1.Эталонная модель ISO/OSI. Назначение уровней.

Сетевая модель OSI - сетевая модель стека сетевых протоколов OSI/ISO Посредством данной модели различные сетевые устройства могут взаимодействовать друг с другом. Модель определяет различные уровни взаимодействия систем. Каждый уровень выполняет определенные функции при таком взаимодействии.

1.Прикладной обеспечивает взаимодействие пользовательских приложений с сетью.2.Представительский (представления) обеспечивает преобразование протоколов и кодирование/декодирование данных 3.Сеансовый обеспечивает поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. 4.Транспортный предназначен для обеспечения надежной передачи данных от отправителя к получателю. 5.Сетевой предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию. 6.Канальный предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками, которые могут возникнуть. 7. Физический определяет метод передачи данных, представленных в двоичном виде, от одного устройства к другому. Любой протокол модели OSI должен взаимодействовать либо с протоколами своего уровня, либо с протоколами на единицу выше и/или ниже своего уровня. Взаимодействия с протоколами своего уровня называются горизонтальными, а с уровнями на единицу выше или ниже — вертикальными. Любой протокол модели OSI может выполнять только функции своего уровня и не может выполнять функций другого уровня, что не выполняется в протоколах альтернативных моделей. Каждому уровню с некоторой долей условности соответствует свой операнд — логически неделимый элемент данных, которым на отдельном уровне можно оперировать в рамках модели и используемых протоколов: на физическом уровне мельчайшая единица — бит, на канальном уровне информация объединена в кадры, на сетевом — в пакеты (датаграммы), на транспортном — в сегменты.

2. Архитектура TCP/IP. Иерархия сетевых протоколов. Назначение основных протоколов.

Стек протоколов TCP/IP — набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет. Название TCP/IP происходит из двух наиважнейших протоколов семейства — Transmission Control Protocol и Internet Protocol, которые были разработаны и описаны первыми в данном стандарте. Протоколы работают друг с другом в стеке — это означает, что протокол, располагающийся на уровне выше, работает «поверх» нижнего,

используя механизмы инкапсуляции. Например, протокол TCP работает поверх протокола IP.
Стек протоколов TCP/IP включает в себя четыре уровня: 1)прикладной уровень (application layer), 2)транспортный уровень (transport layer), 3) сетевой уровень (Internet layer), 4)канальный уровень (link layer).
Протоколы этих уровней полностью реализуют функциональные возможности модели OSI. На стеке протоколов TCP/IP построено всё взаимодействие пользователей в IP-сетях. Стек является независимым от физической среды передачи данных. На прикладном уровне работает большинство сетевых приложений. Эти программы имеют свои собственные протоколы обмена информацией, например, HTTP для WWW, FTP, SMTP, SSH, DNS
Протоколы транспортного уровня могут решать проблему негарантированной доставки сообщений («дошло ли сообщение до адресата?»), а также гарантировать правильную последовательность прихода данных. (TCP, UDP)
Сетевой уровень изначально разработан для передачи данных из одной (под)сети в другую. (IP (вспомогательные протоколы, вроде ICMP и IGMP)
Канальный уровень (Link layer) описывает, каким образом передаются пакеты данных через физический уровень, включая кодирование (то есть специальные последовательности бит, определяющих начало и конец пакета данных). (Ethernet)

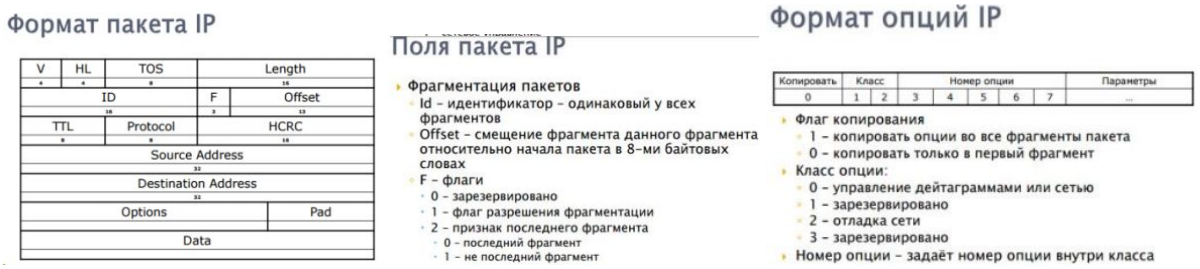
3. IP-адресация. Классы сетей, маска сети, зарезервированные адреса.

IP-адрес – это уникальный числовой адрес, однозначно идентифицирующий узел, группу узлов или сеть. IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел (так называемых «октетов»), разделенных точками, каждое из которых может принимать значения в диапазоне от 0 до 255. Существует 5 классов IP-адресов – A, B, C, D, E. Принадлежность IP-адреса к тому или иному классу определяется значением первого октета.

Класс IP-адреса	A	B	C	D	E
Диапазон первого октета	1-126	128-191	192-223	224-239	240-247

Поддерживается: 1)Индивидуальная адресация 2) Широковещательная адресация 3) Групповая адресация
Адрес 0.0.0.0 - В таблицах маршрутизации – маршрут по умолчанию. При адресации – данная сеть.
Все узлы данной локальной сети - Формат адреса: 255.255.255.255
Для более гибкого определения границ между разрядами номеров сети и узла внутри IP-адреса используются так называемые маски подсети. Маска подсети – это 4-байтовое число специального вида, которое используется совместно с

IP-адресом. "Специальный вид" маски подсети заключается в следующем: двоичные разряды маски, соответствующие разрядам IP-адреса, отведенным под номер сети, содержат единицы, а в разрядах, соответствующих разрядам номера узла – нули.



4. Сетевой протокол IP. Назначение и основные функции.

