

ЭТАЛОННАЯ МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ

1. ПОНЯТИЕ «ОТКРЫТАЯ СИСТЕМА»

2. ПРИКЛАДНОЙ УРОВЕНЬ

3. УРОВЕНЬ ПРЕДСТАВЛЕНИЯ

4. СЕАНСОВЫЙ УРОВЕНЬ

5. ТРАНСПОРТНЫЙ УРОВЕНЬ

6. СЕТЕВОЙ УРОВЕНЬ

7. УРОВЕНЬ ЗВЕНА ДАННЫХ

8. ФИЗИЧЕСКИЙ УРОВЕНЬ

9. ПРИМЕР СООТВЕТСТВИЯ СТЕКА

HTTP/TCP/IP/Ethernet УРОВНЯМ МОДЕЛИ OSI

1. ПОНЯТИЕ «ОТКРЫТАЯ СИСТЕМА»

Обмен информацией в любых службах электросвязи должен осуществляться по определённым, заранее оговорённым правилам. Эти правила (стандарты) разрабатываются рядом международных организаций электросвязи.

В начале 1980-х годов организации ISO, ITU-T и некоторые другие – разработали модель, которая сыграла значительную роль в развитии сетей. Эта модель называется моделью взаимодействия открытых систем (ВОС) (Open System Interconnection, OSI) или моделью OSI. Модель OSI определяет различные уровни взаимодействия систем, даёт им стандартные имена и указывает, какие функции должен выполнять каждый уровень. Модель OSI была разработана на основании большого опыта, полученного при создании компьютерных сетей, в основном глобальных, в 1970-е годы.



Модель OSI, как это следует из её названия, описывает взаимосвязи **открытых систем**. В широком смысле открытой системой может быть названа любая система (компьютер, вычислительная сеть, операционная система, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями.

Под термином «спецификация» (в вычислительной технике) понимают формализованное описание аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, ограничений и особых характеристик. При этом, не всякая спецификация является стандартом. В свою очередь, под открытыми спецификациями понимаются опубликованные, общедоступные спецификации, соответствующие стандартам и принятые в результате достижения согласия после всестороннего обсуждения всеми заинтересованными сторонами.



Использование при разработке систем открытых спецификаций позволяет третьим сторонам разрабатывать для этих систем различные аппаратные или программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей.

Для реальных систем полная открытость является недостижимым идеалом. Как правило, даже в системах, называемых открытыми, этому определению соответствуют лишь некоторые части, поддерживающие внешние интерфейсы.

Модель OSI касается только одного аспекта открытости, а именно открытости средств взаимодействия устройств, связанных в вычислительную сеть. Здесь под открытой системой понимается сетевое устройство, готовое взаимодействовать с другими сетевыми устройствами с использованием стандартных правил, определяющих формат, содержание и значение принимаемых и отправляемых сообщений.



Если две сети построены с соблюдением открытости, то это даёт следующие преимущества:

- возможность построения сети из аппаратных средств различных производителей, придерживающихся одного и того же стандарта;
- возможность безболезненной замены отдельных компонентов сети другими, более совершенными, что позволяет сети развиваться с минимальными затратами;
- возможность лёгкого сопряжения одной сети с другой;
- простота освоения и обслуживания сети.

Ярким примером открытой системы является международная сеть Internet. Эта сеть развивалась в полном соответствии с требованиями, предъявляемыми к открытым системам. В разработке её стандартов принимали участие тысячи специалистов-пользователей этой сети из различных университетов, научных организаций и фирм-производителей



вычислительной аппаратуры и программного обеспечения, работающих в разных странах. Само название стандартов, определяющих работу сети Internet – Request For Comments (RFC), что можно перевести как «запрос на комментарии», - показывает гласный и открытый характер принимаемых стандартов. В результате сеть Internet сумела объединить в себе самое разнообразное оборудование и программное обеспечение огромного числа сетей, разбросанных по всему миру.



1.1 СТРУКТУРА МОДЕЛИ OSI

В модели OSI (рисунок 1.1 – следующий слайд) средства взаимодействия делятся на семь уровней: прикладной, представления, сеансовый, транспортный, сетевой, звена данных, физический. Каждый уровень имеет дело с одним определённым аспектом взаимодействия сетевых устройств.

Взаимодействие уровней происходит с помощью протоколов и интерфейсов.

Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются **протоколом**.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с чётко определёнными правилами и с помощью стандартизованных форматов сообщений.



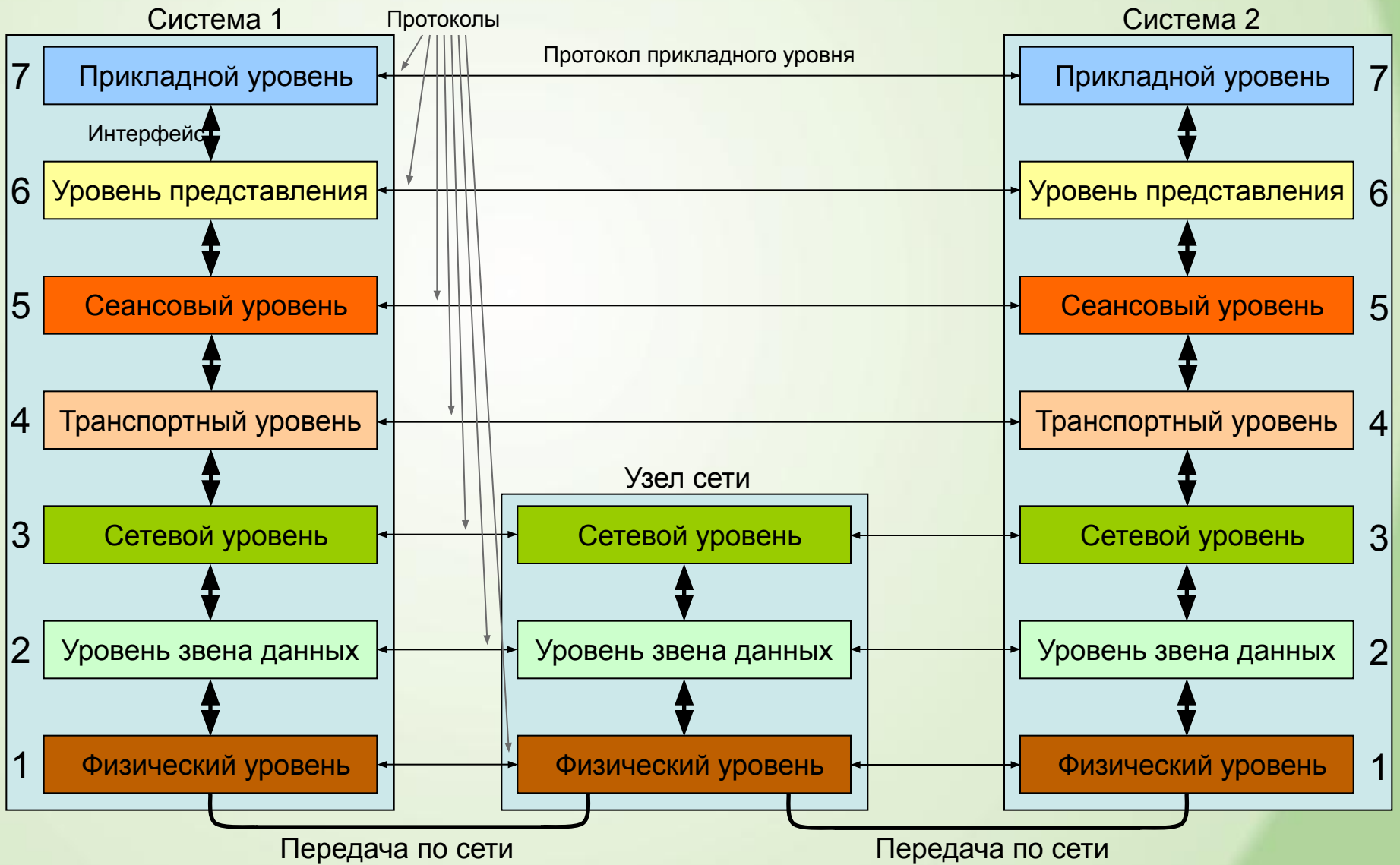


Рисунок 1.1 – Структура модели OSI



Эти правила принято называть **интерфейсом**. Интерфейс определяет набор сервисов, предоставляемый данным уровнем соседнему уровню. В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют **правила** взаимодействия модулей одного уровня в разных узлах, а интерфейсы – модулей соседних уровней в одном узле.

