

## Лабораторная работа № 2

### ИЗУЧЕНИЕ МОДЕЛИ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ

**Цель работы:** Изучить эталонную сетевую модель OSI.

#### Краткие сведения из теории

**Сетевая модель** - это модель взаимодействия сетевых протоколов. А протоколы в свою очередь, это стандарты, которые определяют каким образом, будут обмениваться данными различные программы.

OSI расшифровывается как Open System Interconnection. На русском языке это звучит следующим образом: Сетевая модель взаимодействия открытых систем (эталонная модель). Эту модель можно смело назвать стандартом. Именно этой модели придерживаются производители сетевых устройств, когда разрабатывают новые продукты.

Сетевая модель OSI состоит из 7 уровней, причем принято начинать отсчёт с нижнего.

Перечислим их:

- 7. Прикладной уровень (application layer)
- 6. Представительский уровень или уровень представления (presentation layer)
- 5. Сеансовый уровень (session layer)
- 4. Транспортный уровень (transport layer)
- 3. Сетевой уровень (network layer)
- 2. Канальный уровень (data link layer)
- 1. Физический уровень (physical layer)

Сетевая модель – это модель взаимодействия сетевых протоколов (стандартов), на каждом уровне присутствуют свои протоколы.

#### **Прикладной уровень**

Прикладной уровень или уровень приложений (application layer) – это самый верхний уровень модели. Он *осуществляет связь пользовательских приложений с сетью*. Эти приложения нам всем знакомы: просмотр веб-страниц (HTTP), передача и приём почты (SMTP, POP3), приём и получение файлов (FTP, TFTP), удаленный доступ (Telnet) и т.д.

#### **Представительский уровень**

Представительский уровень или уровень представления данных (presentation layer) – он *преобразует данные в соответствующий формат*. **Пример:** картинки (все изображения) которые вы видите на экране, передаются при пересылке файла в виде единиц и нулей (битов). Когда отправляете фотографию по электронной почте, протокол Прикладного уровня SMTP отправляет фотографию на нижний уровень, т.е. на уровень Представления. Где фотография преобразуется в удобный вид данных для более низких уровней, например, в биты.

Именно таким же образом, когда у получателя именно уровень Представления преобразует биты в полноценную фотографию, например, JPEG.

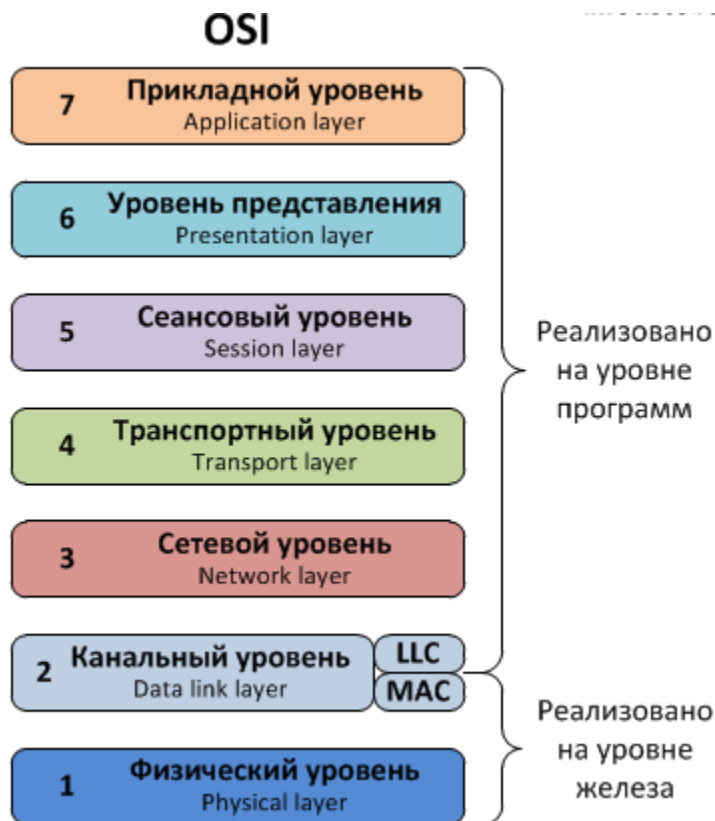


Рисунок 1 – Уровни модели OSI

### Сеансовый уровень

Сеансовый уровень или уровень сессий (session layer) – *организует сеанс связи между компьютерами*. Хорошим примером будут служить аудио и видеоконференции, на этом уровне устанавливается, каким кодеком будет кодироваться сигнал, причем этот кодек должен присутствовать на обеих машинах. Еще примером может служить протокол SMPP (Short message peer-to-peer protocol), с помощью него отправляются СМС и USSD запросы. И последний пример: PAP (Password Authentication Protocol) – это старый протокол для отправки имени пользователя и пароля на сервер без шифрования.

### Транспортный уровень

Транспортный уровень (transport layer) – этот уровень *обеспечивает надёжность передачи данных от отправителя к получателю*. Например, вы общаетесь с помощью веб-камеры. Нужна ли здесь надежная доставка каждого бита переданного изображения? Конечно нет, если потеряется несколько битов из потокового видео Вы даже этого не заметите, даже картинка не изменится (м.б. изменится цвет одного пикселя из 900000 пикселей, который промелькнет со скоростью 24 кадра в секунду).

А теперь приведем такой пример: пересылается (например, через почту) в архиве важная информация или программа. Вы скачиваете себе на

компьютер этот архив. Вот здесь надёжность нужна 100%, т.к. если пару бит при закачке архива потеряются – Вы не сможете затем его разархивировать, т.е. извлечь необходимые данные. Или представьте себе отправку пароля на сервер, и в пути один бит потерялся – пароль уже потеряет свой вид и значение изменится.

Таким образом, когда мы смотрим видеоролики в интернете, иногда мы видим некоторые артефакты, задержки, шумы и т.п. А когда мы читаем текст с веб-страницы – потеря (или искажение) букв не допустима, и когда скачиваем программы – тоже все проходит без ошибок.

На этом уровне выделяются два протокола: UDP и TCP. UDP протокол (User Datagram Protocol) передает данные без установления соединения, не подтверждает доставку данных и не делает повторы. TCP протокол (Transmission Control Protocol), который перед передачей устанавливает соединение, подтверждает доставку данных, при необходимости делает повтор, гарантирует целостность и правильную последовательность загружаемых данных.

Следовательно, для музыки, видео, видеоконференций и звонков используем UDP (передаем данные без проверки и без задержек), а для текста, программ, паролей, архивов и т.п. – TCP (передача данных с подтверждением о получении, затрачивается больше времени).

### **Сетевой уровень**

Сетевой уровень (network layer) – этот уровень *определяет путь, по которому данные будут переданы*. Это третий уровень Сетевой модели OSI, и существуют такие устройства, которые как раз и называют устройствами третьего уровня – маршрутизаторы.