Сетевой протокол IP является базовым строительным элементом всей сети Интернет, построенной на базе стека протоколов TCP/IP. Он обеспечивает работу базовой службы доставки пакетов, все протоколы сетевого и соседних уровней используют протокол IP для доставки данных.
Протокол IP выполняет ряд важных функций:
Определяет базовую единицу передачи информации в сети Интернет — дейтограмму; 1) Определяет схему интернет-адресации (IP-адрес); 2)Осуществляет обмен данными между уровнем доступа к сети и транспортным уровнем;3) Выполняет маршрутизацию пакетов, адресованных удаленным узлам;4) Отвечает за разбиение и сборку дейтаграмм.
Особенностью протокола IP является то, что он не проверяет были ли данные успешно доставлены. Иными словами, данный протокол работает без создания логических соединений. Установка логических соединений делегируется протоколам других уровней. Помимо этого, при обнаружении и исправлении ошибок протокол IP также полагается на другие протоколы.
Формат пакета, определяемый протоколом IP называется дейтаграммой.
ARP (англ. Address Resolution Protocol — протокол определения адреса) — протокол в компьютерных сетях, предназначенный для определения я AC-адреса по известному IP-адресу. Рассмотрим суть функционирования ARP на простом примере. Компьютер А (IP-адрес 10.0.0.1) и компьютер Б (IP 10.22.22.2) соединены сетью Ethernet. Компьютер А желает переслать пакет данных на компьютер Б, IP-адрес компьютера Б ему известен. Однако сеть Ethernet, которой они соединены, не работает с IP-адресами. Поэтому компьютеру А для осуществления передачи через Ethernet требуется узнать адрес компьютера Б в сети Ethernet (MAC-адрес в терминах Ethernet). Для этой задачи и используется протокол ARP. По этому протоколу компьютер А отправляет широковещательный запрос, адресованный всем компьютерам в одном с ним широковещательном домене. Суть запроса: «компьютер с IP-адресом 10.22.22.2, сообщите свой MAC-адрес компьютеру с IP-адресом

10.0.0.1». Сеть Ethernet доставляет этот запрос всем устройствам в том же сегменте Ethernet, в том числе и компьютеру Б. Компьютер Б отвечает компьютеру А на запрос и сообщает свой MAC-адрес (напр. 00:ea:d1:11:f1:11) Теперь, получив MAC-адрес компьютера Б, компьютер А может передавать ему любые данные через сеть Ethernet. Перед тем как передать пакет сетевого уровня через сегмент Ethernet, сетевой стек проверяет кэш ARP, чтобы выяснить, не зарегистрирована ли в нём уже нужная информация об узле-получателе Записи в кэше ARP могут быть статическими и динамическими. Пример, данный выше, описывает динамическую запись кэша. Можно также создавать статические записи в таблице ARP. Это можно сделать при помощи команды: arp -s В системах семейства Windows до NT 6.0 записи в таблице ARP, созданные динамически, остаются в кэше в течение 2-х минут. Если в течение этих двух минут произошла повторная передача данных по этому адресу, то время хранения записи в кэше продлевается ещё на 2 минуты. Эта процедура может повторяться до тех пор, пока запись в кэше просуществует до 10 минут. После этого запись будет удалена из кэша, и будет отправлен повторный запрос ARP
Дейтаграмма содержит множество различных полей: IP-адрес отправителя и IP-адрес получателя, Время жизни пакета (TTL), Контрольная сумма заголовка, Флаги фрагментации и др.

5. Механизмы связи сетевого и канального уровня в TCP/IP. Протоколы ARP.

Связь с канальным уровнем осуществляется через ARP-таблицу. Она состоит из IP-адрес, MAC-адрес, Тип записи.
Типы записей:
- Динамические записи
- Статические записи

ARP - протокол в компьютерных сетях, предназначенный для определения MAC-адреса, имея IP-адрес другого компьютера. Цель – заполнение ARP-таблицы. Осуществляется широковещательный запрос с установленными полями ip и mac, а для получателя устанавливается только IP адрес, а MAC оставлен пустым. Запрос осуществляется всем узлам, узлы сравнивают запрашиваемый IP адрес и если он совпал с их адресом, то заполняют MAC адрес и отправляют arp ответ (индивидуальный).
Заполнение ARP-таблицы:
◦ На приемной стороне при получении ARP-запроса
◦ На запрашивающей стороне при получении ответа

6. Управляющий протокол ICMP. Типы пакетов.

ICMP (англ. Internet Control Message Protocol — протокол межсетевых управляющих сообщений) — сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают. Также на ICMP возлагаются некоторые сервисные функции. ICMP основан на протоколе IP. Каждое ICMP-сообщение инкапсулируется непосредственно в пределах одного IP-пакета, и, таким образом, как и UDP и в отличие от TCP, ICMP является т. н. «ненадежным» (не контролирующим доставку и её правильность). В отличие от UDP, где реализация надёжности возложена на ПО прикладного уровня, ICMP (в силу специфики применения) обычно не нуждается в реализации надежной доставки Нотификационные: 3- получатель недостижим 11-превышено время ожидания 12-ошибка в заголовке Управляющие: 4-управление скоростью 5-изменение маршрута Тестовые и диагностические: 0 и 8- эхо-запрос и эхо-ответ 13 и 14 – временные метки 17 и 18 – маска адреса
Предназначен для решения задач: 1)Управления 2)Нотификации об ошибках и проблемах 3) Тестирования и мониторинга
Инкапсулирован в IP. Типы: 1) Нотификационные сообщения а)Получатель недостижим б) Превышено время в)Ошибка параметра 2)Управляющие сообщения а)Подавление источника б) Изменение маршрута 3) Тестовые и контрольные сообщения а) Запрос эха и ответ на запрос эха б)Запрос и ответ временной метки в)Запрос и ответ маски адреса

7. Транспортный протокол TCP. Основные особенности и алгоритм функционирования.

