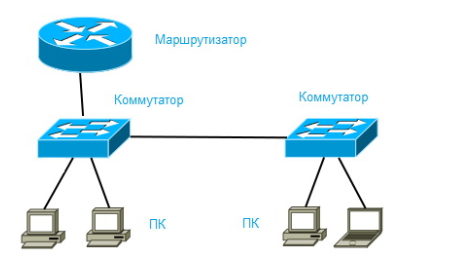
***Билет №1***

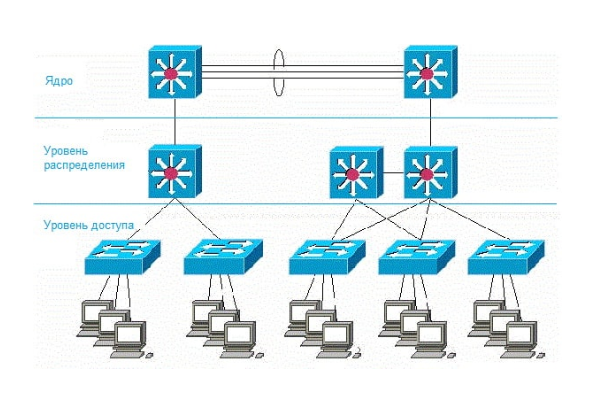
**1.Коммутаторы. Назначение и применение в локальных сетях.**

Коммутаторы одно из важнейших устройств при построении локальной сети.

Так выглядит стандартная схема использования коммутаторов:



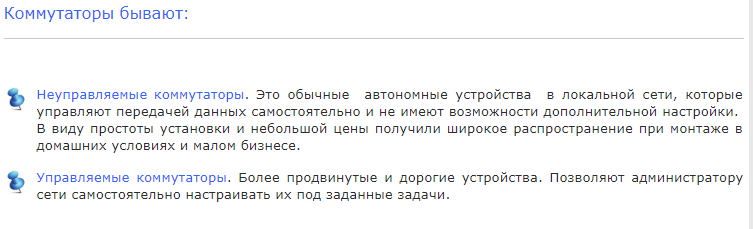
Коммутаторы доступа подключены к конечным пользователям и дают им доступ к ресурсам локальной сети.

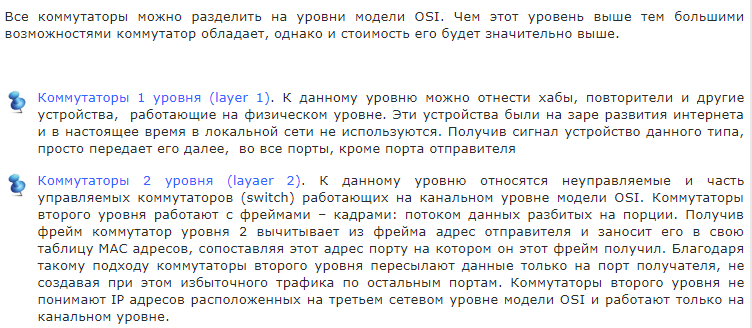
Однако в крупных локальных сетях они выполняют след. Функции: 

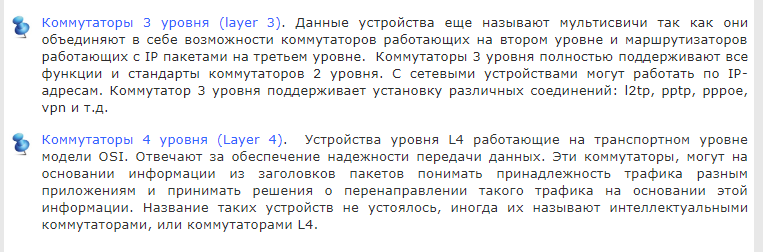
**Уровень доступа сети**. Здесь коммутаторы также как и выше, подключены к конечным пользователям. В крупных локальках фреймы коммутатор не взаимодействуют с другими коммутаторами напрямую, они передаются через уровень расперделения

**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.

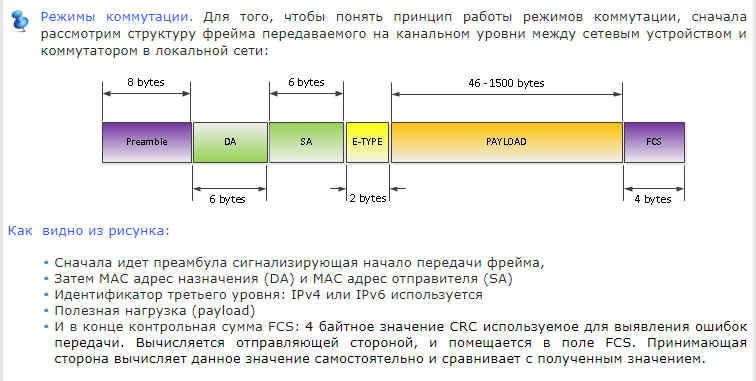






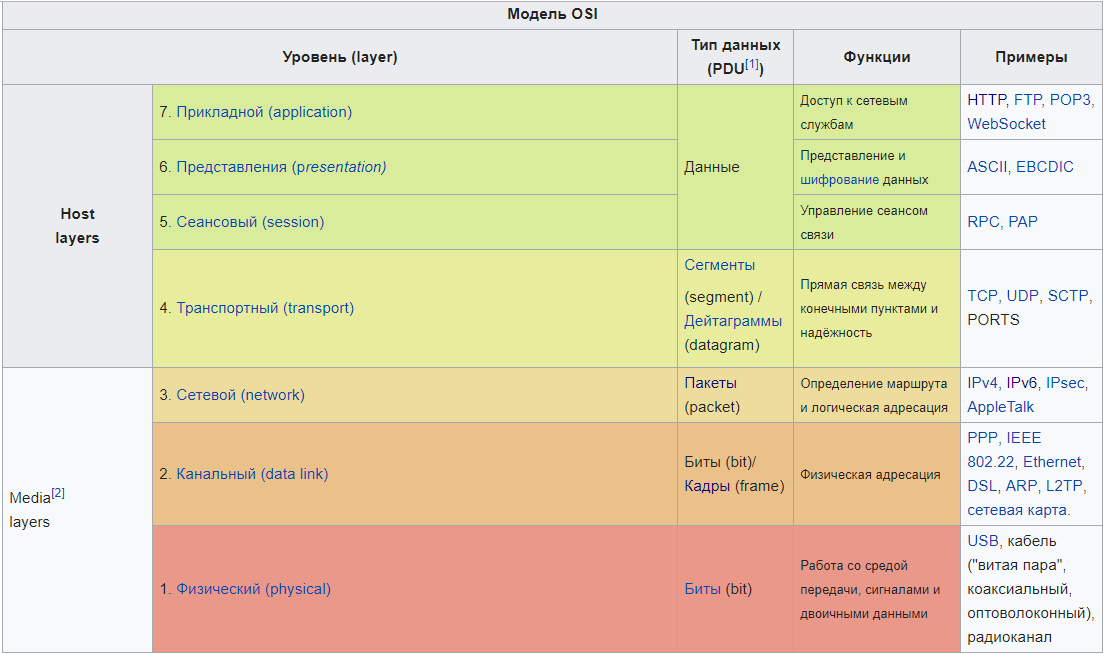
Основные характеристики коммутаторов:

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***

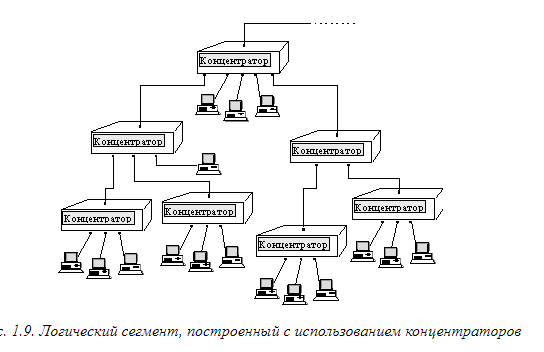
**Повторители. Назначение и применение в локальных сетях**

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.

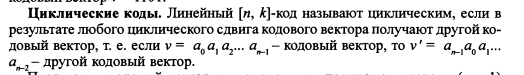


Появление устройств, централизующих соединения между отдельными сетевыми устройствами, потенциально позволяет улучшить управляемость сети и ее эксплуатационные характеристики (модифицируемость, ремонтопригодность и т.п.). С этой целью разработчики концентраторов часто встраивают в свои устройства, кроме 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**.



Порождающий полином

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)

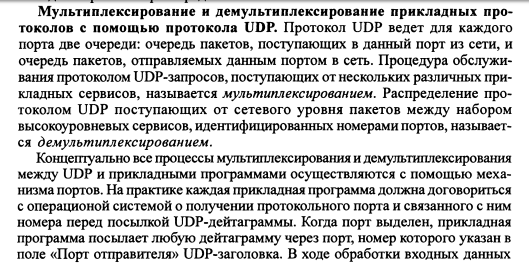
Основная идея технологии расширения спектра (Spread Spectrum, SS) заключается в том, чтобы от узкополосного спектра сигнала, возникающего при обычном потенциальном кодировании, перейти к широкополосному спектру. Именно это позволяет значительно повысить помехоустойчивость передаваемых данных.  Бит кодовой 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.

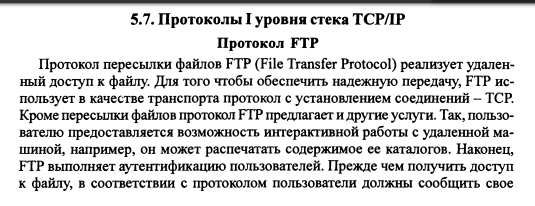
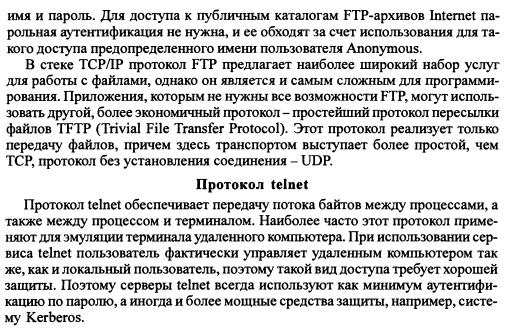
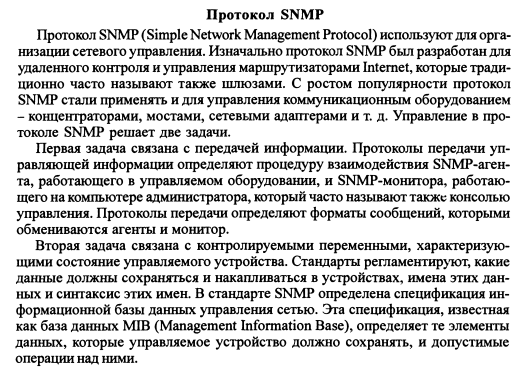
**Псевдослучайная перестройка рабочей частоты** (FHSS — [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***f****requency-****h****opping****s****pread****s****pectrum*) — метод передачи информации по [радио](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B8%D0%BE), 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**



**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***

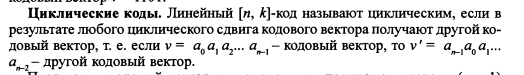
**Мост. Назначение и применение в локальных сетях**.