В набор правил включается следующее:

1. Функции выполняемые данными протоколами или интерфейсами.

Существуют стеки протоколов различных технологий, если это протоколы одного уровня их можно сопоставлять.

Если известно к какому уровню относится данная технология, следовательно известно и какие функции она выполняет.

Например, протокол IP – выполняет функции сетевого



уровня, а Ethernet – функции 1 и 2 уровней. Следовательно эти протоколы дополняют друг друга.

Например, протокол X.25 – выполняет функции 1, 2, 3 уровней, протокол frame relay – выполняет функции 1 и 2 уровней. Значит эти протоколы можно сравнивать относительно уровней 1 и 2 модели OSI.

При сравнении протоколов необходимо учитывать для передачи какой информации они предназначены.

Например, протокол IP – может переносить и пользовательскую информацию и управляющую.

Протокол QoS7 предназначен для переноса сигнальной информации.

Протокол RIP (Routing Information Protocol) – протокол маршрутной информации, работающий на сетевом уровне, предназначен для обмена маршрутной информацией между маршрутизаторами в IP-сетях.

Протокол SNMP (Simple Network Management Protocol) –



простой протокол управления сетью, работающий на прикладном уровне, предназначен, как это понятно из названия, для управления сетью. Протокол CMIP (Common Management Information Protocol) – протокол общей управляющей информации, относящийся к прикладному уровню, также предназначен для управления сетью. Протоколы SNMP и CMIP можно сравнивать.

Например, протокол SNMP нельзя сравнивать с ОКС7, но можно сравнивать с протоколом OMAP (Operations, Maintenance and Administration Part) – подсистема эксплуатации, технического обслуживания и административного управления.

OMAP является одним из протоколов, составляющих стек протоколов ОКС7, и позволяет контролировать и управлять оборудованием, связанным с сетью сигнализации ОКС7.



2. Сообщения и их параметры для выполнения функций протоколов или интерфейсов.

Каждый уровень модели OSI имеет свои, характерные именно для него сообщения.

Чем ниже уровень протокола, тем больше участия аппаратных средств (физический уровень – концентраторы, повторители, усилители и т. д.). Чем выше уровень, тем больше программного обеспечения, соответственно на каждом уровне реализуются свои сообщения.

3. Форматы и коды сообщений.

Различные протоколы имеют разные форматы и коды сообщений. Причём протоколы, соответствующие одинаковым уровням модели OSI, часто, имеют разные форматы и коды, но несмотря на это, такие протоколы выполняют одни и те же, или похожие функции. Это вызвано тем, что одни и те же функции, можно реализовать разными способами, в данном случае разными форматами.



Например, протокол общей управляющей информации CMIP может выполнять более широкий и более сложный набор функций, по сравнению с простым протоколом управления сетью SNMP, следовательно форматы этих протоколов будут различаться, так как для выполнения более сложных функций потребуется более сложная структура формата сообщений.

4. Процедура обмена сообщениями.

При обмене сообщениями реализуется определённый алгоритм.

На примере, приведённом на следующем слайде показан несложный алгоритм обмена сообщениями между подсистемами ISUP (подсистемами пользователей ISDN) общеканальной сигнализации номер 7 (ОКС7). Для понимания этого примера не требуется знания подсистемы ISUP. Смысл примера заключается в иллюстрации принципа обмена сообщениями.



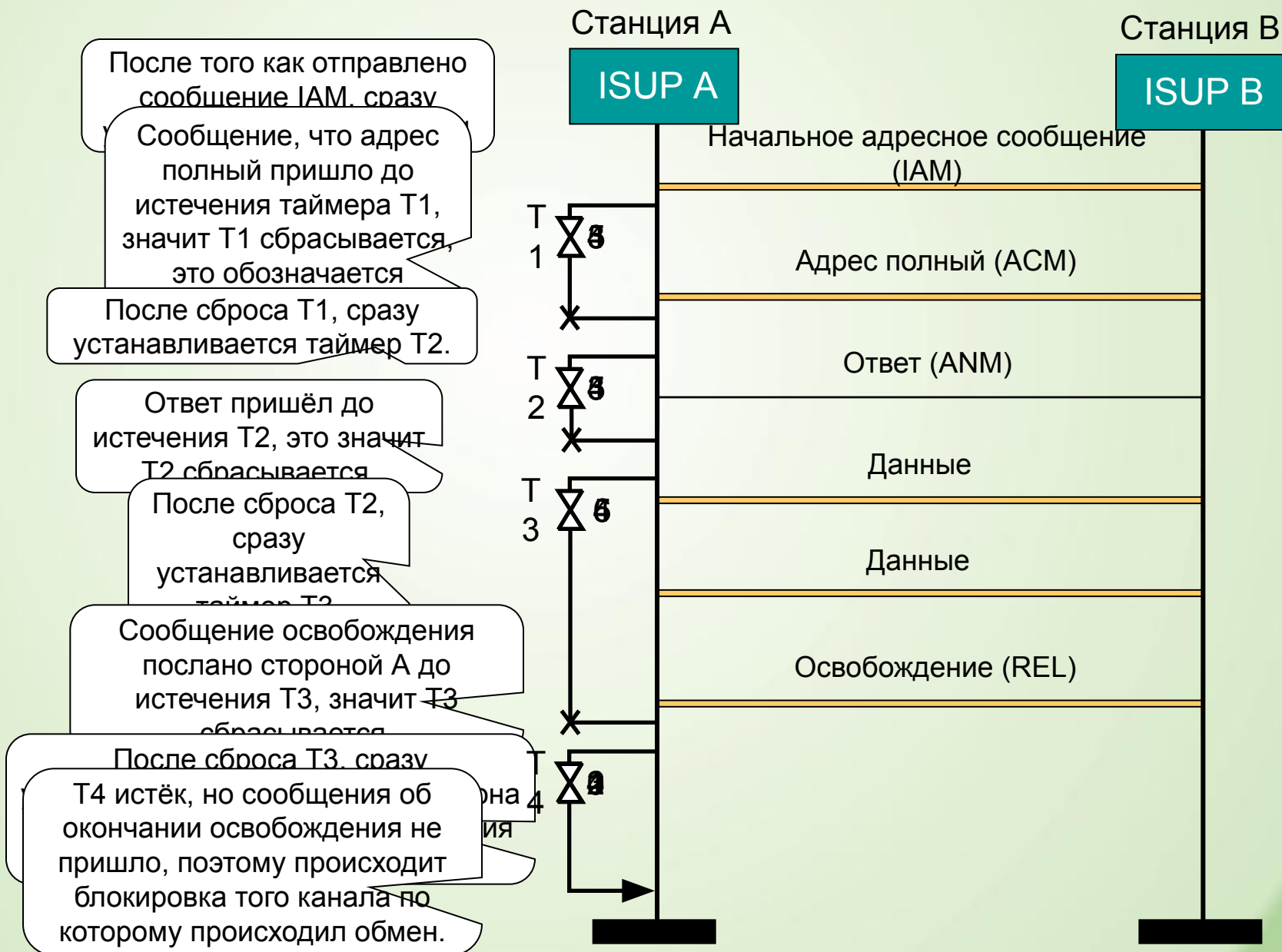


Рисунок 1.2 – Иллюстрация процесса обмена сообщениями



Средства каждого уровня должны обрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

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

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней – как правило, чисто программными средствами.

