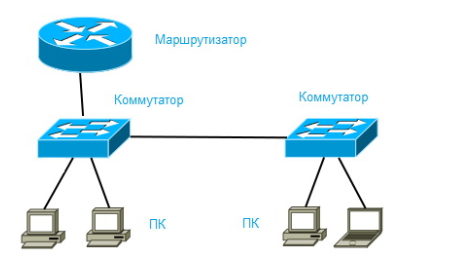
***Билет №1***

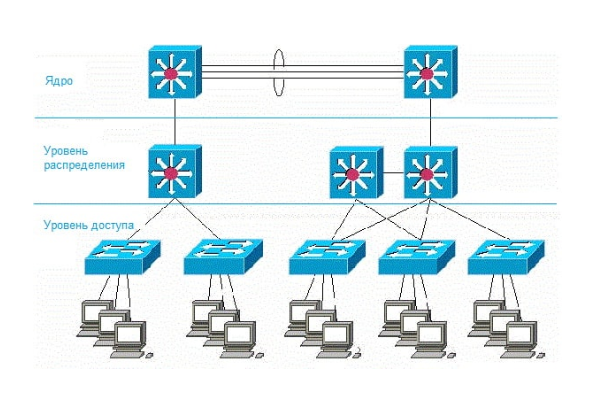
**1.Коммутаторы. Назначение и применение в локальных сетях.**

Коммутаторы одно из важнейших устройств при построении локальной сети.

Так выглядит стандартная схема использования коммутаторов:



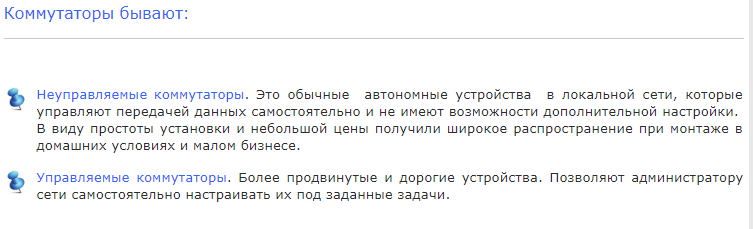
Коммутаторы доступа подключены к конечным пользователям и дают им доступ к ресурсам локальной сети.

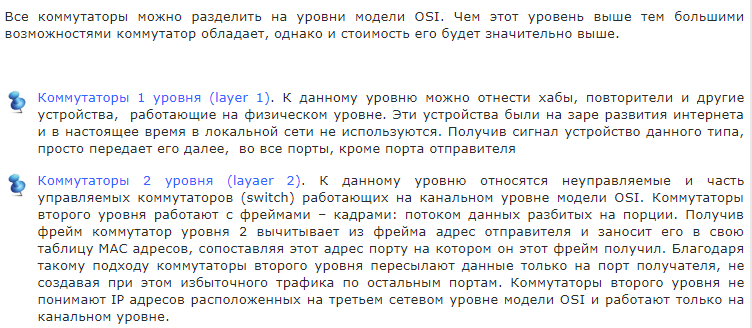
Однако в крупных локальных сетях они выполняют след. Функции: 

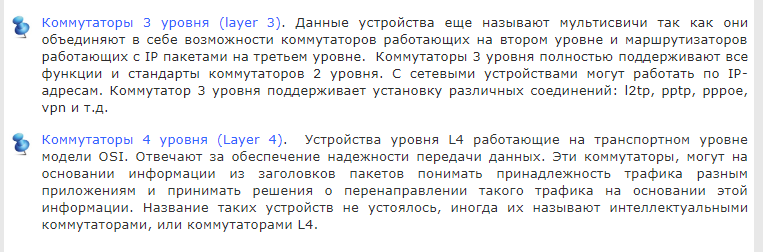
**Уровень доступа сети**. Здесь коммутаторы также как и выше, подключены к конечным пользователям. В крупных локальках фреймы коммутатор не взаимодействуют с другими коммутаторами напрямую, они передаются через уровень расперделения

**Уровень распределения сети.** Пересылают трафик между коммутаторами уровня доступа, но не подключены к пользователям напрямую

**Уровень ядра системы.** Устройства на данном уровне объединяет крупные локальные сети и обеспечивают быструю скорость передачи между ними.