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.

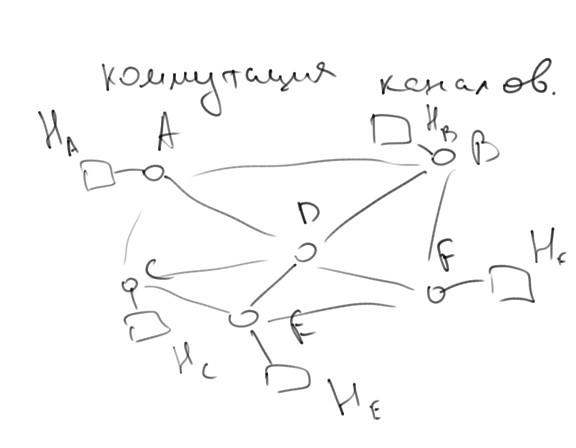
Региональные(Глобальные):

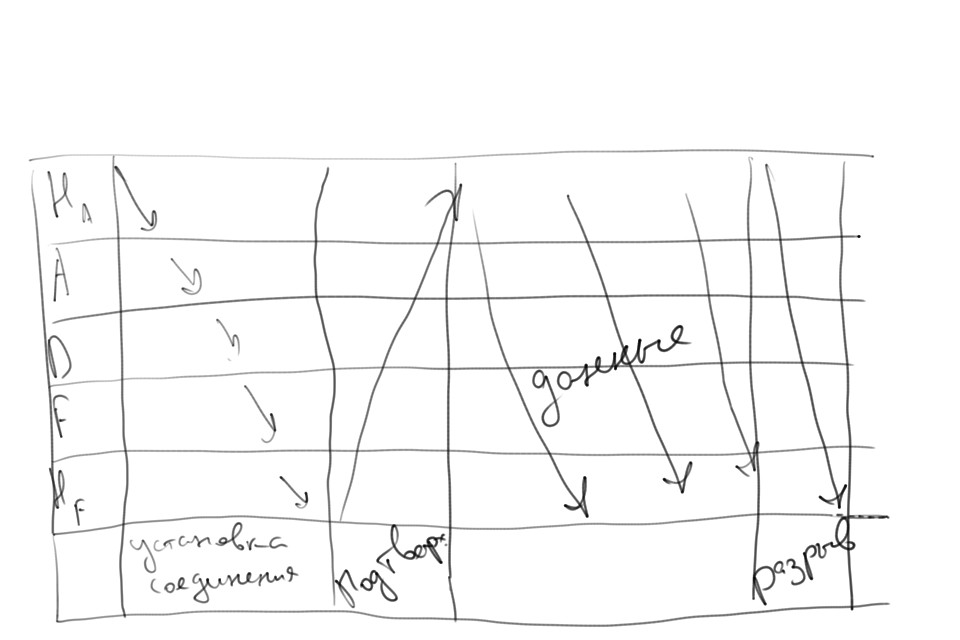
Характер:

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)



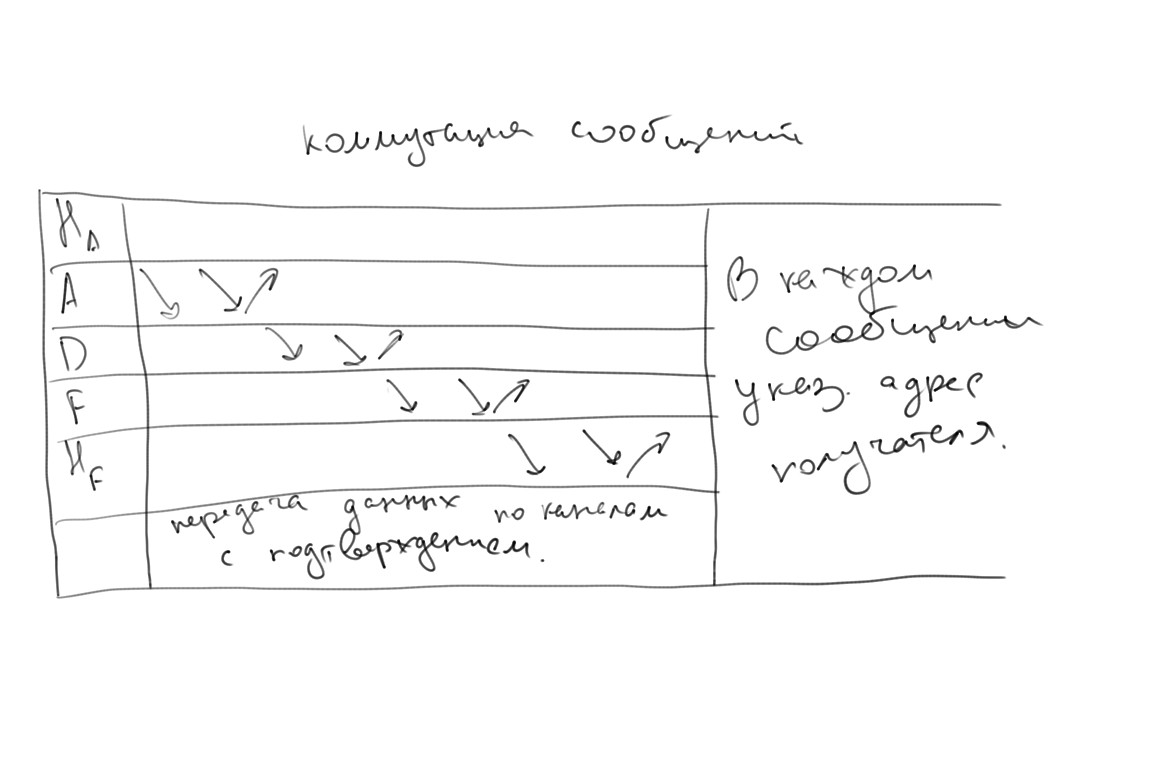


Недостатки:

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:

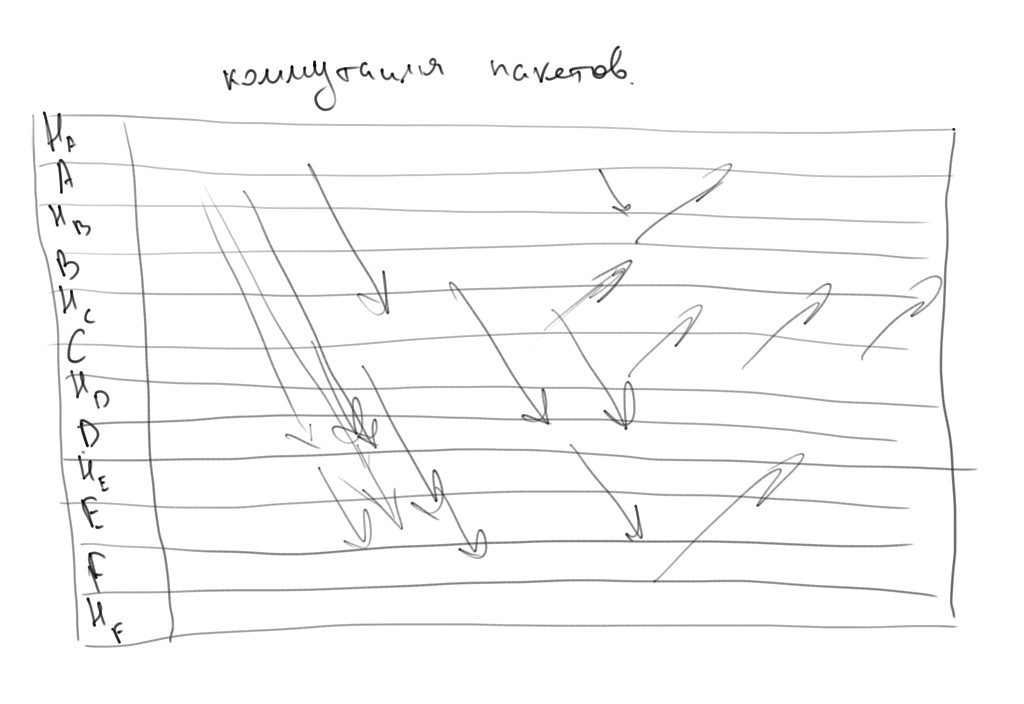
1. После формирования канала задержек нет, передача файлов в реальном времени
2. Адрес назначения указ только 1 раз.

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

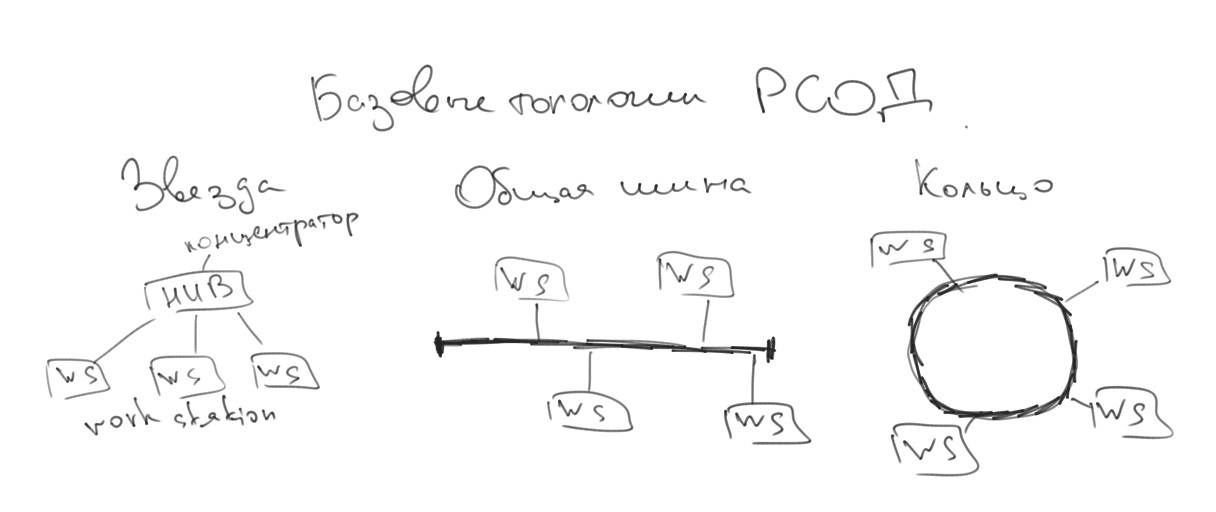


Сообщение разбивается на блоки (пакеты), котор. Имеет адрес назначения, каждый передается по различным каналам. Обеспечивает равномерную нагрузку сети. Пропускная способность увеличилась!