Программный модуль, реализующий некоторый протокол, часто для краткости также называют «протоколом». При этом соотношение между протоколом – формально определённой процедурой и протоколом – программным модулем, реализующим эту процедуру, аналогично соотношению между алгоритмом решения некоторой задачи и программой,



решающей эту задачу.

Понятно, что один и тот же алгоритм может быть запрограммирован с разной степенью эффективности. Точно так же и протокол может иметь несколько программных реализаций. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программных решений. Более того, на эффективность взаимодействия устройств в сети влияет качество всей совокупности протоколов, составляющих стек, в частности, насколько рационально распределены функции между протоколами разных уровней и насколько хорошо определены интерфейсы между ними.

Протоколы реализуются не только терминалами пользователей, но и другими сетевыми объектами, в компьютерных сетях это такие объекты как: концентраторы, мосты, коммутаторы, маршрутизаторы и т. д.; в телефонных сетях это такие объекты как: АТС, концентраторы, подстанции и



т. д. В общем случае связь терминалов пользователей в сети осуществляется не напрямую, а через различные коммуникационные устройства. В зависимости от типа устройства в нём должны быть встроенные средства, реализующие тот или иной набор протоколов.

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Свои собственные протоколы взаимодействия приложения реализуют, обращаясь к системным средствам. Поэтому необходимо различать уровень взаимодействия приложений и прикладной уровень.

Следует также иметь в виду, что приложение может взять на себя функции некоторых верхних уровней модели OSI. Например, некоторые СУБД имеют встроенные средства удалённого доступа к файлам. В этом случае приложение,



выполняя доступ к удалённым ресурсам, не использует системную файловую службу; оно обходит верхние уровни модели OSI и обращается напрямую к системным средствам, ответственным за транспортировку сообщений по сети, которые располагаются на нижних уровнях модели OSI.



1.2 ПРИНЦИП РАБОТЫ МОДЕЛЬ OSI

Пусть приложение обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Обычное сообщение состоит из заголовка и поля данных. Заголовок содержит служебную информацию, которую необходимо передать через сеть прикладному уровню машины-адресата, чтобы сообщить ему, какую работу надо выполнить. В данном случае заголовок, очевидно, должен содержать информацию о месте нахождения файла и о типе операции, которую необходимо над ним выполнить. Поле данных сообщения может быть пустым или содержать какие-либо данные, например те, которые необходимо записать в удалённый файл. Но для того чтобы доставить эту информацию по назначению, предстоит решить ещё много задач, ответственность за которые несут нижележащие уровни.



После формирования сообщения прикладной уровень направляет его вниз по стеку уровню представления. Протокол уровня представления на основании информации, полученной из заголовка прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию – заголовок уровня представления, в котором содержатся указания для протокола уровня представления машины-адресата. Полученное в результате сообщение передаётся вниз сеансовому уровню, который в свою очередь добавляет свой заголовок, и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце, в виде так называемого «концевика».) Наконец, сообщение достигает нижнего, физического уровня, который собственно и передаёт его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рисунок 1.3 – следующий слайд).



Когда сообщение по сети поступает на машину-адресат, оно принимается её физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня,

Сообщение четвёртого уровня

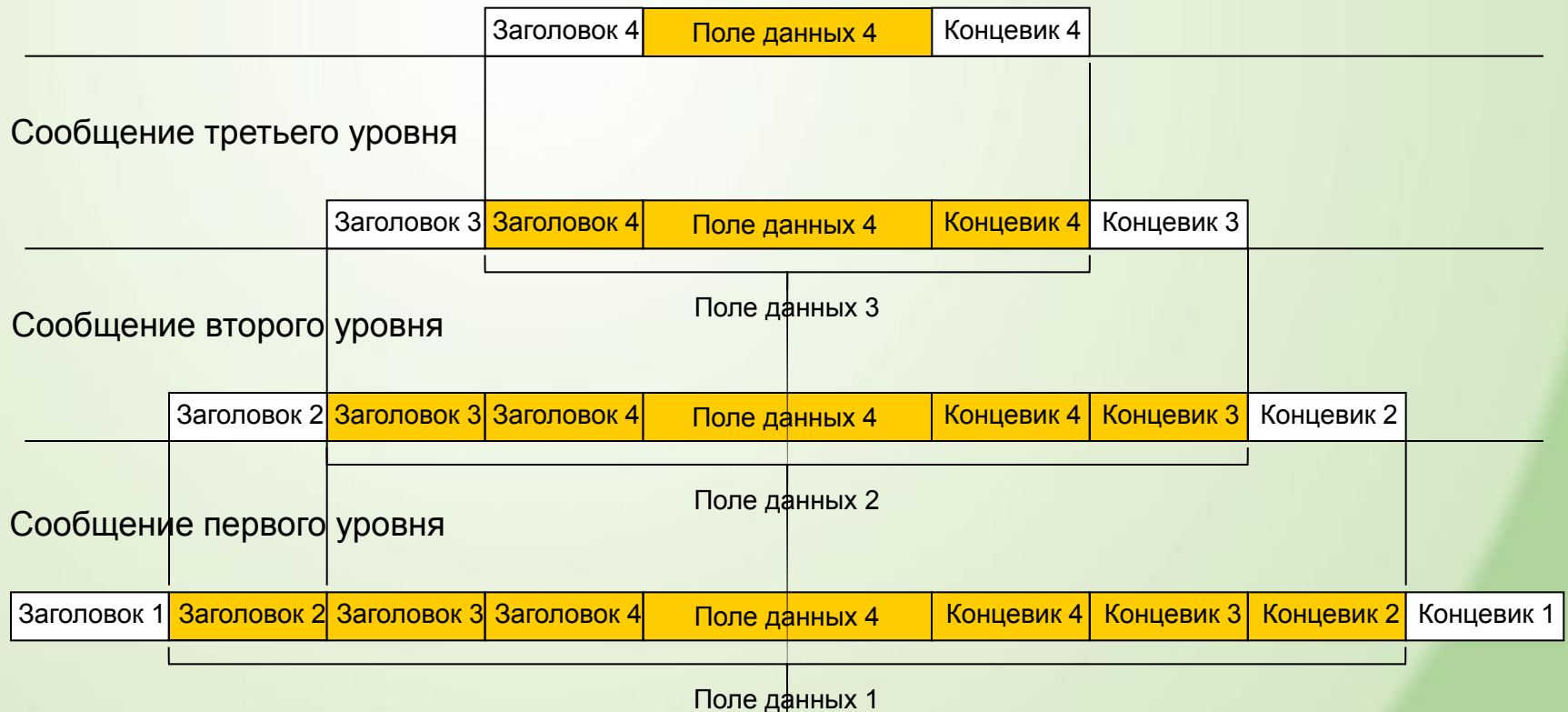


Рисунок 1.3 – Вложенность сообщений различных уровней



выполняя соответствующие данному уровню функции, а затем удаляет заголовок и передаёт сообщение вышележащему уровню.

Наряду с термином сообщение (message) существуют и другие термины, применяемые сетевыми специалистами для обозначения единиц данных в процедурах обмена. В стандартах ISO для обозначения единиц данных, с которыми имеют дело протоколы разных уровней, используется общее название **протокольный блок данных** (Protocol Data Unit, PDU). Для обозначения блоков данных определённых уровней часто используются специальные названия: кадр (frame), пакет (packet), дейтаграмма (datagram), сегмент (segment).

В модели OSI различают два основных типа протоколов. В протоколах с установлением соединения (connection-oriented) перед обменом данными отправитель и получатель должны сначала установить соединение и, возможно, выбрать некоторые параметры протокола, которые они будут



использовать при обмене данными. После завершения диалога они должны разорвать это соединение. Телефонная связь – это пример взаимодействия, основанного на установлении соединения.

Вторая группа протоколов – протоколы без предварительного установления соединения (connectionless). Такие протоколы называются также дейтаграммными протоколами. Например, в компьютерных сетях используются протоколы обоих типов.



