Теормин по сетям 2019-2020

1) **Модель OSI** имеет уровневую организацию. Она включает в себя семь уровней: физический, канальный, сетевой, транспортный, сессии, представления и прикладной. Это модель взаимодействия открытых систем (Open Systems Interconnection), на практике не используется.

2) **Под сервисом** понимают услуги, которые нижерасположенный уровень оказывает по запросам вышерасположенного.

3) **Интерфейс** определяет формирование и передачу запроса на услугу.

4) Правила и соглашения по установлению соединения, его поддержанию и обмену данными по нему между активностями, расположенными на одинаковом уровне на разных машинах, называется **протоколом**.

5) Активные элементы уровня, т.е. элементы, которые могут сами совершать действия, в отличие от элементов, над которыми совершают действия, называются **активностями**. Активности могут быть программными и аппаратными.

6) **Физический уровень** предназначен для передачи битов по каналу связи. Этот уровень никак не анализирует информацию, основная задача: определить способ представления информации в виде сигналов, которые будут передаваться по среде передачи данных.

7) **Канальный уровень** передает уже целые сообщения, он должен уметь в потоке бит отделять сообщения. Также он обеспечивает обнаружение и коррекцию ошибок. В широковещательных сетях канальный уровень обеспечивает физическую адресацию и управляет доступом к разделяемой среде передачи данных.

8) **Сетевой уровень** предназначен для построения крупных составных сетей на основе различных сетевых технологий. Обеспечивается согласование различий в разных технологиях канального уровня, общая адресация и поиск маршрутов в крупной составной сети.

9) **Транспортный уровень** обеспечивает передачу данных между процессами, которые находятся на разных хостах. Может обеспечивать еще более высокую надежность. Является сквозным, так как сообщения напрямую передаются от отправителя к получателю.

10) **Сеансовый уровень** позволяет устанавливать сеансы связи. Предотвращает одновременное выполнение критичной операции (управление маркерами), а также синхронизация и вохобновления передачи в случае сбоя.

11) **Уровень представления** обеспечивает согласование синтаксиса и семантики передаваемых данных. А также шифрование и дешифрование.

12) **Прикладной уровень** - это набор приложений, которые могут использовать пользователи сети

13) Одной из основных целей этого проекта была разработка унифицированных способов соединения сетей для создания систем передачи данных, обладающих высокой живучестью. Так появилась **модель TCP/IP**: протокол управления передачей – TCP (Transmission Control Protocol) и межсетевой протокола – IP (Internet Protocol). Уровни стека: канальный, сетевой (межсетевой), транспортный, прикладной.

14) **Модель сервиса IP**. Internet Protocol («межсетевой протокол») – маршрутизируемый протокол сетевого уровня стека TCP/IP. Именно IP стал тем протоколом, который объединил отдельные компьютерные сети во всемирную сеть Интернет. Неотъемлемой частью протокола является адресация сети. IP не гарантирует надёжной доставки пакета до адрсата.

15) Transmission Control Protocol (**TCP**, протокол управления передачей)— один из основных протоколов передачи данныхинтернета, предназначенный для управления передачей данных. В стеке протоколов TCP/IP выполняет функции 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.

16) В современной сети Интернет используется IP четвёртой версии, также известный как **IPv4**. В протоколе IP этой версии каждому узлу сети ставится в соответствие IP-адрес длиной 4 октета (4 байта).

17) Вводится в эксплуатацию шестая версия протокола – **IPv6**, которая позволяет адресовать значительно большее количество узлов, чем IPv4. Эта версия отличается повышенной разрядностью адреса и встроенной возможностью шифрования.

18) **Возможности IP**:

- предотвращает «зацикливание» пакетов;
- фрагментирует пакеты если они слишком длинные;
- использует контрольную сумму, чтобы сократить возможность доставки в неправильное место назначение;
- позволяет добавлять новые опции к заголовку;
- работает над любой физической средой;

19) **Сокет** - пара IP-адрес хоста и порт. Доступ к TCP-сервису происходит через сокет. Каждое соединение идентифицируется парой сокетов (отправителем и получателем), между которыми оно установлено. Один и тот же сокет может быть использован для разных соединений.

20) **Порт** - 16-разрядный локальный номер на хосте, который является TSAP (transmission service access point) для TCP.

21) Процедура **трехкратного рукопожатия** - установка TCP-соединения:

- Есть клиент (инициатор соединения) и сервер (тот, к кому клиент хочет обратиться).
- *1-ое рукопожатие*
  - Клиент посылает серверу пакет, в котором установлен флаг SYN и указан порядковый номер передаваемого байта в SN.
- *2-ое рукопожатие*
  - Сервер получает пакет и в ответ отправляет другой пакет, в котором:
    - ACK и полученный порядковый номер + 1 в AN
    - SYN с новым ожидаемым порядковым номером в SN
- *3-е рукопожатие*
  - Клиент получает пакет и в ответ отправляет пакет, в котором ACK и ожидаемый порядковый номер + 1 в AC.
- После этого соединение считается установленным.