­­**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

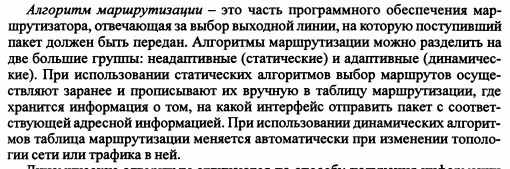
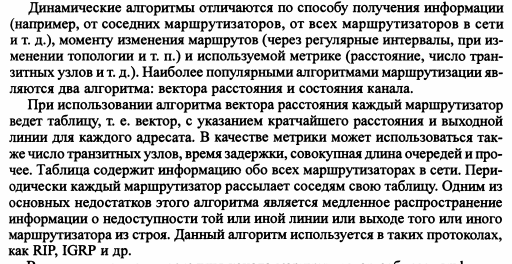
***Базовые топологии:***



**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 конкретного пакета, при этом он передает пакет со входного интерфейса на соответствующий выходной интерфейс.

Стр.250 книги

***Билет №7***

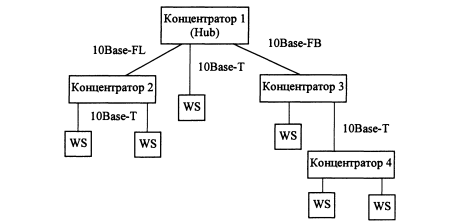
**Применение правил «5-4-3» и «4-х хабов» в локальных сетях.**

Стандарты lOBase-5 и lOBase-2 разрешают использование в сети не более 4 повторителей и, соответственно, не более 5 сегментов кабеля. При максимальной длине сегмента кабеля 500 м максимальная длина сети lOBase-5 составляет 2500 м. В случае стандарта lOBase-2 максимальная длина сети равна 5 х185 = 925 м.



Правило применения повторителей в сети Ethernet называется «правило 5-4-3»: 5 сегментов, 4 повторителя, 3 нагруженных сегмента.

Для обеспечения синхронизации станций при реализации процедур доступа CSMA/CD и надежного распознавания станциями коллизий в стандарте lOBase-T определено максимально число концентраторов между любыми двумя станциями сети - 4. Это правило носит название «правило 4 хабов» и оно заменяет «правило 5-4-3», применяемое к сетям lOBase-5 и lOBase-2. Очевидно, что если между любыми двумя узлами сети не должно быть больше 4 повторителей, то максимальный диаметр сети lOBase-T составляет 5 х 100 = 500 м. На рис. 3.6 представлена такая структура, образующая общую область столкновений - один домен коллизий. При создании сети lOBase-T

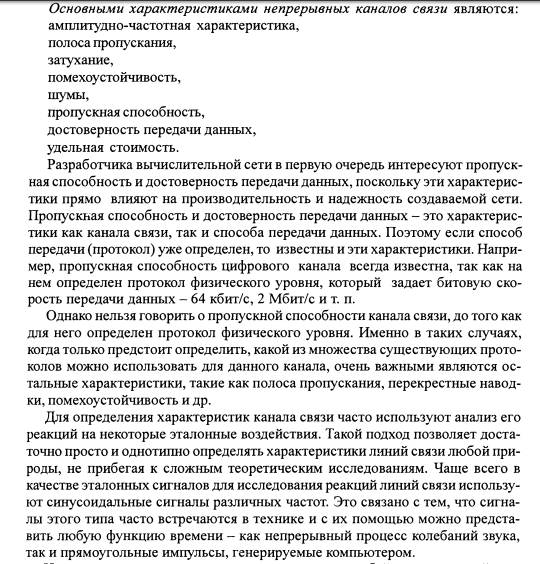


**Методы коммутации в региональных РСОД. Сравнительный анализ.**

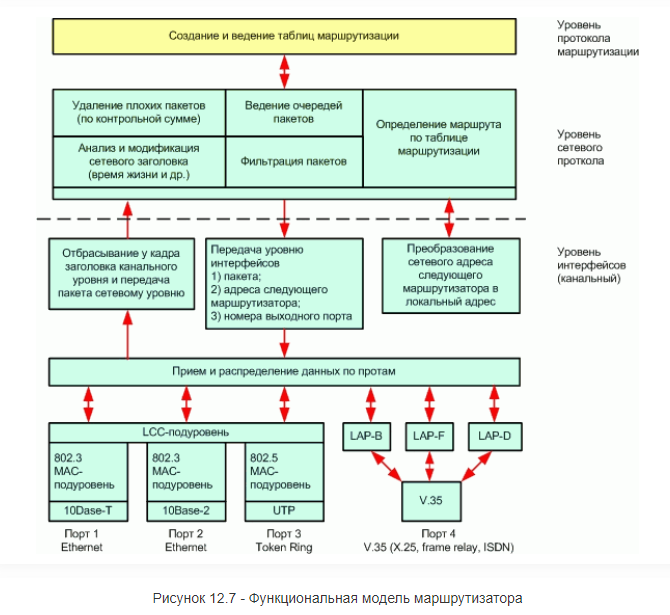
См. Билет №6

***Билет №8***

**Основные характеристики непрерывного канала связи.**



**Функциональная модель маршрутизатора.**



<http://siblec.ru/index.php?dn=html&way=bW9kL2h0bWwvY29udGVudC84c2VtLzA5NC8xMi0zLTMuaHRt>

***Билет №9***

**Методы доступа в локальных РСОД.**

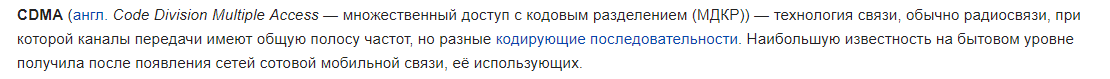


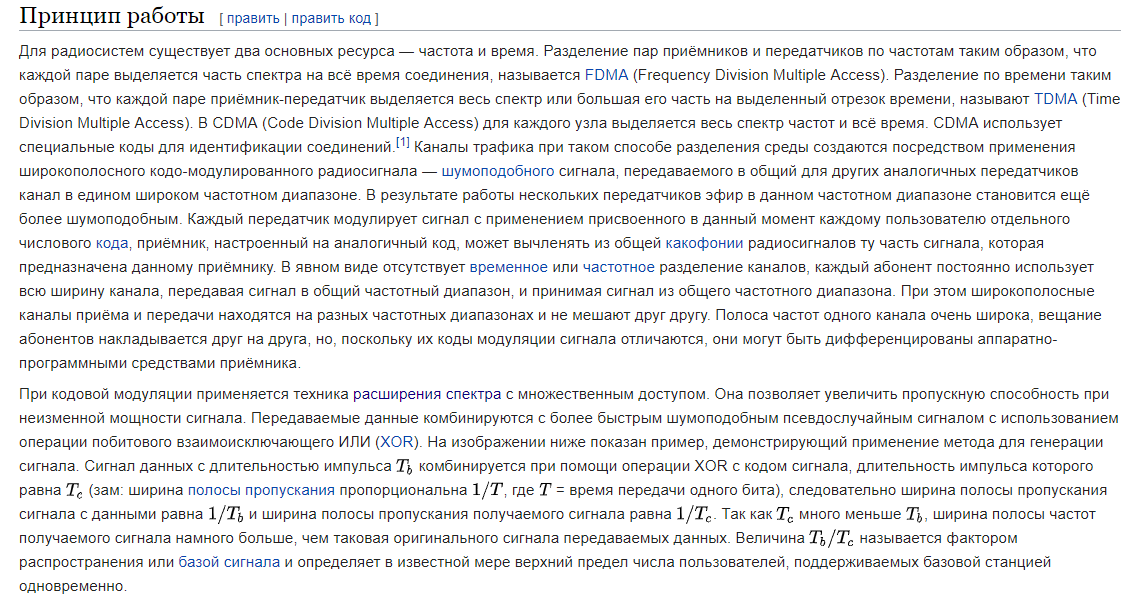
**Разбиение IP-сети на подсети при помощи маски. Пример**.

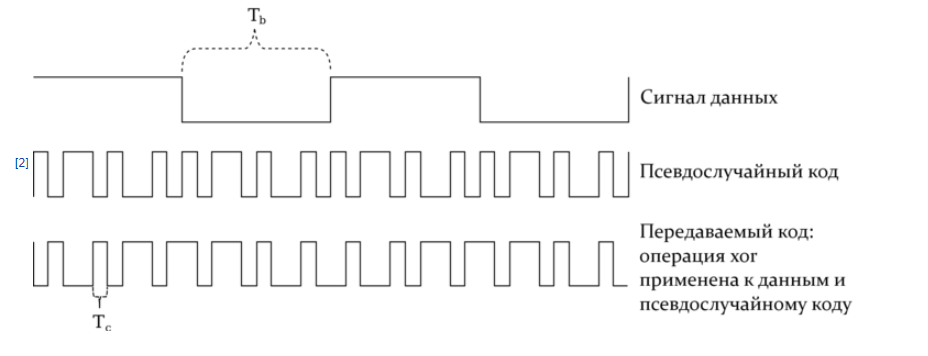
Здесь воля фантазии, рисуем сеть.

***Билет №10***

**Кодовое разделение каналов. Прямое последовательное расширение спектра**









### 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.

**Идентификация соединения в интернет при помощи сокетов**.

По определению сокет- это пара идентификаторов, включающая в себя IP-адрес и номер порта (в терминологии TCP/IP портами называют системные очереди в которые операционной системой организуются пакеты, поступающие на транспортный уровень). Фактически сокет это конечная точка сетевых коммуникаций. Каждый использующийся сокет имеет тип и ассоциированный с ним процесс. Сокеты существуют внутри коммуникационных доменов. Домены это абстракции, которые подразумевают конкретную структуру адресации и множество протоколов, которое определяет различные типы сокетов внутри домена. Примерами коммуникационных доменов могут быть: UNIX домен, Internet домен, и т.д. Два сокета, один для хоста-получателя, другой для хоста-отправителя, определяют соединение для протоколов, ориентированных на установление связи, таких, как TCP.

Сокеты (англ. socket — разъём) — название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут исполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью. Сокет — абстрактный объект, представляющий конечную точку соединения.