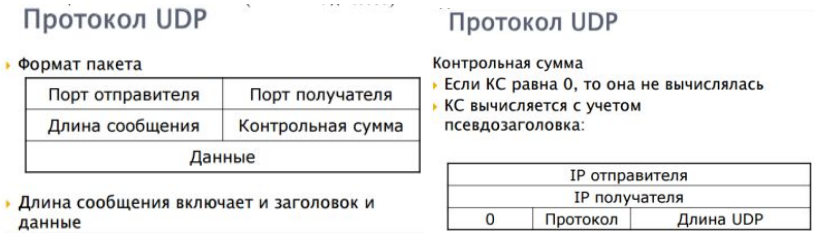
Transmission Control Protocol (протокол управления передачей) — один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных. В стеке протоколов IP, TCP выполняет функции протокола транспортного уровня модели OSI. Механизм TCP предоставляет поток данных с предварительной установкой соединения, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета, гарантируя тем самым, в отличие от

UDP, целостность передаваемых данных и уведомление отправителя о результатах передачи. Адресация приложений осуществляется с помощью TCP-портов. Данные для передачи хранятся в буфере: Данные от приложения добавляются конец буфера, Данные для передачи в сеть берутся из начала буфер. Пересылаемая порция данных называется – сегмент. Каждый передаваемый байт – пронумерован. Сегменту присваивается номер его первого байта (номер очереди). Каждый посланный в сеть байт должен быть подтвержден. При получении подтверждения от сегмента подтвержденными считаются все байты сегмента. Если подтверждение не получено в течение определенного времени – сегмент из буфера повторной передачи посылается заново. Подтверждение содержит номер следующего ожидаемого байта В TCP отрицательные квитанции не посылаются. Основные поля пакета: Порт отправителя и получателя, номер подтверждения и очереди, флаги, контрольная сумма, опции, окно
Флаги: 1) URG – задействовано поле «Указатель срочности» 2) ACK - задействовано поле «Подтверждение» 3) PSH – включена функция проталкивания 4) RST – перезагрузка соединения 5) SYN – синхронизация номеров очередей 6)FIN – завершение соединения

8. Транспортный протокол UDP. Основные особенности.

UDP (англ. User Datagram Protocol — протокол пользовательских датаграмм) — один из ключевых элементов TCP/IP, набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. UDP использует простую модель передачи, без неявных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. Таким образом, UDP предоставляет ненадёжный сервис, и датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в системах реального времени. Для адресации используется UDP-порты.
По сравнению с другими транспортными протоколами имеет:
◦ Более высокую скорость
◦ Меньшую надежность
Пакет состоит из: Порт отправителя и получателя, длина сообщения, контрольная сумма.

Контрольная сумма вычисляется с учетом псевдозаголовка, в нем есть IP отправителя и получателя, протокол и длина udp.



9. Основные задачи маршрутизации в TCP/IP. Статическая маршрутизация. Таблицы маршрутизации.

Маршрутизация – это функция сетевого уровня, заключается в доставке пакетов через сеть от одного узла к другому. Маршрутизация — процесс определения маршрута следования данных в сетях связи. Бывает индивидуальная и групповая. Статическая маршрутизация - 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:1) Адрес сети (на которую маршрутизируется трафик), маска сети 2) Адрес шлюза (узла), который способствует дальнейшей маршрутизации (или подключен к маршрутизируемой сети напрямую) 3) (опционально) метрика (иногда именуется также "ценой") маршрута.
При наличии нескольких маршрутов на одну и ту же сеть некоторые маршрутизаторы выбирают маршрут с минимальной метрикой
Таблица маршрутизации — электронная таблица (файл) или база данных, хранящаяся на маршрутизаторе или сетевом компьютере, которая описывает соответствие между адресами назначения и интерфейсами, через которые следует отправить пакет данных до следующего маршрутизатора. Является простейшей формой правил маршрутизации.
Таблица маршрутизации обычно содержит:1) адрес сети или узла назначения, либо указание, что маршрут является маршрутом по умолчанию 2)маску сети назначения 3) шлюз, обозначающий адрес маршрутизатора в сети, на который необходимо отправить пакет, следующий до указанного адреса назначения 4) интерфейс, через который доступен шлюз 5) метрику — числовой показатель, задающий предпочтительность маршрута. Чем меньше число, тем более предпочтителен маршрут (интуитивно представляется как расстояние).

- Атрибуты маршрутных записей
  - Сеть/узел назначения
  - Сетевой интерфейс
  - Маршрутизатор
  - Метрика маршрута
  - Флаги
- Псевдомаршруты
  - Дополнительные записи в таблице маршрутизации
  - Используются для унификации процедуры поиска маршрута
  - Два типа:
    - Псевдомаршрут на IP-адреса собственных сетевых интерфейсов
    - Псевдомаршрут на подключенные IP-сети

Назначение	Интерфейс	Маршрутизатор	Метрика	Тип
195.19.212.0 /24	eth1	195.19.214.6	5	S
195.19.213.128 /25	ppp0	195.19.215.1	7	S
195.19.213.0 /25	tok0	195.19.16.1	7	R
0.0.0.0/0	eth1	195.19.214.1	5	S

- В некоторых системах поддерживается несколько таблиц маршрутизации
- В таких таблицах используется коммутация по адресу источника:
  - В зависимости от адреса источника выбирается подчиненная таблица маршрутизации

10. Динамическая маршрутизация в сетях TCP/IP. Поиск кратчайшего пути. Алгоритмы Беллмана-Форда и Дейкстры.