22) **UDP** (англ. User Datagram Protocol— протокол пользовательских датаграмм)— один из ключевых элементов набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемыедатаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. 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.

23) **Структура UDP-заголовка**:

Биты	0 - 15	16 - 31
0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
64-...	Данные (Data)	

24) **ICMP** (англ. Internet Control Message Protocol — протокол межсетевых управляющих сообщений[1]) — 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.

25) **Домен** – это область пространства иерархических имен сети Интернет, которая обслуживается набором серверов доменных имен (DNS) и централизованно администрируется. Домен идентифицируется именем домена.

26) **Доменное имя** (domain name) – это адрес сетевого соединения, который идентифицирует владельца адреса.

27) **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.

28) Составные части доменного адреса называются **сегментами** и образуют иерархическую систему.

29) Для определения по доменному адресу IP-адреса на специальных узлах Сети имеются Таблицы соответствия. Такие узлы называются **серверами DNS** (Domain Name Service, «служба доменных имен»).

30) **Коммутация** - процесс соединения абонентов через транзитные узлы. Выделяют два основных способа коммутации потоков данных: коммутацию каналов и коммутацию пакетов.

31) **Сквозная задержка (end-to-end delay)** – время от момента когда послан первый бит до момента когда придет последний бит. В общем случае:

$$t = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} + Q_i(t) \right)$$

- Первое слагаемое - задержка пакетизации
- Второй слагаемое - задержка распространения
- Третье слагаемое - задержка буферизации

32) **Задержка распространения** – время распространения одного бита по каналу длиной l со скоростью c. Эта задержка не имеет переменных характеристик и однозначно определяется материалом кабеля и его длиной.

33) **Задержка пакетизации** – время, за которое все биты пакета с первого до последнего переданы в канал с фиксированной пропускной способностью.

34) **Задержка буферизации** - время, которое пакет находится в буфере. Основной источник неопределенности в итоговой e2e-задержке.

35) **Свойства очередей** (далее речь идет о задержке в очереди):

1. Нерегулярность увеличивает задержку

2. Случайность увеличивает задержку

36) **Ethernet** - это технология проводных локальных сетей. Работает на физическом (регламентированы конкретные провода) и канальном уровне.

37) **Коммутатор (Ethernet-коммутатор, сетевой коммутатор)** — устройство, предназначенное для соединения нескольких узлов в пределах одного или нескольких сегментов сети. Работает на канальном уровне. В общем случае коммутатор выполняет 2 базовые операции: поиск соответствия в таблице коммутации и ередачу на надлежащий выходной порт.

38) Коммутатор содержит в себе **таблицу коммутации**, в которой указывается соответствие MAC-адреса узла порту коммутатора. При включении коммутатора таблица пустая, поэтому коммутатор работает в режиме обучения.

39) **Max-min справедливость** - распределение имеющегося ресурса так, чтобы максимизировать минимальный поток.

40) **Max-min распределение** - это распределение, в котором нельзя увеличить скорость какого-нибудь потока, не снизив скорости другого меньшего потока

41) **Распределение потоков** - разделение пропускной способности каждой линии сети между различными потоками.

42) Линия называется **насыщенной**, если сумма скоростей проходящих через неё потоков равна пропускной способности линии.

43) Линия называется **критичной для источника**, если линия насыщена и источник имеет на этой линии наибольшую долю пропускной способности среди всех потоков, проходящих через эту линию.

44) **Теорема о max-min распределении**. Распределение потоков является max-min тогда и только тогда, когда у каждого источника есть критичная линия.

Важно! Если max-min распределение существует, то оно единственно для заданной топологии.

45) **Алгоритм текущего ведра** - это алгоритм, с помощью которого можно придать трафику некоторую форму, т.е. ограничить полосу пропускания канала и гарантировать определенную скорость передачи. Должен располагаться на хосте.

46) **Алгоритм текущего ведра с маркерами**. Модификация алгоритма текущего ведра. Вместе с пакетами, но независимо от них, в ведро поступают маркеры. Теперь пакеты выходят только при наличии определенного числа маркеров. В таком случае можно накапливать маркеры и кратковременно ускорять передачу пакетов в сеть.

47) **Управление потоком** - механизм, который притормаживает передачу данных от отправителя при неготовности получателя. Существует 2 механизма: Stop and wait и Скользящее окно.

48) **Stop and wait** (модельный, а не реальный подход). В одно и то же время передается не более одного пакета.

