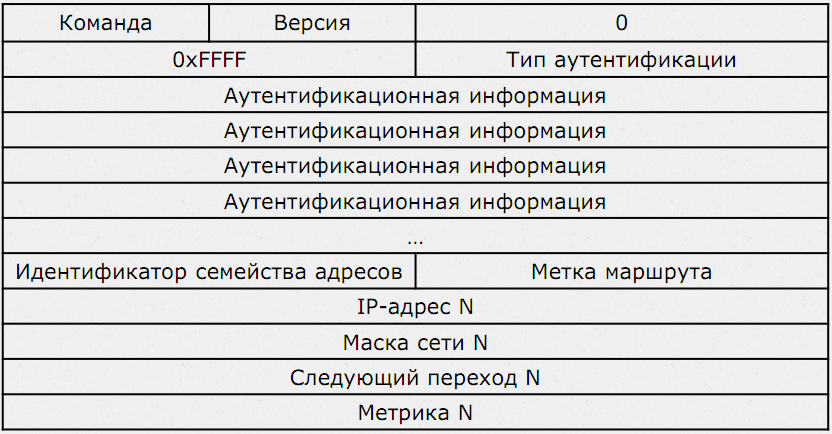
На практике – используют несколько технологий ! Никакие технологии не спасают от петли маршрутизации, включающей более 2 узлов !

Протокол маршрутизации RIP-II

Стандарт RFC 1721, 1994 год

Нововведения:O Аутентификация O Маска сети O Групповая адресация O Метки маршрута O Ссылка на следующий маршрутизатор

O Для адресации соседних маршрутизаторов используется групповая адресация O Адрес рассылки обновлений – 224.0.0.9 O Для совместимости оставлена возможность использовать широковещание O Пакет RIP-II использует UDP, порт 520 O В один пакет – до 24 маршрутных записей.

Формат пакета:

Идентификатор семейства адресов

O 0xFFFF – для аутентификации

Аутентификационная информация

O Используется для проверки подлинности пакета

Маска сети

O Используется для передачи маршрутов на подсети

Следующий переход

O Для исключения лишних шагов в маршрутизации

O Для возможности использовать информацию из других источников

Метка маршрута

O Для упрощения взаимодействия с протоколами EGP

Формат пакета совместим с RIP-1.

**Билет 22. Маршрутизация в сетях TCP/IP. Протоколы маршрутизации IGRP**.

Протокол IGRP (англ. Interior Gateway Routing Protocol) — протокол маршрутизации, разработанный фирмой Cisco, для своих многопротокольных маршрутизаторов в середине 80-х годов для маршрутизации в пределах автономной системы (AS), имеющей сложную топологию и разные характеристики полосы пропускания и задержки. IGRP является протоколом внутренних роутеров (IGP) с вектором расстояния.

Для выбора маршрута в IGRP используется комбинация показателей, таких как задержка сети, полоса пропускания, надежность и загруженность сети. Весовой коэффициент этих показателей может выбираться автоматически или задаваться администратором сети. Для надежности и загруженности сети это значения от 1 до 255, полоса пропускания — от 1200 бит/сек до 10 Гбит/сек, задержка может принимать значение до 24-го порядка.

Для повышения стабильности работы IGRP предусматривает такие механизмы, как удержание изменений, расщепленный горизонт (split-horizon) и корректировка отмены.

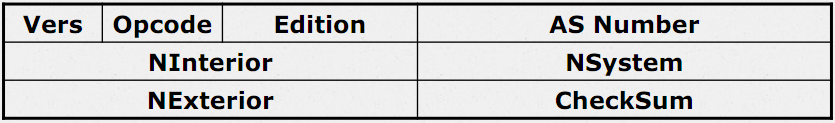
Удержание изменений Когда в сеть поступает информация об изменениях маршрутов (например, об обрыве связи) от одного из роутеров, то изменения в таблицы маршрутизации поступают не мгновенно, а в течение некоторого времени. В этот период роутер, ещё не получивший информацию об изменениях, может продолжать распространять информацию об уже несуществующем маршруте.

Расщепленный горизонт (split-horizon) Суть этого механизма состоит в том, для предотвращения зацикливания маршрутов между соседними роутерами, информация об изменении маршрута не должна распространяться в направлении того роутера, от которого она пришла.

Корректировка отмены маршрута (route-poisoning) — это принудительное удаление маршрута и перевод в состояние удержания, применяется для борьбы с маршрутными петлями.

Таймеры Таймер корректировки определяет, как часто должны отправляться сообщения о корректировке маршрутов. Таймер недействующих маршрутов определяет, сколько времени должен ожидать роутер при отсутствии сообщений о корректировке какого-нибудь конкретного маршрута, прежде чем объявить этот маршрут недействующим.

Формат заголовка:



Vers – версия протокола IGRP (1)

Opcode – код операции O 1 – обновление O 2 - запрос

Edition – последовательный номер обновления

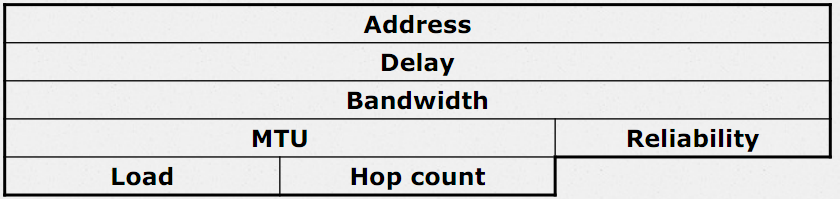
AS Number – номер автономной системы

NInterior, NSystem, NExterior – число маршрутных

записей в обновлении

CheckSum – контрольная сумма

Формат строки обновления:



Address – старшие 3 октета адреса

Delay – задержка. D = Delay \* 10мкс

Bandwidth – полоса пропускания. B = 1010/Bandwidth

MTU – размер MTU в байтах

Reliability – надежность. R = Reliability/255

Load – загрузка. L = Load/255

Hop count – число пересылок

**Билет 23. Маршрутизация в сетях TCP/IP. Протокол маршрутизации OSPF.**

OSPF (англ. Open Shortest Path First) — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала (link-state technology) и использующий для нахождения кратчайшего пути Алгоритм Дейкстры (Dijkstra’s algorithm).