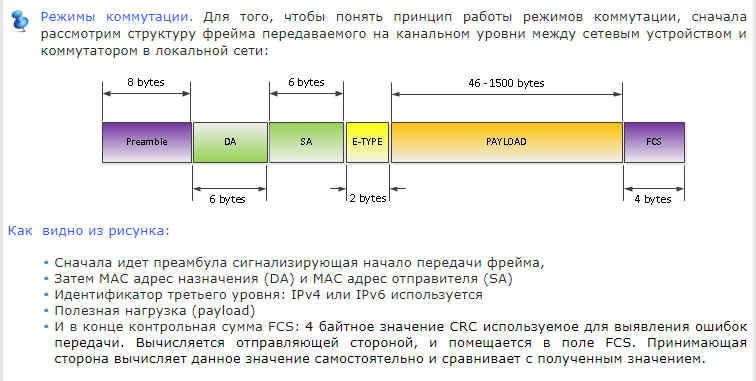






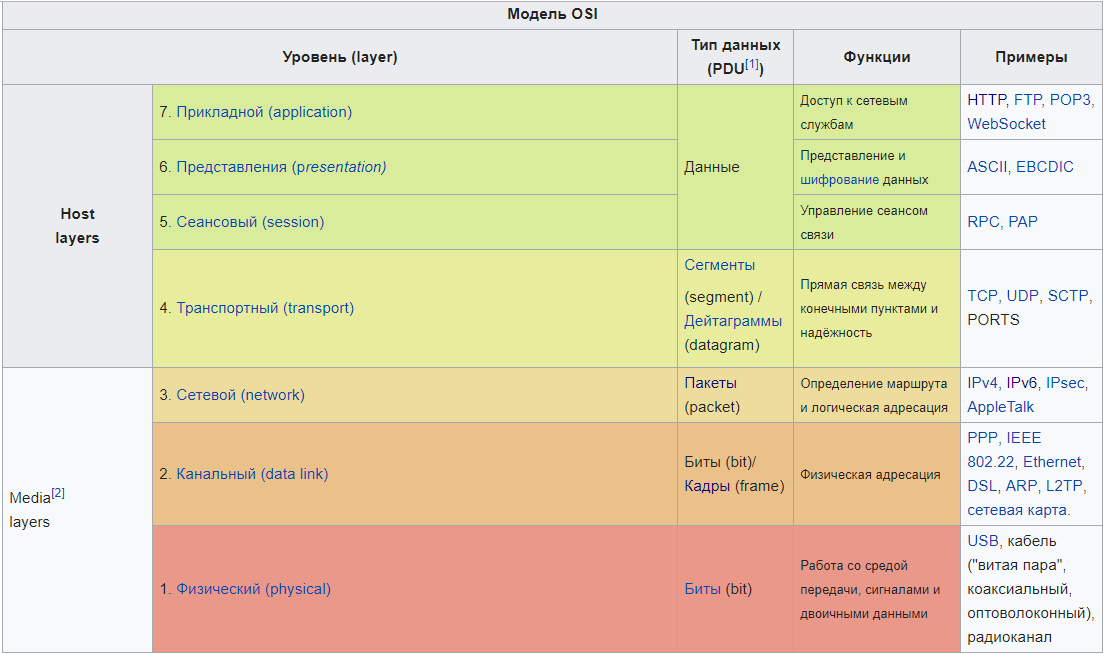
Основные характеристики коммутаторов:

1. Количество портов
2. Скорость передачи данных
3. Режим передачи Full/half-duplex.
4. Внутренняя пропускная способность
5. Размер таблицы mac-адресов



http://bp-kuban.ru/stati/item/switch-lan

**2.Модель OSI/ISO**



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.

**Прикладной уровень**

*Основная статья:*[***Прикладной уровень***](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Прикладной уровень (уровень приложений; [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *application layer*) — верхний уровень модели, обеспечивающий взаимодействие пользовательских приложений с сетью:

* позволяет приложениям использовать сетевые службы:
  + удалённый доступ к файлам и базам данных,
  + пересылка электронной почты;
* отвечает за передачу служебной информации;
* предоставляет приложениям информацию об ошибках;
* формирует запросы к уровню представления.

Протоколы прикладного уровня: [RDP](https://ru.wikipedia.org/wiki/Remote_Desktop_Protocol), [HTTP](https://ru.wikipedia.org/wiki/HTTP), [SMTP](https://ru.wikipedia.org/wiki/SMTP), [SNMP](https://ru.wikipedia.org/wiki/SNMP), [POP3](https://ru.wikipedia.org/wiki/Post_Office_Protocol), [FTP](https://ru.wikipedia.org/wiki/FTP), [XMPP](https://ru.wikipedia.org/wiki/XMPP), [OSCAR](https://ru.wikipedia.org/wiki/OSCAR), [Modbus](https://ru.wikipedia.org/wiki/Modbus), [SIP](https://ru.wikipedia.org/wiki/SIP), [TELNET](https://ru.wikipedia.org/wiki/TELNET) и другие.

**Уровень представления**

*Основная статья:*[***Уровень представления***](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C_%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F)

Зачастую ошибочно называемый представительским уровнем, этот уровень ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *presentation layer*) обеспечивает преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с прикладного уровня, на уровне представления преобразуются в формат для передачи по сети, а полученные из сети данные преобразуются в формат приложений. На этом уровне может осуществляться сжатие/распаковка или шифрование/дешифрование, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

Уровень представлений обычно представляет собой промежуточный протокол для преобразования информации из соседних уровней. Это позволяет осуществлять обмен между приложениями на разнородных компьютерных системах прозрачным для приложений образом. Уровень представлений обеспечивает форматирование и преобразование кода. Форматирование кода используется для того, чтобы гарантировать приложению поступление информации для обработки, которая имела бы для него смысл. При необходимости этот уровень может выполнять перевод из одного формата данных в другой.

Уровень представлений имеет дело не только с форматами и представлением данных, он также занимается структурами данных, которые используются программами. Таким образом, уровень 6 обеспечивает организацию данных при их пересылке.

**Сеансовый уровень** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *session layer*) модели обеспечивает поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. Уровень управляет созданием/завершением сеанса, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений.

**Транспортный уровень**[[править](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&section=5) | [править код](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=5)]

*Основная статья:*[***Транспортный уровень***](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Транспортный уровень ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *transport layer*) модели предназначен для обеспечения надёжной передачи данных от отправителя к получателю. При этом уровень надёжности может варьироваться в широких пределах. Существует множество классов протоколов транспортного уровня, начиная от протоколов, предоставляющих только основные транспортные функции (например, функции передачи данных без подтверждения приёма), и заканчивая протоколами, которые гарантируют доставку в пункт назначения нескольких пакетов данных в надлежащей последовательности, мультиплексируют несколько потоков данных, обеспечивают механизм управления потоками данных и гарантируют достоверность принятых данных. Например, [UDP](https://ru.wikipedia.org/wiki/UDP) ограничивается контролем целостности данных в рамках одной датаграммы и не исключает возможности потери пакета целиком или дублирования пакетов, нарушение порядка получения пакетов данных; [TCP](https://ru.wikipedia.org/wiki/TCP) обеспечивает надёжную непрерывную передачу данных, исключающую потерю данных или нарушение порядка их поступления или дублирования, может перераспределять данные, разбивая большие порции данных на фрагменты и наоборот, склеивая фрагменты в один пакет.

**Сетевой уровень**[[править](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&section=6) | [править код](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=6)]

*Основная статья:*[***Сетевой уровень***](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Сетевой уровень ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *network layer*) модели предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и «заторов» в сети.

Протоколы сетевого уровня маршрутизируют данные от источника к получателю. Работающие на этом уровне устройства ([маршрутизаторы](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80)) условно называют устройствами третьего уровня (по номеру уровня в модели OSI).