IP-адрес – это логический адрес в сети.

На этом уровне достаточно много протоколов.

Команда ping – это работает протокол ICMP.

Те самые маршрутизаторы используют протоколы этого уровня для маршрутизации пакетов (RIP, EIGRP, OSPF).

### **Канальный уровень**

Канальный уровень (data link layer) – *отвечает за доставку кадров между устройствами, подключенными к одному сетевому сегменту* (может использоваться для обнаружения и исправления ошибок, возникших на физическом уровне). Кадры канального уровня не пересекают границ сетевого сегмента.

MAC-адресе является физическим адресом. Устройства канального уровня – коммутаторы, концентраторы и т. п.

IEEE (Institute of Electrical and Electronics Engineers – Институт инженеров по электротехнике и электронике) определяет канальный уровень двумя подуровнями: LLC и MAC.

LLC – управление логическим каналом (Logical Link Control), создан для взаимодействия с верхним уровнем.

MAC – управление доступом к передающей среде (Media Access Control), создан для взаимодействия с нижним уровнем.

Пример: в компьютере имеется сетевая карта, для взаимодействия с ней существует драйвер. Драйвер – это некоторая **программа** - верхний подуровень канального уровня, через которую как раз и можно связаться с нижними уровнями, а точнее с микропроцессором (**железо**) – нижний подуровень канального уровня.

Типичных представителей на этом уровне много. PPP (Point-to-Point) – это протокол для связи двух компьютеров напрямую. FDDI (Fiber Distributed Data Interface) – стандарт передаёт данные на расстояние до 200 километров.

### Физический уровень

Физический уровень (physical layer) – самый нижний уровень, *непосредственно осуществляющий передачу потока данных*. Протоколы: Bluetooth, IRDA (Инфракрасная связь), медные провода (витая пара, телефонная линия), Wi-Fi, и т.д.

### Инкапсуляция

*Инкапсуляция – это процесс передачи данных с верхнего уровня приложений вниз (по стеку протоколов) к физическому уровню, чтобы быть переданными по сетевой физической среде (витая пара, оптическое волокно, Wi-Fi, и др.).* Причём на каждом уровне различные протоколы добавляют к передающимся данным свою информацию.

Сетевая модель OSI состоит из 7 уровней (уровень приложений, уровень представления, сеансовый, транспортный, сетевой, канальный и физический). Все сетевые устройства работают согласно модели OSI, только некоторые используют все 7 уровней, а другие меньше. Это позволяет обрабатывать поступающие данные в несколько раз быстрее.

Например, компьютер использует все 7 уровней, маршрутизатор – 3 нижних уровня, коммутатор – только 2 нижних уровня.

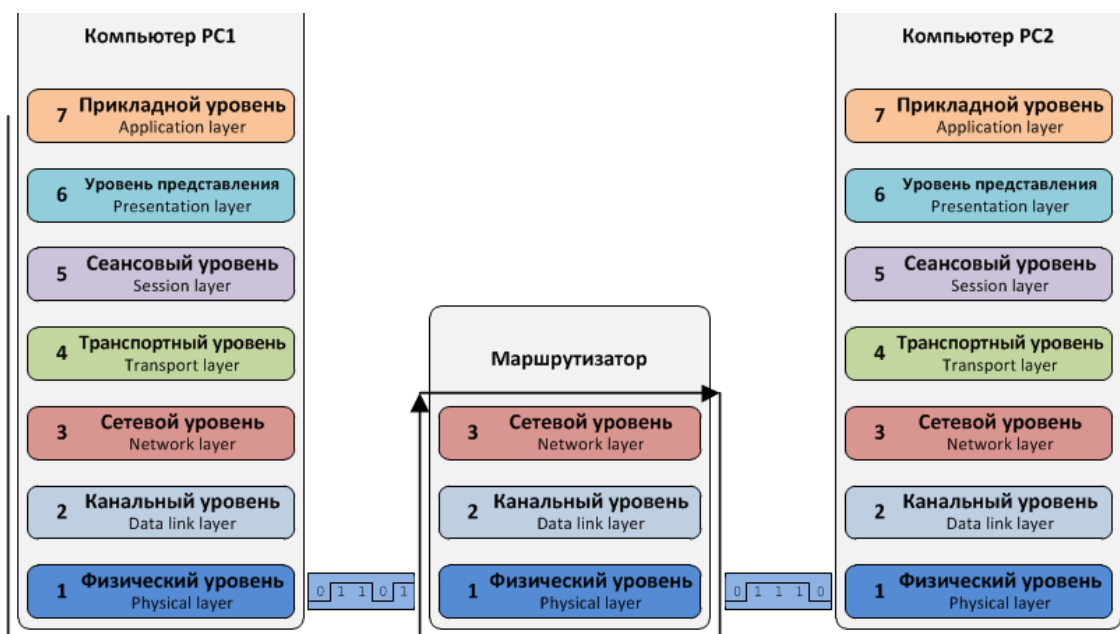


Рисунок 2 – Уровни модели OSI различных сетевых устройств

На рисунке 2 показано взаимодействие двух компьютеров, между которыми находится маршрутизатор. Компьютерами PC1 и PC2 могут быть как домашние компьютеры, так и сервера. Маршрутизатор, как и говорилось выше, работает только на трех уровнях модели, их (трех уровней) достаточно, чтобы проложить маршрут в любой сети.

Теперь перейдем к самому процессу инкапсуляции, декапсуляции.

### Инкапсуляция и декапсуляция

Проще будет разобрать эти процессы инкапсуляции и декапсуляции на примере. Допустим, Вы захотели посмотреть какую-то веб-страничку, ввели в адресную строку браузера адрес сайта и нажали кнопку Enter. После этого браузер должен отправить запрос на сервер (на котором хранится эта веб-страничка), с целью получения данных. Вот как раз на этом этапе, введенный Вами адрес сайта является данными, которые должны передаться на сервер в виде запроса.



Рисунок 3 – Прикладной уровень модели OSI

Эти данные опускаются с уровня приложений, на уровень представления данных.

На этом уровне Ваш компьютер преобразует строку введенного текста (адреса) в формат удобный для передачи далее на нижний уровень.



Рисунок 4 – Уровень представления модели OSI

Далее данные (уже не текст) поступают на сеансовый уровень, но на нём (в данном случае) нам нет необходимости использовать протоколы (этого уровня), и поэтому данные передаются далее.