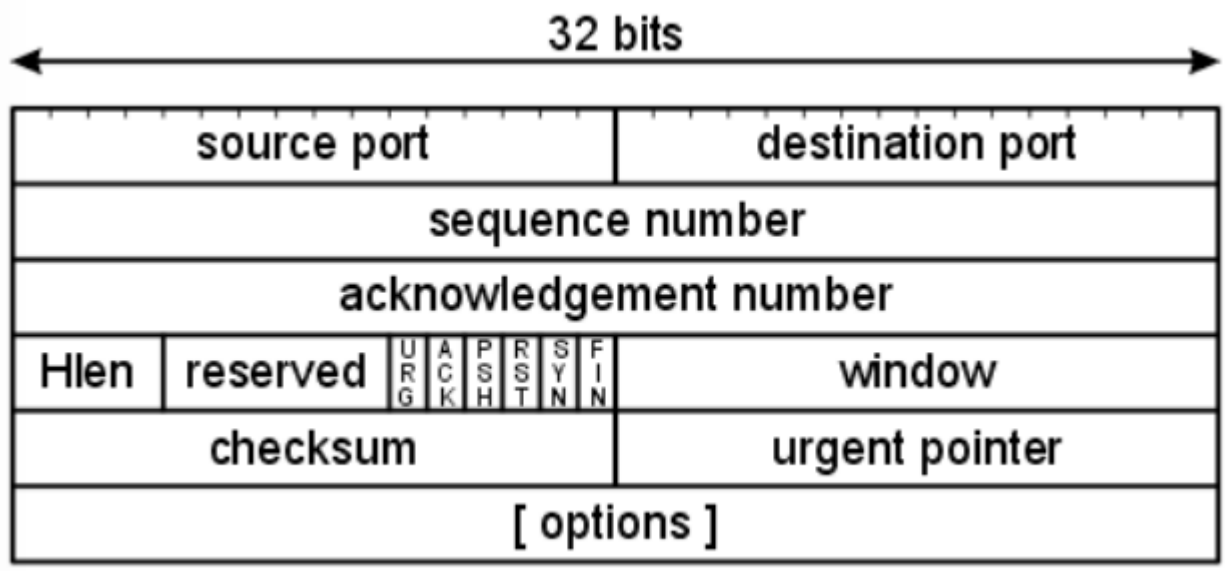
Возникает проблема “дублирования”, решение - используем счетчик на 1 бит, чтобы отличить новые данные от дубликатов.

Главный недостаток Stop And Wait - неэффективное использование канала.

49) **Скользящее окно** - обобщение механизма Stop and wait. Разрешим использовать сразу несколько неподтвержденных сегментов, тогда **размер окна** - количество разрешенных неподтвержденных сегментов.

50) **Заголовок IP:**

51) **Заголовок TCP:**



52) **Псевдозаголовок** - содержит 32-разрядные IP-адреса отправителя и получателя, номера протокола и число байтов в TCP-сегменге, включая заголовок. Псевдозаголовок нужен только для работы с адресами отправителя и получателя на уровне TCP. Он не передается по сети, а создается при передаче пакета с сетевого на транспортный уровень.

53) **PDU** - Protocol Data Unit, самостоятельная единица данных протокола. Разные протоколы на разных уровнях работают с PDU разной длины.

54) **Фрагментация** - разбиение исходного PDU на более короткие фрагменты. Длины всех фрагментов кроме последнего должны быть кратны 8-ми байтам. Проблема фрагментации состоит в том, что происходит увеличение накладных расходов при передаче пакета.

55) **Характер ошибок**: Одиночные, Множественные, Групповые.

56) **Особенности борьбы с ошибками на разных уровнях**:

- L2 (канальный) - защищает весь кадр
- L3 (сетевой - IP) - только заголовок
- L4 (транспортный - TCP) - часть заголовка и тело

57) **Перегрузка** - явление, при котором из-за нерегулярности потоков данных нагрузка на выходных линиях коммутатора начинает превосходить пропускную способность этих линий.

58) **Источник проблемы перегрузки** – несбалансированность пропускных способностей каналов в сравнении со скоростью передаваемых данных.

59) **Управление перегрузками** – это такая организация потоков в транспортной среде, при которой потоки соответствуют пропускной способности подсети и не превышают ее.

60) **Методы, предотвращающие перегрузки** Методы ориентированы на минимизацию перегрузок при первых признаках их проявлений, а не на борьбу с перегрузками, когда они уже случились.

Существуют следующие методы:

1. Сдерживание нарастания трафика посредством управления потоком (скользящее окно)
2. Корректная организация очередей на коммутаторах и маршрутизаторах
3. Выбор метода сброса пакетов
4. Регулирование времени жизни пакета в сети

61) **Требования к алгоритму управления перегрузками:**

- Каналы сети должны быть как можно более загруженными
- Распределение пропускной способности было справедливым
- Возможность быстро реагировать на изменения в топологии сети
- Распределенное управление

62) **AIMD** (Additive Increase Multiple Decrease - аддитивное увеличение, мультипликативное уменьшение) - механизм для оценки окна перегрузки.