**Канальный уровень**[[править](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&section=7) | [править код](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=7)]

*Основная статья:*[***Канальный уровень***](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Канальный уровень ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *data link layer*) предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля ошибок, которые могут возникнуть. Полученные с физического уровня данные, представленные в битах, он упаковывает в [кадры](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B4%D1%80_(%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8)), проверяет их на целостность и, если нужно, исправляет ошибки (формирует повторный запрос повреждённого кадра) и отправляет на сетевой уровень. Канальный уровень может взаимодействовать с одним или несколькими физическими уровнями, контролируя и управляя этим взаимодействием.

Спецификация [IEEE 802](https://ru.wikipedia.org/wiki/IEEE_802) разделяет этот уровень на два подуровня: [MAC](https://ru.wikipedia.org/wiki/Media_Access_Control) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *media access control*) регулирует доступ к разделяемой физической среде, [LLC](https://ru.wikipedia.org/wiki/Logical_link_control) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *logical link control*) обеспечивает обслуживание сетевого уровня.

На этом уровне работают [коммутаторы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D1%82%D0%B0%D1%82%D0%BE%D1%80), [мосты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BC%D0%BE%D1%81%D1%82) и другие устройства. Эти устройства используют адресацию второго уровня (по номеру уровня в модели OSI).

**Физический уровень**[[править](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&veaction=edit&section=8) | [править код](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI&action=edit&section=8)]

*Основная статья:*[***Физический уровень***](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)

Физический уровень ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *physical layer*) — нижний уровень модели, который определяет метод передачи данных, представленных в двоичном виде, от одного устройства (компьютера) к другому. Составлением таких методов занимаются разные организации, в том числе: [Институт инженеров по электротехнике и электронике](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%BE%D0%B2_%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B8_%D0%B8_%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%B8%D0%BA%D0%B8), [Альянс электронной промышленности](https://ru.wikipedia.org/wiki/Electronic_Industries_Alliance), [Европейский институт телекоммуникационных стандартов](https://ru.wikipedia.org/wiki/%D0%95%D0%B2%D1%80%D0%BE%D0%BF%D0%B5%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82_%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D1%85_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BE%D0%B2) и другие. Осуществляют передачу электрических или оптических сигналов в кабель или в радиоэфир и, соответственно, их приём и преобразование в биты данных в соответствии с [методами кодирования цифровых сигналов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BA%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

На этом уровне также работают [концентраторы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BE%D0%BD%D1%86%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80), [повторители](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D1%82%D0%B5%D0%BB%D1%8C_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B5_%D0%BE%D0%B1%D0%BE%D1%80%D1%83%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) сигнала и [медиаконвертеры](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D0%B0%D0%BA%D0%BE%D0%BD%D0%B2%D0%B5%D1%80%D1%82%D0%B5%D1%80" \o "Медиаконвертер).

Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом. К физическому уровню относятся физические, электрические и механические интерфейсы между двумя системами. Физический уровень определяет такие виды сред передачи данных как [оптоволокно](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B2%D0%BE%D0%BB%D0%BE%D0%BA%D0%BD%D0%BE), [витая пара](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%82%D0%B0%D1%8F_%D0%BF%D0%B0%D1%80%D0%B0), [коаксиальный кабель](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B0%D0%BA%D1%81%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%B0%D0%B1%D0%B5%D0%BB%D1%8C), спутниковый канал передач данных и т.п. Стандартными типами сетевых интерфейсов, относящимися к физическому уровню, являются: [V.35](https://ru.wikipedia.org/wiki/V.35), [RS-232](https://ru.wikipedia.org/wiki/RS-232), [RS-485](https://ru.wikipedia.org/wiki/RS-485), [RJ-11](https://ru.wikipedia.org/wiki/RJ-45), [RJ-45](https://ru.wikipedia.org/wiki/RJ-45), разъёмы [AUI](https://ru.wikipedia.org/wiki/AUI) и [BNC](https://ru.wikipedia.org/wiki/BNC-%D0%BA%D0%BE%D0%BD%D0%BD%D0%B5%D0%BA%D1%82%D0%BE%D1%80).

***Билет №2***

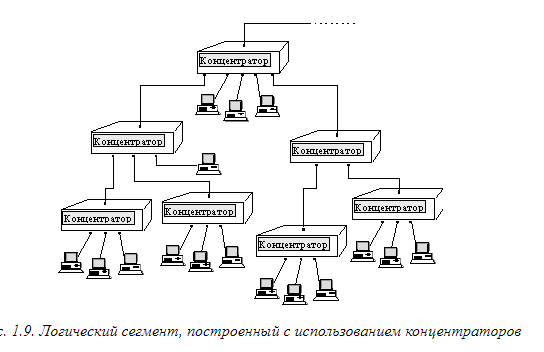
**Повторители. Назначение и применение в локальных сетях**

Основная функция *повторителя* (repeater), как это следует из его названия - повторение сигналов, поступающих на один из его портов, на всех остальных портах (Ethernet) или на следующем в логическом кольце порте (Token Ring, FDDI) синхронно с сигналами-оригиналами. Повторитель улучшает электрические характеристики сигналов и их синхронность, и за счет этого появляется возможность увеличивать общую длину кабеля между самыми удаленными в сети станциями.

Многопортовый повторитель часто называют *концентратором* (hub, concentrator), что отражает тот факт, что данное устройство реализует не только функцию повторения сигналов, но и концентрирует в одном центральном устройстве функции объединения компьютеров в сеть. Практически во всех современных сетевых стандартах концентратор является необходимым элементом сети, соединяющим отдельные компьютеры в сеть.

Отрезки кабеля, соединяющие два компьютера или какие либо два других сетевых устройства называются *физическими сегментам*. Таким образом, концентраторы и повторители, которые используются для добавления новых физических сегментов, являются средством физической структуризации сети.