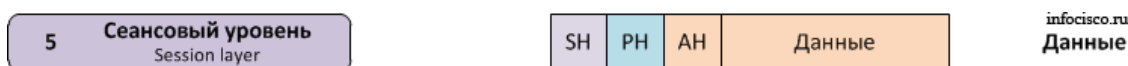


Рисунок 5 – Сеансовый уровень модели OSI

Транспортный уровень получает данные и определяет, что дальше они должны быть переданы используя протокол TCP. Перед передачей транспортный уровень разбивает данные на кусочки данных и добавляет к каждому кусочку заголовок, в котором содержится информация о логических портах компьютеров (с какого данные были посланы (например - 1223) и для какого предназначаются (в данном случае 80)). На транспортном уровне эти кусочки данных с заголовком называются сегментами. Сегменты передаются дальше вниз к сетевому уровню.

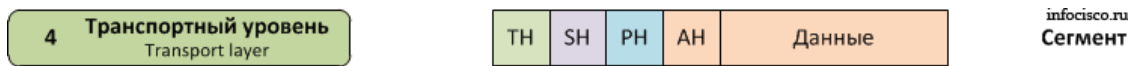


Рисунок 6 – Транспортный уровень модели OSI

Сетевой уровень, получая каждый сегмент, разделяет его на еще более маленькие части и к каждой части добавляет свой заголовок. В заголовке сетевого уровня указываются логические сетевые адреса отправителя (Ваш компьютер) и получателя (Сервер).

Логические сетевые адреса – это всем известные IP-адреса.

Эти маленькие кусочки данных уже с несколькими заголовками (на верхних уровнях тоже добавляются специфичные заголовки) на сетевом уровне называются пакетами, которые в свою очередь передаются на канальный уровень.

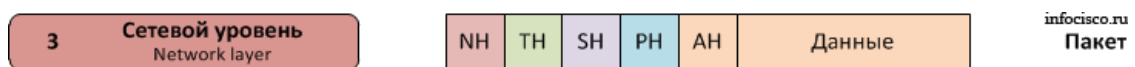


Рисунок 7 – Сетевой уровень модели OSI

На канальном уровне пакеты разделяются на еще более маленькие кусочки данных, и к ним помимо опять добавляемого заголовка, только уже канального уровня, добавляется еще и трейлер. На этом уровне в заголовках содержатся физические адреса устройств – передающего и для кого они предназначаются, а в трейлере находится вычисленная контрольная сумма, некий код (информация), который используется для определения целостности данных.

Физические адреса устройств – это MAC-адреса.

Эти очень маленькие кусочки данных именуется кадрами или фреймами (одно и то же). Далее кадры передаются на физический уровень.



Рисунок 8 – Канальный уровень модели OSI

На физический уровень кадры передаются уже в виде сигналов битов и следуют через другие сетевые устройства в пункт назначения.

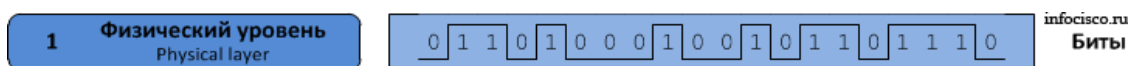


Рисунок 9 – Физический уровень модели OSI

Весь процесс преобразования данных (с верхнего уровня) в сигналы (на нижний уровень) называется инкапсуляцией. Посмотрите на рисунок ниже, там представлена общая схема инкапсулирования с верхнего уровня на нижний:

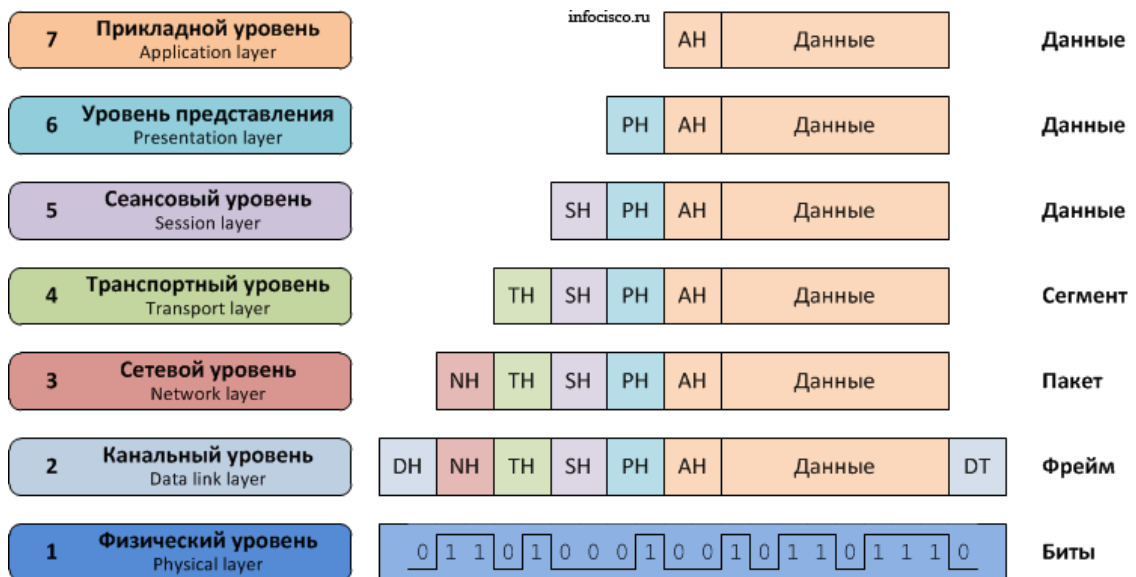


Рисунок 10 – Иерархия уровней модели OSI

Далее сигналы, проходя через несколько сетевых устройств (в нашем случае это маршрутизатор и коммутатор), доходят до получателя, в данном случае до сервера.