Идея - как можно быстрее нащупываем предел имеющегося ресурса, после чего прекращаем наращивать скорость по следующим правилам:

● Если пакет успешно получен, то с каждым последующим пакетом W = W + 1/W Т.к. пакетов в окне W, то по исчерпании окна его размер будет увеличен на 1

● Если пакет был сброшен, то W = W/2

63) **Особенности AIMD**: очень чувствителен к вероятности потери пакетов и ущемляет потоки с большим RTT.

64) **RTT** – время круговой задержки.

65) **CWND** – окно перегрузки

66) **MSS** – максимальный размер сегмента.

67) Стратегия **TCP Tahoe** позволяет оценить CWND и заключается в следующем:

1. Используя медленный старт, быстро «нащупать» доступную пропускную способность сети.

2. Приблизившись к насыщению сети, перейти в режим предотвращения перегрузки.

Общий график работы алгоритма

68) **TCP Reno и TCP New Reno** являются модификацией алгоритма TCP Tahoe и вводят 2 новые фазы: быстрая повторная передача и быстрое восстановление.

69) **Основной задачей сетевого уровня** является маршрутизация пакетов.

70) **Алгоритм маршрутизации** реализует ПО маршрутизатора на сетевом уровне, т.е. он отвечает за определение, по какой из линий отправлять пакет дальше. При этом независимо от способа выбора маршрута алгоритм маршрутизации **должен обладать следующими свойствами:** корректность, простота, устойчивость, стабильность, справедливость, оптимальность.

71) **Алгоритмы маршрутизации можно разбить на два больших класса:** неадаптивные (статическая маршрутизация) и адаптивные (динамическая маршрутизация).

72) **Свойство оптимального пути**: если маршрутизатор B находится на оптимальном пути между маршрутизаторами A и C, то оптимальный маршрут между B и C принадлежит этому оптимальному пути. Это следствие из того, что конкатенация непересекающихся оптимальных маршрутов - тоже оптимальный маршрут.

73) Следствием из этого свойства является утверждение, что все маршруты к заданной точке транспортной среды образуют дерево с корнем в этой точке. Это дерево называют **деревом захода.**

74) Один из наиболее популярных алгоритмов динамической маршрутизации – **маршрутизация по вектору расстояния**. Этот алгоритм, построенный на идеях алгоритма Беллмана–Форда для нахождения кратчайшего пути и алгоритма Форда–Фалкерсона, определяющего максимальный поток в графе**.**

75) **Алгоритм маршрутизации по вектору расстояния** теоретически работает хорошо, но у него есть один **недостаток**: он очень медленно реагирует на разрушения каналов в транспортной среде.

76) **Разделение направлений (Split Horizon Hack)** - решение проблемы бесконечного счетчика.

77) **Маршрутизация по состоянию канала**: каждый маршрутизатор собирает информацию о своих непосредственных соседях, а потом делится этой информацией со всеми маршрутизаторами ⇒ все знают друг о друге.

78) **OSPF (Open Shortest Path First)** — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала и использующий для нахождения кратчайшего пути алгоритм Дейкстры.

79) **Виды соединений в OSPF:**

- Каналы типа точка-точка
- Системы передачи данных (СПД) на основе каналов с множественным доступом и вещанием
- СПД на основе коммутации каналов или коммутации пакетов

80) **Автономная система (АС)** - сеть, охватывающая единую территорию, находящаяся под единым административным управлением и имеющая единую систему правил маршрутизации (политику маршрутизации) по отношению к остальным сетям (т.е. от какого бы устройства внутри АС не поступил пакет, он будет маршрутизирован в другую АС с учетом одного и того же набора правил для всех устройств данной АС).

АС является единицей иерархии в Интернете. Цель иерархии - добиться простоты наращивания размера сети без дополнительных переделок существующего и доп. Инвестиций.

81) Для взаимодействия между АС используется протокол **BGP** (Border Gateway Protocol).

82) BGP-маршрутизатор видит мир как множество других BGP-маршрутизаторов, как-то связанных между собой. **Сети для такого маршрутизатора делятся на три категории**:

● Тупиковые - не могут использоваться для передачи через них трафика

● Сети с множественными соединениями - могут использоваться для транзита, если это допускается политикой определенной АС

● Транзитные сети - могут и должны использоваться для передачи трафика, возможно, с некоторыми ограничениями

83) **Структура Интернета.** Весь Интернет представляет собой множество независимых автономных систем, взаимодействующих через BGP. В Интернете нет какой-либо предопределенной структуры взаимодействия. Там можно выделить несколько магистральных сетей, к которым подключены региональные, а к ним подключены локальные сети организаций.

84) В реальности используются следующие **протоколы групповой маршрутизации**:

IGMP (Internet Group Management Protocol) ,

DVMRP (Distance Vector Multicast Routing Protocol) ,

PIM (Protocol Independent Multicast).

85) **IP-адреса класса D** выделены для групповой адресации.

86) **Данные** - то, с помощью чего мы описываем явление.

87) **Сигнал** - это представление данных.