OSPF имеет следующие приемущества:

Высокая скорость сходимости по сравнению с дистанционно-векторными протоколами маршрутизации;

Поддержка сетевых масок переменной длины (VLSM);

Оптимальное использование пропускной способности (т. к. строится минимальный остовный граф по алгоритму Дейкстры);

Задается числом 0..65535

Метрика определяется как количество секунд, требуемое для передачи 100 Мбит через физическую среду данной сети O 10Base-T Ethernet – 10 O 56 кбит/с – 1785

O Метрика канала со скоростью передачи данных 100 Мбит/с и выше – 1

Для каждого канала связи метрика может задаваться администратором.

Рассылаются пакеты LSA – Link State Advertisement

O Содержат информацию о подключенных

каналах и их состояниях (метриках)

Рассылка осуществляется при изменении состояния какого-либо канала. В результате лавинного обмена все узлы получают информацию о всех каналах сети.

Все маршрутизаторы строят LSD – Link State Database

Условия корректного функционирования протокола – все LSD идентичны

Каждый маршрутизатор по LSD строит LST – Link State Tree. O В качестве корня использует себя.

Причины рассылки LSA:

O Изменилось состояние интерфейса O Произошли изменения в маршрутизаторе сети O Произошло изменение состояния одного из соседних маршрутизаторов O Изменилось состояние одного из внутренних маршрутов O Изменение состояния межзонного маршрута O Появление нового маршрутизатора, подключенного к сети O Возникли изменения одного из внешних маршрутов O Маршрутизатор перестал быть пограничным для данной as (например, перезагрузился). O Возраст маршрута достиг предельного значения (30 минут) O И т.п.

Таблица маршрутизации состоит из:

IP-адрес места назначения и маска

Тип места назначения (сеть, граничный маршрутизатор и т.д.)

Тип сервиса (TOS)

Домен маршрутизации

Тип пути:

O Внутренний O Межобластной O Внешний, ведущий к AS

Метрика

Следующий маршрутизатор

Объявляющий маршрутизатор (используется для межобластных обменов и для связей автономных систем друг с другом).

В OSPF можно назначить несколько видов маршрутизаторов:

DR (Designated Router) – назначенный маршрутизатор

O Используется, как основной адрес передачи LSA маршрутизаторами O Рассылает обновления всем маршрутизаторам

BDR (Backup Designated Router) – запасной назначенный маршрутизатор

O Заменяет DR в случае выхода DR из строя.

Достоинства протокола OSPF

O Гибкая метрика O Быстрая сходимость O Низкая загрузка каналов служебным трафиком O Отсутствие петель маршрутизации O Поддержка доменов маршрутизации O Поддержка различных маршрутов для разных типов обслуживания.

Недостатки протокола OSPF

O Высокие требования к ресурсам: O Быстродействие

O Объемы памяти для хранения LSD, LST O Высокая сложность конфигурирования.

**Билет 24. Маршрутизация в сетях TCP/IP. Протоколы маршрутизации EIGRP.**

EIGRP (англ. Enhanced Interior Gateway Routing Protocol) — протокол маршрутизации, разработанный фирмой Cisco на основе протокола IGRP той же фирмы. Релиз протокола состоялся в 1994 году. EIGRP использует механизм DUAL для выбора наиболее короткого маршрута.

Более ранний и практически не используемый ныне протокол IGRP был создан как альтернатива протоколу RIP (до того, как был разработан OSPF). После появления OSPF, Cisco представила EIGRP — переработанный и улучшенный вариант IGRP, свободный от основного недостатка дистанционно-векторных протоколов — особых ситуаций с зацикливанием маршрутов — благодаря специальному алгоритму распространения информации об изменениях в топологии сети. Несмотря на то что в общем случае протоколы состояния связей (OSPF) отрабатывают изменения в топологии сети быстрее, чем EIGRP, а также OSPF имеет ряд дополнительных возможностей, EIGRP более прост в реализации и менее требователен к вычислительным ресурсам маршрутизатора.

Основные особенности:

O Механизм обнаружения соседей O Посылка обновлений таблиц O Вычисление вероятных заместителей O Алгоритм активного поиска DUAL.

Метрика EIGRP соответствует метрике IGRP.

Обнаружение соседей:

Посылается пакет Hello

O 1 раз в 5 сек. для быстрых сетей O 1 раз в 60 сек. для медленных сетей

По получаемым сообщениями делается вывод об имеющихся соседях.

На основе полученных данных составляется таблица топологии, содержащая поля:

Содержит поля :

O Минимальная полоса пропускания в пути O Общая задержка пути O Надежность пути O Загрузка пути O Минимум MTU на пути O Текущая дистанция O Отчетная дистанция O Источник маршрута

В качестве маршрута выбирается путь с наименьшей текущей дистанцией.

Алгоритм DUAL:

DUAL – Diffuse Update Algorithm

Используется для активного поиска маршрута в случает удаления его из таблицы маршрутизации

Станция, потерявшая маршрут посылает

запрос соседям

Если сосед имеет вероятного заместителя

посылает ответ

Если сосед не имеет – сам начинает процедуру активного поиска

O Использует все интерфейсы, кроме входящего.

В стандарте EIGRP содержится 5 видов пакетов:

O Приветствие (Hello) O Обновление (Update) O Запрос (Query) O Ответ (Reply) O Запрос информации (Request).

Достоинства протокола EIGRP

O Очень малое использование сети в нормальном состоянии (только передача Hello) O При изменениях посылаются только изменения маршрутных таблиц O Низкое время сходимости O Обход петель маршрутизации O Поддержка маски сети O Простота реализации.

Недостатки протокола EIGRP

O Несколько худшая по сравнению с OSPF сходимость.

**Билет 25. Именование ресурсов в сетях TCP/IP. Доменная система имен**.

DNS (англ. Domain Name System — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене (SRV-запись).

Распределённая база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определённому протоколу.

Основой DNS является представление об иерархической структуре доменного имени и зонах. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу (с административной точки зрения — другой организации или человеку), что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

DNS обладает следующими характеристиками:

Распределённость администрирования. Ответственность за разные части иерархической структуры несут разные люди или организации.

Распределённость хранения информации. Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности и (возможно) адреса корневых DNS-серверов.