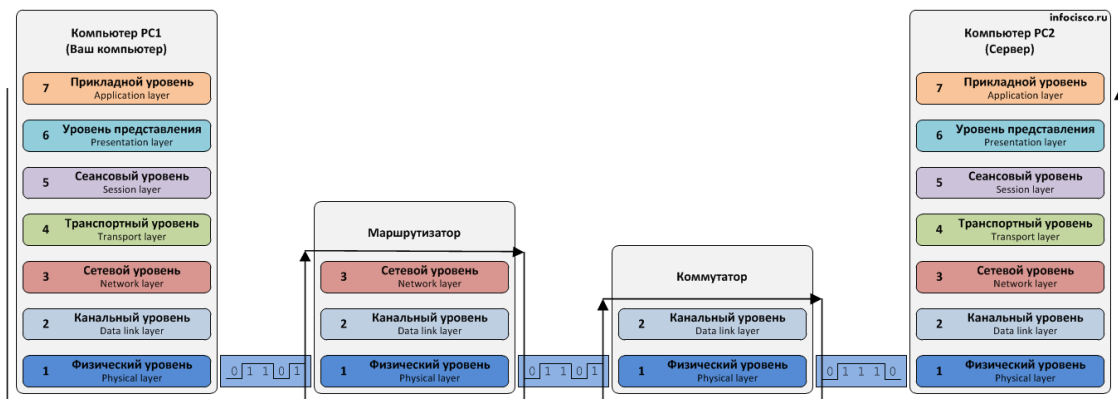


Рисунок 11 – Уровни модели OSI различных сетевых устройств

Сетевая карта сервера принимает биты (на физическом уровне) и преобразует их в кадры (для канального уровня). Канальный уровень в обратной последовательности должен преобразовать кадры в пакеты (для сетевого уровня), только перед преобразованием уровень сначала смотрит на MAC-адрес (физический адрес) получателя, он должен совпадать с MAC-адресом сетевой карты, иначе кадр будет уничтожен. Затем канальный уровень (в случае совпадения MAC-адреса) высчитывает сумму полученных данных и сравнивает полученное значение со значением трейлера. Значение трейлера высчитывалось на Вашем компьютере, а теперь оно, после передачи по проводам, сравнивается с полученным значением на сервере и, если они совпадают, кадр преобразуется в пакет. Если проверочный код целостности данных рознится – кадр незамедлительно уничтожается.

На сетевом уровне происходит проверка логического адреса (IP-адреса), в случае успешной проверки пакет преобразуется в сегмент, попадая на транспортный уровень.

На транспортном уровне проверяется информация из заголовка, что это за сегмент, какой используется протокол, для какого логического порта предназначается и т.п. Протокол использовался TCP, поэтому назад на Ваш компьютер посылается уведомление о прибытии сегмента. Как говорилось выше (когда данные упаковывали в сегмент) в том случае использовался 80 порт назначения. Т.к. на веб-сервере как раз открыт этот порт, данные передаются дальше на верхний уровень.

На верхних уровнях запрос (введенный адрес сайта) обрабатывается веб-сервером (проверяется, доступна-ли запрашиваемая веб-страничка).

Этот *процесс преобразования сигналов из провода в данные называется процессом декапсуляции.*

После того, как страница будет найдена на сервере, она (текст, изображения, музыка) преобразуется в цифровой код, удобный для инкапсулирования. Большой объем данных делится на части и поступает ниже на уровень – транспортный. Там кусочек данных преобразуется в сегмент, только порт назначения теперь будет тот, с которого вы посылали (1223). Сегмент преобразуется в пакет, в заголовке которого содержится IP-адрес вашего компьютера и переходит ниже. На канальном уровне пакет в свою очередь преобразуется в кадры и добавляется заголовок и трейлер. В заголовок помещается MAC-адрес назначения (в данном случае это будет адрес шлюза), а в трейлер проверочный код на целостность данных. Далее сетевая карта посылает кадры в виде сигналов по кабелю по направлению к Вашему компьютеру.

Так и происходит сетевой обмен данными, инкапсуляция и декапсуляция.

## **PDU**

Те кусочки данных (вместе с заголовками), которые переходят с уровня на уровень (с добавлением заголовков или наоборот) называются **Protocol Data Unit** или **PDU**. Если перевести литературно на русский язык, то получается **фрагмент данных на каждом уровне модели.**

## **Описание сетевых устройств**

### **Сетевая карта**

Сетевая карта нужна вашему компьютеру для взаимодействия (обмена данными) с другими сетевыми устройствами.

По способу реализации сетевой карты в Ваш компьютер можно разделить их на 3 типа: встроенные, внутренние и внешние.

Встроенная карта – это когда часть сетевой платы встроена в материнскую плату, т.е. в материнскую плату встроены разъем для подключения кабеля и чип для обработки информации (дополнительные функции, которые должны обрабатываться на сетевой плате обрабатываются



центральным процессором с помощью специального драйвера). Сейчас все ноутбуки имеют такую реализацию, да и для нормальной платы нет места внутри корпуса. Домашние компьютеры в настоящее время не лишены решения встроенной сетевой карты. Производители материнских плат позаботились о том, чтобы при покупке у Вас сразу была и сетевая карта, и звуковая карта, и видео карта, и еще какая-нибудь карта.

Внутренняя – это когда сетевая карта устанавливается внутрь компьютера (системного блока). Это значит, что имеется отдельная сетевая карта, с сетевым разъёмом, вставленная через специальный разъем (PCI, PCI-E, ISA) в материнскую плату. Она может быть извлечена из одного компьютера и вставлена в другой.

Внешняя – сейчас такую реализацию можно встретить, когда подключают планшет или коммуникатор через сетевой интерфейс к локальной сети, также на ноутбуках без сетевого разъёма или с устаревшим (или не работающим) сетевым разъёмом. Чаще всего это сетевая карта в пластиковом корпусе с USB кабелем для подключения к компьютеру.

### **Отличие модема от адаптера**

Часто можно встретить такое название сетевого устройства как модем. В одном магазине мы видим Wi-Fi-модем, а в другом Wi-Fi-адаптер.

### **Исследование: Чем является модем и в чём отличие от адаптера.**

Итак, проведем небольшое исследование и выясним что такое модем, что такое адаптер, и чем они отличаются.

Разберем, что обозначают сами слова модем (modem) и адаптер (adapter).

Модем (modem). Это слово является акронимом, т.е. аббревиатурой (сокращенным написанием) образованной от начальных букв английских слов mo(dulator) и dem(odulator). Модулятор (лат. modulator — соблюдающий ритм) — устройство, изменяющее параметры несущего сигнала в соответствии с изменениями передаваемого (информационного) сигнала. Демодулятор - детектор радиоприёмного устройства, восстанавливающий информацию из радиосигнала, заложенную в него модулятором.

Проще говоря, модем преобразовывает биты данных в сигнал (соответствующий среде передачи данных) и наоборот, сигнал – в данные.

Адаптер (adapter или adaptor). В переводе на русский получается: переходник, переходное устройство, приспособление, переходная муфта, соединительное устройство, звукосниматель, тот, кто переделывает литературное произведение (адаптирует его). Так как слово Адаптер происходит от *адаптировать*, рассмотрим и его. Адаптировать - это значит приспособить, облегчить, настроить.

Так же, как и с модемом, говоря простыми словами, адаптер принимает сигнал из кабеля (или любой другой передающей среды) и преобразовывает его в биты данных и наоборот.

Слово «модем» используется только как устройство модуляции/демодуляции: модем, софт-модем, беспроводной модем, голосовой модем, спутниковый модем, факс-модем.