88) **Передача** - это процесс взаимодействия передатчика и приемника с целью получения приемником сигналов от передатчика.

89) **Полоса пропускания канала** - спектр частот, которые канал пропускает без существенного снижения мощности сигнала.

90) **Пропускная способность канала** - максимальная скорость, с которой канал способен передавать сигнал.

91) **Сигнальная скорость** - скорости изменения значения сигнала. Измеряется в бод.

92) **Витая пара** – два медных изолированных провода, один из которых обвит вокруг другого. Может быть использована для передачи как цифровых, так и аналоговых сигналов.

93) **Цифровой сигнал** — это дискретная последовательность импульсов по напряжению, каждый из которых имеет ступенчатую форму. Каждый импульс — это единичный сигнал.

94) **Скорость передачи данных** — это количество бит в секунду, которые передают с помощью сигналов. Эту скорость также называют битовой скоростью.

95) **Продолжительность или длина бита** — это интервал времени, который нужно передатчику, чтобы испустить надлежащий единичный сигнал.

96) **АЦП** (Аналогово-Цифровой Преобразователь) преобразует аналоговые данные в цифровую форму.

97) **ЦАП** (Цифро-Аналоговый Преобразователь) выполняет обратную процедуру.

98) **Кодек** (кодер-декодер) - устройство, объединяющее в себе функции и АЦП, и ЦАП.

99) **Модем** (Модулятор-Демодулятор) преобразует цифровой сигнал в аналоговый в надлежащем диапазоне частот и наоборот.

100) Есть **три основных метода модуляции**:

- амплитудная модуляция
- частотная модуляция
- фазовая модуляция

101) **Несущий сигнал** - сигнал, один или несколько параметров которого изменяются в процессе модуляции, т.е. в процессе наложения информационного сигнала. Имеет свою амплитуду и частоту.

102) **Электромагни́тные во́лны** — электромагнитные колебания, распространяющиеся в пространстве с конечной скоростью, зависящей от свойств среды.

103) Одним из первых протоколов, разработанных для беспроводных локальных сетей, является **MACA** - Multiple Access with Collision Avoidance – множественный доступ с предотвращением столкновений.

Идея, лежащая в основе этого протокола, заключается в том, что отправитель перед началом передачи основной информации заставляет получателя передать короткий кадр, чтобы окружающие получателя станции могли услышать эту передачу и воздержаться от действий на время, требуемое для приема большого информационного кадра.

104) В **IEEE 802.11** описаны семейство стандартов, определяющих функционирование беспроводных локальных сетей - WiFi.

105) Первой и ключевой технологией стандарта 802.11 является технология расширения спектра передачи методом прямой последовательности (Direct Sequence Spread Spectrum – **DSSS**).

106) **Идея метода DSSS:**

- Пусть имеется канал с широкой полосой пропускания.
- Разобьем его на n полос.
- Введем избыточность для каждого бита - каждому значению бита сопоставим определенный код с длиной n. Отельный бит данного кода назовем чип. Существует большое число методов, позволяющих осуществить кодирование.
- Теперь будем передавать каждый бит, параллельно передавая его код, причем каждый чип в своей полосе.

107) **Беспроводные стандарты используют формат кадра**, отличный от Ethernet. В WiFi определены 3 типа кадров: 1. контрольные 2. управляющие 3. кадры данных.

108) Стандарт **IEEE 802.3** описывает технологию пакетной передачи Ethernet.

109) **Три общие класса сервиса,** которые может обеспечить канальный уровень**:**

1. Сервис без уведомления и без соединения
2. Сервис с уведомлением и без соединения
3. Сервис с уведомлением и с соединением

110) Общие допущения, действующие для всех протоколов канального уровня:

● Данные передаются только в одном направлении.

● Получатель и отправитель всегда готовы к отправке и получению данных.

● Буфер неограниченного размера.

111) Протоколы, реализующие идею начала передачи только после определения, занят канал или нет, называются **протоколами с обнаружением несущей** – CSMA (Carrier Sensitive Multiple Access).

112) **Обнаружение коллизий** – это аналоговый процесс, поэтому, чтобы обнаруживать их, необходимо использовать специальные кодировки на физическом уровне.

113) **Репитеры** - это устройство физического уровня, которое отвечает за очистку, усиление и передачу сигнала. Репитеры не могут отстоять более чем на 2,5 км, и на одном сегменте их не может быть более четырех.

114) **Трансивер** отвечает за обнаружение несущей частоты и коллизий. Когда трансивер обнаруживает коллизию, он посылает специальный сигнал по кабелю, гарантирующий, что другие трансиверы услышат эту коллизию.

115) **Сетевой коммутатор** представляет собой устройство с несколькими портами, к которым можно подключать сегменты с множественным доступом. По отношению к трафику каждый порт коммутатора может быть как входным, так и выходным.