Кеширование информации. Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.

Иерархическая структура, в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.

Резервирование. За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

Домены первого уровня: .ru, .tv, .de.

Домены второго уровня: .gov, .com, .biz.

**Билет 26. Архитектура DNS. Рекурсивные и нерекурсивные серверы имен. Ретрансляторы.**

По способу ответа на запрос

Рекурсивные серверы: O Самостоятельно выполняют весь поиск O Кэшируют полученную информацию

Нерекурсивные серверы: O Указывают, где есть необходимая информация O Не кэшируют информацию.

DNS-запрос может быть рекурсивным — требующим полного поиска, — и нерекурсивным — не требующим полного поиска.

Аналогично, DNS-сервер может быть рекурсивным (умеющим выполнять полный поиск) и нерекурсивным (не умеющим выполнять полный поиск). Некоторые программы DNS-серверов, например, BIND, можно сконфигурировать так, чтобы запросы одних клиентов выполнялись рекурсивно, а запросы других — нерекурсивно.

При ответе на нерекурсивный запрос, а также — при неумении или запрете выполнять рекурсивные запросы, — DNS-сервер либо возвращает данные о зоне, за которую он ответствен, либо возвращает адреса серверов, которые обладают большим объёмом информации о запрошенной зоне, чем отвечающий сервер, чаще всего — адреса корневых серверов.

В случае рекурсивного запроса DNS-сервер опрашивает серверы (в порядке убывания уровня зон в имени), пока не найдёт ответ или не обнаружит, что домен не существует. (На практике поиск начинается с наиболее близких к искомому DNS-серверов, если информация о них есть в кеше и не устарела, сервер может не запрашивать другие DNS-серверы.).

При рекурсивной обработке запросов все ответы проходят через DNS-сервер, и он получает возможность кэшировать их. Повторный запрос на те же имена обычно не идет дальше кэша сервера, обращения к другим серверам не происходит вообще.

Ретрансляторы:

Используются для сокращения внешнего трафика Кэшируют информацию всех запросов Должны конфигурироваться как ретрансляторы.

**Билет 27. Архитектура DNS. Прямой поиск.**

DNS (англ. Domain Name System — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене (SRV-запись).

Распределённая база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определённому протоколу.

Основой DNS является представление об иерархической структуре доменного имени и зонах. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу (с административной точки зрения — другой организации или человеку), что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

DNS обладает следующими характеристиками:

Распределённость администрирования. Ответственность за разные части иерархической структуры несут разные люди или организации.

Распределённость хранения информации. Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности и (возможно) адреса корневых DNS-серверов.

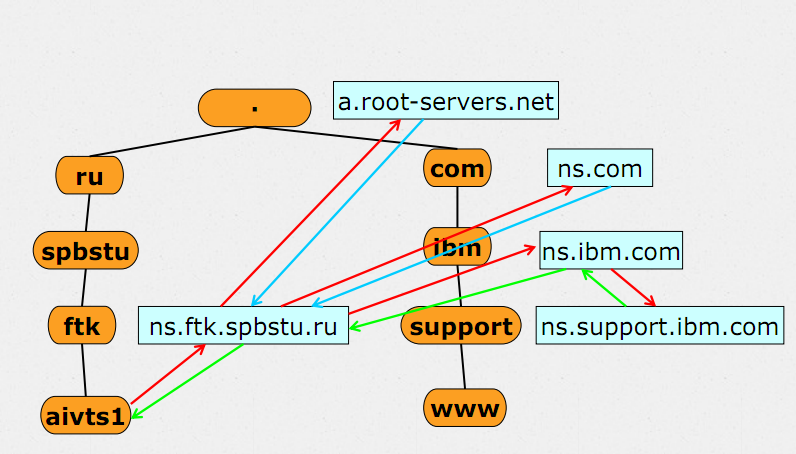
Кеширование информации. Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.

Иерархическая структура, в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.

Резервирование. За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение.

Прямой запрос

Прямой (forward) запрос — запрос на преобразование имени (символьного адреса) хоста в IP-адрес.



**Билет 28. База данных DNS. Ресурсные записи DNS. Адресные записи, записи о сервере имен.**

База данных DNS состоит из ресурсных записей (RR)

Каждая запись хранит определенный тип информации

Каждая запись содержит следующие поля:

O Имя (необязательное). В случае отсутствия используется предыдущее O Класс записи (для Интернета - IN) O Тип записи O Время актуальности (необязательное). В случае отсутствия используется значение по умолчанию O Параметры записи (зависят от типа).

Примеры записей:

Тип - Расшифровка названия (англ.) - Код -Описание - Употребимость

A – Address – 1 - Адресная запись, соответствие между именем и IP-адресом - одна из самых часто используемых записей

AAAA - A+1+1+1 (A использовался для IPv4, AAAA для IPv6) – 28 - Адрес в формате IPv6 - эквивалента А для IPV4

Формат bind: O name IN A address

Пример: O aivt IN A 195.19.212.16.

NS - Authoritative name server – 2 - Адрес узла, отвечающего за доменную зону. - Критически важна для функционирования самой системы доменных имён DNS

Формат bind: O domain IN NS address

Пример: O ftk.spbstu.ru. IN NS 195.19.212.13

*PTR - Domain name pointer – 12 - Реализует механизм переадресации - широко используется для IPv4-адресов в домене in-addr.arpa, для IPv6 — в ip6.arpa.*

**Билет 29. База данных DNS. Ресурсные записи DNS. Главная ресурсная запись. Маршрутизация электронной почты.**

DNS. Главная ресурсная запись. Обозначается «SOA» (start of authority)

Параметры:O Имя домена O Имя первичного DNS-сервера O Почтовый адрес администратора O Серийный номер зоны (Serial) O Период обновления (Refresh) O Время валидности данных зоны (Expire) O Период повторных попыток (Retry) O Значение по умолчанию (DefaultTTL)

Должна быть одна на зону.

Формат bind: O domain IN SOA ns e-mail (Serial Refresh Retry Expire DefaultTTL)

Пример: O ftk.spbstu.ru. IN SOA ns.ftk.spbstu.ru. admin.ftk.spbstu.ru (2008121801 3600 900 2592000 900).