## Принципы сокетов

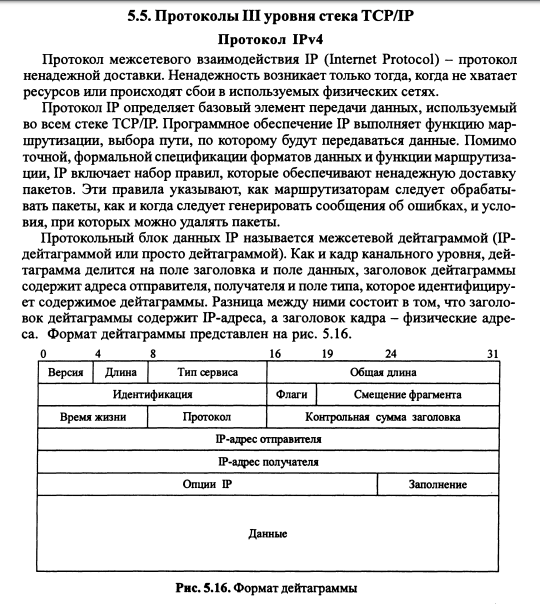
Каждый процесс может создать слушающий сокет (серверный сокет) и привязать его к какому-нибудь порту операционной системы (в UNIX непривилегированные процессы не могут использовать порты меньше 1024). Слушающий процесс обычно находится в цикле ожидания, то есть просыпается при появлении нового соединения. При этом сохраняется возможность проверить наличие соединений на данный момент, установить тайм-аут для операции и т.д.

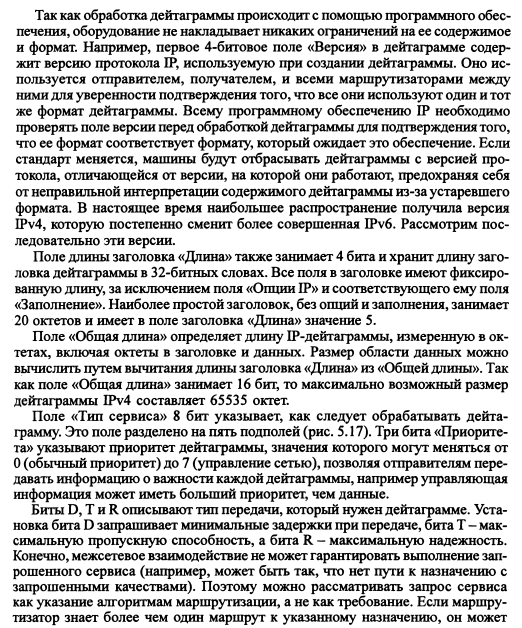
***Билет №11***

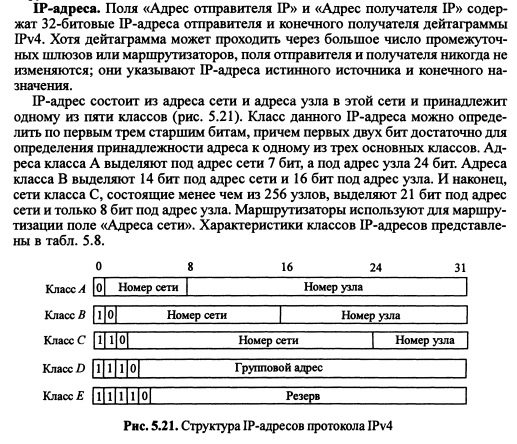
**Расширение спектра скачкообразной перестройкой частоты**

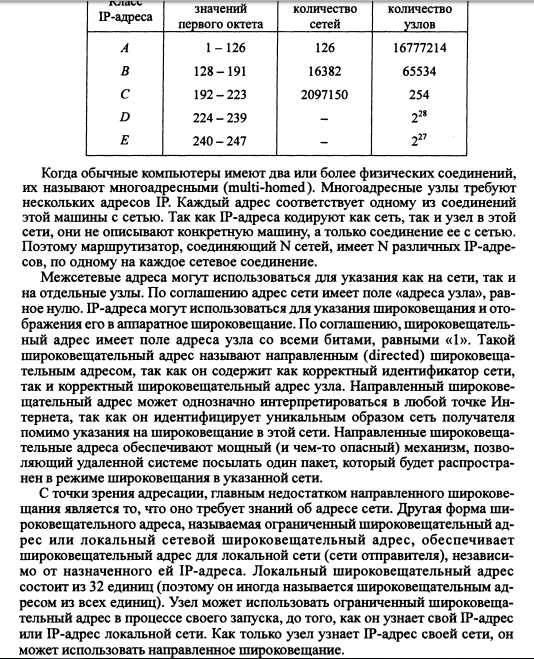
См. Билет №3

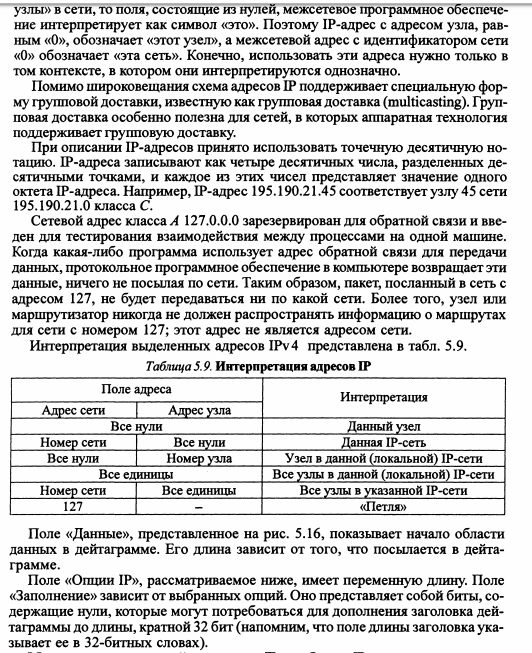
**IPv4-адреса. Классы адресов и их характеристика**







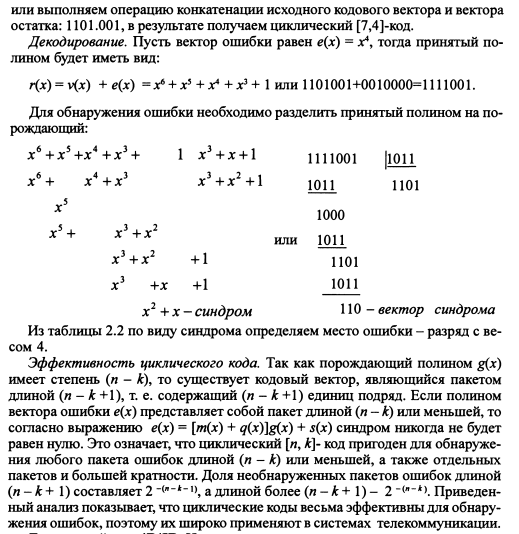




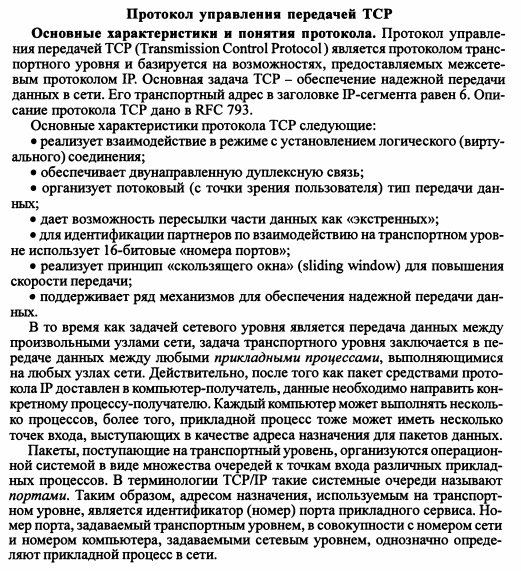
***Билет №12***

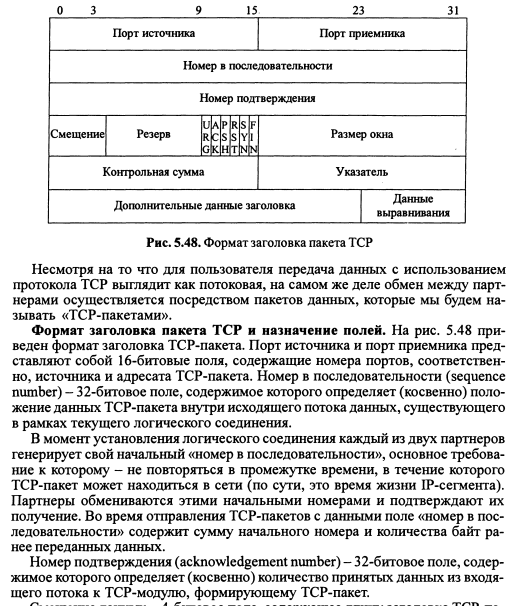
**Построение циклического кода. Правило кодирования**