А вот со словом «адаптер» дела обстоят интереснее: блок питания, адаптер сети, DOF-адаптер, Адаптер-переходник (переходное кольцо для фотооптики), сетевая карта (плата), интерфейс между устройствами, видеоадаптер, графический адаптер (преобразует графический образ в иную форму). И еще несколько вырезок из предложений, где встречается это слово:

Слово «модем» используется только как устройство модуляции/демодуляции. «Адаптер» же намного более универсальное слово, хоть и имеет общий смысл. Везде, где требуется что-то конвертировать, преобразовать, изменить и приспособить, можно использовать слова «адаптер». Собственно, и функции модема подходят под определение адаптера, поэтому мы можем считать эти слова (в пределах сетевых технологий) одинаковыми по значению и при необходимости заменять одно другим. Причем, так как «модем» более ‘узкое’ и специализированное слово, им заменять слово «адаптер» следует с осторожностью. «Адаптером» же наоборот, можно заменить любой «модем».

При передаче данных в какой-либо физической среде происходит постепенное затухание сигнала.

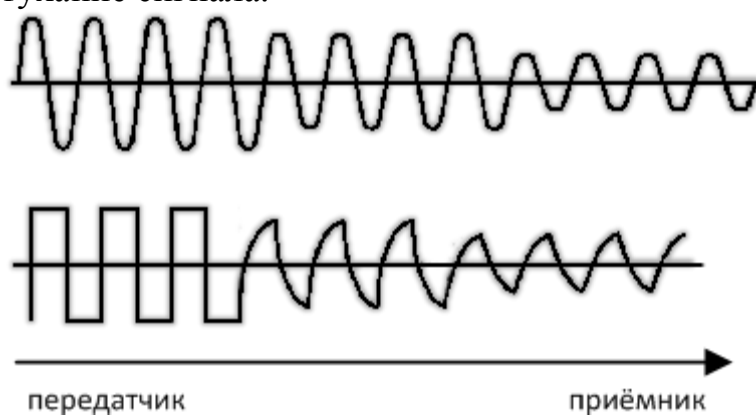


Рисунок 12 – Искажение импульсов в линии связи

Длина физической среды варьируется в зависимости от используемой технологии. Например, для коаксиального кабеля соответствующего технологии «Толстый Ethernet» максимальная длина сегмента 500 метров, для коаксиального кабеля «Тонкий Ethernet» - 185 метров, для кабеля витой пары – 200 метров, многомодовое оптоволокно – до 500 – 2000 метров (в зависимости от технологии), одномодовое оптоволокно – до 50 километров, Bluetooth или блютуз – до 200 метров, Wi-Fi – до 300 метров, с 29 июля 2011 года IEEE выпустил стандарт IEEE 802.22 (Super Wi-Fi) – в радиусе до 100 км, WiMAX – до 80 км, GSM – от 400 м до 50 км.

### **Повторитель**

Из его названия становится ясно, что он повторяет полученный сигнал. Другими словами, там, где сигнал уже почти становится не распознаваемым (лучше до такого состояния не доводить, будет куча ошибок, уменьшится скорость, стабильность и т.д.), туда устанавливается повторитель. В него поступает затухающий сигнал, репитер его распознаёт и отправляет в

первоначальном виде дальше. Таким образом, длина увеличивается на необходимое нам расстояние.

Под повторителем подразумевается устройство с одним входом и одним выходом, позволяя соединить два сетевых устройства (например, две сетевые карты, установленные на компьютерах).

### **Концентратор**

Чтобы соединить несколько компьютеров в локальную сеть можно воспользоваться концентратором, или как его еще называют хабом (hub). По принципу работы концентратор можно безошибочно назвать многопортовым повторителем. Работает он следующим образом: как только на один порт приходит сигнал, концентратор распознаёт сигнал и передаёт его на все активные порты. Каждый подключенный компьютер получает этот сигнал, распознаёт его до канального уровня модели OSI, и проверяет, предназначались ли эти данные ему (если не ему, то отбрасывает эти данные), и т.д.

На данный момент концентраторы совсем не популярны (хотя и до сих пор используются), т.к. существуют проблемы, связанные именно с ними.

Во-первых, если на два (и больше) порта хаба придет одновременно какой-либо сигнал, то произойдет столкновение этих сигналов, они смешаются друг с другом и восстановить информацию станет не возможным. В маленьких сетях это будет происходить редко, а вот с постепенным увеличением числа компьютеров будут увеличиться и столкновения.

Процесс столкновения сигналов называется **коллизией**.

Во-вторых, из-за того, что полученный сигнал распространяется на все порты, злоумышленник может подключиться к локальной сети и начать собирать всю получаемую информацию с концентратора.

### **Сетевой мост**

Сетевой мост (bridge, бридж). Переход на следующий уровень обозначает переход с физического (первого) уровня, на канальный (второй).

Мост - это уже устройство второго уровня сетевой модели OSI и это самое главное отличие от концентраторов и репитеров (напомню, что последние два, устройства первого (физического) уровня, и когда сигнал приходит на их порт, они его распознают и повторно отправляют во все активные порты).

Когда на сетевой мост приходит сигнал, он на первом уровне распознаёт сигнал, а затем преобразовывает (декапсулирует) сигналы в кадры (фреймы). Здесь уже начинается работа второго уровня, на нём к PDU добавлялись хидер (заголовок) и трейлер (прицеп). Мост высчитывает контрольную сумму кадра и сравнивает полученный результат со значением в трейлере, если значения не совпадают, фрейм уничтожается. Если сумма совпадает, то мост изучает заголовок кадра, в нем содержится физический адрес (MAC) получателя. У самого моста в памяти хранится таблица, в которой содержится информация – на какой порт отправить кадр, если MAC-адрес такой-то. И если в таблице физический адрес найден, то кадр

передаётся на соответствующий порт. Если не найден, включается механизм поиска MAC-адреса в подключенных сегментах сети.

Вот так мост и работает, плюсы очевидны:

- все фреймы, поступившие в сетевой мост с ошибками в контрольной сумме (CRC) и не соответствующие стандарту отбрасываются (уничтожаются);
  - если одновременно на разные порт приходит сигнал, то мост обрабатывает оба в порядке очереди;
  - поддержка дополнительных протоколов;
- Минус тоже есть: задержка передачи данных, т.к. каждый фрейм приходится обрабатывать центральным процессором.

### **Коммутатор**

Улучшить ситуацию (весьма значительно) может сетевой коммутатор (switch, свитч, переключатель). Он тоже является устройством второго уровня и даже иногда можно услышать, что его называют многопортовым мостом. У коммутатора тоже есть таблица, связывающая MAC-адреса с портами – таблица коммутации, но в отличие от моста коммутатор ищет MAC-адрес аппаратно, а мост – программно.