Концентраторы образуют из отдельных физических отрезков кабеля общую среду передачи данных - *логический сегмент* (рис. 1.8). Логический сегмент также называют доменом коллизий, поскольку при попытке одновременной передачи данных любых двух компьютеров этого сегмента, хотя бы и принадлежащих разным физическим сегментам, возникает блокировка передающей среды. Следует особо подчеркнуть, что какую бы сложную структуру не образовывали концентраторы, например, путем иерархического соединения (рис. 1.9), все компьютеры, подключенные к ним, образуют единый логический сегмент, в котором любая пара взаимодействующих компьютеров полностью блокирует возможность обмена данными для других компьютеров.

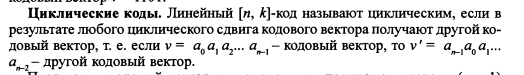


Появление устройств, централизующих соединения между отдельными сетевыми устройствами, потенциально позволяет улучшить управляемость сети и ее эксплуатационные характеристики (модифицируемость, ремонтопригодность и т.п.). С этой целью разработчики концентраторов часто встраивают в свои устройства, кроме основной функции повторителя, ряд вспомогательных функций, весьма полезных для улучшения качества сети.

Различные производители концентраторов реализуют в своих устройствах различные наборы вспомогательных функций, но наиболее часто встречаются следующие:

* Объединение сегментов с различными физическими средами (например, коаксиал, витая пара и оптоволокно) в единый логический сегмент.
* Автосегментация портов - автоматическое отключение порта при его некорректном поведении (повреждение кабеля, интенсивная генерация пакетов ошибочной длины и т.п.).
* Поддержка между концентраторами резервных связей, которые используются при отказе основных.
* Защита передаваемых по сети данных от несанкционированного доступа (например, путем искажения поля данных в кадрах, повторяемых на портах, не содержащих компьютера с адресом назначения).
* Поддержка средств управления сетями - протокола SNMP, баз управляющей информации MIB

**Циклический код. Синдром ошибки**.



Тут короче вспоминаем ДЗ

***Билет №3***

**Концентраторы. Назначение и применение в локальных сетях**.

См. Билет №2.

**Технология расширенного спектра в беспроводных сетях. Общие принципы.**

### Технология расширения спектра методом прямой последовательности (DSSS)

Основная идея технологии расширения спектра (Spread Spectrum, SS) заключается в том, чтобы от узкополосного спектра сигнала, возникающего при обычном потенциальном кодировании, перейти к широкополосному спектру. Именно это позволяет значительно повысить помехоустойчивость передаваемых данных.  Бит кодовой последовательности называют чипом. В стандарте IEEE 802.11 в качестве кодовой последовательности используется 11-ти элементный код Баркера, который складывается по модулю 2 с каждым битом информации[[3]](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D1%80%D0%B0%D1%81%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%B8%D1%8F_%D1%81%D0%BF%D0%B5%D0%BA%D1%82%D1%80%D0%B0_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BC_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D0%B9_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8#cite_note-3). В результате спектр сигнала расширяется в 11 раз. При приёме полученная последовательность чипов декодируется путём сложения по модулю 2 принятой последовательности чипов с той же кодовой последовательностью.

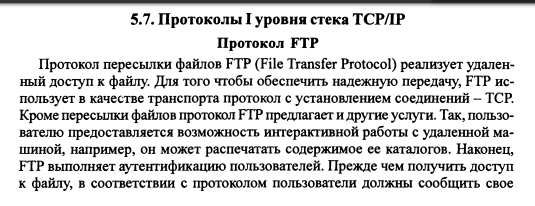
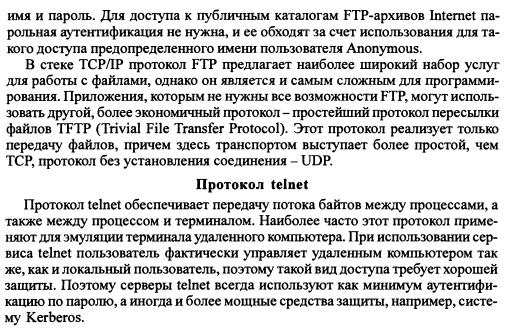
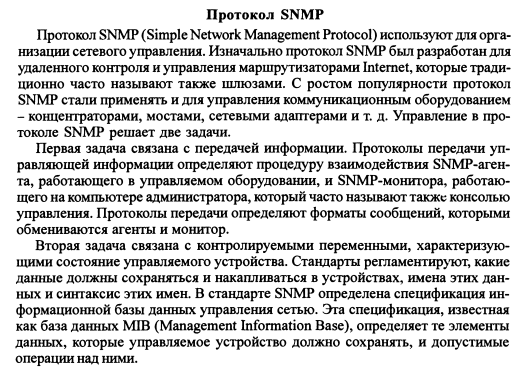
**Псевдослучайная перестройка рабочей частоты** (FHSS — [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***f****requency-****h****opping****s****pread****s****pectrum*) — метод передачи информации по [радио](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B8%D0%BE), особенность которого заключается в частой смене [несущей частоты](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D1%81%D1%83%D1%89%D0%B8%D0%B9_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB). Частота меняется в соответствии с [псевдослучайной последовательностью чисел](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB), известной как отправителю, так и получателю. Метод повышает помехозащищённость канала связи. Основным недостатком ППРЧ является низкая скорость передачи данных

**Линейная частотная модуляция (CSS)** сигнала — это вид [частотной модуляции](https://ru.wikipedia.org/wiki/%D0%A7%D0%B0%D1%81%D1%82%D0%BE%D1%82%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F), при которой частота [несущего сигнала](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D1%81%D1%83%D1%89%D0%B8%D0%B9_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB) изменяется по линейному закону.