За́пись MX (от англ. mail exchanger) — это один из типов записей в DNS, указывающий способ маршрутизации электронной почты. MX-записи для данного домена указывают серверы, на которые нужно отправлять электронную почту, предназначенную для адресов в данном домене. Кроме того, MX-записи указывают приоритет каждого из возможных серверов для отправки.

Имя хоста, указанного в записи MX, должно содержать IP-адрес, определённый с помощью записи IN A. Псевдонимы IN CNAME не могут иметь своих MX-записей.

Чтобы отправить электронную почту на определённый адрес, сервер-отправитель делает DNS-запрос, запрашивая MX-запись домена получателя электронного сообщения (то есть части адреса после символа «@»). В результате запроса возвращается список имён хостов почтовых серверов, принимающих входящую почту для данного домена, а также величину приоритета для каждого из хостов. Сервер-отправитель затем пытается установить SMTP-соединение с одним из этих хостов, начиная с того, у кого значение величины приоритета наименьшее, перебирая каждый из них, пока не удастся установить соединение хотя бы с одним из них. Если же имеется несколько хостов с одинаковыми приоритетами, то должны быть предприняты попытки установить соединение с каждым из них.

Механизм записей MX предоставляет возможность использовать множество серверов для одного домена и упорядочивания их использования в целях уменьшения нагрузки и увеличения вероятности успешной доставки почты. Кроме того, такой механизм предоставляет возможность распределить обработку входящей почты среди нескольких физических серверов.

**Билет 30. База данных DNS. Ресурсные записи DNS. Записи о псевдонимах, сервисах.**

CNAME-запись

Запись типа CNAME (Canonical Name — Каноническое имя) позволяет присваивать хосту мнемонические имена. Мнемонические имена, или псевдонимы, широко применяются для связывания с хостом какой-либо функции, либо просто для сокращения имени.

Реальное имя иногда называют каноническим.

Если для хоста есть запись типа CNAME, которая содержит его мнемонические имена, другие записи для данного хоста должны ссылаться на его реальное (каноническое) имя, а не на мнемоническое. Когда программы DNS встречают запись CNAME, они прекращают свои запросы по мнемоническому имени и переключаются на реальное имя.

Например, недопустима конструкция вида:

ns1.test.ru. CNAME ns.test.ru.

office.test.ru. NS ns1.test.ru.

Кроме того, если данное имя использовано в качестве псевдонима, то на него нельзя занести записи любого другого типа.

Т.е. недопустима конструкция вида:

domain CNAME имя\_хоста

domain MX 10 почтовый сервер

Мнемонические имена полезны, например, в случае, когда имя хоста изменилось, и вы хотите разрешить пользователям, знающим старое имя, получить доступ к хосту.

Запись типа CNAME имеет следующий формат:

мнемоимя [TTL] CNAME имя\_хоста

Служебная запись (SRV запись) стандарт в DNS определяющий местоположение, то есть имя хоста и номер порта, серверов для определенных служб.

Формат bind: O \_Service.\_Proto.Name IN SRV Priority Weight Port Target

Параметры O Service – имя сервиса O Proto – имя протокола O Priority – приоритет O Weight – вес (для балансировки нагрузки) O Port – номер порта O Target – имя узла, на котором находится сервис

Пример: O \_xmpp-server.\_tcp.example.ru. 3600 IN SRV 20 0 5269 jabber1.ru

Обратный DNS-запрос

DNS используется в первую очередь для преобразования символьных имён в IP-адреса, но он также может выполнять обратный процесс. Для этого используются уже имеющиеся средства DNS. Дело в том, что с записью DNS могут быть сопоставлены различные данные, в том числе и какое-либо символьное имя. Существует специальный домен in-addr.arpa, записи в котором используются для преобразования IP-адресов в символьные имена. Например, для получения DNS-имени для адреса 11.22.33.44 можно запросить у DNS-сервера запись 44.33.22.11.in-addr.arpa, и тот вернёт соответствующее символьное имя. Обратный порядок записи частей IP-адреса объясняется тем, что в IP-адресах старшие биты расположены в начале, а в символьных DNS-именах старшие (находящиеся ближе к корню) части расположены в конце.

Обеспечивает преобразование имен из домена in-addr.arpa. в доменное имя

Параметры:O Имя узла в домене in-addr.arpa. O Доменное имя узла

Формат bind: O in-addr-name IN PTR name

Примеры:

O 11.12.19.195.in-addr.arpa. IN PTR ya.ru.

O 16 IN PTR aivt.ftk.spbstu.ru.

**Билет 31.DNS. Обратный поиск**

CNAME-запись

Запись типа CNAME (Canonical Name — Каноническое имя) позволяет присваивать хосту мнемонические имена. Мнемонические имена, или псевдонимы, широко применяются для связывания с хостом какой-либо функции, либо просто для сокращения имени.

Реальное имя иногда называют каноническим.

Если для хоста есть запись типа CNAME, которая содержит его мнемонические имена, другие записи для данного хоста должны ссылаться на его реальное (каноническое) имя, а не на мнемоническое. Когда программы DNS встречают запись CNAME, они прекращают свои запросы по мнемоническому имени и переключаются на реальное имя.

Например, недопустима конструкция вида:

ns1.test.ru. CNAME ns.test.ru.

office.test.ru. NS ns1.test.ru.

Кроме того, если данное имя использовано в качестве псевдонима, то на него нельзя занести записи любого другого типа.

Т.е. недопустима конструкция вида:

domain CNAME имя\_хоста

domain MX 10 почтовый сервер

Мнемонические имена полезны, например, в случае, когда имя хоста изменилось, и вы хотите разрешить пользователям, знающим старое имя, получить доступ к хосту.

Запись типа CNAME имеет следующий формат:

мнемоимя [TTL] CNAME имя\_хоста

Служебная запись (SRV запись) стандарт в DNS определяющий местоположение, то есть имя хоста и номер порта, серверов для определенных служб.