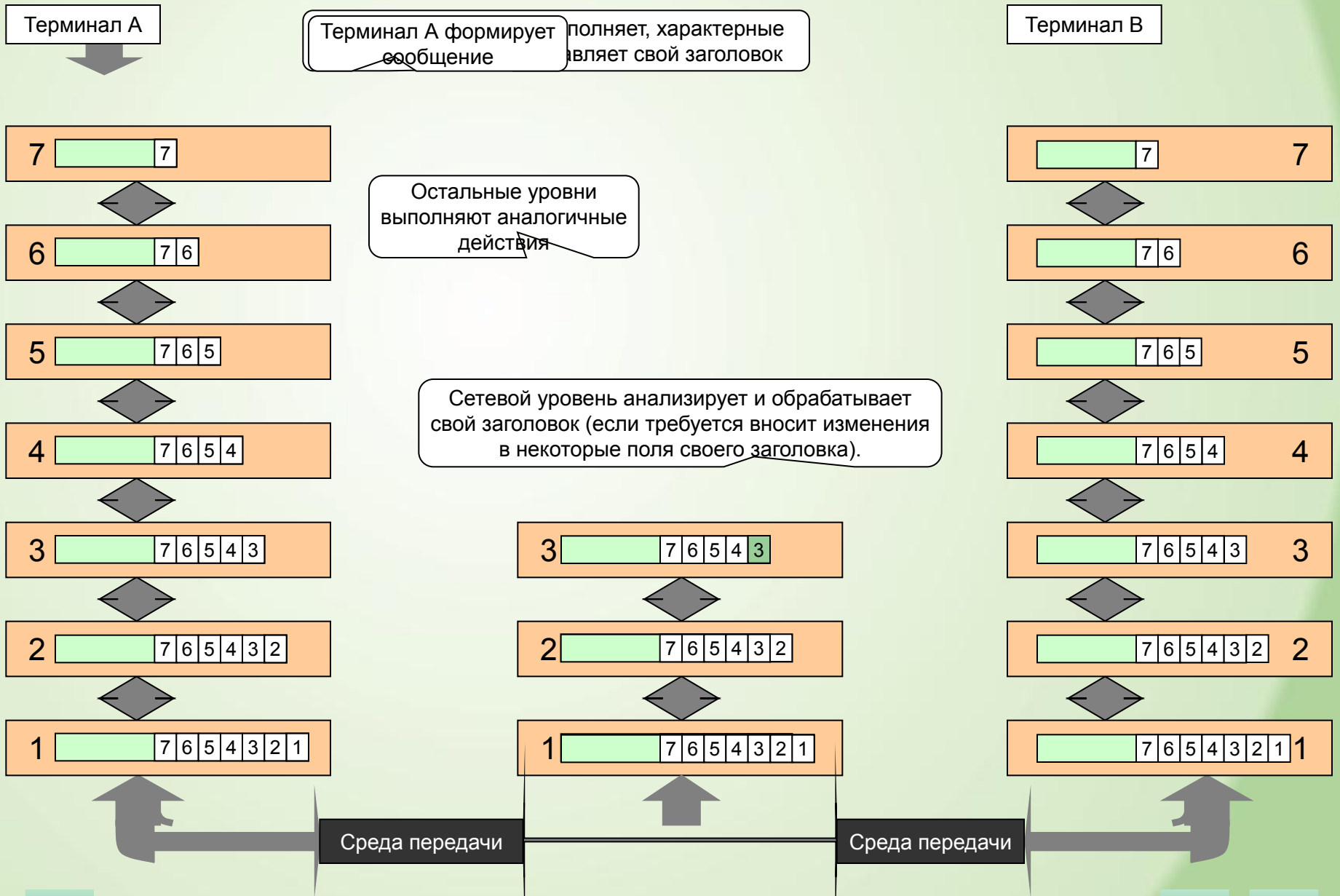


Рисунок 1.4 - Иллюстрация принципа работы модели OSI.



2. ПРИКЛАДНОЙ УРОВЕНЬ

Цель прикладного уровня, как самого высокого в модели OSI, состоит в том, чтобы обеспечить средства обслуживания для **прикладного процесса**, который обращается к **среде OSI**.

Прикладной процесс – это элемент в **реальной открытой системе**, который выполняет обработку информации для приложения.

Среда OSI – абстрактное представление набора понятий, элементов, функций, услуг, протоколов и т. д., определённых моделью OSI и стандартами, которые определяют взаимодействие открытых систем.

Реальная открытая система – это реальная система, которая выполняет требования стандартов модели OSI при связи с другими реальными системам.



Прикладные процессы обмениваются информацией посредством прикладных **объектов**, которые используют услуги прикладных протоколов и услуги уровня представления.

Объект уровня с номером N - это активный элемент процесса уровня N, выполняющий набор возможностей, определённых для N уровня модели OSI.

Единица данных, которой оперирует прикладной уровень, обычно называется **сообщением (message)**.

Средства обслуживания прикладного уровня в режиме с установлением соединения:

1. Идентификация заданных партнеров по обмену.

В любой сети связи необходима идентификация пользователей или пользовательских терминалов, по крайней мере, для того, чтобы устанавливать соединения между пользователями, однозначно определёнными с помощью идентификаторов. Кроме того, при предоставлении разных



услуг зачастую необходимо идентифицировать сами услуги.

Примером службы, которая идентифицирует пользователя, может послужить система доменных имён DNS (Domain Name System). DNS – это распределённая база данных, поддерживающая иерархическую систему имён для идентификации узлов в сети Internet. Служба DNS предназначена для автоматического поиска IP-адреса по известному символьному имени узла.

Протокол DNS является служебным протоколом прикладного уровня. Этот протокол несимметричен – в нём определены DNS-серверы и DNS-клиенты. DNS-серверы хранят часть распределённой базы данных о соответствии символьных имён и IP-адресов. Клиенты сервера DNS сообщают ему известное символьное имя и просят вернуть соответствующий ему IP-адрес. Например, DNS-клиент сообщает DNS-серверу символьное имя: example.ru, а DNS-сервер отправляет DNS-клиенту IP-адрес: 101.45.80.51, соответствующий символьному



имени.

Ещё одним примером идентификации может служить идентификация абонента и его подвижной станции в цифровой сотовой системе подвижной радиосвязи стандарта GSM.

В этом стандарте каждый подвижный абонент на время пользования системой связи получает стандартный модуль подлинности абонента (SIM-карту), который содержит международный идентификационный номер (IMSI), с помощью которого сеть идентифицирует абонента.

Для идентификации терминала абонента используется международный идентификационный номер оборудования подвижной станции (IMEI). Регистр идентификации оборудования (EIR), содержит централизованную базу данных номеров IMEI. Например, если известно, что подвижная станция украдена, то её номер IMEI заносится в чёрный список, при попытке воспользоваться услугами связи такая подвижная станция будет идентифицирована как украденная и ей будет



отказано в обслуживании.

2. Аутентификация заданных партнеров по обмену.

Для исключения несанкционированного использования ресурсов системы связи вводятся механизмы аутентификации – удостоверения подлинности абонента.

Проблема аутентификации заключается в том, чтобы помешать кому-либо, кроме санкционированного пользователя (отправителя), изменять канал, то есть получатель должен быть уверен, что в настоящий момент он принимает сообщение от санкционированного пользователя.

Примером может послужить механизм аутентификации в беспроводной системе связи стандарта DECT. Механизм заключается в следующем – во время сеанса аутентификации базовая станция проверяет аутентификационный ключ, не передавая его по эфиру. Принцип нераскрытия идентификационной информации заключается в следующем: базовая станция посылает трубке случайное число, которое



называется «запрос». Трубка рассчитывает «ответ», комбинируя аутентификационный ключ с полученным случайным числом, и передаёт «ответ» базовой станции. Базовая станция также просчитывает «ответ» и сравнивает его с полученным «ответом». В результате сравнения происходит либо продолжение установления связи либо разъединение.