116) Если адрес получателя кадра находится в том же сегменте сети, что и отправитель, коммутатор сбрасывает кадр. Этот процесс называется **фильтрацией**.

117) Если адрес получателя находится в другом сегменте, коммутатор пересылает кадр на порт, к которому подключен соответствующий сегмент. Это **пересылка**.

118) Если у коммутатора нет записи об адресе получателя, то он передаст кадр всем портам, кроме того порта, с которого кадр был получен, и номер соответствующего порта коммутатора в таблицу MAC-адресов - это процесс обучения. Это **лавинная рассылка**.

119) **Основное предназначение STP (Spanning Tree Protocol)** – удаление петель в топологии сети и автоматическое управление топологией сети с дублирующими каналами. Для своей работы STP строит дерево.

120) **Иерархическая маршрутизация** – маршрутизация с иерархией сетей подобной иерархии коммутаторов в телефонной сети.

121) В некоторых приложениях возникает потребность пересылки одного и того же сообщения всем машинам, такой режим передачи называется **вещанием**.

122) **Групповая передача** используется, когда надо обеспечить взаимодействие группы взаимосвязанных процессов, разбросанных по сети. Алгоритм групповой маршрутизации основывается на дереве связей.

123) Адреса выделяет только **N IC – Network Information Center**.

124) **Есть две таблицы.** Первая показывает как достичь интересующей сети. Вторая – как достичь узел внутри сети. Когда поступает IP пакет, маршрутизатор ищет его адрес доставки в таблице маршрутизации.

125) Управление функционированием Internet происходит через маршрутизаторы с помощью протокола **ICMP.** Этот протокол выявляет и рассылает сообщения о десятках событий в сети.

126) Протокол определения адреса. Как зная 32-разрядный сетевой IP-адрес получить адрес канального уровня, например, MAC-адрес. В подсеть посылается запрос: “У кого такой IP-адрес?”. Машина с указанным адресом шлет ответ. Протокол, который реализует рассылку запросов и сбор ответов - **ARP** протокол.

127) Обратный протокол определения адреса - **Reverse Address Resolution Protocol (RARP).**

128) Dynamic Host Configuration Protocol (**DHCP**, протокол динамической настройки узла) - сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

129) Проблема адресации состоит в том, как указать с каким удаленным прикладным процессом надо установить соединение? Для этого используется **TSAP – Transport Service Access Point.**

130) **Информация** - сведения (сообщения, данные) независимо от формы их представления.

131) **Безопасность информации** - состояние информационных ресурсов и информационных систем, при котором с требуемой вероятностью обеспечивается защита информации от утечки, хищения, утраты, несанкционированного уничтожения, искажения, модификации, копирования и блокирования.

132) **Целостность информации** - состояние защищенности информации, характеризуемое способностью КС обеспечивать сохранность и неизменность конфиденциальной информации при попытках несанкционированных или случайных воздействий на нее в процессе обработки или хранения.

133) **Информационная система** - совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств.

134) **Уязвимость информационной системы** – недостаток или слабое место в системном или прикладном программном (программно-аппаратном) обеспечении автоматизированной информационной системы, которые могут быть использованы для реализации угрозы безопасности информации.

135) **Несанкционированный доступ к информации (НСД)** - доступ к информации, нарушающий правила разграничения доступа с использованием штатных средств, предоставляемых средствами вычислительной техники или автоматизированными системами.

136) **Идентификация** - это процесс распознавания элемента системы, обычно с помощью заранее определенного идентификатора или другой уникальной информации - каждый субъект или объект системы должен быть однозначно идентифицируем.

137) **Аутентификация** - это проверка подлинности идентификации пользователя, процесса, устройства или другого компонента системы (обычно осуществляется перед разрешением доступа).

138) **Авторизация** - процедура предоставления субъекту определенных прав доступа к ресурсам системы после успешного прохождения им процедуры аутентификации. Для каждого субъекта в системе определяется набор прав, которые он может использовать при обращении к её ресурсам.

139) **Администрирование** - процесс управления доступом субъектов к ресурсам системы.

- создание идентификатора субъекта (создание учётной записи пользователя) в системе;
- управление данными субъекта, используемыми для его аутентификации (смена пароля, издание сертификата и т. п.);
- управление правами доступа субъекта к ресурсам системы.

140) **Аудит** - процесс контроля (мониторинга) доступа субъектов к ресурсам системы, включающий протоколирование действий субъектов при их работе с ресурсами.

141) Документ (**документированная информация**) - зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать.

142) **Информационные ресурсы** - документы и массивы документов в информационных системах.

143) **Информационные технологии** - процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов.

144) **Информационно-телекоммуникационная сеть** - технологическая система, предназначенная для передачи по линиям связи информации, доступ к которой осуществляется с использованием средств вычислительной техники.

145) **Доступ к информации (доступ)** - ознакомление с информацией, её обработка, в частности копирование, модификация или уничтожение информации.