Формат bind: O \_Service.\_Proto.Name IN SRV Priority Weight Port Target

Параметры O Service – имя сервиса O Proto – имя протокола O Priority – приоритет O Weight – вес (для балансировки нагрузки) O Port – номер порта O Target – имя узла, на котором находится сервис

Пример: O \_xmpp-server.\_tcp.example.ru. 3600 IN SRV 20 0 5269 jabber1.ru

Обратный DNS-запрос

DNS используется в первую очередь для преобразования символьных имён в IP-адреса, но он также может выполнять обратный процесс. Для этого используются уже имеющиеся средства DNS. Дело в том, что с записью DNS могут быть сопоставлены различные данные, в том числе и какое-либо символьное имя. Существует специальный домен in-addr.arpa, записи в котором используются для преобразования IP-адресов в символьные имена. Например, для получения DNS-имени для адреса 11.22.33.44 можно запросить у DNS-сервера запись 44.33.22.11.in-addr.arpa, и тот вернёт соответствующее символьное имя. Обратный порядок записи частей IP-адреса объясняется тем, что в IP-адресах старшие биты расположены в начале, а в символьных DNS-именах старшие (находящиеся ближе к корню) части расположены в конце.

Обеспечивает преобразование имен из домена in-addr.arpa. в доменное имя

Параметры:O Имя узла в домене in-addr.arpa. O Доменное имя узла

Формат bind: O in-addr-name IN PTR name

Примеры:

O 11.12.19.195.in-addr.arpa. IN PTR ya.ru.

O 16 IN PTR aivt.ftk.spbstu.ru.

**Билет 32. Доменная система имен. Динамические обновления; нотификации об изменениях; инкрементальные обновления**

DNS. Динамические обновления

Предпосылка – наличие динамических систем управления адресами (DHCP)

DNS Dynamic Update - RFC 2136

Авторизованный клиент посылает серверу DNS запрос изменение информации о ресурсной записи Если сервер не первичный, то запрос пересылается наверх

Когда запрос дойдет до первичного сервера

O Первичный заносит информацию в свою БД

O Инкрементируется серийный номер зоны

Вторичные серверы позже получают информацию от Первичного

DHCP (англ. Dynamic Host Configuration Protocol — протокол динамической конфигурации узла) — это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP, и получает от него нужные параметры.

DNS. Нотификация

Асинхронное информирование об изменении информации Первичный сервер посылает обновление всем известным ему вторичным Вторичные посылают всем другим вторичным Каждое сообщение, полученное от первичного, подтверждается Вторичные серверы осуществляют запрос зоны как при периодическом обновлении.

**Билет 33. Доменная система имен. Утилиты контроля DNS. Регистрация доменов**.

DNS (англ. Domain Name System — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене (SRV-запись).

Распределённая база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определённому протоколу.

Основой DNS является представление об иерархической структуре доменного имени и зонах. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу (с административной точки зрения — другой организации или человеку), что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

DNS обладает следующими характеристиками:

Распределённость администрирования. Ответственность за разные части иерархической структуры несут разные люди или организации.

Распределённость хранения информации. Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности и (возможно) адреса корневых DNS-серверов.

Кеширование информации. Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.

Иерархическая структура, в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.

Резервирование. За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение.

nslookup :O unix O Windows

dig:O unix

host:O unix

DNS. Регистрация доменных имен

InterNIC – на территории США

Reseaux IP Europeens (RIPE) – в Европе

African Network Information Centre (AfriNIC) – в Африке

Asia Pacific Network Information Centre (APNIC) – в Азии и Океании

Regional Latin-American and Caribbean IP

Address Registry (LACNIC) – в Латинской Америке

Russian Institute for Public Networks (RIPN) – в России

**Билет 34. Прикладные протоколы. Протоколы удаленных терминалов.**

Прикладные протоколы TCP/IP: O Удаленные терминалы O Электронная почта O Передача файлов

O Передача новостей O Обмен мгновенными сообщениями O Передача объектов

Среди протоколов удаленных терминалов можно выделить:

Протокол RLOGIN (англ. Remote LOGIN — удалённый вход в систему) — протокол прикладного уровня (7ой уровень модели OSI), часть стека TCP/IP. Позволяет пользователям UNIX подключаться к системам UNIX на других машинах и работать так же, как при прямом подключении терминала к машине. Этот протокол обеспечивает такой же сервис, как протокол TELNET.

TELNET (англ. TErminaL NETwork) — сетевой протокол для реализации текстового интерфейса по сети (в современной форме — при помощи транспорта TCP). Название «telnet» имеют также некоторые утилиты, реализующие клиентскую часть протокола.

Назначение протокола TELNET в предоставлении достаточно общего, двунаправленного, восьмибитного байт-ориентированного средства связи. Его основная задача заключается в том, чтобы позволить терминальным устройствам и терминальным процессам взаимодействовать друг с другом. Предполагается, что этот протокол может быть использован для связи вида терминал-терминал («связывание») или для связи процесс-процесс («распределенные вычисления»).

Хотя в сессии Telnet выделяют клиентскую и серверную сторону, протокол на самом деле полностью симметричен. После установления транспортного соединения (как правило, TCP) оба его конца играют роль «сетевых виртуальных терминалов», обменивающихся двумя типами данных:

Прикладными данными;Командами протокола Telnet.

SSH (англ. Secure SHell — «безопасная оболочка»[1]) — сетевой протокол сеансового уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Схож по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

SSH позволяет безопасно передавать в незащищенной среде практически любой другой сетевой протокол. Таким образом, можно не только удаленно работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео (например, с веб-камеры)[2]. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удаленного запуска клиентов X Window System.

X Window System (англ. X Window System core protocol) — формат взаимодействия системы X Window, сетевой оконной системы для растровых видеотерминалов. X Window основана на клиент-серверной модели, то есть один сервер управляет всем вводом/выводом, таким как экран(ы), клавиатура и «мышь», все приложения работают как клиенты, взаимодействуя с пользователем и другими клиентами через сервер. Это взаимодействие и обеспечивается корневым протоколом. Существуют также другие протоколы, которые являются как «надстройками» над корневым, так и совершенно независимыми.

Корневой протокол системы X Window предусматривает только 4 типа пакетов данных, посылаемых асинхронно через сеть: запросы, отклики, события и сообщения об ошибках.

**Билет 35. Электронная почта. Механизм работы. Система MIME.**

Электро́нная по́чта (англ. email, e-mail, от англ. electronic mail) — технология и предоставляемые ею услуги по пересылке и получению электронных сообщений (называемых «письма» или «электронные письма») по распределённой (в том числе глобальной) компьютерной сети.