Протоколы динамической маршрутизации предназначены для автоматизации процесса построения маршрутных таблиц маршрутизаторов. Принцип их использования достаточно прост: маршрутизаторы с помощью устанавливаемого протоколом порядка рассылают определенную информацию из своей таблицы маршрутизации другим и корректируют свою таблицу на основе полученных от других данных. Такой метод построения и поддержки маршрутных таблиц существенно упрощает задачу администрирования сетей, в которых могут происходить изменения (например, расширение) или в ситуациях, когда какие-либо маршрутизаторы и/или подсети выходят из строя. Следует отметить, что использование протоколов динамической маршрутизации не отменяет возможность «ручного» внесения данных в таблицы маршрутизаторов. Внесенные таким образом записи называют статическими, а записи, полученные в результате обмена информацией между маршрутизаторами – динамическими. В любой таблице маршрутизации всегда присутствует, по крайней мере, одна статическая запись – маршрут по умолчанию. Динамическая маршрутизация обоснована часто меняющимися параметрами в сети: меняется топология сети, каналы связи, узлы сети, также может изменяться в сети и нагрузка. Задача оптимальной маршрутизации – найти оптимальный путь для пакета в данный момент времени. А поиск маршрута сводится к поиску кратчайшего пути на графе. Критерии оптимальности могут быть разные: минимальное время досатвки, минимальная стоимость доставки, минимальная задержка и т.д. Для нахождения оптимальных маршрутов используют алгоритмы Бэллмана-Форда и Дейкстры

Алгоритм Бэллмана-Форда

Количество узлов – N, текущий шаг – h

Вес дуг между узлами (вершинами) равен метрикам.

Граф дополняется до полного, вес новых дуг – бесконечность. В начале путь до всех вершин равен бесконечности, а в начальной вершине – 0. На каждом шаге для всех вершин пересчитывается путь до них (выбирается минимальное число из: вес соседней вершины + вес дуги). Если h = N, то выход из алгоритма, если оценки не изменились – выход. N-1 – число шагов; N –

операций при минимизации на каждом шаге; Сложность W = O(N^3);
Достоинства алгоритма:
1) Хорошо распараллеливается 2) Просто реализуется 3) Не требует ресурсов памяти 4) Требуется информация только о соседних вершинах 5) Часто заканчивается раньше N-1 итерации ;Недостатки алгоритма: В худшем случае количество операций - ~ N^3

Алгоритм Дейкстры

Основная идея – на каждом шаге выбор любого кратчайшего пути
m – номер текущего шага; P – множество помеченных вершин; n – номер вершины, которая добавляется в P
В начале в множество P входит только начальная вершина. Ее вес равен 0, остальные вершины равны путям до них от начальной вершины. На каждом шаге выбирается вершина с минимальным весом и добавляется в P, потом для всех вершин, которые не находятся в P происходит перерасчет весов: выбирается минимальное число из предыдущего веса и веса вершины n + пути от нее до пересчитываемой вершины. Если m = N, то выход.
N-1 – число шагов; N – операций при пересчете оценок на каждом шаге;
W = O(N^2); Достоинства алгоритма: Высокая скорость (~N^2); Недостатки алгоритма: 1)Плохо распараллеливается 2)Требуется иметь информацию о топологии всей сети 3) Требует существенных ресурсов памяти (~N^2)

11. Протоколы динамической маршрутизации RIP, OSPF. Автономные системы.

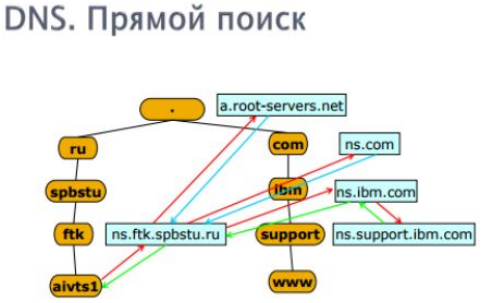
Автономная система - часть сети, управляющая из одного центра управления, реализующая одну политику маршрутизации. Внутри АС обеспечиваются одинаковые протоколы маршрутизации. Протоколы маршрутизации внутри АС - IGP – Interior Gateway Protocol, а вне - EGP – Exterior Gateway Protocol.
Протокол RIP Является IGP. Использует алгоритм вычисления маршрута Бэллмана-Форда. Тип протокола – однопутевой. Метрика в нем это целое число из диапазона 0..15. Измеряется числом промежуточных маршрутизаторов до сети назначении. Для непосредственно подсоединенных сетей – значение «0». Значение «16» – сеть недоступна. Метрика не зависит от задержек, пропускной способности, надежности, загрузки. 1 раз в 30 сек маршрутизаторы рассылают свою таблицу маршрутизации с помощью RIP пакетов. У протокола плохая сходимость, поэтому в течении некоторого времени маршруты находятся в некорректном состоянии. Пакет RIP инкапсулируется в UDP. Для адресации соседних маршрутизаторов используется широковещательная адресация.
Достоинства ◦ Простота реализации ◦ Низкие требования к вычислительным ресурсам маршрутизаторов ◦ Низкие требования к объемам памяти маршрутизаторов ◦ Простота настройки; Недостатки ◦ Неэффективность

метрики маршрутов ◦ Высокая загрузка каналов ◦ Ограниченный диаметр сети ◦ Медленная сходимость ◦ Отсутствие маски подсети ◦ Отсутствие подтверждения подлинности ◦ Отсутствие шифрования
Протокол OSPF Использует алгоритм Дэйкстра. Является IGP протоколом. Метрика задается числом 0..65535. Метрика определяется как количество секунд, требуемое для передачи 100 Мбит через физическую среду данной сети. Рассылаются пакеты LSA , они 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. Каждый маршрутизатор по LSD строит LST –Link State Tree (В качестве корня использует себя). DR (Designated Router) – назначенный маршрутизатор, используется, как основной адрес передачи LSA маршрутизаторами. Рассылает обновления всем маршрутизаторам. OSPF инкапсулируется непосредственно в IP. Достоинства ◦ Гибкая метрика ◦ Быстрая сходимость ◦ Низкая загрузка каналов служебным трафиком ◦ Отсутствие петель маршрутизации ◦ Поддержка доменов маршрутизации ◦ Поддержка различных маршрутов для разных типов обслуживания Недостатки ◦ Высокие требования к ресурсам: Быстродействие, Объемы памяти для хранения LSD, LST ◦ Высокая сложность конфигурирования

12. Методы именования ресурсов в сетях TCP/IP. Доменная система имен.