Если кто-то подслушивает по эфирному интерфейсу, для того чтобы украсть аутентификационный ключ, ему необходимо знать алгоритм для выявления ключа из «запроса» и «ответа». Этот «обратный» алгоритм требует огромной компьютерной мощности. Поэтому стоимость извлечения ключа подслушиванием процедуры аутентификации невероятно высока.



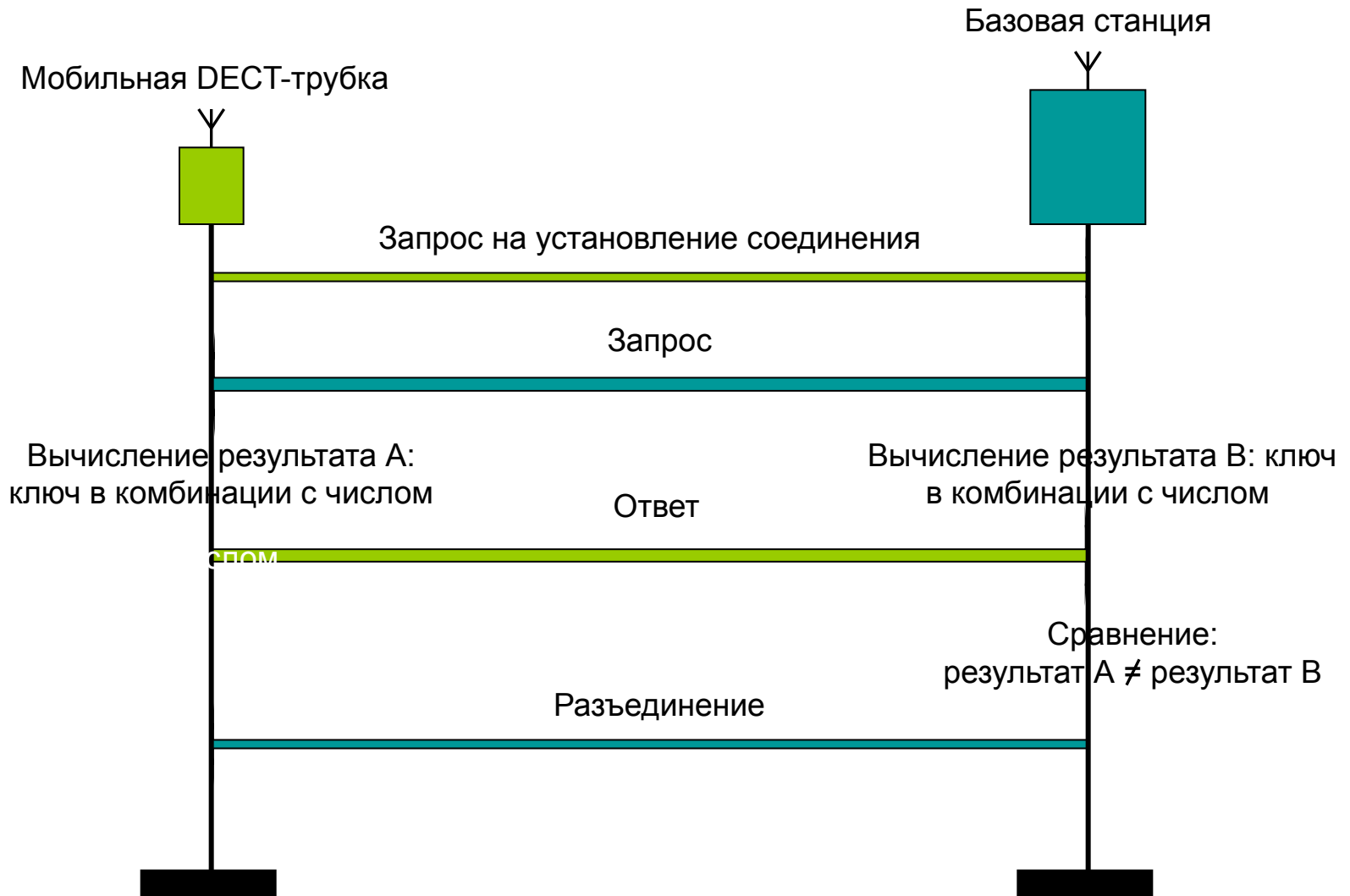


Рисунок 2.1 - Иллюстрация механизма аутентификации



3. Синхронизация взаимодействующих прикладных процессов.

Примером протокола, обеспечивающего синхронизацию, может послужить протокол реального времени RTP (Real-time Transport Protocol), относящийся к прикладному уровню.

Протокол RTP предназначен для доставки чувствительной к задержкам информации.

Отправитель помечает каждый RTP-пакет временной меткой, а получатель извлекает её и вычисляет суммарную задержку. Разница в задержке пакетов позволяет определить джиттер (величину колебания задержки) и смягчить его влияние – все пакеты будут выдаваться приложению с одинаковой задержкой.

Протокол RTP совместно с протоколом резервирования ресурсов (Resource Reservation Protocol, RSVP) используется для передачи синхронизированной мультимедиаинформации с определённым уровнем качества обслуживания.



4. Определение качества услуг.

Выполнение этих функций можно рассмотреть на примере двух совместно используемых протоколов: протокола RTP, который уже упоминался, и протокола управления транспортировкой в реальном времени RTCP (Real-Time Transport Control Protocol).

RTP поддерживается RTCP, который формирует дополнительные отчёты, содержащие информацию о сеансах связи RTP.

Протокол RTCP обеспечивает обратную связь с отправителями, а получателям потоков он предоставляет некоторые возможности повышения качества услуг, информацию о пакетах (потери, задержки, джиттер) и о пользователе (приложении, потоке). Для управления потоком существуют отчёты двух типов – генерируемые отправителями и генерируемые получателями. Например, информация о доле потерянных пакетов и абсолютном количестве потерь



позволяет отправителю при получении отчёта обнаруживать, что перегрузка канала может заставить получателей не принимать потоки пакетов, которые они ожидали. В этом случае отправитель имеет возможность понизить скорость кодирования, чтобы уменьшить перегрузку и улучшить приём.

5. Соглашение об ответственности за восстановление при ошибках.

Например, в телефонной сети общего пользования (ТфОП) при низком качестве обслуживания абонент решает перезвонить ему или нет.

В сети Internet ответственность ложится на программы, например, такие как различные менеджеры загрузки файлов из Internet. Такие программы при обрыве связи не теряют ту часть файла, которая уже загружена, а при возобновлении связи продолжают загружать файл.



6. Соглашение по аспектам защиты.

Компонентами того соглашения могут быть: идентификация, управление доступом, целостность данных.

Управление доступом заключается в определении того, что можно ли пользователю предоставить доступ. Причины отказа в доступе могут быть следующие: пользователь не прошёл идентификацию или аутентификацию, у пользователя нет средств для оплаты услуги, которой он решил воспользоваться.

Пример отказа в доступе можно привести из сотовой сети связи стандарта GSM. Если номер IMEI подвижной станции абонента внесён в чёрный список, то абоненту, владеющему такой станцией, будет отказано в обслуживании.

Целостность обеспечивает защиту от **подслушивания** и **манипулирования** данными, поддерживая конфиденциальность и неизменность передаваемой информации.

Подслушивание – во время передачи данных о пользователе (пользовательских идентификаторов и паролей) или частных



конфиденциальных данных по незащищённым каналам эти данные можно подслушать и впоследствии злоупотреблять ими. Методы шифровки данных снижают вероятность этой угрозы.

Манипулирование данными – заключается в том, что данные, которые передаются по каналам связи, в принципе могут быть изменены.

7. Выбор режима диалога.

На прикладном уровне может быть выбран режим с установлением соединения, а может быть выбран режим без установления соединения.

8. Определение методики распределения затрат.

Например – определение того, кто будет оплачивать услугу – абонент вызывающий или вызываемый.

9. Идентификация синтаксисов прикладного уровня.

Прикладной уровень идентифицирует набор синтаксисов



представления информации и затем сообщает уровню представления какой, или какие, синтаксисы он будет использовать.