Электронная почта по составу элементов и принципу работы практически повторяет систему обычной (бумажной) почты, заимствуя как термины (почта, письмо, конверт, вложение, ящик, доставка и другие), так и характерные особенности - простоту использования, задержки передачи сообщений, достаточную надёжность и в то же время отсутствие гарантии доставки.

Формат почтового адреса: name@host.domain name:

O Имя пользователя O Название почтового ящика O Название списка рассылки

Формат сообщения - RFC 822

Расширенные сообщения – MIME.

Multipurpose Internet Mail Extensions (MIME (произн. «майм»), англ. Multipurpose Internet Mail Extension — многоцелевое расширение интернет-почты) — стандарт, описывающий передачу различных типов данных по электронной почте, а также, шире, спецификация для кодирования информации и форматирования сообщений таким образом, чтобы их можно было пересылать по Интернету.

MIME определяет механизмы для передачи разного рода информации внутри текстовых данных (в частности, с помощью электронной почты), а именно: текст на языках, для которых используются кодировки, отличные от ASCII, и нетекстовый контент, такой как картинки, музыка, фильмы и программы. MIME является также фундаментальным компонентом коммуникационных протоколов, таких как HTTP.

**Билет 36. Электронная почта. Протокол передачи электронной почты SMTP.**

Электро́нная по́чта (англ. email, e-mail, от англ. electronic mail) — технология и предоставляемые ею услуги по пересылке и получению электронных сообщений (называемых «письма» или «электронные письма») по распределённой (в том числе глобальной) компьютерной сети.

Электронная почта по составу элементов и принципу работы практически повторяет систему обычной (бумажной) почты, заимствуя как термины (почта, письмо, конверт, вложение, ящик, доставка и другие), так и характерные особенности - простоту использования, задержки передачи сообщений, достаточную надёжность и в то же время отсутствие гарантии доставки.

SMTP (англ. Simple Mail Transfer Protocol — простой протокол передачи почты) — это сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.

Сервер SMTP — это конечный автомат с внутренним состоянием.

2ХХ — команда успешно выполнена

3XX — ожидаются дополнительные данные от клиента

4ХХ — временная ошибка, клиент должен произвести следующую попытку через некоторое время

5ХХ — неустранимая ошибка

Клиент <–> MTA

MTA <–> MTA

Использует TCP, порт 25

Не поддерживает шифрования

Базовая версия не поддерживает аутентификацию .

Пример взаимодействия:

Соединение //происходит соединение с сервером

HELO <домен> //Ответ от сервера, если всё ОК

MAIL FROM: <адрес> //Адрес отправителя

RCPT TO: <адрес> //Адрес получателя

DATA //Начало передачи сообщения, текст

<Текст письма>

. //Все письма должны оканчиваться точкой на новой строке

NOOP //Пустая команда

QUIT //Выход

ESMTP (англ. Extended SMTP) — масштабируемое расширение протокола SMTP. В настоящее время под «протоколом SMTP», как правило, подразумевают ESMTP и его расширения.

VRFY <mailbox> //Проверка существования адреса

RSET - прерывает текущую процедуру посылки

почтового сообщения.

Протокол SMTP. Аутентификация

Команда AUTH <механизм> [<строка>]

Секция команды MAIL FROM:

O MAIL FROM: <адрес> AUTH=строка

O MAIL FROM: <адрес> AUTH=<>

**Билет 37. Электронная почта. Маршрутизация почты. Борьба со спамом.**

Электро́нная по́чта (англ. email, e-mail, от англ. electronic mail) — технология и предоставляемые ею услуги по пересылке и получению электронных сообщений (называемых «письма» или «электронные письма») по распределённой (в том числе глобальной) компьютерной сети.

Электронная почта по составу элементов и принципу работы практически повторяет систему обычной (бумажной) почты, заимствуя как термины (почта, письмо, конверт, вложение, ящик, доставка и другие), так и характерные особенности - простоту использования, задержки передачи сообщений, достаточную надёжность и в то же время отсутствие гарантии доставки.

Маршрутизация почты – это процесс, согласно которому почтовые сервера осуществляют доставку почтовых сообщений. Маршрутизация почты происходит на основании MX записей домена.

Пример 1:

Почтовый сервер домена george.beatles.org пытается доставить сообщение, адресованное домену john.beatles.org.

Почтовый сервер делает DNS -запрос о MX-записях узла george.beatles.org.

Из ответа на DNS-запрос он узнает, что домен john.beatles.org имеет три MX записи.

Почтовый сервер попытается доставить почту серверу john.beatles.org, как MX с наивысшим приоритетом (наименьшее числовое значение). Если он не сможет связаться с john.beatles.org, он может (но не обязательно) попытаться связаться с poul.beatles.org, и, если тот не отвечает, попытается связаться с ringo.beatles.org.

В случае, когда никакой из почтовых серверов, указанных в MX записях для домена не доступен, почтовый сервер отправителя может попробовать доставить почту серверу, на который указывает A-запись домена получателя.

Борьба со спамом:

Технология Sender ID

Идея:

O Владелец домена публикует в открытом доступе список адресов, с которых можно отправлять почту от имени данного домена

O Получатель почты проверяет сообщение, руководствуясь заголовками письма и записями о разрешенных именах домена

Purported Responsible Address (PRA)— предполагаемый адрес отправителя

Задача – извлечение из письма PRA и последующая проверка его

Информация о разрешенных адресах домена хранится в DNS. Поиск PRA осуществляется по цепочкам заголовков письма

Если PRA найден: O На основе PRA определяется PRD (Purported Responsible Domain) O Осуществляется запрос к DNS-серверу, отвечающему за домен PRD O Принимается решение о дальнейших действиях

Если PRA не найден: O Письмо посылается на доп. Проверку.

Sender Policy Framework (структура политики отправителя) — расширение для протокола отправки электронной почты SMTP.

SPF позволяет владельцу домена указать в TXT записи, соответствующей имени домена, специальным образом сформированную строку, указывающую список серверов, имеющих право отправлять email-сообщения с обратными адресами в этом домене.

Агенты передачи почты получающие почтовые сообщения могут запрашивать SPF-информацию с помощью простого DNS-запроса, верифицируя таким образом сервер отправителя.