DNS-запрос (запрос от клиента (или сервера) серверу. Запрос может быть рекурсивным или не рекурсивным . Термином Рекурсия в DNS обозначают алгоритм поведения DNS-сервера, при котором сервер выполняет от имени клиента полный поиск нужной информации во всей системе DNS, при необходимости обращаясь к другим DNS-серверам. При ответе на нерекурсивный запрос, а также при неумении или запрете выполнять рекурсивные запросы, DNS-сервер либо возвращает данные о зоне, за которую он ответственен, либо возвращает ошибку. Настройки не рекурсивного сервера, когда при ответе выдаются адреса серверов, которые обладают большим объёмом информации о запрошенной зоне, чем отвечающий сервер (чаще всего — адреса корневых серверов), являются некорректными, и такой сервер может быть использован для организации DoS-атак. 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 кэше 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 кэшасервера, обращения к другим серверам не происходит вообще. Допустимое время хранения ответов в кэше приходит вместе с ответами



Задачи серверов DNS: Для всех серверов: 1) хранение инфы о зоне 2) обслуживание запросов об этой зоне 3) перенаправление запросов о других зонах вышестоящим или подчиненным серверам 4) кэширование инфы о других зонах. Первичные серверы:1) копирование зонной инфы вторичным 2) модификация первичной инфы Вторичные серверы: периодический опрос первичных.

13. Прямой поиск в системе DNS. Рекурсивные и нерекурсивные серверы имен. Ключевые ресурсные записи в системе DNS. Обратный поиск.

При прямом поиске происходит преобразование символических имен в адреса. При обратном поиске адреса преобразуются в символические имена.
DNS сервера разбиваются на 2 типа по способу ответа на запрос:
Рекурсивные серверы - самостоятельно выполняют весь поиск
и кэшируют полученную информацию.
Нерекурсивные серверы - указывают, где есть необходимая информация и не кэшируют информацию.
База данных DNS состоит из ресурсных записей (RR). Каждая запись хранит определенный тип информации. Каждая запись содержит следующие поля: 1) Имя (необязательное). В случае отсутствия используется предыдущее 2)Класс записи (для Интернета - IN) 3)Тип записи 4)Время актуальности (необяз). В случае отсутствия используется значение по умолчанию 5)Параметры записи (зависят от типа) ; Ключевые записи: Адресная запись - доменное имя и адрес ; Запись о сервере имен – имя домена и адрес сервера; Главная ресурсная запись - параметры зоны днс ;Запись о сервере электронной почты – имя почтового сервера, имя почтового домена и приоритет ;Запись о псевдониме – доменное имя узла (псевдоним) и реальное; Обратный поиск используется, чтобы убедиться в валидности IP-адреса. ;Для обратного поиска введена ресурсная запись PTR. Она обеспечивает преобразование имен из домена in-addr.arpa. в доменное имя.

14. Методы организации опосредованного доступа к сети. Прокси-серверы и трансляция адресов с помощью технологии NAT.

Proxy

Прокси-сервер (от англ. proxy — «представитель, уполномоченный») — промежуточный сервер (комплекс программ) в компьютерных сетях, выполняющий роль посредника между пользователем и целевым сервером (при этом о посредничестве могут как знать, так и не знать обе стороны), позволяющий клиентам как выполнять косвенные запросы (принимая и передавая их через прокси-сервер) к другим сетевым службам, так и получать ответы. Сначала клиент подключается к прокси-серверу и запрашивает какой-либо ресурс (например, e-mail), расположенный на другом сервере.
Решаемые задачи: ◦ Кэширование информации ◦ Сокрытие внутренней части сети ◦ Сокращение времени доступа в сеть
Используют протоколы HTTP и FTP. Для доступа к серверу-посреднику обычно используется HTTP.
Иерархии кэшей: ◦ Два типа отношений:
Родительские/дочерние (parent)
Родственные (sibling)
◦ Сокращение трафика ◦ Сокращение времени ◦ Ограничение на уровень иерархии
Клиентский компьютер имеет настройку (конкретной программы или операционной системы), в соответствии с которой все сетевые соединения по некоторому протоколу совершаются не на IP-адрес сервера (ресурса), выделяемый из DNS-имени ресурса, или напрямую заданный, а на IP-адрес (и другой порт) прокси-сервера.
При необходимости обращения к любому ресурсу по этому протоколу, клиентский компьютер открывает сетевое соединение с прокси-сервером (на нужном порту) и совершает обычный запрос, как если бы он обращался непосредственно к ресурсу.
Распознав данные запроса, проверив его корректность и разрешения для клиентского компьютера, прокси-сервер, не разрывая соединения, сам открывает новое сетевое соединение непосредственно с ресурсом и делает тот же самый запрос. Получив данные (или сообщение об ошибке), прокси-сервер передаёт их клиентскому компьютеру.
Таким образом прокси-сервер является полнофункциональным сервером и клиентом для каждого поддерживаемого протокола и имеет полный контроль над всеми деталями реализации этого протокола, имеет возможность применения заданных администратором политик доступа на каждом этапе работы протокола.

NAT

Механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов. Преобразование адреса методом NAT может производиться почти любым маршрутизирующим устройством — маршрутизатором[1], сервером доступа, межсетевым экраном. Наиболее популярным является SNAT, суть механизма которого состоит в замене адреса источника (англ. source) при прохождении пакета в одну сторону и обратной замене адреса назначения (англ. destination) в ответном пакете. Наряду с адресами источник/назначение могут также заменяться номера портов источника и назначения. Принимая пакет от локального компьютера, роутер смотрит на IP-адрес назначения. Если это локальный адрес, то пакет пересылается другому локальному компьютеру. Если нет, то пакет надо переслать наружу в интернет. Но ведь обратным адресом в пакете указан локальный адрес компьютера, который из интернета будет недоступен. Поэтому роутер «на лету» транслирует (подменяет) обратный IP-адрес пакета на свой внешний (видимый из интернета) IP-адрес и меняет номер порта (чтобы различать ответные пакеты, адресованные разным локальным компьютерам). Комбинацию, нужную для обратной подстановки, роутер сохраняет у себя во временной таблице. Используется в неанонсированных сетях. Защищает клиентские станции от внешних сетей. Поддерживает протоколы TCP, UDP, ICMP. FTP в активном режиме не может использовать NAT. Для транспортных протоколов выделяются ассоциированные порты, имитирующие участников соединения

15. Конфигурирование компьютерных сетей. Протокол DHCP. Утилиты ping, traceroute, nslookup, ifconfig/ipconfig, netstat.