У коммутатора есть три режима коммутации:

- Store and Forward – Сохранить и передать. В этом режиме свитч получая сигнал, распознаёт его. Далее проверяет контрольную сумму фрейма (CRC). Затем узнаёт из заголовка фрейма физический адрес получателя. Ищет этот адрес в таблице коммутации и отправляет фрейм (перед отправкой преобразовав его в сигналы) в соответствующий порт (или в некоторых случаях – порты, об этом Вы узнаете в отдельной статье про коммутаторы, с примерами в лабораторных).
- Cut-through – Сквозной. Это самый быстрый способ передачи фрейма на коммутаторе. Когда свитч получает сигнал, он его распознаёт и преобразует в фрейм, затем сразу считывает заголовок фрейма и получив физический адрес получателя отправляет фрейм на соответствующий порт. Контрольную сумму не высчитывает, а, следовательно, могут появиться ошибки.
- Fragment-free – Безфрагментный. Этот режим тоже не проверяет контрольную сумму, но обнаружить ошибку может. Перед отправкой кадра на соответствующий порт коммутатор производит проверку первых 64 байтов кадра. Часто ошибка возникает именно вначале фрейма.

### **Маршрутизатор**

Маршрутизатор, роутер, сетевой шлюз, просто шлюз, router, gateway, а также (не всегда) брандмауэр, brandmauer, firewall, файервол – это всё одно устройство (примечание: шлюз, firewall и т.п. не обязательно являются роутерами, и эти названия соответствуют выполняемым задачам, которые в свою очередь может выполнять роутер).

Маршрутизатор – это устройство третьего ( сетевого) уровня сетевой модели OSI. Роутер уже пересылает пакеты по сети. Также он, руководствуясь правилами как статически, так и динамически может выбирать маршрут (путь) следования пакета в пункт назначения (получателю).

Так как роутер работает на 3 уровне, он уже обрабатывает логические адреса хостов в сети – IP-адреса. У роутеров имеется специальная таблица маршрутизации, которой он руководствуется при выборе наилучшего пути. Заполняется эта таблица специальными протоколами маршрутизации.

Роутер способен разделять сети на подсети, а также объединять сети с разной физической средой передачи данных (например, оптика, медь и т.д.).

## Как работают сетевые устройства согласно сетевой модели OSI

Передача данных между двумя соединенными компьютерами.

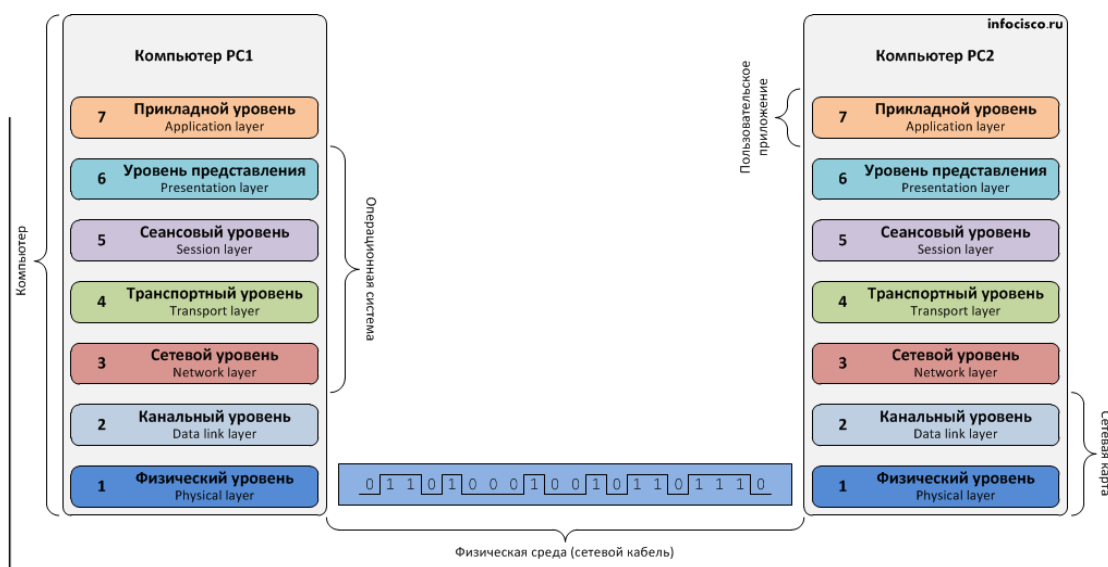


Рисунок 13 – Непосредственная связь двух компьютеров

Приложение на компьютере PC1 отправляет данные другому приложению находящемуся на другом компьютере PC2. Начиная с верхнего уровня (уровень приложений) данные направляются к сетевой карте на канальный уровень. На нём сетевая карта преобразует фреймы в биты и отправляет в физическую среду (например, кабель витую пару). На другой стороне кабеля поступает сигнал, и сетевая карта компьютера PC2 принимает эти сигнала, распознавая их в биты и формируя из них фреймы. Данные (содержащиеся в фреймах) декапсулируются к верхнему уровню, и когда доходят до уровня приложений, соответствующая программа на компьютере PC2 получает их.

### Повторитель. Концентратор.

*Репитер* и *концентратор* работают на одном и том же уровне, поэтому касательно сетевой модели OSI они изображаются одинаково. Для

удобства представлений сетевых устройств будем их отображать между нашими компьютерами.

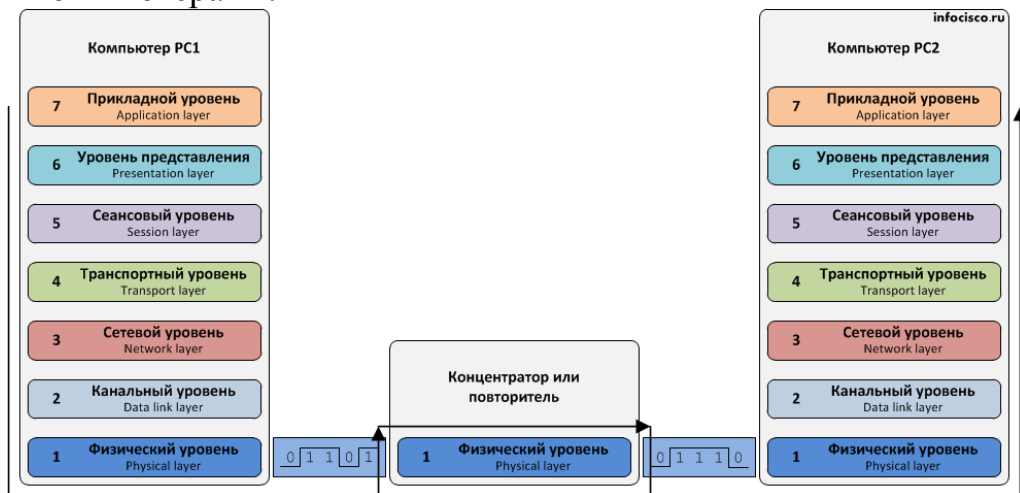


Рисунок 14 – Связь двух компьютеров через концентратор

Репитер и концентратор устройства первого (физического) уровня. Они принимают сигнал, распознают его, и пересылают сигнал далее во все активные порты.