***Билет №4***

**Механизм мультиплексирования в протоколах стека TCP/IP**

Протокол UDP ведет для каждого порта две очереди: очередь пакетов, поступающих в данный порт из сети, и очередь пакетов, отправляемых данным портом в сеть. Процедура обслуживания протоколом UDP-запросов, поступающих от нескольких различных прикладных сервисов, называется мультиплексированием. Распределение протоколом UDP поступающих от сетевого уровня пакетов между набором высокоуровневых сервисов, идентифицированных номерами портов, называется демультиплексированием. Концептуально все процессы мультиплексирования и демультиплексирования между UDP и прикладными программами осуществляются с помощью механизма портов. На практике каждая прикладная программа должна договориться с операционой системой о получении протокольного порта и связанного с ним номера перед посьшкой UDP-дейтаграммы. Когда порт выделен, прикладная программа посьшает любую дейтаграмму через порт, номер которого указан в поле «Порт отправителя» UDP-заголовка. В ходе обработки входных данных 409 5. Сетевые протоколы UDP принимает приходящие от IP дейтаграммы и демультиплексирует их по портам назначения. UDP использует номер порта получателя UDP для выбора соответствующего получателя для пришедшей дейтаграммы. Порт UDP легче всего представить в виде очереди. В большинстве реализащш, когда прикладная программа договаривается с операщюной системой об использовании данного порта, операщюнная система создает внутреннюю очередь, которая хранит приходящие сообщения. Часто приложение может указать или изменить размеры очереди. Когда UDP получает дейтаграмму, он проверяет, нет ли порта назначения с таким номером среди используемых портов. Если нет, он посьшает ICMP-сообщение об ошибке «порт недоступен» и уничтожает дейтаграмму. Если есть, UDP добавляет новую дейтаграмму в очередь порта, где прикладная программа может ее получить. Конечно, если очередь порта уже переполнена, то тогда UDP уничтожает новую дейтаграмму. К услугам протокола UDP может обратиться любое приложите, однако многие из них предпочитают иметь дело с более сложным протоколом транспортного уровня - TCP. Дело в том, что протокол UDP выступает простым посредником между сетевым уровнем и прикладными сервисами, и, в отличие от TCP, не берет на себя никаких фуншщй по обеспечению надежности передачи. Протокол UDP, как уже отмечалось, является дейтаграммным протоколом, т. е. он не устанавливает логического соединения, не нумерует и не упорядочивает пакеты данных. С другой стороны, фунюдиональная простота протокола UDP обусловливает простоту его алгоритма, компактность и высокое быстродействие. Поэтому те приложения, в которых реализован собственный, достаточно надежный механизм обмена сообщениями, основанный на установлении соединения, предпочитают для непосредственной передачи данных по сети использовать менее надежные, но более быстрые средства транспортировки, каким является протокол UDP. Протокол UDP можно использовать и в том случае, когда хорошее качество каналов связи обеспечивает достаточный уровень надежности и без применения дополнительных приемов типа установления логического соединения и квитирования передаваемых пакетов.

**Синхронизация на канальном уровне. Синхронные протоколы канального уровня**

На канальном уровне происходит синхронизация  передачи кадров. Приемник должен распознать начало первого байта кадра, границы полей кадра и признак окончания кадра. Существует три способа синхронизации передачи кадров:

- асинхронная (Asynchronous)

- синхронная (Synchronous)

- изохронная (Isochronous)

Способ синхронизации имеет важное значение в сети, так как неверный выбор может привести к задержкам и уменьшению производительности.

**Асинхронная передача**

При асинхронной передаче часы приемника и передатчика не синхронизированы, хотя и используют одинаковые временные интервалы. Каждый кадр посылается отдельно, он начинается стартовым битом и заканчивается стоповым битом. Кадр относительно короток, поэтому не происходит ошибок, связанных с несогласованностью часов приемника и передатчика.

Такой метод часто используется при передаче символов, которые передаются через случайные промежутки времени, например в терминале. Стартовый и стоповый биты не только управляют запуском таймера приемника, но и устанавливают границы кадра.

Хотя асинхронна передача и кажется относительно простой, в ней часто возникают ошибки. Для контроля ошибок к концу каждого байта в кадре добавляется специальный бит, называемый битом четности (parity bit). Он равен единице, если в байте находится четное число единиц. Главный недостаток такого метода – он не может определять множественные ошибки. На практике такой метод применяется редко из-за низкой надежности и большой избыточности информации.

В следующей таблице приведены преимущества и недостатки асинхронной передачи.

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Простая технология и недорогое оборудование | Добавляют 20-30% избыточной информации |
|  | Не опознаются множественные битовые ошибки |
|  | Низкая скорость передачи |

**Синхронная передача**

Этот вид передачи требует общего отсчета времени, что может быть реализовано двумя способами:

- с помощью передачи специальной строки битов (бит-ориентированные протоколы) или нескольких символов (байт-ориентированные протоколы, SYNC)

- используя отдельный канал синхронизации

Здесь не существует стартовых и стоповых битов, поэтому отдельные символы при синхронной передаче пересылать нельзя. Все обмены данных осуществляются кадрами, которые имеют заголовок и концевик. Это напоминает асинхронную передачу, но там стартовые и стоповые биты служили для других целей. Благодаря синхронизации таймеров приемника и передатчика синхронная передача более устойчива к сбоям. Для поддержки связи при отсутствии данных, в такой передаче используются биты заполнения (fill bits). С помощью битов заполнения передача информации по каналу не прерывается.

Синхронная передача более эффективна при большом трафике, так как количество избыточной информации в ней меньше чем при асинхронной передаче.

Для контроля ошибок используют метод циклического избыточного контроля (CRC, Cyclic Redundancy Check, контрольная сумма). К каждому участку данных добавляется код CRC, полученный по специальному алгоритму. При приеме CRC код вновь подчитывается и сравнивается с полученным. Код CRC намного более эффективно обнаруживает ошибки чем использование четности.

Байт-ориентированные протоколы в основном используются для передачи блоков символов, например текстовых файлов. Синхронизация осуществляется с помощью управляющих символов Synch, которые добавляются перед каждым блоком символов.

Но для передачи двоичных данных невыгодно добавлять символы в поле данных кадра, так как добавляется много избыточной информации. Сейчас применяется более универсальный метод – бит-ориентированный протокол. Здесь используются не символы, а специальные флаги – группа битов, поэтому длина кадра может быть не кратна 8 битам.