Для конфигурирования используются 2 подхода – ручная и автоматизированная настройка. Для автоматизированной используются протоколы RARP, BOOTP, DHCP.
Протокол RARP используется для только для получения IP-адреса.
Протокол BOOTP способен получать: IP-адрес, Маску сети, Маршрутизатор «по умолчанию.
Протокол DHCP является расширением BOOTP: Информация размещается в поле опций BOOTP.
Поддерживает 3 режима:
◦ Ручное распределение
◦ Автоматическое распределение
◦ Динамическое распределение
Управляет следующими параметрами: ◦ Маска сети ◦ Default gateway ◦ DNS ◦ HostName ◦ DomainName, Параметры узла (Default TTL, IP forwarding),

Параметры интерфейсов, Параметры TCP, Параметры приложений, Параметры аренды
nslookup — утилита, предоставляющая пользователю интерфейс командной строки для обращения к системе DNS (проще говоря, DNS-клиент).
ping — утилита для проверки соединений в сетях на основе TCP/IP
traceroute — это служебная компьютерная программа, предназначенная для определения маршрутов следования данных в сетях TCP/IP.
ifconfig/ipconfig - утилита командной строки для управления сетевыми интерфейсами.
netstat - утилита командной строки выводящая на дисплей состояние TCP-соединений (как входящих, так и исходящих), таблицы маршрутизации, число сетевых интерфейсов и сетевую статистику по протоколам.

16. Устройство и назначение электронной почты. Протокол передачи почты SMTP. Протоколы доступа к почтовым ящикам POP3 и IMAP4.

Электронная почта— технология и служба по пересылке и получению электронных сообщений (называемых «письма») между пользователями компьютерной сети. **SMTP—** это широко используемый сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.
Электронная почта представлена почтовым клиентом (MUA, mail user agent — пользовательский почтовый агент) для почтового сервера (MSA, mail submission agent — агент отправки электронной почты) с помощью SMTP по TCP-порту 587. Оттуда MSA доставляет почту своим агентам передачи сообщений (MTA, mail transfer agent). Граничный MTA должен найти целевой хост. Он использует систему доменных имен (DNS) для поиска записей почтового обменника (mail exchanger — MX) домена получателя (часть адреса, находящаяся справа от символа @). Возвращаемая запись почтового MX содержит имя целевого хоста. Затем MTA подключается к серверу обмена в качестве SMTP-клиента.
Как только цель MX принимает входящее сообщение, она передает его агенту доставки почты (mail delivery agent — MDA) для локальной доставки

сообщения. MDA предусматривает возможность сохранять сообщения в соответствующем формате почтового ящика. Приём почты, опять же, может быть проведен как несколькими, так и одним компьютером — изображение показывает два ближайших ящика для каждого случая. MDA может доставлять сообщения прямо на хранение или передавать их по сети с помощью SMTP или любых других средств, в том числе протокола локальной пересылки почты (Local Mail Transfer Protocol — LMTP) — производного от SMTP, предназначенного для этой цели.
После доставки на локальный почтовый сервер сообщение хранится для пакетного поиска по аутентифицированным почтовым клиентам (MUA).
POP3 - Стандартный интернет-протокол прикладного уровня, используемый клиентами электронной почты для получения почты с удалённого сервера по TCP-соединению. POP поддерживает простые требования «загрузи-и-удали» для доступа к удалённым почтовым ящикам. Хотя большая часть POP-клиентов предоставляет возможность оставить почту на сервере после загрузки, использующие POP клиенты обычно соединяются, извлекают все письма, сохраняют их на пользовательском компьютере как новые сообщения, удаляют их с сервера, после чего разъединяются.
IMAP (Internet Message Access Protocol) — протокол прикладного уровня для доступа к электронной почте. 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 обычно протокол SMTP, так как собственная команда отправки протокола IMAP, называемая APPEND, считается «неудачной» и «небезопасной». При использовании POP3 клиент подключается к серверу только на промежуток времени, необходимый для загрузки новых сообщений. При использовании IMAP соединение не разрывается, пока пользовательский интерфейс активен, а сообщения загружаются только по требованию клиента. Это позволяет уменьшить время отклика для пользователей, в чьих ящиках имеется много сообщений большого объёма.
Протокол POP требует, чтобы текущий клиент был единственным подключенным к ящику. IMAP позволяет одновременный доступ нескольких клиентов к ящику и предоставляет клиенту возможность отслеживать изменения, вносимые другими клиентами, подключенными одновременно с ним. Благодаря системе флагов, определенной в IMAP4, клиент может

отслеживать состояние сообщения (прочитано, отправлен ответ, удалено и т. д.); данные о флагах хранятся на сервере. Клиенты IMAP4 могут создавать, переименовывать и удалять ящики и перемещать сообщения между ящиками. Кроме того, можно использовать расширение IMAP4 Access Control List (ACL) Extension (RFC 4314) для управления правами доступа к ящикам.

17. Основные способы передачи файлов. Протокол передачи файлов FTP. Активный и пассивный режимы работы FTP.