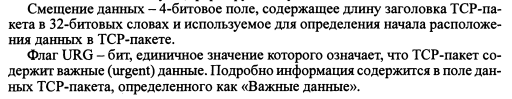


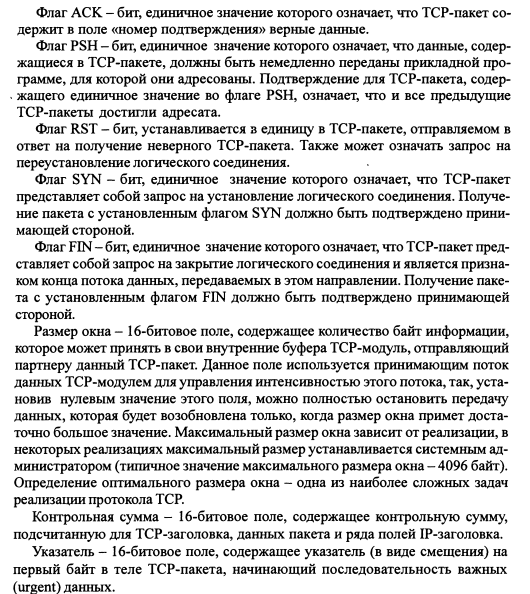


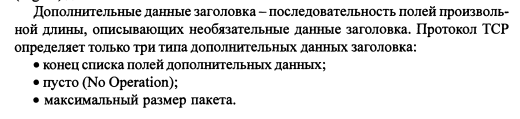
**ТСР-протокол**

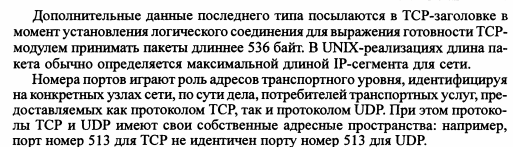


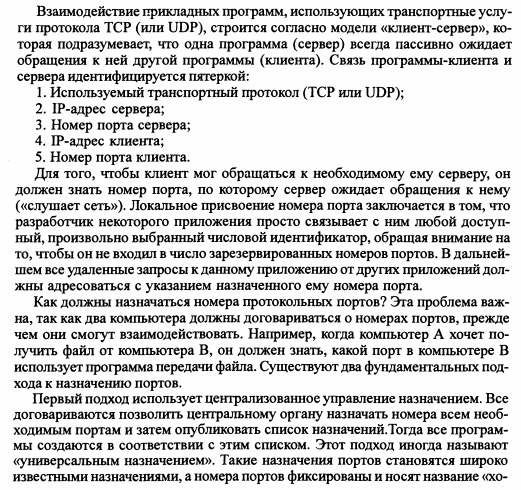


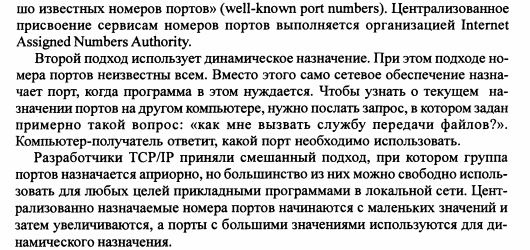


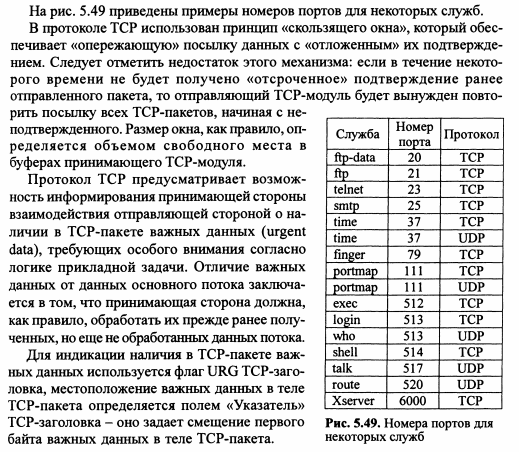


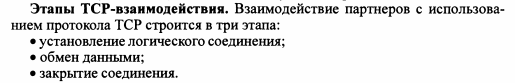












***Билет №13***

**Резервирование связей в локальных сетях. STP-протокол**

Есть 2 способа повышения надёжности сети:  
1. Добавление альтернативных связей( в случае неисправности основных связей, подключаются альтернативные)

2.Добавление связей для балансировки нагрузки в сети( доп связи тоже работают)

В первом случае повышается надёжность, во втором как надёжность, так и продуктивность сети

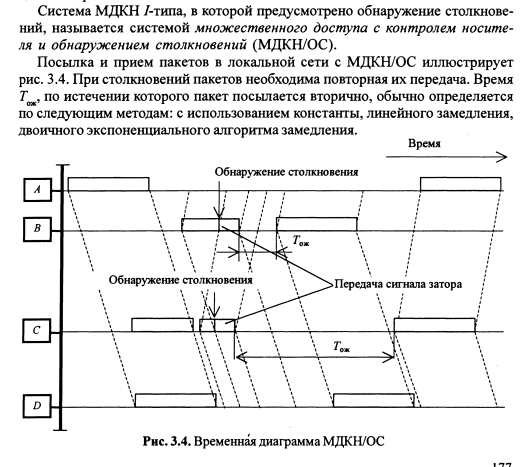
STP протокол реализующий алгоритм покрывающего дерева STA(spanning tree algorithm). STA формирует сеть в виде графа, коммутаторы и сегменты – узлы. Короче это алгоритм реализует дерево от любого коммутатора или сегмента до нужного коммутатора или сегмента в виде дерева и находит самый короткий путь. В качестве расстояние используется метрика (обратно пропорциональная к пропускной способности сегмента).

**RIP-протокол**

Протокол маршрутной информации. Один из самых простых протоколов маршрутизации, позволяет маршрутизаторам динамически обновлять маршрутную информацию, получая её от соседних маршрутизаторов.

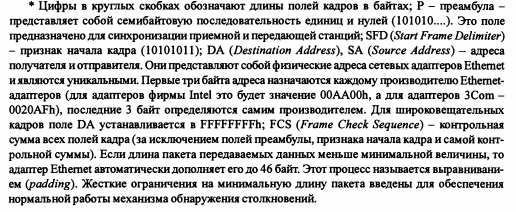
***Билет №14***

**Метод доступа МДКН/ОС**



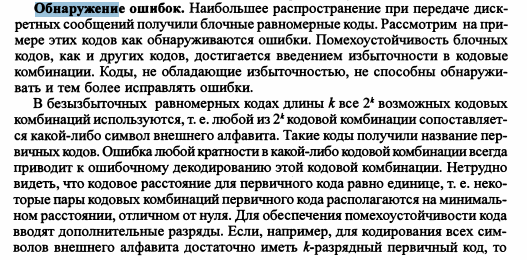
**Форматы и структуры кадров Ethernet. Назначение полей**

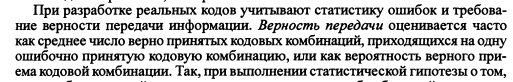


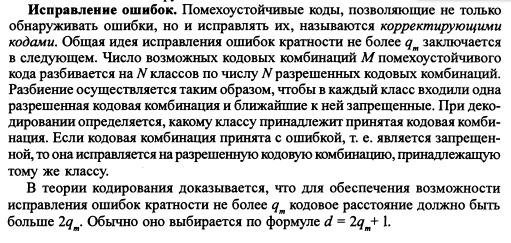


***Билет №15***

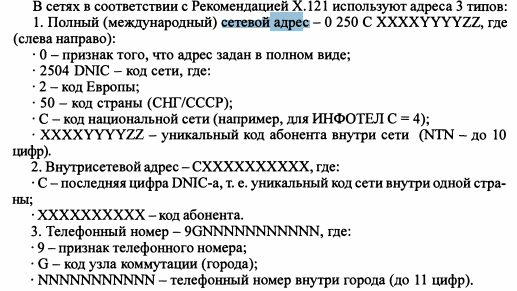
Помехи в канале. Условия обнаружения и исправления ошибок.





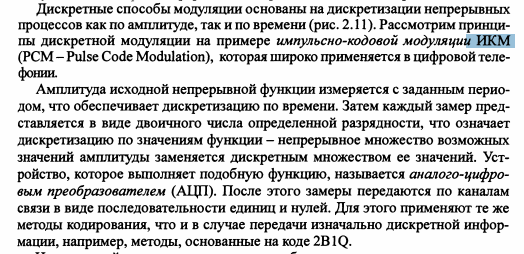


Сетевые адреса. Назначение. Примеры.



***Билет №16***

**ИКМ-модуляция**



**Принцип идентификации двух процессов в сетевом взаимодействии**

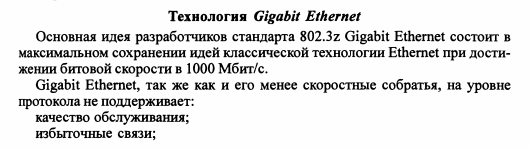
Непонятный вопрос.

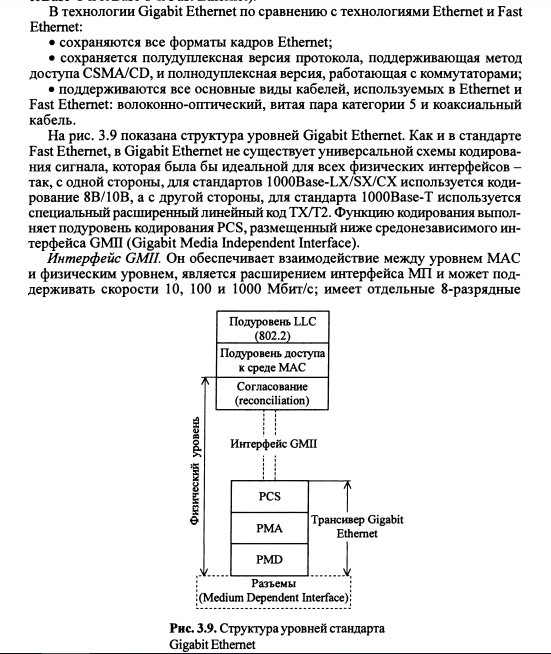
***Билет №17***

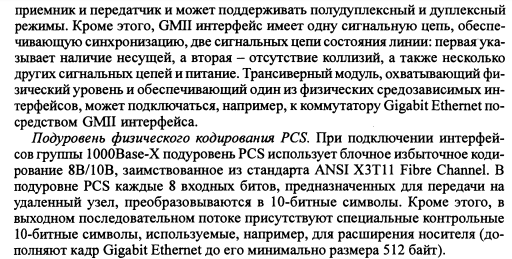
**Асинхронный и синхронный режим передачи**

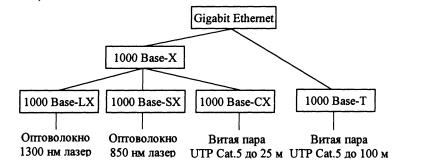
См. Билет 4.