Набор средств обслуживания прикладного уровня в режиме без установления соединения такой же как и с установлением соединения, исключая функцию синхронизации.

Функции прикладного уровня.

Прикладной уровень включает в себя все те функции, которые нужно выполнить для того, чтобы организовать связь в любом режиме между открытыми системами и которые не выполнены нижними уровнями. Прикладной уровень включает как функции выполняемые программами, так и функции выполняемые людьми.

Фактически, в функции прикладного уровня входит обеспечение средств обслуживания, которые были перечислены выше.



3. УРОВЕНЬ ПРЕДСТАВЛЕНИЯ

Главной целью уровня представления (Presentation Layer) является обеспечение представления данных, передаваемых между прикладными объектами разных открытых систем. Это освобождает взаимодействующие прикладные объекты от решения проблем связанных с представлением информации.

Услуги, предоставляемые прикладному уровню:

1. Идентификация набора синтаксисов переноса.

Под синтаксисом переноса понимается синтаксис, который используется объектами представления взаимодействующих открытых систем для связи между собой.

Идентификация набора синтаксисов переноса обеспечивает одно или более средств для преобразования синтаксиса прикладного уровня.



2. Выбор синтаксиса переноса.

Выбор синтаксиса переноса обеспечивает средства для первоначального выбора синтаксиса переноса и впоследствии изменение этого выбора.

3. Доступ к услугам сеанса (сеансового уровня).

Услуги сеанса предоставлены прикладным объектам в форме услуг представления.

Функции уровня представления:

Уровень представления выполняет следующие функции, которые помогают обеспечивать услуги представления:

1. Выбор синтаксиса переноса и пересмотр этого выбора.

Эта функция состоит в том, чтобы объекты представления взаимодействующих систем выбрали взаимоприемлемый синтаксис переноса и при необходимости пересмотрели бы его.



2. Преобразование синтаксиса, выбранного прикладными объектами в синтаксис переноса, включая специальные функции.

Благодаря этой функции решаются проблемы семантики и синтаксиса передаваемой информации.

Дело в том, что на разных машинах используются разные способы кодировки - ASCII, Unicode, EBCDIC и т. д.

ASCII (American Standard Code for Information Interchange) - Американский стандартный код для обмена информацией. Поддерживает до 256 символов.

Unicode - универсальная кодировка, 16-разрядный символьный код, позволяющий поддерживать до 64000 различных символов.

EBCDIC (Extended Binary-Coded Decimal Interchange Code) - расширенный двоично-кодированный десятичный код для обмена информацией (поддерживающий до 256 символов). EBCDIC - код фирмы IBM для представления символов в виде



чисел.

Пользователи как правило используют структуры данных, а не случайный набор байт. Уровень представления работает со структурами данных, преобразуя их во внутреннее для конкретной машины и из внутреннего, машинного представления, - в стандартное представление для передачи по сети – то есть говоря в терминах модели OSI, происходит преобразование синтаксиса, используемого прикладным уровнем в синтаксис переноса, используемый уровнем представления.

При преобразовании информации уровень представления может выполнять специальные функции такие как сжатие данных, шифрование и дешифрование данных.

3. Шифрование и дешифрование данных.

На уровне представления может выполняться шифрование и дешифрование данных, благодаря которому секретность



обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол Secure Socket Layer (SSL), разработанный фирмой Netscape, как протокол обеспечивающий защиту данных между сервисными протоколами (например HTTP, FTP) и транспортным стеком протоколов TCP/IP.

4. Использование услуг сеансового уровня.

Уровень представления использует услуги сеансового уровня как нижележащего, так как в модели OSI нижележащий уровень предоставляет свои услуги вышележащему.

Функции уровня представления часто реализуются в прикладных протоколах, примером могут служить, уже упоминавшиеся протоколы: SNMP, CMIP, RTP, HTTP, FTP, SIP (несмотря на то, что эти протоколы называются прикладными). Протокол RTP, кроме функций 7 и 6 уровней, ещё, выполняет функции 5 уровня.



А если требуется дополнить уже имеющийся набор функций, относящихся к уровню представления, то используется отдельный протокол, например, такой как SSL, который дополняет, например, протокол HTTP.



4. СЕАНСОВЫЙ УРОВЕНЬ

Цель сеансового уровня (Session Layer) заключается в том, чтобы обеспечить средства, необходимые для взаимодействия объектов представления, а также обеспечить синхронизацию и управление обменом данными объектов представления. Чтобы реализовать это, сеансовый уровень обеспечивает следующие функции: установление сеанса связи между объектами представления, поддержка этого сеанса и разъединение сеанса.

Услуги, предоставляемые уровню представления в режиме с установлением соединения:

1. Установление соединения сеансового уровня.

Установление соединения сеансового уровня даёт возможность объектам представления установить сеанс связи между собой, а также договориться о параметрах сеанса связи



во время его установления.

Важной особенностью является то, что запрос на установление сеанса связи инициирует объект представления, а сеансовый объект выполняет этот запрос. То есть самостоятельно инициировать запрос на установление соединения сеансовый объект не может.

Примером протокола, выполняющего часть функций сеансового уровня может послужить прикладной протокол инициирования сеансов связи SIP (Session Initiation Protocol). Несмотря на то, что SIP называется прикладным протоколом он ещё выполняет функции уровня представления и сеансового уровня. В протоколе SIP нельзя чётко определить, где кончаются функции прикладного уровня, а где начинаются функции сеансового уровня потому, что эти функции объединены в этом протоколе.

Для организации сеанса связи протокол SIP использует запрос INVITE, который приглашает пользователя принять



участие в сеансе связи (набор сообщений протокола SIP делится на запросы (команды) клиента серверу и ответы сервера клиенту).

Функцию установления соединения на сеансовом уровне в протоколе SIP можно трактовать следующим образом – прикладной уровень сообщает уровню представления о том, что нужно установить соединение, после этого уровень представления передаёт запрос на установление сеанса связи сеансовому уровню, а сеансовый уровень уже устанавливает соединение. Важно понимать, что инициатором установления сеанса, в данном случае, является прикладной уровень, поэтому протокол SIP и относят к прикладному.

2. Перенос нормальных данных.

Эта услуга позволяет передающему объекту представления посылать **сервисный блок данных** сеанса (session-service-data-unit) принимающему объекту представления. Примером



нормальных данных может служить данные пользователя.

Сервисный блок данных уровня N (service-data-unit) – количество информации, подлинность которой сохраняется при переносе между объектами уровня с номером (N+1) и которое не интерпретируется объектами уровня с номером N.

PDU уровня N является для (N-1) уровня сервисным блоком данных.

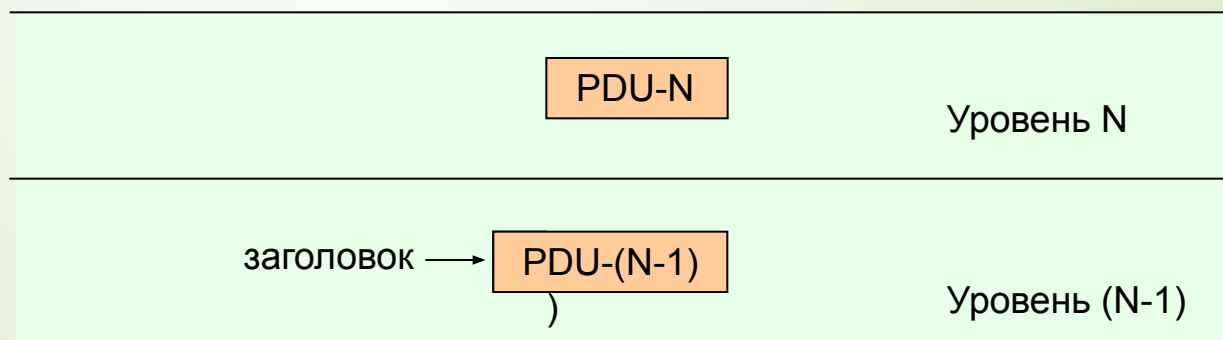


Рисунок 4.1 – Иллюстрация понятий PDU и SDU



3. Перенос срочных данных.