В следующей таблице приведены преимущества и недостатки синхронной передачи.

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Более эффективный способ передачи по сравнению с асинхронным | Более сложное и дорогое аппаратное обеспечение приемника и передатчика |
| Более высокая скорость передачи |  |
| Улучшенный контроль ошибок |  |

**Изохронная передача**

При изохронной передаче используется таймер с постоянной и заранее заданной частотой. Сигнал таймера передается через сеть на все остальные устройства.

В соответствии с установленными правилами сетевые устройства определяют свободные временные промежутки и посылают кадры во время этих промежутков. В данном случае сигнал синхронизации не присутствует в начале кадра как при асинхронной передаче и не посылается в начале строки как при синхронной. Сигнал передает отдельное устройство, которое само не участвует в передаче данных.

В следующей таблице приведены преимущества и недостатки изохронной передачи.

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Гарантированная определенная скорость передачи | Требуется безотказный внешний таймер |
| Передается крайне небольшое количество дополнительной информации |  |

***Билет №5***

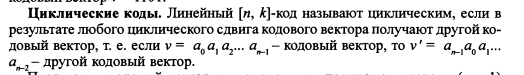
**Мост. Назначение и применение в локальных сетях**.

Мост - это устройство, которое обеспечивает взаимосвязь двух (реже нескольких) локальных сетей посредством передачи кадров из одной сети в другую с помощью их промежуточной буферизации. Мост, в отличие от повторителя, не старается поддержать побитовый синхронизм в обеих объединяемых сетях. Вместо этого он выступает по отношению к каждой из сетей как конечный узел. Он принимает кадр, буферизует его, анализирует адрес назначения кадра и только в том случае, когда адресуемый узел действительно принадлежит другой сети, он передает его туда.

Мост не только снижает нагрузку в объединенной сети, но и уменьшает возможности несанкционированного доступа, так как пакеты, предназначенные для циркуляции внутри одного сегмента, физически не появляются на других, что исключает их "прослушивание" станциями других сегментов.

**Книга:** Мосты (bridge) имеют много отличий от повторителей. Повторители передают все пакеты, а мосты только те, которые необходимы. Если пакет не нужно передавать в другой сегмент, он фильтруется. Для мостов существуют многочисленные алгоритмы (правила) передачи и фильтрации пакетов. Минимальным требованием является фильтрация пакетов по МАС-адресу получателя.

**Циклический код. Пример**



ДЗ

***Билет №6***

**Архитектура РСОД**.

РСОД (Распределенная система обработки данных) – связанные между собой ЭВМ, с помощью которых происходит обработка информации

Она подразумевает тот или иной вид сети, который обеспечивает распределение и обработку:

1. БД
2. Вычисл. Мощность
3. Управление числ.

Функции:

1. Доступ к вычислительным ресурсам в режиме файл-сервер
2. Выполнение заданий и коммуникация клиентов в режиме клиент-сервер

Клиент-сервер: На сервере данные, все вычисления. Клиент получает результаты, а на файл-сервере по другому.

1. Сбор статистики о функционировании системы
2. Обеспечение живучести и надёжности системы

Признаки:

1. По степени однородности  
   1. Полностью не однородные,   
   2. Частично однородные,   
   3. Полностью однородные РСОД.
2. По архитектуре особенностями:
3. РСОД на основе систем телеобработки
4. РСОД на основе сетевой технологии
5. По степени распределения:
6. Региональные
7. Локальные

Полн. Не однородные – Разноархитектурная сеть, возможно разная ОС.

Частично не однородная – Различн ОС, но одинаковая архитектура или наоборот.

Полностью не однородная – Один ОС и одна архитектура.

РСОД на основе систем телеком – не обеспечивает полного симметричн. Обработки информации, как следствие ограниченные возможности.

РСОД на осн. Сетевой технологии - такая форма взаим. ЭВМ, при котором любой из процессов одной из машин может установить лигочн. Связь с любым процессом в любой другой ЭВМ.

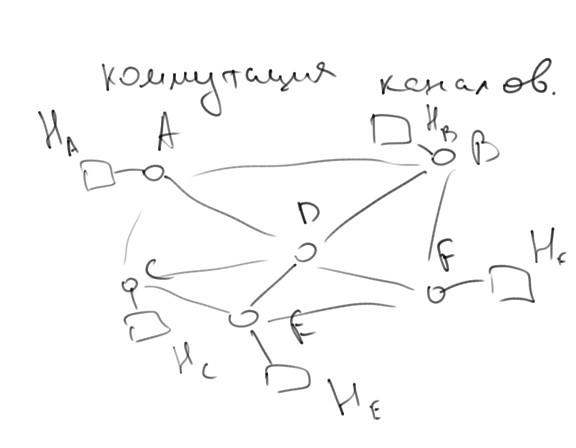
Региональные(Глобальные):

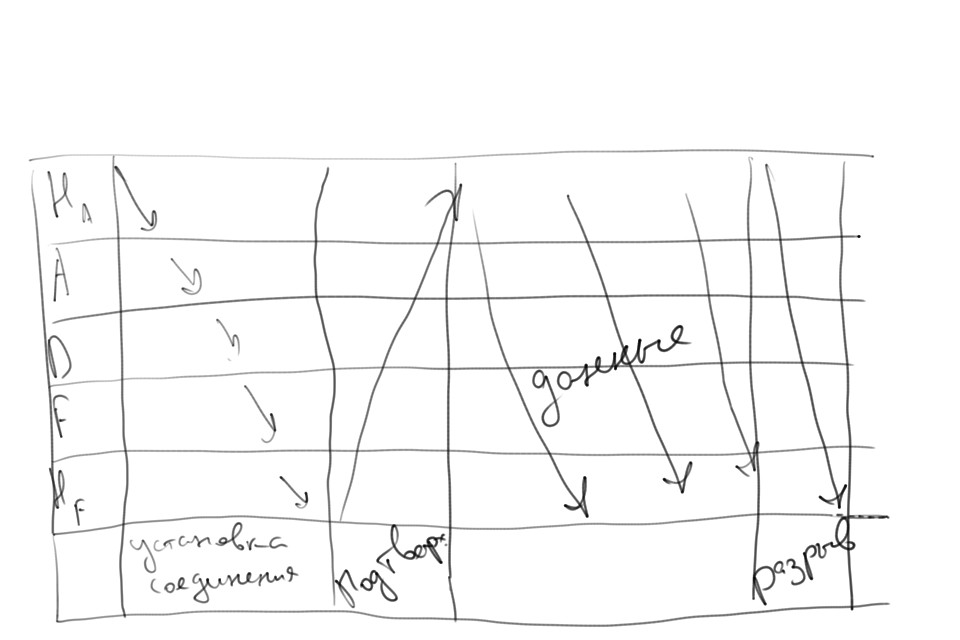
Характер:

1. Неогранич. Географич распределенностью
2. Наличие тех или иных механизмов маршрутизации
3. Каждые смежные 2 узла связаны собствен. Каналом, и отсутствие проблемы его распределения
4. Широкий диапазон скоростной передачи 103-108 бит/с
5. Произвольной топологией

В них можно выделить несколько способов орг. Взаимодействовать между ЭВМ.

1. Коммуникации каналов (КК)
2. Коммуникации саообщен. (КС)
3. Коммуникации пакетов (КП)



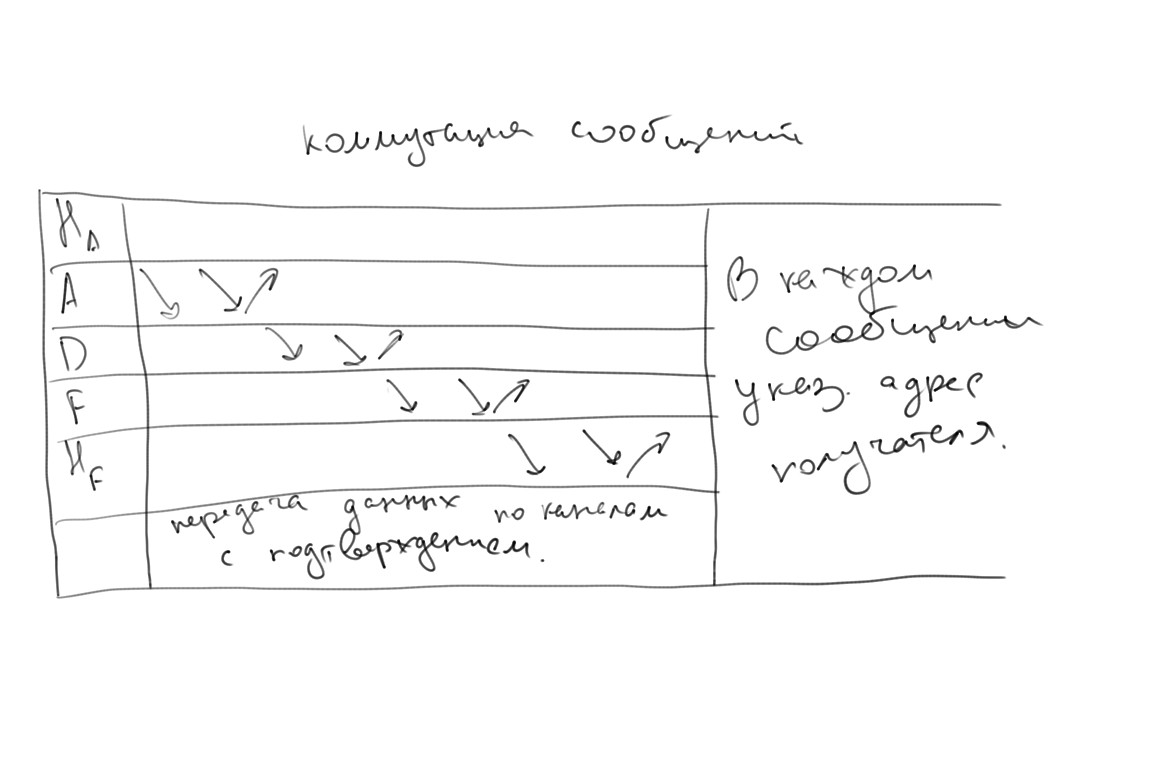


Недостатки:

1. Сформированный (скомутированный) канал закрепляется на всем участке передачи на всё это время (они заняты и не могут пользоваться этими каналами)
2. Скорость передачи и приёма должны бы согласованы (совместимы)

Плюсы:

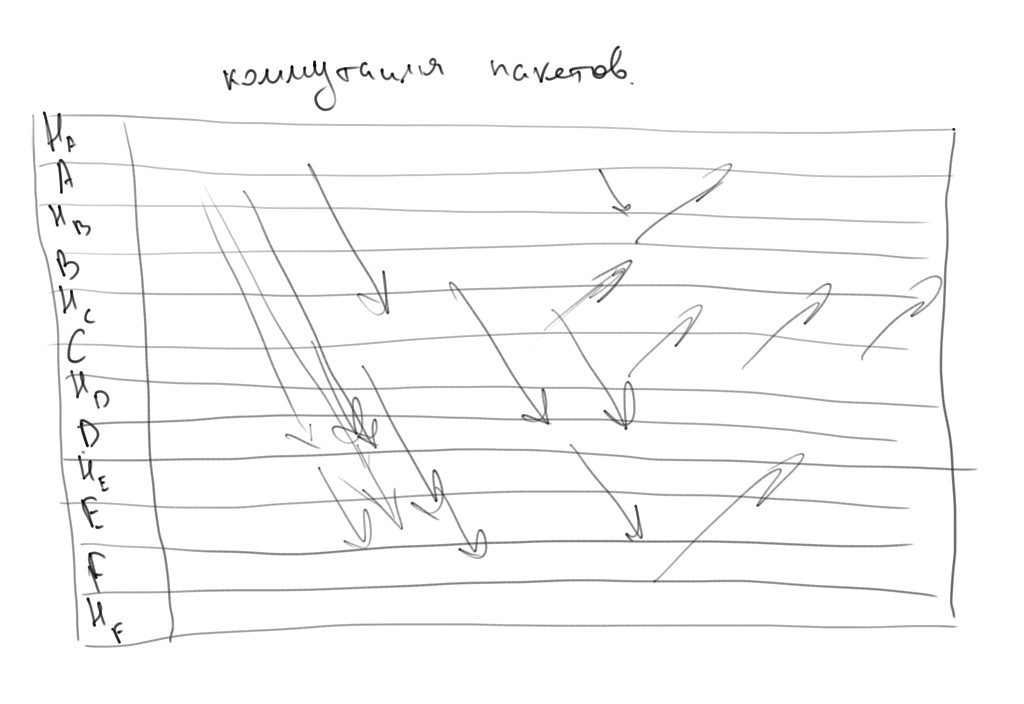
1. После формирования канала задержек нет, передача файлов в реальном времени
2. Адрес назначения указ только 1 раз.