**Билет 38. Электронная почта. Протокол доступа к почтовым ящикам POP3.**

Электро́нная по́чта (англ. email, e-mail, от англ. electronic mail) — технология и предоставляемые ею услуги по пересылке и получению электронных сообщений (называемых «письма» или «электронные письма») по распределённой (в том числе глобальной) компьютерной сети.

Электронная почта по составу элементов и принципу работы практически повторяет систему обычной (бумажной) почты, заимствуя как термины (почта, письмо, конверт, вложение, ящик, доставка и другие), так и характерные особенности - простоту использования, задержки передачи сообщений, достаточную надёжность и в то же время отсутствие гарантии доставки.

POP3 (англ. Post Office Protocol Version 3 — протокол почтового отделения, версия 3) используется почтовым клиентом для получения сообщений электронной почты с сервера. Обычно используется в паре с протоколом SMTP.

Позволяет:

O Доступ к удаленному почтовому ящику O Аутентификация пользователя O Просмотр списка писем O Копирование писем в локальный ящик O Удаление писем с сервера O Ящик представляется одной почтовой папкой (Inbox) O Используется протокол TCP, порт 110.

Протокол POP-3. Команды

USER <имя>

O -ERR - если не поддерживается plaintext

authentication

O +OK - в остальных случаях

PASS <пароль>

O +OK – в случае успешной аутентификации

O -ERR - в случае неуспешной аутентификации

STAT

O +OK <N> <M> - N писем общей длиной M

LIST

O +OK

1 <длина 1>

N <длина N>

LIST <N>

O +OK

<N> <длина>

O -ERR - сообщение отсутствует

RETR <number>

O +OK

<текст сообщения>

.

O -ERR – сообщение отсутствует

DELE <N>

O +OK

O -ERR – сообщение отсутствует

RSET

O +OK

QUIT

O +OK

TOP <N> <M>

O +OK

<заголовок сообщения N>

<пустая строка>

<M строк сообщения>

O -ERR – сообщение отсутствует

Аутентификация:

APOP <имя> <дайджест>

O +OK

O -ERR - ошибочная аутентификация

AUTH <тип\_аутентификации> (RFC 1734)

O -ERR – неизвестный метод аутентификации

O + - ответы сервера

O +OK

Достоинства:O простота реализации O большая

распространенность

Недостатки: O отсутствие шифрования O аутентификация O блокировка ящика O отсутствие папок O отсутствие атрибутов сообщений.

**Билет 39. Электронная почта. Протокол доступа к почтовым ящикам IMAP4.**

IMAP (англ. Internet Message Access Protocol) — протокол прикладного уровня для доступа к электронной почте.

Он базируется на транспортном протоколе TCP и использует порт 143.

IMAP предоставляет пользователю обширные возможности для работы с почтовыми ящиками, находящимися на центральном сервере. Почтовая программа, использующая этот протокол, получает доступ к хранилищу корреспонденции на сервере так, как будто эта корреспонденция расположена на компьютере получателя. Электронными письмами можно манипулировать с компьютера пользователя (клиента) без постоянной пересылки с сервера и обратно файлов с полным содержанием писем.

Для отправки писем используется протокол SMTP.

Особенности: O Позволяет хранить удаленную структуру папок сообщений O Обеспечивает асинхронный обмен командами O Уникальный номер команды и ответа O Флаги сообщений O Уникальные идентификаторы сообщений O Механизмы копирования и перемещения сообщений O Средства поиска сообщений O Варианты аутентификации (login и authenticate).

Флаги сообщения

Этот атрибут представляет собой список из нуля или более именованных лексем, соотнесенный данному сообщению. Флаг устанавливается путем его добавления к этому списку и обнуляется путем его удаления.

\seen — сообщение прочитано

\answered — на сообщение отправлен ответ

\flagged — сообщение отмечено как «важное»

\deleted — сообщение отмечено как удаленное

\draft — сообщение отмечено как черновик

\recent — недавнее сообщение (впервые появилось в ящике в ходе текущей сессии).

Команды:

На всех стадиях

O CAPABILITY – запрос списка возможностей

O NOOP

O LOGOUT

Стадия «Неаутентифицирован»

O LOGIN <username> <password>

O AUTHENTICATE <method>

На стадия «Аутентифицирован»

O SELECT <имя\_ящика> - выбор ящика

O EXAMINE <имя\_ящика> - выбор ящика (RO)

O CREATE <имя\_ящика> - создание ящика

O DELETE <имя\_ящика> - удаление ящика

O RENAME <старое\_имя> <новое\_имя> - переименование ящика

O SUBSCRIBE <имя\_ящика> - подписка на ящик

O UNSUBSCRIBE <имя\_ящика> - отмена подписки на ящик

На стадия «Аутентифицирован»

O LIST <база> <имя\_ящика> - выдача списка ящиков

O LSUB - <база> <имя\_ящика> - выдача списка подписанных ящиков

O STATUS <имя\_ящика> [<имена\_эл-тов\_состояния>] – выдача состояния ящика

O APPEND <имя\_ящика> [(флаги)] […] – добавить сообщение в ящик.

На стадия «Выбран»

O CHECK – проверка ящика

O CLOSE – удаление помеченных сообщений и закрытие ящика

O EXPUNGE - удаление помеченных сообщений

O SEARCH [CHARSET] <критерии> - поиск сообщени

O FETCH <набор\_сообщений> <эл-ты данных>

O STORE <набор\_сообщений> <значение> - изменение флагов сообщений

O COPY <набор\_сообщений> <имя\_ящика> - копирование сообщений в ящик

O UID <команда> - выдача идентификаторов сообщений

**Билет 40. Протокол передачи файлов FTP. Активный режим.**

FTP (англ. File Transfer Protocol — протокол передачи файлов) — протокол, предназначенный для передачи файлов в компьютерных сетях. FTP позволяет подключаться к серверам FTP, просматривать содержимое каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверами.

Протокол FTP относится к протоколам прикладного уровня и для передачи данных использует транспортный протокол TCP. Команды и данные, в отличие от большинства других протоколов, передаются по разным портам. Исходящий порт 20, открываемый на стороне сервера, используется для передачи данных, порт 21 для передачи команд. Порт для приема данных клиентом определяется в диалоге согласования. В случае, если передача файла была прервана по каким-либо причинам, протокол предусматривает средства для докачки файла, что бывает очень удобно при передаче больших файлов.