### Сетевой мост. Коммутатор.

*Сетевой мост* и *коммутатор* тоже работают на одном уровне (канальном) и изображаются они соответственно одинаково.

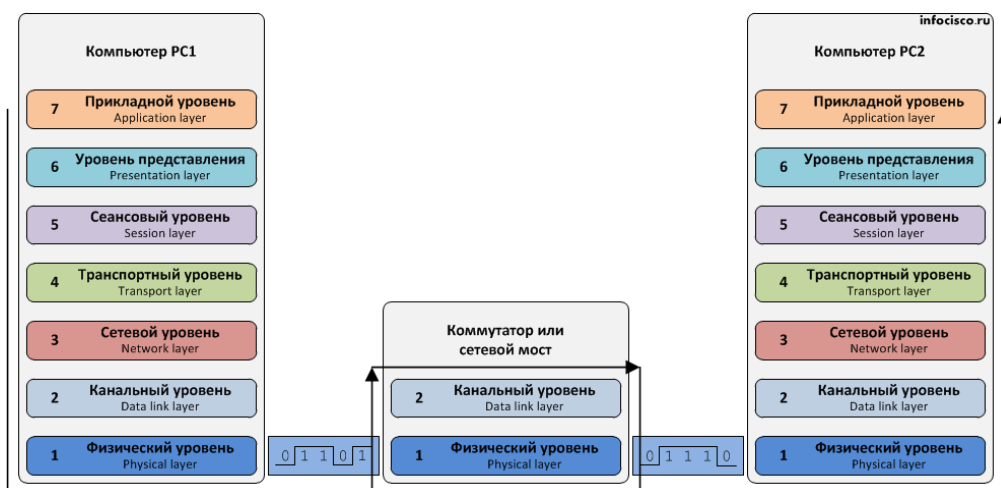


Рисунок 15 – Связь двух компьютеров через коммутатор

Оба устройства уже второго уровня, поэтому помимо распознавания сигнала (подобно концентраторам на первом уровне) они декапсулируют его (сигнал) в фреймы. На втором уровне сравнивается контрольная сумма трейлера (прицепа) фрейма. Затем из заголовка фрейма узнаётся MAC-адрес получателя, и проверяется его наличие в коммутируемой таблице. Если адрес присутствует, то фрейм обратно инкапсулируется в биты и отправляется (уже

в виде сигнала) на соответствующий порт. Если адрес не найден, происходит процесс поиска этого адреса в подключенных сетях.

### Маршрутизатор.

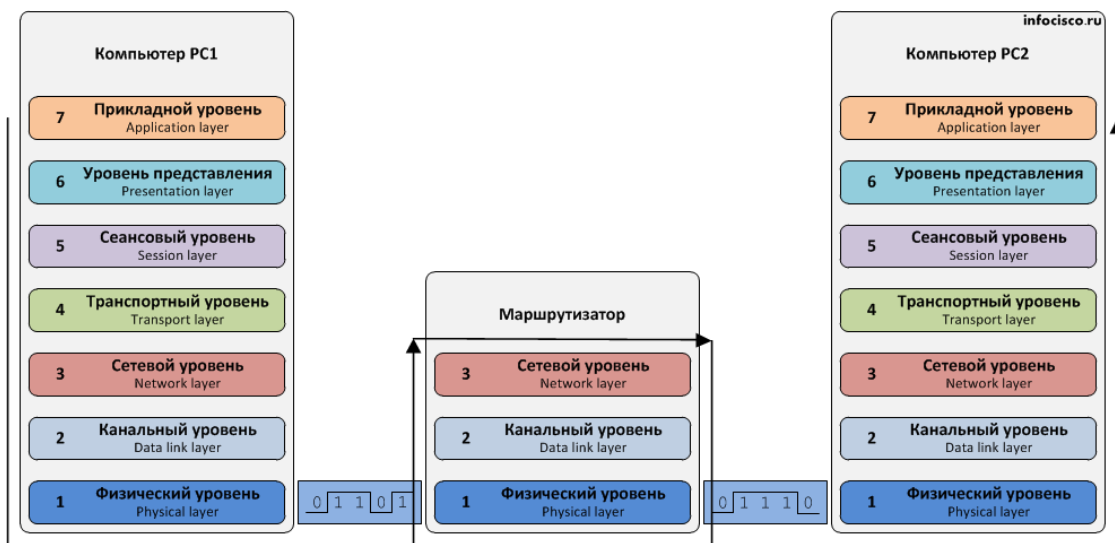


Рисунок 16 – Связь двух компьютеров через маршрутизатор

**Маршрутизатор** (или роутер) – это устройство третьего уровня. Вот как примерно роутер функционирует: На порт поступает сигнал, и роутер распознаёт его. Распознанный сигнал (биты) формируют фреймы (кадры). Сверяется контрольная сумма в трейлере и MAC-адрес получателя. Если все проверки прошли успешно, фреймы формируют пакет. На третьем уровне маршрутизатор исследует заголовок пакета. В нем присутствует IP адрес пункта назначения (получателя). На основе IP-адреса и собственной таблицы маршрутизации роутер выбирает наилучший путь следования пакетов к получателю. Выбрав путь, роутер инкапсулирует пакет в фреймы, а затем в биты и отправляет их в виде сигналов на соответствующий порт (выбранный в таблице маршрутизации).

**Интерфейс** – формально определенная логическая и/или физическая граница между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

**Протокол** – набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется **стеком протоколов**.

В сущности, термины «**протокол**» и «**интерфейс**» выражают одно и то же понятие – формализованное описание процедуры взаимодействия двух объектов, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы — правила взаимодействия модулей соседних уровней в одном узле.