146) **Доступность информации** - состояние информации, характеризуемое способностью КС обеспечивать беспрепятственный доступ к информации субъектам, имеющим на это полномочия.

147) **Несанкционированный доступ к информации (НСД)** - доступ к защищаемой информации с нарушением установленных прав или правил доступа, приводящий к получению субъектом возможности ознакомления с информацией и/или воздействия на нее.

148) **Защита информации** - деятельность, направленная на предотвращение утечки, разглашения, несанкционированного доступа (НСД) защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию.

149) **Защищаемая информация** - информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственником информации.

150) **Угроза безопасности КС** - потенциально возможное воздействие на информацию или другие компоненты КС, которое может прямо или косвенно привести к нанесению ущерба интересам субъектов информационных отношений.

151) **Угроза безопасности информации** - совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации.

152) **Фактор, воздействующий на защищаемую информацию** - явление, действие или процесс, результатом которого могут быть утечка, искажение, уничтожение защищаемой информации, блокирование доступа к ней.

153) **Источник угрозы безопасности информации** - субъект (физическое лицо, материальный объект или физическое явление), являющийся непосредственной причиной возникновения угрозы безопасности информации.

154) **Уязвимость (информационной системы); брешь** - свойство информационной системы, обусловливающее возможность реализации угроз безопасности обрабатываемой в ней информации.

155) **Атака на компьютерную систему** – это действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании уязвимости.

156) **Классификация угроз:**

1. по направленности реализации

2. по временным характеристикам воздействия

3. по объекту воздействия

157) **Монитор безопасности (МБ)** - субъект, активизирующийся при возникновении потока от любого субъекта к любому объекту и который разрешает поток, только из множества легального доступа L. МБ является механизмом реализации политики безопасности в КС. (L - подмножество потоков, характеризующих легальный доступ)

158) **Требования к МБ:**

1. Полнота: Монитор безопасности должен вызываться (активизироваться) при каждом обращении за доступом любого субъекта к любому объекту, и не должно быть никаких способов его обхода.

2. Изолированность: Монитор безопасности должен быть защищен от отслеживания и перехвата своей работы.

3. Верифицируемость: Монитор безопасности должен быть проверяемым (само- или внешне тестируемым) на предмет выполнения своих функций.

4. Непрерывность: Монитор безопасности должен функционировать любых штатных и нештатных, в том числе и аварийных, ситуациях.

159) **Политика безопасности** - общий принцип (методология, правило, схема) безопасной работы (доступа) коллектива пользователей с общими информационными ресурсами.

Выделяется две основных (базовых) политики безопасности: дискреционная (матричная) и мандатная (полномочий).

160) **Шифрование** – метод, используемый для преобразования данных в шифрованный текст для того, чтобы они были прочитаны только пользователем, обладающим соответствующим ключом шифрования для расшифровки содержимого.

161) Искусство создания шифра называют **криптографией**, а искусство его вскрытия – **криптоанализом**. Вместе эти дисциплины образуют **криптологию**.

162) **Методы обнаружения аномалий** используют описание нормального поведения наблюдаемых объектов в сети, и любое отклонение от нормального поведения считается аномальным – нарушением.

163) **Методы обнаружения злоупотреблений** используют описание запрещенных действий объектом в сети, например описание известных атак, и если наблюдаемое поведение некоторого объекта сети совпадает с описанием запрещенного, то действие объекта блокируют.

164) **Доменная система имен** – это метод, при котором в сетевой группе выделяется абонентская машина, отвечающая за назначение имен машинам в группе и обладающая полнотой информации о всех именах машин группы и их IP-адресах.

165) **Anycast** - метод рассылки пакетов “кому угодно” из получателей - на самом деле, ближайшему из них. В протоколе IP anycast реализован путём публикации одинакового маршрута из различных точек сети через протокол BGP.

Как это работает: BGP-роутер может знать несколько маршрутов, которые достигают сети с одним и тем же IP. При выборе пути для очередного пакета BGP будет выбирать кратчайший из них.

В настоящее время anycast используется в сети Internet для уменьшения времени отклика и для балансировки нагрузки корневых DNS-серверов

166) **Всемирная паутина WWW** основывается на использовании гипертекста.

167) Специальный протокол **HTTP** (HyperText Transmission Protocol) используется в Интернете с 1990 г., основан на парадигме запрос-ответ.

168) Для описания документов и связывания их гиперссылками служит язык **HTML** (Hyper Text Markup Language).

169) **Браузеры**- специальные программы для просмотра документов.

170) Для адресации на прикладном уровне большинство протоколов используют универсальные идентификаторы ресурса – **URI (Universal Resource Identifier)**.

171) **URL** – это идентификатор URI, который помимо идентификации ресурса предоставляет еще и информацию о ее местонахождении.

172) **URN** – это идентификатор URI, который идентифицирует ресурс в определенном пространстве имен.