**Gigabit Ethernet. Основные характеристики технологии**









***Билет №18***

**Обнаружение и коррекция ошибок на канальном уровне**

Канальный уровень должен обнаруживать ошибки передачи данных, связанные с искажением бит в принятом кадре данных или с потерей кадра, и по возможности их корректировать.  
Большая часть протоколов канального уровня выполняет только первую задачу ? обнаружение ошибок, считая, что корректировать ошибки, то есть повторно передавать данные, содержавшие искаженную информацию, должны протоколы верхних уровней. Так работают такие популярные протоколы локальных сетей, как Ethernet, Token Ring, FDDI и другие. Однако существуют протоколы канального уровня, например LLC2 или LAP-B, которые самостоятельно решают задачу восстановления искаженных или потерянных кадров.  
Для сетей, в которых искажения и потери кадров очень редки, разрабатываются протоколы типа Ethernet, в которых не предусматриваются процедуры устранения ошибок, наличие которых требует от конечных узлов дополнительных вычислительных затрат, что при надежной работе сети является избыточным.  
Напротив, если в сети искажения и потери случаются часто, то желательно уже на канальном уровне использовать протокол с коррекцией ошибок. Протоколы верхних уровней, работая с большими тайм-аутами, восстановят потерянные данные с большой задержкой. Поэтому нельзя считать, что один протокол лучше другого потому, что он восстанавливает ошибочные кадры, а другой протокол ? нет. Каждый протокол должен работать в тех условиях, для которых он разработан.  
Методы обнаружения ошибок  
Все методы обнаружения ошибок основаны на передаче в составе кадра данных служебной избыточной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. Эту служебную информацию принято называть контрольной суммой (Frame Check Sequence, FCS). Контрольная сумма вычисляется как функция от основной информации. Принимающая сторона повторно вычисляет контрольную сумму кадра по известному алгоритму и в случае ее совпадения с принятой контрольной суммой, делает вывод о том, что данные были переданы через сеть корректно. Существует несколько распространенных алгоритмов вычисления контрольной суммы, отличающихся вычислительной сложностью и способностью обнаруживать ошибки в данных.  
Контроль по паритету представляет собой наиболее простой метод контроля данных. В то же время это наименее мощный алгоритм контроля, так как с его помощью можно обнаружить только одиночные ошибки в проверяемых данных. Метод заключается в суммировании по модулю 2 всех бит контролируемой информации. Например, для данных 100101011 результатом контрольного суммирования будет значение 1. Результат суммирования также представляет собой один бит данных, который пересылается вместе с контролируемой информацией. При искажении пересылки любого бита исходных данных (или контрольного разряда) результат суммирования будет отличаться от принятого контрольного разряда, что говорит об ошибке. Однако двойная ошибка, например 110101010, будет неверно принята за корректные данные. Поэтому контроль по паритету применяется к небольшим порциям данных, что дает коэффициент избыточности для этого метода 1/8. Метод редко применяется в вычислительных сетях из-за его большой избыточности и невысоких диагностических способностей.  
Циклический избыточный контроль (Cyclic Redundancy Check, CRC) является в настоящее время наиболее популярным методом контроля в вычислительных сетях. Метод основан на рассмотрении исходных данных в виде одного многоразрядного двоичного числа. Например, кадр стандарта Ethernet, состоящий из 1024 байт, будет рассматриваться как одно число, состоящее из 8192 бит. В качестве контрольной информации рассматривается остаток от деления этого числа на известный делитель R. Обычно в качестве делителя выбирается семнадцати- или тридцати трехразрядное число, чтобы остаток от деления имел длину 16 разрядов (2 байт) или 32 разряда (4 байт). При получении кадра данных снова вычисляется остаток от деления на тот же делитель R, но при этом к данным кадра добавляется и содержащаяся в нем контрольная сумма. Если остаток от деления на R равен нулю, то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.  
Этот метод обладает более высокой вычислительной сложностью, но его диагностические возможности гораздо выше, чем у методов контроля по паритету. Метод CRC обнаруживает все одиночные ошибки, двойные ошибки и ошибки в нечетном числе бит. Метод обладает также невысокой степенью избыточности. Например, для кадра Ethernet размером в 1024 байт контрольная информация длиной в 4 байт составляет только 0,4 %.  
Методы восстановления искаженных и потерянных кадров основаны на повторной передаче кадра данных в том случае, если кадр не доходит до адресата или приемник обнаружил в нем искажение информации. Чтобы убедиться в необходимости повторной передачи данных, отправитель нумерует отправляемые кадры и для каждого кадра ожидает от приемника так называемой положительной квитанции ? служебного кадра, извещающего о том, что исходный кадр был получен и данные в нем оказались корректными. Время этого ожидания ограничено ? при отправке каждого кадра передатчик запускает таймер, и, если по его истечении положительная квитанция на получена, кадр считается утерянным. Приемник в случае получения кадра с искаженными данными может отправить отрицательную квитанцию ? указание на то, что данный кадр нужно передать повторно.  
Существуют два подхода к организации процесса обмена квитанциями: с простоями и с организацией «окна».  
Метод с простоями (Idle Source) требует, чтобы источник, пославший кадр, ожидал получения квитанции (положительной или отрицательной) от приемника и только после этого посылал следующий кадр (или повторял искаженный). Если же квитанция не приходит в течение тайм-аута, то кадр (или квитанция) считается утерянным и его передача повторяется. На рис.15, а видно, что в этом случае производительность обмена данными существенно снижается, ? хотя передатчик и мог бы послать следующий кадр сразу же после отправки предыдущего, он обязан ждать прихода квитанции. Снижение производительности этого метода коррекции особенно заметно на низкоскоростных каналах связи, т.е в территориальных сетях.

Второй метод называется методом «скользящего окна» (sliding window). В этом методе для повышения коэффициента использования линии источнику разрешается передать некоторое количество кадров в непрерывном режиме, то есть в максимально возможном для источника темпе, без получения на эти кадры положительных ответных квитанций. Количество кадров, которые разрешается передавать таким образом, называется размером окна. Рис.15, б иллюстрирует данный метод для окна размером в W кадров.  
В начальный момент, когда еще не послано ни одного кадра, окно определяет диапазон кадров с номерами от 1 до W включительно. Источник начинает передавать кадры и получать в ответ квитанции. Для простоты предположим, что квитанции поступают в той же последовательности, что и кадры, которым они соответствуют. В момент t1 при получении первой квитанции К1 окно сдвигается на одну позицию, определяя новый диапазон от 2 до (W+1).  
Процессы отправки кадров и получения квитанции идут достаточно независимо друг от друга. Рассмотрим произвольный момент времени tn, когда источник получил квитанцию на кадр с номером п. Окно сдвинулось вправо и определило диапазон разрешенных к передаче кадров от (n+1) до (W+n). Все множество кадров, выходящих из источника, можно разделить на перечисленные ниже группы.  
• Кадры с номерами от 1 до n уже были отправлены и квитанции на них получены, то есть они находятся за пределами окна слева.  
• Кадры, начиная с номера (n+1) и кончая номером (W+n), находятся в пределах окна и потому могут быть отправлены не дожидаясь прихода какой-либо квитанции. Этот диапазон может быть разделен еще на два поддиапазона:  
• кадры с номерами от (n+1) до m, которые уже отправлены, но квитанции на них еще не получены;  
• кадры c номерами от m до (W+n), которые пока не отправлены, хотя запрета на это нет.  
• Все кадры с номерами, большими или равными (W+n+1), находятся за пределами окна справа и поэтому пока не могут быть отправлены.  
Перемещение окна вдоль последовательности номеров кадров показано на рис.15, в. Здесь t0 ? исходный момент, t1 и tn ? моменты прихода квитанций на первый и n-й кадр соответственно. Каждый раз, когда приходит квитанция, окно сдвигается влево, но его размер при этом не меняется и остается равным W. Заметим, что хотя в данном примере размер окна в процессе передачи остается постоянным, в реальных протоколах (например, TCP) можно встретить варианты данного алгоритма с изменяющимся размером окна.  
При отправке кадра с номером n источнику разрешается передать еще W-1 кадров до получения квитанции на кадр n, так что в сеть последним уйдет кадр с номером (W+n-1). Если же за это время квитанция на кадр n так и не пришла, то процесс передачи приостанавливается, и по истечении некоторого тайм-аута кадр n (или квитанция на него) считается утерянным, и он передается снова.  
Если же поток квитанций поступает более-менее регулярно, в пределах допуска в W кадров, то скорость обмена достигает максимально возможной величины для данного канала и принятого протокола.  
Метод скользящего окна более сложен в реализации, т.к. передатчик должен хранить в буфере все кадры, на которые пока не получены положительные квитанции. Кроме того, требуется отслеживать несколько параметров алгоритма: размер окна W, номер кадра, на который получена квитанция, номер кадра, который еще можно передать до получения новой квитанции.  
Приемник может не посылать квитанции на каждый принятый корректный кадр. Если несколько кадров пришли почти одновременно, то приемник может послать квитанцию только на последний кадр. При этом подразумевается, что все предыдущие кадры также дошли благополучно.  
Некоторые методы используют отрицательные квитанции. Отрицательные квитанции бывают двух типов ? групповые и избирательные. Групповая квитанция содержит номер кадра, начиная с которого нужно повторить передачу всех кадров, отправленных передатчиком в сеть. Избирательная отрицательная квитанция требует повторной передачи только одного кадра.  
Метод скользящего окна реализован в протоколах: LLC2, LAP-B, X.25, TCP, Novell NCP Burst Mode.  
Метод с простоями является частным случаем метода скользящего окна, когда размер окна равен единице.  
Метод скользящего окна имеет два параметра, которые могут заметно влиять на эффективность передачи данных между передатчиком и приемником, ? размер окна и величина тайм-аута ожидания квитанции. В надежных сетях для повышения скорости обмена данными размер окна нужно увеличивать, т.к. при этом передатчик будет посылать кадры с меньшими паузами. В ненадежных сетях размер окна уменьшают, т.к. при частых потерях и искажениях кадров резко возрастает объем вторично передаваемых через сеть кадров, а значит, пропускная способность сети будет расходоваться вхолостую. Выбор тайм-аута зависит не от надежности сети, а от задержек передачи кадров сетью.

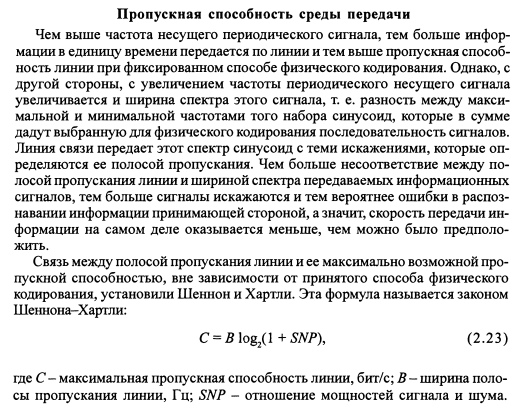
**Протокол межсетевого взаимодействия – IP**