Эта услуга обеспечивает срочную обработку срочного сервисного блока данных сеанса. На такой блок данных накладывается ограничение по его размеру.

Срочный блок данных (expedited-data-unit) – сервисный блок данных, который передаётся и (или) обрабатывается с приоритетом над нормальным сервисным блоком данных. Перенос срочных данных может использоваться для целей сигнализации и прерывания (например, прерывание сеанса связи).

Поток срочных данных не зависит от потока нормальных данных, хотя данные, которые содержатся в этих потоках могут быть логически связаны. В соединении могут организовываться два подканала, один для срочных данных, другой для нормальных.



4. Управление маркером.

Управление маркером – средство сеанса связи, которое позволяет взаимодействующим объектам представления явно управлять правом на использование определённых услуг между собой. То есть тот объект, который владеет маркером имеет право на использование определённой услуги, а другой объект не имеет такого права по отношению к этой же услуге.

Определены четыре маркера:

1. Маркер данных.
2. Маркер разъединения.
3. Маркер минорной синхронизации.
4. Мажорный маркер.

Маркер всегда находится в одном из следующих состояний:

1. Маркер доступен:

- а) Маркером владеет один пользователь услуг сеанса (сеансового уровня), который имеет исключительное



право на использование услуги (при условии, что нет никаких других ограничений).

б) Маркером не владеет другой пользователь услуг сеанса, который не имеет право пользоваться услугой в данный момент, но может приобрести это право позже.

2. Маркер не доступен ни одному пользователю услуг сеанса, тогда, никакой пользователь услуг сеанса не имеет исключительное право на использование услуги. Тогда услуга становится, либо доступной обоим пользователям услуг сеанса (перенос данных, разъединение), либо недоступна обоим пользователям услуг сеанса (например, услуга синхронизации).

5. Синхронизация сеансового соединения.

Эта услуга позволяет объектам представления:

1. Определить и идентифицировать точки синхронизации.
2. Сбросить сеанс связи, который останется в определённом



состоянии и договориться о точке ресинхронизации (в этом случае возможна потеря данных).

Пользователи услуг сеанса могут вставлять точки синхронизации в данные, которые они передают. Есть два метода идентификации точек синхронизации. Точка синхронизации может быть идентифицирована одним серийным номером, если право на вставку точек синхронизации управляется маркером. Другой метод заключается в том, что точки синхронизации могут быть идентифицированы двумя серийными номерами, один для каждого направления потока. Это двойная схема нумерации называется симметричной синхронизацией, она позволяет пользователям услуг сеанса независимо размещать точки синхронизации в своих потоках.

Есть два типа точек синхронизации:

1. Минорные точки синхронизации;
2. Мажорные точки синхронизации.

Мажорные точки синхронизации используются для



структурирования обмена данных в виде ряда диалоговых блоков. Для диалогового блока характерно то, что процесс передачи информации происходящий в его пределах отделён от процесса передачи информации происходящего до него и после него. Мажорная точка синхронизации указывает конец одного диалогового блока и начало следующего. Каждая мажорная точка синхронизации устанавливается явно.

Минорные точки синхронизации используются для структурирования обмена данных в пределах диалогового блока. На рисунке 4.2 (следующий слайд) изображено, как диалоговый блок структурирован минорными точками синхронизации. Каждая минорная точка синхронизации либо устанавливается явно либо не устанавливается явно.

Минорная точка синхронизации, установленная с помощью услуги разделения данных, защищает все данные, посланные до вставки этой точки синхронизации. Это обеспечивает предотвращение повторных запросов синхронизации.



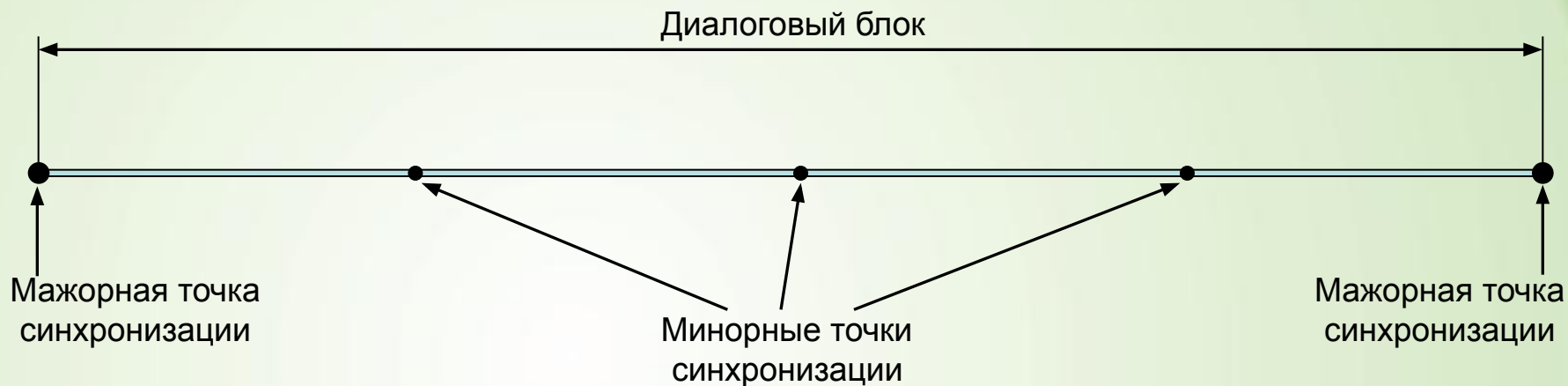


Рисунок 4.2 – Пример структурированного диалогового блока

6. Ресинхронизация.

Ресинхронизация может быть инициализирована любым пользователем услуг сеанса. При ресинхронизации сеанс связи переходит в определённое состояние, и поэтому происходит перераспределение маркеров и установка нового (новых) значения (значений) серийного (серийных) номера (номеров) точки (точек) синхронизации.



Когда используется симметричная синхронизация, пользователь услуг сеанса может повторно запрашивать синхронизацию для одного направления потока данных, или для обоих направлений. Если синхронизация запрашивается для обоих направлений, то обоим серийным номерам будут присвоены новые значения; иначе, новое значение будет присвоено тому серийному номеру, который связан с потоком для которого запрашивалась синхронизация. Ресинхронизация производит удаление всех недоставленных данных потока, для которого производился запрос.

Если симметричная синхронизация не используется, то оба потока всегда повторно синхронизируются. Единственный серийный номер точки синхронизации принимает новое значение. Все недоставленные данные удаляются.

Определены три варианта ресинхронизации:

1. Вариант отказа, который используется для установки серийного номера точки синхронизации в неиспользуемое



1. значение.
2. Вариант перезапуска, который используется для установки серийного номера точки синхронизации в любое используемое значение, которое больше чем значение серийного номера точки синхронизации, которое идентифицировало последнюю мажорную точку синхронизации.
3. Вариант установки, который используется для установки серийного номера точки синхронизации в любое значение выбранное пользователем услуг сеанса.

При использовании симметричной синхронизации, любой из трёх перечисленных вариантов ресинхронизации доступен пользователю услуг сеанса для каждого потока данных, для которого запрошена ресинхронизация.

7. Отчёт об исключениях.

Данная услуга обеспечивает уведомление объектов представления о исключительных ситуациях. Например, если



какая-либо услуга не может быть до конца выполнена из-за ошибок протокола сеансового уровня (или сбоев), объекты представления будут информированы об этом с помощью сообщения исключения.

8. Разрушение соединения сеансового уровня.

Эта услуга позволяет взаимодействующим объектам представления разрушать сеанс связи без потери данных. С помощью этой услуги любой из взаимодействующих объектов представления может прервать сеанс связи, но в этом случае данные могут быть потеряны. Сеанс связи может быть также прерван одним из объектов сеансового уровня.