SFTP (*SSH File Transfer Protocol*) — протокол прикладного уровня, предназначенный для копирования и выполнения других операций с файлами поверх надёжного и безопасного соединения. Протокол как расширение к SSH-2, однако SFTP допускает реализацию и с использованием иных протоколов сеансового уровня. **TFTP** (*Trivial File Transfer Protocol*) используется главным образом для первоначальной загрузки бездисковых рабочих станций. TFTP, в отличие от FTP, не содержит возможностей аутентификации (хотя возможна фильтрация по IP-адресу) и основан на транспортном протоколе UDP.
FTP работает в двух режимах: в пассивном и активном.
Активный режим. Происходит установление tcp – соединения клиента к 21 порту сервера, далее происходит процесс аутентификации клиента. Затем отправляется команда PORT с номером порта клиента Y, по которому будет осуществляться передача данных. Сервер выделяет новый порт(20) для tcp – соединения под передачу данных и устанавливает соединение с портом Y.
Пассивный режим. Клиент устанавливает tcp - соединение к 21 порту, отправляет команду PASV, сервер в ответ PASV OK и номер порта Z, к которому нужно будет подключиться клиенту. Далее клиент устанавливает tcp - соединение по порту Z.

18. Протокол HTTP. Основные отличия HTTP от других протоколов архитектуры TCP/IP.

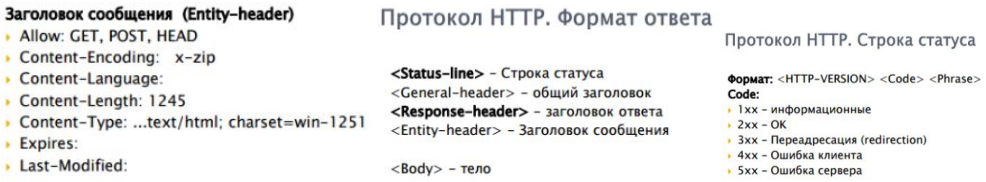
HTTP (HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов в формате HTML, в настоящий момент

используется для передачи произвольных данных). 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.

Большинство протоколов предусматривают установление TCP-сессии, в ходе которой один раз происходит авторизация, и дальнейшие действия выполняются в контексте этой авторизации. HTTP же устанавливает отдельную TCP-сессию на каждый запрос; в более поздних версиях HTTP было разрешено делать несколько запросов в ходе одной TCP-сессии, но браузеры обычно запрашивают только страницу и включённые в неё объекты, а затем сразу разрывают TCP-сессию. Для поддержки авторизованного доступа в HTTP используются cookies; причем такой способ авторизации позволяет сохранить сессию даже после перезагрузки клиента и сервера. При доступе к данным по FTP или по файловым протоколам тип файла (точнее, тип содержащихся в нем данных) определяется по расширению имени файла, что не всегда удобно. HTTP перед тем, как передать сами данные, передаёт заголовок «Content-Type: тип/подтип», позволяющую клиенту однозначно определить, каким образом обрабатывать присланные данные. Это особенно важно при работе с CGI-скриптами, когда расширение имени файла указывает не на тип присылаемых клиенту данных, а на необходимость запуска данного файла на сервере и отправки клиенту результатов работы программы, записанной в этом файле (при этом один и тот же файл в зависимости от аргументов запроса и своих собственных соображений может порождать ответы разных типов — в простейшем случае картинки в разных форматах).

Кроме того, HTTP позволяет клиенту прислать на сервер параметры, которые будут переданы запускаемому CGI-скрипту. Для этого же в HTML были введены формы.

Перечисленные особенности HTTP позволили создавать поисковые машины (первой из которых стала AltaVista, созданная фирмой DEC), форумы и Internet-магазины. Это коммерциализировало Интернет, появились компании, основным полем деятельности которых стало предоставление доступа в Интернет (провайдеры) и создание сайтов.



19. Управление в сетях TCP/IP. Управляющий протокол SNMP.

SNMP (Simple Network Management Protocol — простой протокол сетевого управления) — стандартный интернет - протокол для управления устройствами в IP -сетях на основе архитектур TCP /UDP . Протокол обычно используется в системах сетевого управления для контроля подключенных к сети устройств на предмет условий, которые требуют внимания администратора. SNMP определен Инженерным советом интернета (IETF) как компонент TCP /IP. Он состоит из набора стандартов для сетевого управления, включая протокол прикладного уровня, схему баз данных и набор объектов данных. SNMP предоставляет данные для управления в виде переменных, описывающих конфигурацию управляемой системы. Эти переменные могут быть запрошены (а иногда и заданы) управляющими приложениями. При использовании SNMP один или более административных компьютеров (где функционируют программные средства, называемые менеджерами) выполняют отслеживание или управление группой хостов или устройств в компьютерной сети. На каждой управляемой системе есть постоянно запущенная программа, называемая агент, которая через SNMP передаёт информацию менеджеру. Менеджеры SNMP обрабатывают данные о конфигурации и функционировании управляемых систем и преобразуют их во внутренний формат, удобный для поддержания протокола SNMP. Протокол также разрешает активные задачи управления, например, изменение и применение новой конфигурации через удаленное изменение этих переменных. Доступные через SNMP переменные организованы в иерархии. Эти иерархии, как и другие метаданные (например, тип и описание переменной), описываются базами управляющей информации (базы MIB, от англ. Management information base). Управляемые протоколом SNMP сети состоят из трех ключевых компонентов: Управляемое устройство; Агент — программное обеспечение, запускаемое на управляемом устройстве, либо на устройстве, подключенном к интерфейсу управления управляемого устройства; Система сетевого управления (Network Management System, NMS) — программное обеспечение, взаимодействующее с менеджерами для поддержки комплексной структуры данных, отражающей состояние сети[1]. Управляемое устройство — элемент сети (оборудование или программное средство), реализующий интерфейс управления (не обязательно SNMP), который разрешает однонаправленный (только для чтения) или двунаправленный доступ к конкретной информации об элементе. Управляемые