См билет 11

***Билет №19***

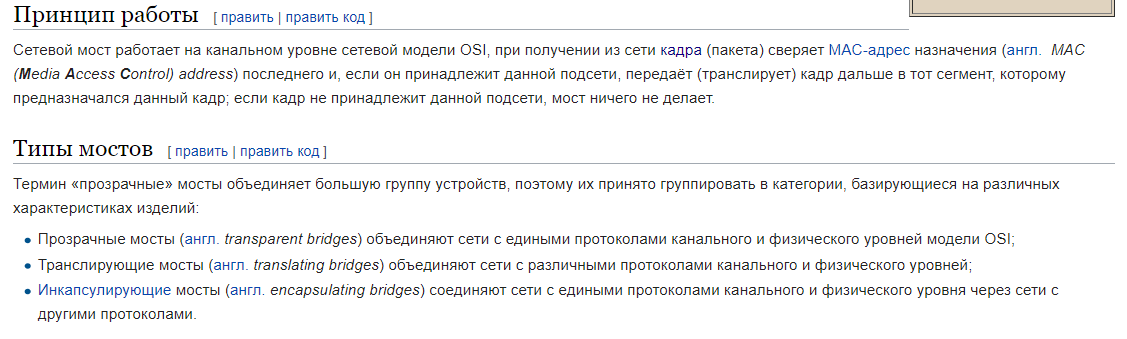
**Пропускная способность канала**

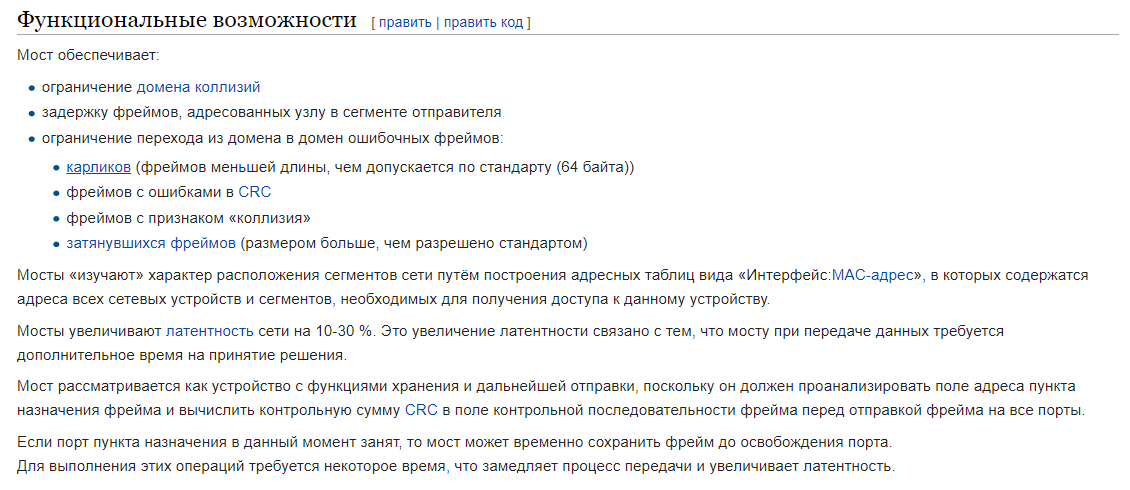




**Мосты. Классификация мостов. Принцип работы**

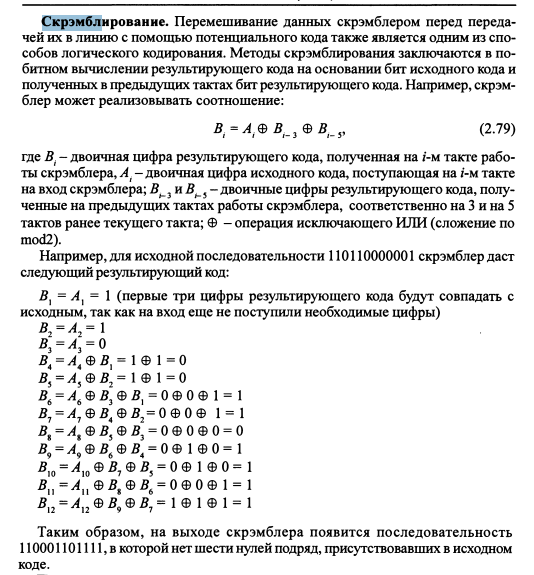
См. Билет 5

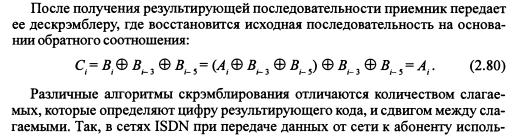




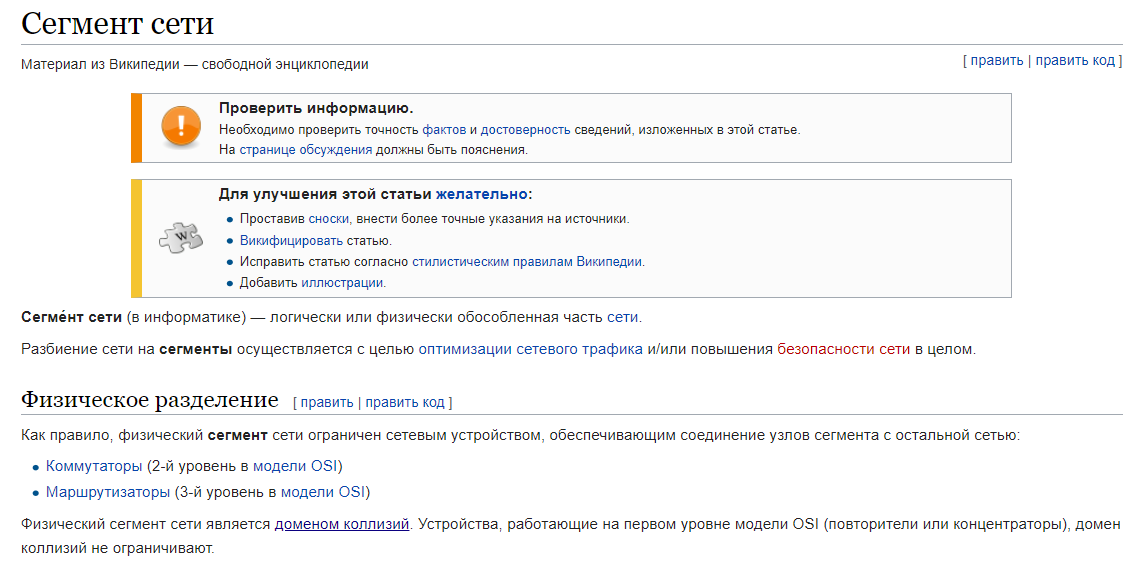
***Билет №20***

**Скрэмблирование как способ логического кодирования. Пример**





**Сегментация локальных сетей**



Физический сегмент – каналы

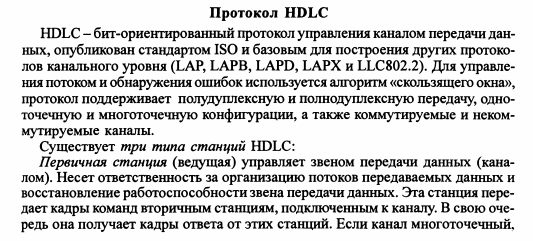
Логический сегмент – группа каналов с общим канальным уровнем.

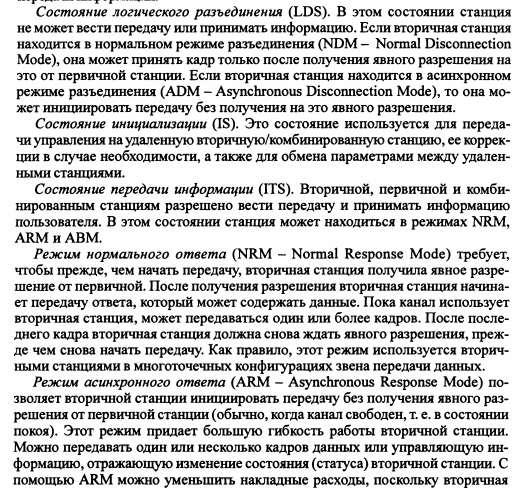
***Билет №21***

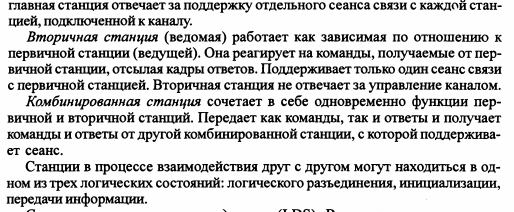
**Синдром ошибки. Определение синдрома однократной ошибки циклического кода**

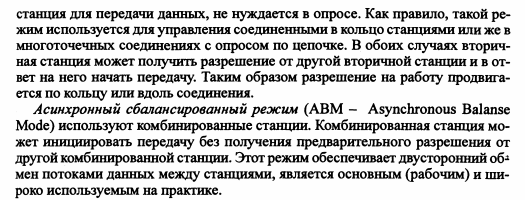
См ДЗ.

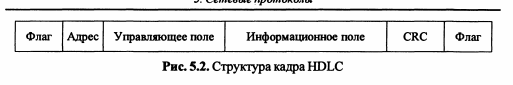
**Протокол HDLC. Структура кадра. Основные режимы работы**