Недостатки:

1. В каждом узле буферной памяти должно хватить памяти для передачи всех данных
2. Узлы должны быть развиты, выбирать маршрут
3. Длительное время передачи
4. Длинные сообщения неэффективны

Плюсы:

1. Короткие сообщения, пропускная способность
2. Скорость передачи не должна быть согласована

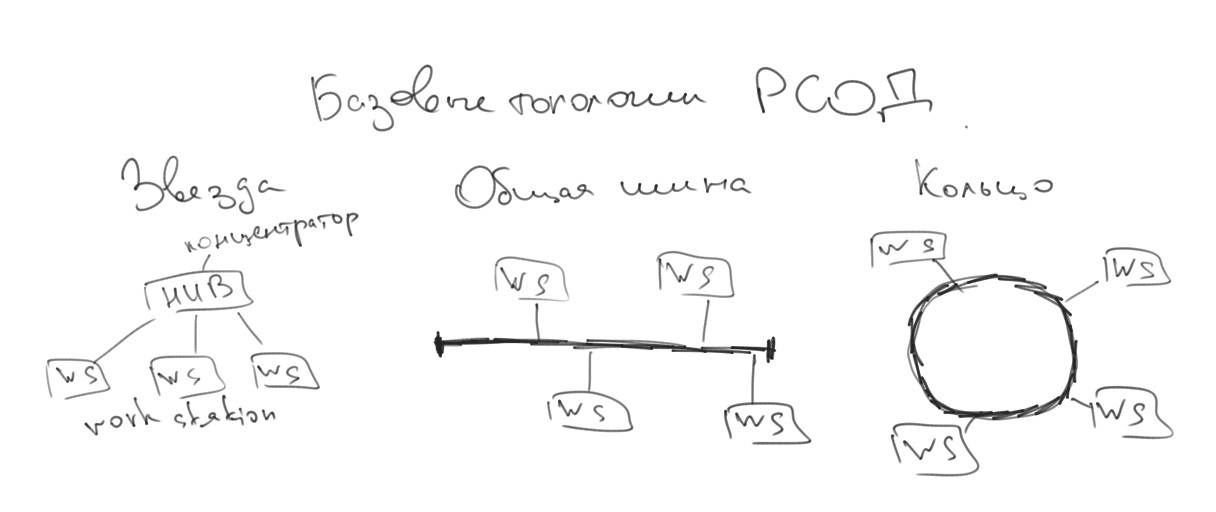


Сообщение разбивается на блоки (пакеты), котор. Имеет адрес назначения, каждый передается по различным каналам. Обеспечивает равномерную нагрузку сети. Пропускная способность увеличилась!

­­**Локальные РСОД**

1. Использование единой коммутации среды (единая шина), полносвязан.
2. Небольшая географич распределенность
3. Высокие и очень высокие скорости обмены 107­­-1010 бит/с
4. Применение спец. Методов и алгоритмов доступа к единой среде для обеспечения выс скорости передачи
5. Ограниченные топология

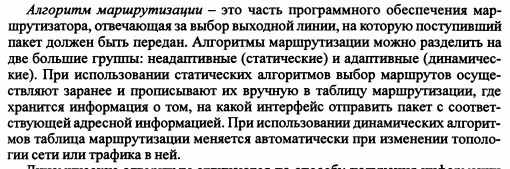
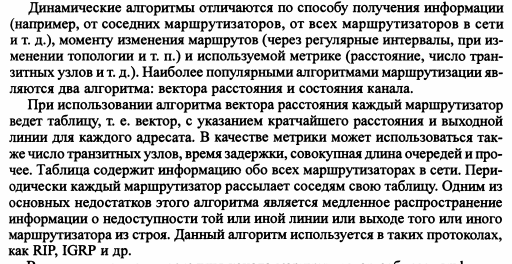
***Базовые топологии:***



**Маршрутизатор. Назначение и применение в сетях**.

маршрутизатор - это устройство третьего уровня, использующее одну и более метрик для определения оптимального пути передачи сетевого трафика на основе информации сетевого уровня. По существу маршрутизатор представляет собой компьютер с необходимым программным обеспечением и устройствами ввода/вьгоода.

Маршрутизатор обеспечивает маршрутизацию, т. е. доставку данных адресату, которую можно разбить на три. Во-первых, сбор информации о других маршрутизаторах и хостах в сети. Для этого маршрутизатор в целях определения марыфута использует тот или иной протокол маршрутизации. Во-вторых, он сохраняет полученную информацию о маршрутах в таблицах маршрутизации. В-третьих, маршрутизатор выбирает наилучший маршрут для каждого 250 3.3. Оборудование локальных сетей конкретного пакета, при этом он передает пакет со входного интерфейса на соответствующий выходной интерфейс. Данные функции он вьшолняет с помопц>ю прото

Стр.250 книги

***Билет №7***

Применение правил «5-4-3» и «4-х хабов» в локальных сетях.

Методы коммутации в региональных РСОД. Сравнительный анализ.