FTP. Активный режим:O Режим «по умолчанию» O Сервер инициирует соединение данных O Невозможно использовать с технологиями типа NAT, Proxy O Обычно запрещён в межсетевых экранах.

В активном режиме, когда клиент говорит «Привет!» он так же сообщает серверу номер порта (из динамического диапазона 1024-65535) для того, чтобы сервер мог подключиться к клиенту для установки соединения для передачи данных. FTP-сервер подключается к заданному номеру порта клиента используя со своей стороны номер TCP-порта 20 для передачи данных.

Команды:O USER <имя> O PASS <пароль> O REIN – реинициализация O ABOR – прервать обмены O QUIT .

Команды, оперирующие с файловой системой:O DELE <имя>– удалить файла O RNFR <имя> - переименовать из O RNTO <имя> - переименовать в O CWD <путь> - сменить каталог O CDUP – перейти в родительский каталог O RMD <имя> - удалить каталог O MKD <имя> - создать каталог O PWD – показать текущий каталог.

Команды, использующие канал данных:O RETR <имя> - получить файл O STOR <имя> - записать файл O LIST [<путь>] – получить список файлов с атрибутами O NLST [<путь>] – получить список имен файлов.

Достоинства:O Эффективность O Гибкость

Недостатки: O Не поддерживает шифрования O Не поддерживает безопасной аутентификации O Не поддерживает современных средств адресации O Сложность работы с защищенными сетями

**Билет 41. Протокол передачи файлов FTP. Пассивный режим.**

FTP (англ. File Transfer Protocol — протокол передачи файлов) — протокол, предназначенный для передачи файлов в компьютерных сетях. FTP позволяет подключаться к серверам FTP, просматривать содержимое каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверами.

Протокол FTP относится к протоколам прикладного уровня и для передачи данных использует транспортный протокол TCP. Команды и данные, в отличие от большинства других протоколов, передаются по разным портам. Исходящий порт 20, открываемый на стороне сервера, используется для передачи данных, порт 21 для передачи команд. Порт для приема данных клиентом определяется в диалоге согласования. В случае, если передача файла была прервана по каким-либо причинам, протокол предусматривает средства для докачки файла, что бывает очень удобно при передаче больших файлов.

PASSIVE MODE

Изначально протокол предполагал встречное TCP-соединение от сервера к клиенту для передачи файла или содержимого каталога. Это делало невозможным общение с сервером, если клиент находится за IP NAT, кроме того, часто запрос соединения к клиенту блокируется файерволом. Чтобы этого избежать, было разработано расширение протокола FTP passive mode, когда соединение для передачи данных тоже происходит от клиента к серверу. Важным моментом является то, что клиент устанавливает соединение с адресом и портом, указанным сервером. Порт сервер выбирает случайным образом из определённого диапазона (49152-65534). Поэтому при нахождении ftp-сервера за NAT, следует явно указать в настройках сервера его адрес.

Команды:O USER <имя> O PASS <пароль> O REIN – реинициализация O ABOR – прервать обмены O QUIT .

Команды, оперирующие с файловой системой:O DELE <имя>– удалить файла O RNFR <имя> - переименовать из O RNTO <имя> - переименовать в O CWD <путь> - сменить каталог O CDUP – перейти в родительский каталог O RMD <имя> - удалить каталог O MKD <имя> - создать каталог O PWD – показать текущий каталог.

Команды, использующие канал данных:O RETR <имя> - получить файл O STOR <имя> - записать файл O LIST [<путь>] – получить список файлов с атрибутами O NLST [<путь>] – получить список имен файлов.

Достоинства:O Эффективность O Гибкость

Недостатки: O Не поддерживает шифрования O Не поддерживает безопасной аутентификации O Не поддерживает современных средств адресации O Сложность работы с защищенными сетями

**Билет 42. Протокол HTTP.**

HTTP (сокр. от англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов). Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом. HTTP в настоящее время повсеместно используется во Всемирной паутине для получения информации с веб-сайтов.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (англ. Uniform Resource Identifier) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы, но ими могут быть логические объекты или что-то абстрактное. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т. д. Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

HTTP — протокол прикладного уровня, аналогичными ему являются FTP и SMTP. Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ». Для идентификации ресурсов HTTP использует глобальные URI. В отличие от многих других протоколов, HTTP не сохраняет своего состояния.

Формат HTTP-запроса

<Request-line> - строка запроса

<General-header> - общий заголовок

<Request-header> - заголовок запроса

<Entity-header> - заголовок сообщения

<Body> - тело

Протокол HTTP. Строка запроса

Формат: <METHOD> <URL> <HTTP-VERSION>

Методы: O GET O POST O HEAD O PUT O DELETE

O OPTIONS O и т.п.

Общий заголовок (General-header)

Присутствует, когда есть тело сообщения

O Connection: O Data O Pragma

O Transfer-encoding: O Upgrade: O no-cache

Заголовок запроса (Request-header)

O Accept: принимаемый контент O Accept-Charset: принимаемый набор символов O Accept-Encoding: compress, zip O Accept-Language: da, ru O Authorization: basic xxx=\*\*\*\*\*\* O From: O Host: O If-modified-since:…

O Referer: O User-agent: O И т.д.

Заголовок сообщения (Entity-header)

O Allow: GET, POST, HEAD O Content-Encoding: x-zip O Content-Language: O Content-Length: 1245 O Content-Type: ...text/html; charset=win-1251 O Expires: O Last-Modified:

Протокол HTTP. Формат ответа

<Status-line> - Строка статуса

<General-header> - общий заголовок

<Response-header> - заголовок ответа

<Entity-header> - Заголовок сообщения

<Body> - тело

Протокол HTTP. Строка статуса

Формат: <HTTP-VERSION> <Code> <Phrase>

Code:

O 1xx – информационные O 2xx – OK O 3xx – Переадресация (redirection) O 4xx – Ошибка клиента O 5xx – Ошибка сервера.

Заголовок ответа (Response-header) O Location: переадресация O Server: спецификация сервера O WWW-Authenticate: basic realm=’localzone’ O Age: Возраст ресурса.