173) **Код состояния HTTP (англ. HTTP status code)** - часть первой строки ответа сервера при запросах по протоколу HTTP. Пример: 201 – Webpage Created.

174) Клиент может не знать все коды состояния, но он обязан отреагировать в соответствии с классом кода. В настоящее время **выделено пять классов кодов состояния**:

- 1xx – Информационные.
- 2xx – Успех.
- 3xx – Перенаправление.
- 4xx – Ошибка клиента.
- 5xx – Ошибка сервера

175) Основная задача системы передачи почтовых сообщений – надежная доставка сообщения от отправителя к получателю. Самым простым способом в этом случае является использование простого протокола передачи почты – **SMTP** (Simple Mail Transfer Protocol).

176) **MIB** (Management Information Base) — локальная база данных.

177) **NAT** (Network Address Translation) – в общем случае технология изменения source или destination IP адреса в пакете на какой-то другой. Позволяет сэкономить IP-адреса.

178) **Центры обработки данных (ЦОД)** – это отказоустойчивая комплексная централизованная система, обеспечивающая автоматизацию бизнес-процессов с высоким уровнем производительности и качеством предоставляемых сервисов. В настоящее время в коммерческой среде наряду с определением Центр обработки данных (ЦОД) используется термин – Дата Центр (ДЦ).

179) **Cloud computing** – технология распределенной обработки данных, при которой некие масштабируемые информационные ресурсы и мощности предоставляются как сервис для многочисленных внешних клиентов посредством Интернет-технологий.

180) **Частное облако** (англ. private cloud) – инфраструктура, предназначенная для использования одной организацией, включающей несколько потребителей (например, подразделений одной организации), возможно также клиентами и подрядчиками данной организации. Частное облако может находиться в собственности, управлении и эксплуатации как самой организации, так и третьей стороны (или какой- либо их комбинации), и оно может физически существовать как внутри, так и вне юрисдикции владельца.

181) **Публичное облако** (англ. public cloud) – инфраструктура, предназначенная для свободного использования широкой публикой. Публичное облако может находиться в собственности, управлении и эксплуатации коммерческих, научных и правительственных организаций (или какой-либо их комбинации). Публичное облако физически существует в юрисдикции владельца – поставщика услуг.

182) **Общественное облако** (англ. community cloud) – вид инфраструктуры, предназначенный для использования конкретным сообществом потребителей из организаций, имеющих общие задачи (например, миссии, требований безопасности, политики, и соответствия различным требованиям). Общественное облако может находиться в кооперативной (совместной) собственности, управлении и эксплуатации од- ной или более из организаций сообщества или третьей стороны (или какой-либо их комбинации), и оно может физически существовать как внутри, так и вне юрисдикции владельца.

183) **Гибридное облако** (англ. hybrid cloud) – это комбинация из двух или более различных облачных инфраструктур (частных, публичных или общественных), остающихся уникальными объектами, но связанных между собой стандартизованными или частными технологиями передачи данных и приложений (например, кратковременное использование ресурсов публичных облаков для балансировки нагрузки между облаками).

184) **Openflow** – протокол управления процессом обработки данных, передающихся по сети маршрутизаторами и коммутаторами, реализующий технологию программно-конфигурируемой сети (ПКС).

185) Протокол используется для управления коммутаторами и маршрутизаторами с центрального устройства - **контроллера сети** (например, с сервера или даже персонального компьютера).

186) Контроллер используется для управления **таблицами потоков коммутаторов**, на основании которых принимается решение о передаче принятого пакета на конкретный порт коммутатора. Таким образом в сети формируются прямые соединения с минимальными задержками и необходимыми параметрами.

187) **Программно-конфигурируемая сеть** (software-defined networking, SDN) — сеть, в которой уровень управления сетью отделен от устройств передачи данных и реализуется программно.

188) **Принцип действия ПКС:**

Разделение тракта данных и тракта управления. OpenFlow отделяет друг от друга функции тракта данных и управления, обычно реализуемые коммутаторами. Функциональность, относящаяся к тракту данных, по-прежнему выполняется на коммутаторе, но за принятие решений о высокоуровневой маршрутизации в OpenFlow отвечает контроллер, как правило организованный на базе стандартного сервера. Коммутатор и контроллер общаются по протоколу OpenFlow Switching Protocol. Контроллер может, например, приказать коммутаторам ввести в действие правила для потоков сетевого трафика. Такие правила могут, в частности, обеспечивать отправку данных по самым быстрым маршрутам или по маршрутам, имеющим минимум транзитных участков.

Интерфейс. OpenFlow предоставляет единый API, с помощью которого администраторы могут программировать работу сети, а также задавать правила маршрутизации пакетов, балансировки нагрузки и управления доступом. В этот API входят два основных компонента: программный интерфейс для контроля пересылки пакетов через сетевые коммутаторы и набор глобальных интерфейсов, на основе которых можно создавать высокоразвитые инструменты управления.