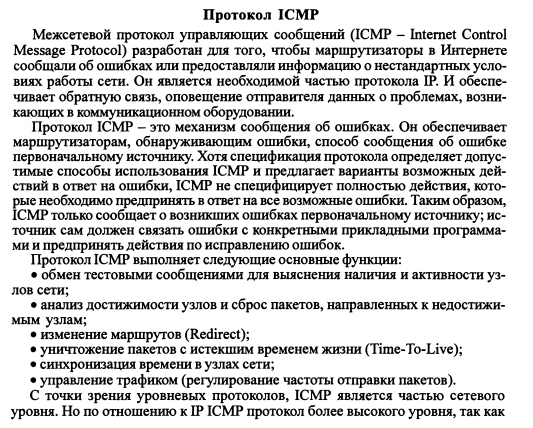


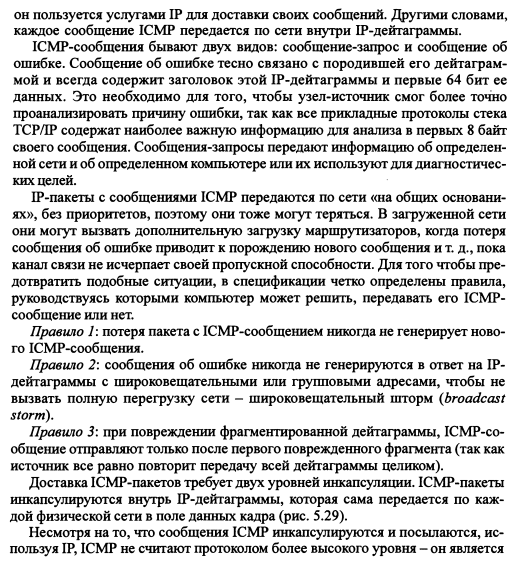




***Билет №22***

**ICMP- протокол. Назначение. Функциональные возможности**



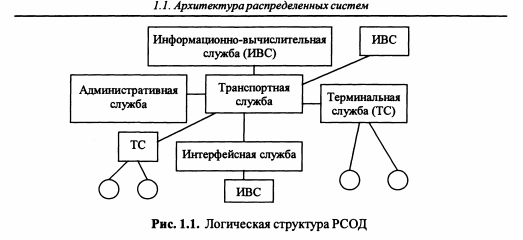


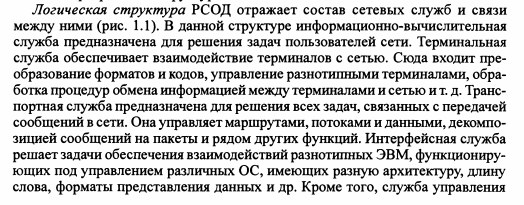
**Коды Хэмминга. Пример**

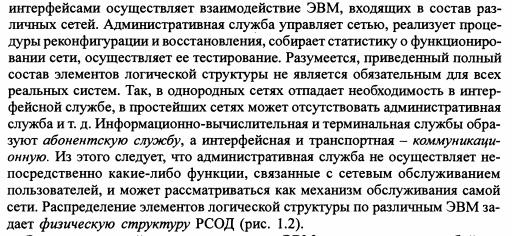
ДЗ

***Билет №23***

**Логическая структура РСОД. Назначение служб**

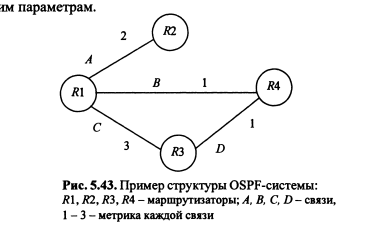


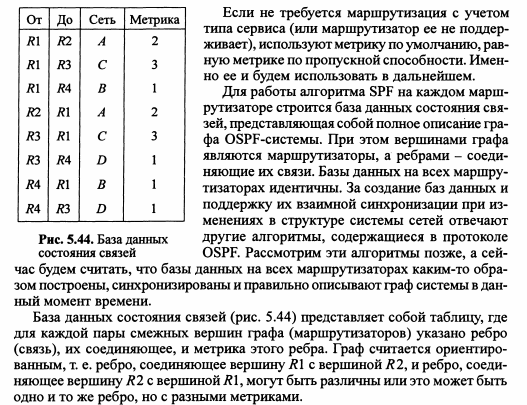


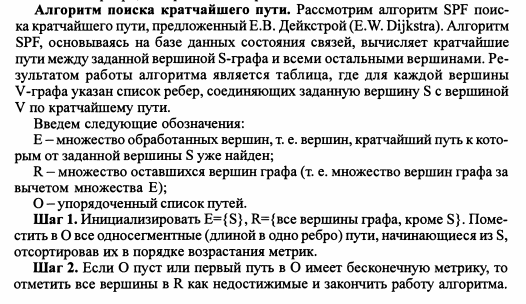


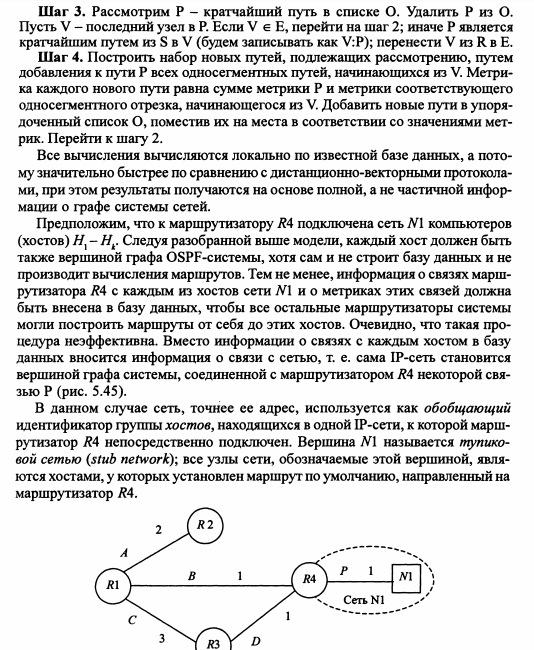
**OSPF-протокол. Назначение и основные принципы работы**











***Билет №24***

**Оборудование LAN**

## **Построение сети**

Существует множество способов классификации сетей. Основным критерием классификации принято считать способ администрирования. То есть в зависимости от того, как организована сеть и как она управляется, её можно отнести к локальной, распределённой, городской или глобальной сети. Управляет сетью или её сегментом [сетевой администратор](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80). В случае сложных сетей их права и обязанности строго распределены, ведётся документация и журналирование действий команды администраторов.

[Компьютеры](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) могут соединяться между собой, используя различные среды доступа: медные проводники ([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)), оптические проводники ([оптические кабели](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)) и через радиоканал (беспроводные технологии). Проводные, оптические связи устанавливаются через [Ethernet](https://ru.wikipedia.org/wiki/Ethernet" \o "Ethernet) и прочие средства. Отдельная локальная вычислительная сеть может иметь связь с другими локальными сетями через шлюзы, а также быть частью глобальной вычислительной сети (например, [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82)) или иметь подключение к ней.

Чаще всего локальные сети построены на технологиях [Ethernet](https://ru.wikipedia.org/wiki/Ethernet" \o "Ethernet). Следует отметить, что ранее использовались протоколы [Frame Relay](https://ru.wikipedia.org/wiki/Frame_Relay" \o "Frame Relay), [Token ring](https://ru.wikipedia.org/wiki/Token_ring" \o "Token ring), которые на сегодняшний день встречаются всё реже, их можно увидеть лишь в специализированных лабораториях, учебных заведениях и службах. Для построения простой локальной сети используются [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), [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), точки беспроводного доступа, беспроводные маршрутизаторы, [модемы](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BC) и сетевые адаптеры. Реже используются преобразователи (конвертеры) среды, усилители сигнала (повторители разного рода) и специальные антенны.

Маршрутизация в локальных сетях используется примитивная, если она вообще необходима. Чаще всего это статическая либо динамическая маршрутизация (основанная на протоколе [RIP](https://ru.wikipedia.org/wiki/RIP2)).

Иногда в локальной сети организуются *рабочие группы* — формальное объединение нескольких компьютеров в группу с единым названием.

[Сетевой администратор](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80) — человек, ответственный за работу локальной сети или её части. В его обязанности входит обеспечение и контроль физической связи, настройка активного оборудования, настройка общего доступа и предопределённого круга программ, обеспечивающих стабильную работу сети.

Технологии локальных сетей реализуют, как правило, функции только двух нижних уровней модели [OSI](https://ru.wikipedia.org/wiki/%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) - физического и канального. Функциональности этих уровней достаточно для доставки кадров в пределах стандартных топологий, которые поддерживают LAN: звезда, общая шина, кольцо и дерево. Однако из этого не следует, что компьютеры, связанные в локальную сеть, не поддерживают протоколы уровней, расположенных выше канального. Эти протоколы также устанавливаются и работают на узлах локальной сети, но выполняемые ими функции не относятся к технологии LAN.

**PPP- протокол. Назначение. Компонентный состав**

**PPP** ([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) *Point-to-Point Protocol*) — двухточечный протокол [канального уровня](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) (Data Link) [сетевой модели OSI](https://ru.wikipedia.org/wiki/%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). Обычно используется для установления прямой связи между двумя узлами сети, причём он может обеспечить аутентификацию соединения, шифрование (с использованием [ECP](https://ru.wikipedia.org/w/index.php?title=Encryption_Control_Protocol&action=edit&redlink=1), [RFC 1968](https://tools.ietf.org/html/rfc1968)) и сжатие данных. Используется на многих типах физических сетей: нуль-модемный кабель, телефонная линия, сотовая связь и т. д.

Часто встречаются подвиды протокола PPP, такие, как Point-to-Point Protocol over Ethernet ([PPPoE](https://ru.wikipedia.org/wiki/PPPoE" \o "PPPoE)), используемый для подключения по Ethernet, и иногда через DSL; и Point-to-Point Protocol over [ATM](https://ru.wikipedia.org/wiki/ATM) ([PPPoA](https://ru.wikipedia.org/wiki/PPPoA" \o "PPPoA)), который используется для подключения по ATM Adaptation Layer 5 (AAL5), который является основной альтернативой [PPPoE](https://ru.wikipedia.org/wiki/PPPoE" \o "PPPoE) для [DSL](https://ru.wikipedia.org/wiki/XDSL).

### Обнаружение закольцованных связей[[править](https://ru.wikipedia.org/w/index.php?title=PPP_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)&veaction=edit&section=4) | [править код](https://ru.wikipedia.org/w/index.php?title=PPP_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)&action=edit&section=4)]

PPP обнаруживает закольцованные связи, используя особенность, включающую [magic numbers](https://ru.wikipedia.org/wiki/Magic_number_(programming)" \o "Magic number (programming)). Когда узел отправляет PPP LCP сообщения, они могут включать в себя магическое число. Если линия закольцована, узел получает сообщение LCP с его собственным магическим числом вместо получения сообщения с магическим числом клиента.

## **Наиболее важные особенности[**[**править**](https://ru.wikipedia.org/w/index.php?title=PPP_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)&veaction=edit&section=5)**|**[**править код**](https://ru.wikipedia.org/w/index.php?title=PPP_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)&action=edit&section=5)**]**

* [Link Control Protocol](https://ru.wikipedia.org/wiki/LCP) устанавливает и завершает соединения, позволяя узлам определять настройки соединения. Также он поддерживает и байт-, и биториентированные кодировки.
* [Network Control Protocol](https://ru.wikipedia.org/wiki/Network_Control_Protocol) используется для определения настроек сетевого уровня, таких как сетевой адрес или настройки сжатия, после того как соединение было установлено.

## **Конфигурационные опции PPP[**[**править**](https://ru.wikipedia.org/w/index.php?title=PPP_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)&veaction=edit&section=6)**|**[**править код**](https://ru.wikipedia.org/w/index.php?title=PPP_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)&action=edit&section=6)**]**

Так как в PPP входит [LCP](https://ru.wikipedia.org/wiki/LCP) протокол, то можно управлять следующими LCP параметрами:

* **Аутентификация**. [RFC 1994](https://tools.ietf.org/html/rfc1994) описывает Challenge Handshake Authentication Protocol ([CHAP](https://ru.wikipedia.org/wiki/Challenge_Handshake_Authentication_Protocol)), который является предпочтительным для проведения аутентификации в PPP, хотя Password Authentication Protocol ([PAP](https://ru.wikipedia.org/wiki/Password_Authentication_Protocol)) иногда ещё используется. Другим вариантом для аутентификации является Extensible Authentication Protocol ([EAP](https://ru.wikipedia.org/wiki/EAP)).
* **Сжатие**. Эффективно увеличивает пропускную способность PPP соединения за счёт сжатия данных в кадре. Наиболее известными алгоритмами сжатия PPP кадров являются Stacker и Predictor.
* **Обнаружение ошибок**. Включает Quality-Protocol и помогает выявить петли обратной связи посредством Magic Numbers [RFC 1661](https://tools.ietf.org/html/rfc1661).
* **Многоканальность**. Multilink PPP (MLPPP, MPPP, MLP) предоставляет методы для распространения трафика через несколько физических каналов, имея одно логическое соединение. Этот вариант позволяет расширить пропускную способность и обеспечивает балансировку нагрузки.

## **PPP-кадр[**[**править**](https://ru.wikipedia.org/w/index.php?title=PPP_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)&veaction=edit&section=7)**|**[**править код**](https://ru.wikipedia.org/w/index.php?title=PPP_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB)&action=edit&section=7)**]**

Каждый кадр PPP всегда начинается и завершается байтом 0x7E. Затем следует байт адреса и байт управления, которые тоже всегда равны 0xFF и 0x03, соответственно. В связи с вероятностью совпадения байтов внутри блока данных с зарезервированными флагами существует система автоматической корректировки «проблемных» данных с последующим восстановлением.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Флаг 0x7E** | **Адрес 0xFF** | **Управление 0x03** | **Данные** | **Контрольная сумма** | **Флаг 0x7E** |
| 1 | 1 | 1 | 1494 | 2 | 1 |

Поля «Флаг», «Адрес» и «Управление» (заголовок кадра [HDLC](https://ru.wikipedia.org/wiki/HDLC)) могут быть опущены и не передаваться, но это произойдёт, если PPP в процессе конфигурирования (используя LCP) договорится об этом. Если PPP инкапсулирован в [L2TP](https://ru.wikipedia.org/wiki/L2TP)-пакеты, то поле «Флаг» не передаётся.