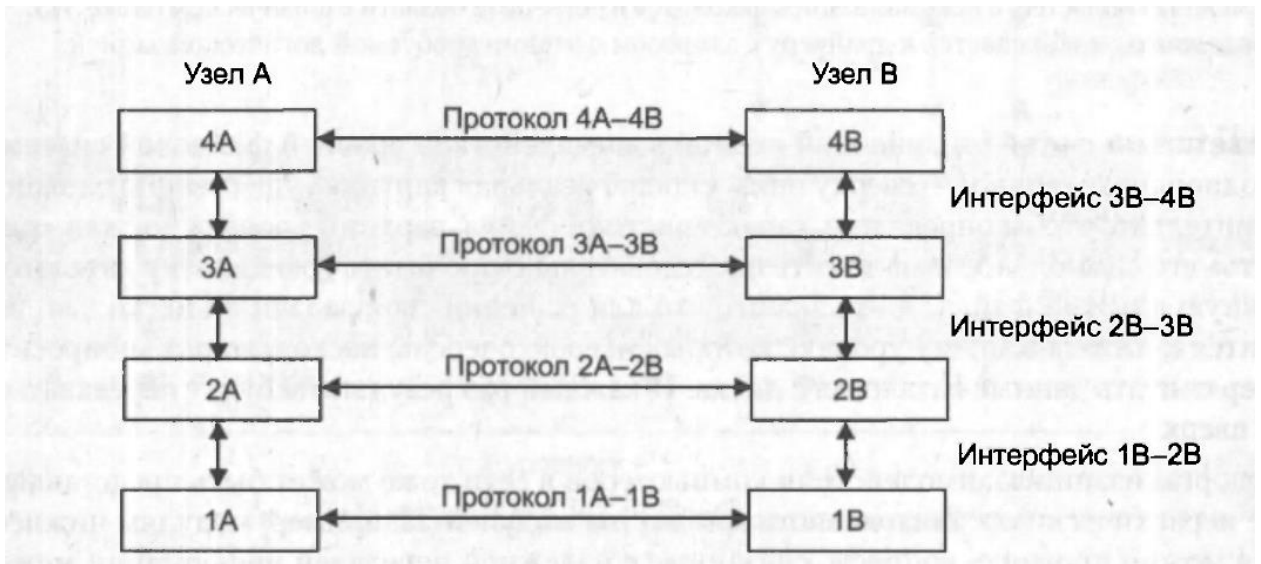


Рисунок 17 – Взаимодействие двух узлов

## Задание

1. Используя программное обеспечение «АСМ» собрать сеть (рисунок 18) из трех подсетей и трех маршрутизаторов (7 хостов). Одна подсеть на концентраторах, вторая на коммутаторах, в третьей один хост.

Адресация в подсетях и между маршрутизаторами различная. Сначала задать все адреса, только потом трафик.

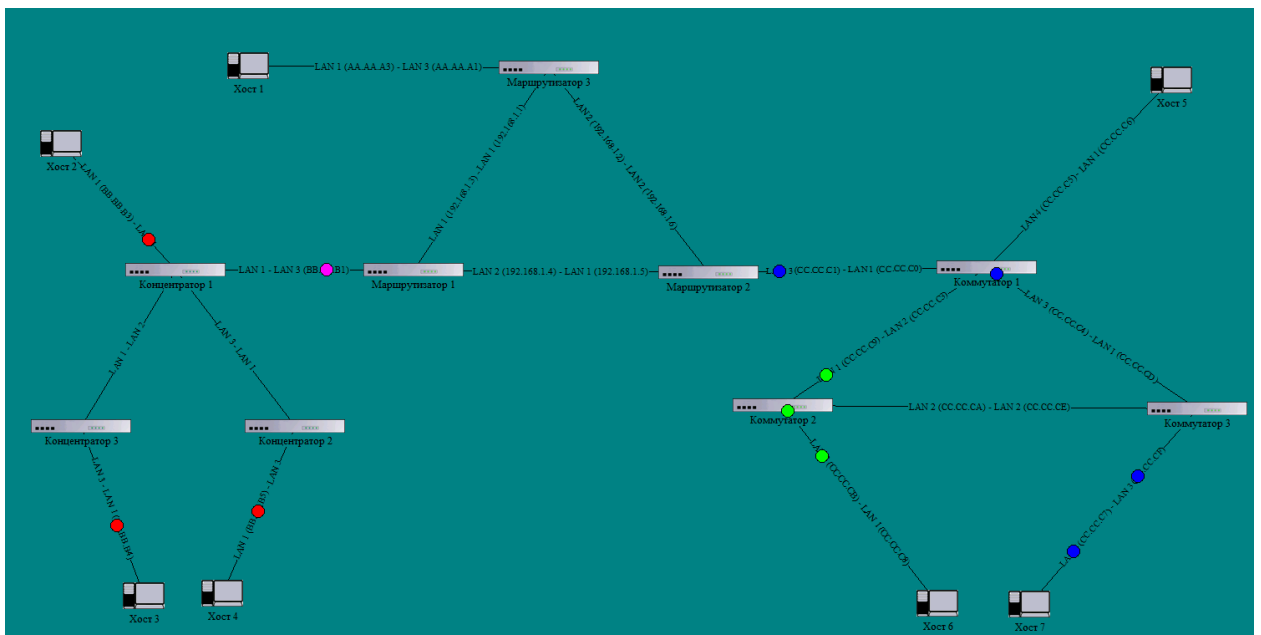


Рисунок 18

Задать трафик между абонентскими устройствами: 1-2, 1-3, 4-7, 5-6.

Вероятности появления пакетов соответственно:  $5+(N \text{ div } 2)$ ;  $10+(N \text{ div } 2)$ ;  $10+(N \text{ div } 3)$ ;  $15+(N \text{ div } 3)$ , %.

Проверить работу сети (без таблицы маршрутизации).

Задать таблицу маршрутизации.



Собрать статистику на узлах коммутации за 100 тактов.

2. Для каждого из маршрутов составить схему (рисунок 19) соответствия функций различных устройств сети уровням модели OSI:

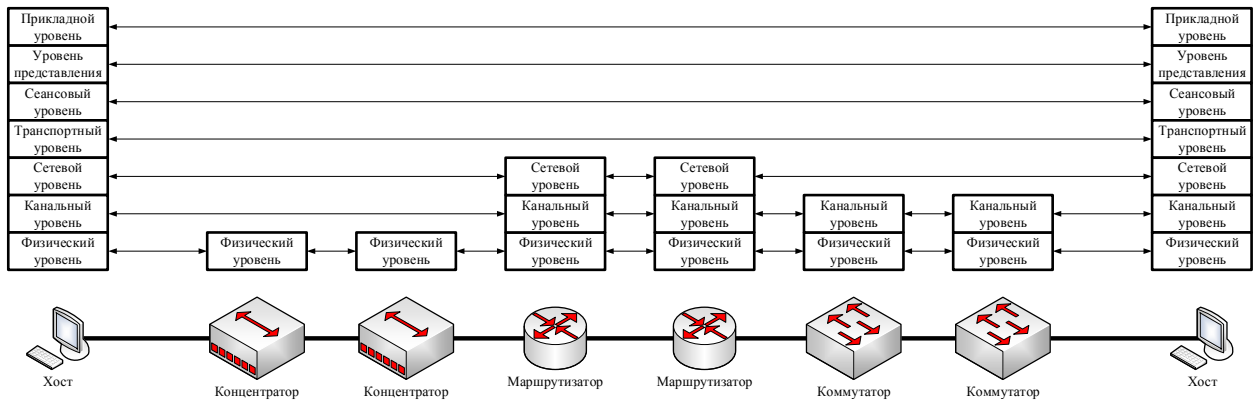


Рисунок 19