устройства обмениваются этой информацией с менеджером. Управляемые устройства могут относиться к любому виду устройств: маршрутизаторы, серверы доступа, коммутаторы, мосты, концентраторы, IP -телефоны, IP -видеокамеры, компьютеры -хосты, принтеры и т. п. Агентом называется программный модуль сетевого управления, располагающийся на управляемом устройстве, либо на устройстве, подключенном к интерфейсу управления управляемого устройства. Агент обладает локальным знанием управляющей информации и переводит эту информацию в специфичную для SNMP форму или из неё (медиация данных). В состав Системы сетевого управления (NMS) входит приложение, отслеживающее и контролирующее управляемые устройства. NMS обеспечивают основную часть обработки данных, необходимых для сетевого управления. В любой управляемой сети может быть одна и более NMS. SNMP работает на прикладном уровне TCP/IP (седьмой уровень модели OSI). Агент SNMP получает запросы по UDP - порту 161. Менеджер может посылать запросы с любого доступного порта источника на порт агента. Ответ агента будет отправлен назад на порт источника на менеджере. Менеджер получает уведомления (Traps и InformRequests) по порту 162. Агент может генерировать уведомления с любого доступного порта

20. Архитектура IPv6. Адресация. Особенности организации сетевого уровня. Транспортные протоколы .

IPv6 — новая версия интернет протокола, призванная решить проблемы, с которыми столкнулась предыдущая версия при её использовании в Интернете, за счёт использования длины адреса 128 бит вместо 32.
Предпосылки развития: 1) малое адресное пространство IPv4 2) неудобный формат адреса 3) сложная маршрутизация 4) низкая защищенность (отсутствие шифрования, отсутствие аутентификации) 5) Низкая эффективность
Длина адреса – 128 разрядов. Формат адреса: Префикс, ID провайдера, ID абонента, ID сети, ID узла.

Существуют различные типы адресов IPv6: одноадресные (Unicast), групповые (Anycast) и многоадресные (Multicast). Пакет, посланный на такой адрес типа Unicast, достигает в точности интерфейса, который этому адресу соответствует. Адреса типа Anycast синтаксически неотличимы от адресов Unicast, но они адресуют группу интерфейсов. Пакет, направленный такому адресу, попадёт в ближайший интерфейс. Адреса Anycast могут использоваться только маршрутизаторами. Адреса типа Multicast идентифицируют группу интерфейсов. Пакет, посланный на такой адрес, достигнет всех интерфейсов, привязанных к группе многоадресного вещания. Широковещательные адреса IPv4 (обычно xxx.xxx.xxx.255) выражаются адресами многоадресного вещания IPv6. Крайние адреса подсети IPv6

являются полноправными адресами и могут использоваться наравне с остальными. Пакеты состоят из управляющей информации, необходимой для доставки пакета адресату, и полезных данных, которые требуется переслать. Управляющая информация делится на содержащуюся в основном фиксированном заголовке, и содержащуюся в одном из необязательных дополнительных заголовков. Полезные данные, как правило, это дейтаграмма или фрагмент протокола более высокого транспортного уровня, но могут быть и данные сетевого уровня. IPv6-пакеты обычно передаются с помощью протоколов канального уровня, таких как Ethernet, который инкапсулирует каждый пакет в кадр. Но IPv6-пакет может быть передан с помощью туннельного протокола более высокого уровня, например в 6to4 или Teredo.

IPv6-пакет — блок информации, форматированный для передачи через компьютерные сети, поддерживающие протокол IPv6. Пакеты состоят из управляющей информации, необходимой для доставки пакета адресату и полезных данных, которые требуется переслать. Управляющая информация делится на содержащуюся в основном фиксированном заголовке, и содержащуюся в одном из необязательных дополнительных заголовков. Полезные данные — это дейтаграмма или фрагмент протокола более высокого транспортного уровня, но могут быть и данные сетевого уровня , или же канального уровня. IPv6-пакеты обычно передаются с помощью протоколов канального уровня, таких как Ethernet, который инкапсулирует каждый пакет в кадр. Фиксированный заголовок IPv6-пакета состоит из 40 октетов (320 бит) и имеет следующий формат: Описание полей: • Version: версия протокола; для IPv6 это значение равно 6 (0110). • Traffic Class: приоритет пакета (8 бит). Это поле состоит из двух значений. Старшие 6 бит используются DSCP для классификации пакетов.[2][3] Оставшиеся два бита используются ECN для контроля перегрузки.[4] • Flow Label: метка потока. • Payload Length: в отличие от поля Total Length в протоколе IPv4 данное поле не включает фиксированный заголовок пакета (16 бит). • Next Header: задает тип расширенного заголовка (англ. IPv6 extension), который идет следующим. В последнем расширенном заголовке поле Next Header задает тип транспортного протокола (TCP, UDP и т. д.) • Hop Limit: аналог поля time to live в IPv4 (8 бит). • Source Address и

Destination Address: адрес отправителя и получателя соответственно; по 128 бит. Расширенные заголовки содержат дополнительную информацию и размещены между фиксированным заголовком и заголовком протокола более высокого уровня[1]. Тип первого расширенного заголовка указывается в поле Next Header фиксированного заголовка, а каждый расширенный заголовок имеет аналогичное поле в котором хранится тип следующего расширенного заголовка. В поле Next Header последнего заголовка находится тип протокола более высокого уровня, находящегося в качестве полезных данных. Каждый расширенный заголовок должен иметь размер в октетах, кратный 8. Некоторые заголовки необходимо расширить до нужного размера. Расширенные заголовки должны быть обработаны только конечным узлом, за исключением заголовка Hop-By-Hop Options, который должен быть обработан каждым промежуточным узлом на пути пакета, включая отправителя и получателя. Если расширенных заголовков в пакете несколько, то рекомендуется отсортировать их как указано в таблице ниже. Отметим, что все расширенные заголовки являются необязательными и не должны появиться в пакете более одного раза, за исключением заголовка Destination Options, который может появиться дважды. Если узел не может обработать какой-то расширенный заголовок, то он должен отбросить пакет и отправить сообщение Parameter Problem (ICMPv6 тип 4, код 1). Если в поле Next Header расширенного заголовка будет 0, то узел должен сделать то же самое.