Например, в протоколе SIP командой BYE оборудование вызываемого или вызывающего пользователей разрушает соединение.

9. Управление диалогом.

Перенос данных между объектами представления может



происходить поочередно в одном направлении, такой режим называется полудуплексным.

Режим переноса при котором объекты представления могут одновременно получать и принимать данные называется дуплексным.

Сеансовый уровень предоставляет услугу выбора режима переноса данных.

Если режим полудуплексный, то сеансовый уровень предоставляет услугу управления активностью объектов представления, то есть предоставляет возможность управлять направлением передачи данных.

Услуги, предоставляемые уровню представления в режиме без установления соединения:

1. Передача данных в режиме без установления соединения, используя режим без установления соединения транспортного уровня.



2. Отчёт об исключениях.

Функции сеансового уровня:

Те, которые должны быть выполнены объектами сеанса для обеспечения услуг сеанса. Кроме функций, заключающихся в выполнении услуг, есть ещё одна функция:

Отображение сеансового соединения на транспортное соединение.

В любой момент времени есть взаимно-однозначное отображение между сеансовым соединением и транспортным соединением. Однако, на протяжении существования транспортного соединения может поддерживаться несколько последовательных сеансовых соединений.



5. ТРАНСПОРТНЫЙ УРОВЕНЬ

На пути от отправителя к получателю пакеты могут быть искажены или потеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надёжным соединением. Цель транспортного уровня (Transport Layer), как транспортной службы состоит в том, чтобы обеспечить прозрачный перенос данных между объектами сеанса и освободить их от знания деталей о созданном для них пути, в котором обеспечивается надёжная и эффективная передача данных.

Транспортный уровень оптимизирует использование доступных сетевых услуг для того, чтобы обеспечить выполнение требований (при минимальных затратах), запрашиваемых каждым объектом сеанса. При этом соблюдаются правила обслуживания параллельно-работающих



объектов сеанса.

Все протоколы транспортного уровня относятся к протоколам, работающим по принципу из конца в конец. Окончания определены как транспортные объекты. Поэтому, транспортный уровень – это оконечная открытая система модели OSI, и транспортные протоколы работают только между оконечными открытыми системами.

Транспортный уровень освобождает вышележащие уровни от проблем связанных с маршрутизацией и ретрансляцией, так как сетевая служба обеспечивает перенос данных от любого транспортного объекта до любого другого.

Услуги, предоставляемые сеансовому уровню в режиме с установлением соединения:

1. Установление транспортного соединения.

Транспортные соединения установленные между объектами сеанса, идентифицируются транспортными адресами.



Объекты сеанса договариваются с транспортной службой о качестве транспортного соединения. Во время создания транспортного соединения из доступного набора классов транспортного обслуживания выбирается требуемый класс (понятия классов транспортного обслуживания и качества транспортного обслуживания рассмотрены в пунктах 5.1 и 5.2 соответственно).

Классы обслуживания характеризуются значениями параметров, которые выбираются пользователями транспортной службы, такими как: пропускная способность, транзитная задержка, задержка установления соединения и гарантируемыми значениями параметров, такими как: коэффициент ошибок, готовность обслуживания.

2. Разрушение транспортного соединения.

При помощи этой услуги любой объект сеанса может разрушить транспортное соединение и сообщить о



разъединении тому объекту сеанса, с которым было установлено соединение.

3. Перенос нормальных данных.

Эта услуга обеспечивает передачу данных в соответствии с согласованным качеством обслуживания. Когда качество обслуживания не может быть обеспечено, и все попытки восстановить это качество оказались неудачными, транспортное соединение прекращается, о чём уведомляются объекты сеанса.

При переносе данных транспортный сервисный блок данных разделяется на некоторое количество сегментов. Затем каждый такой сегмент, один за другим, передаётся от одной **точки доступа транспортного обслуживания** до другой. Причём передача сегментов происходит в той последовательности, которая была изначально (при разбиении блока). Такая услуга относится к управлению потоком данных.



Точка доступа транспортного обслуживания – точка, в которой услуги транспортного уровня, предоставляются транспортным объектом сеансовому объекту.

4. Перенос срочных данных.

Перенос срочных транспортных сервисных блоков данных обеспечивает дополнительное средство информационного обмена в транспортном соединении. Размер каждого из таких блоков ограничен.

Услуги, предоставляемые сеансовому уровню в режиме без установления соединения:

Основной услугой является отображение запроса о передаче транспортного сервисного блока данных на запрос к сетевой службе, которая будет обеспечивать обслуживание в режиме без установления соединения. Другими словами, когда поступает запрос на передачу транспортного сервисного блока



данных, транспортный уровень по этому запросу формирует запрос к сетевому уровню, а сетевой уровень должен обеспечить по пришедшему к нему запросу обслуживание в режиме без установления соединения.

Функции транспортного уровня в режиме с установлением соединения:

1. Отображение транспортного адреса на сетевой адрес.

Когда объект сеанса запрашивает транспортный уровень установить транспортное соединение с другим объектом сеанса, который идентифицируется транспортным адресом, транспортный уровень определяет сетевой адрес, идентифицирующий транспортный объект, который обслуживает соответствующий объект сеанса.

Так как услуги, предоставляются транспортными объектами по принципу из конца в конец, то нет никаких транспортных объектов между взаимодействующими оконечными



транспортными объектами. Поэтому транспортный уровень отображает транспортные адреса на сетевые адреса, которые идентифицируют конечные транспортные объекты (рисунок 5.1).

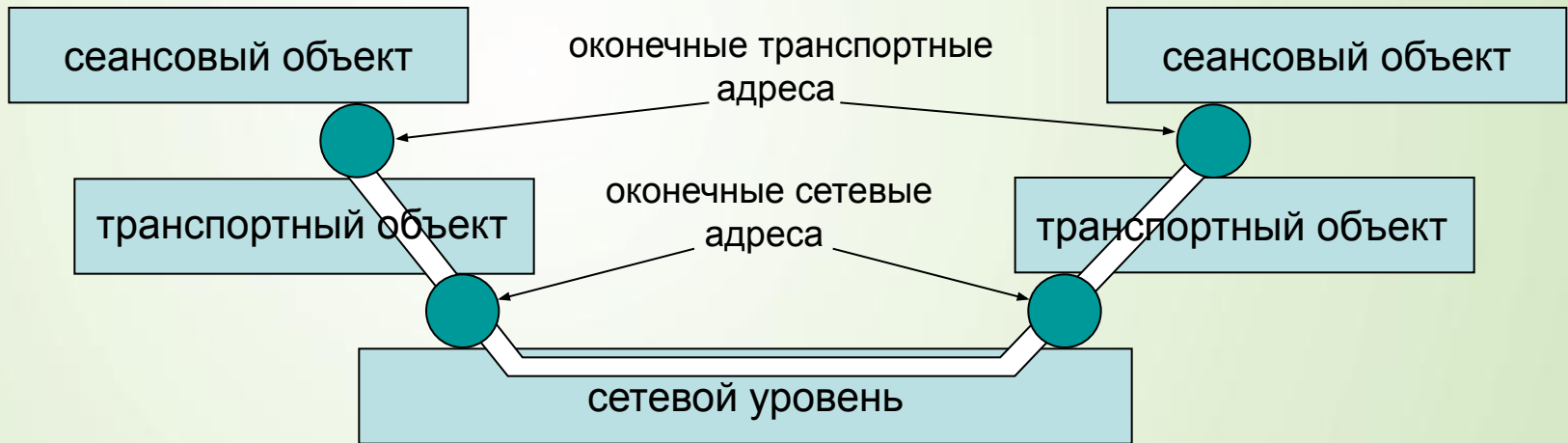


Рисунок 5.1 – Связь транспортных и сетевых адресов

Один транспортный объект может обслуживать больше одного объекта сеанса. Через один транспортный объект



несколько транспортных адресов могут быть связаны с одним сетевым адресом. Для того, чтобы обеспечить эту связь транспортный уровень выполняет функции отображения (рисунок 5.2).

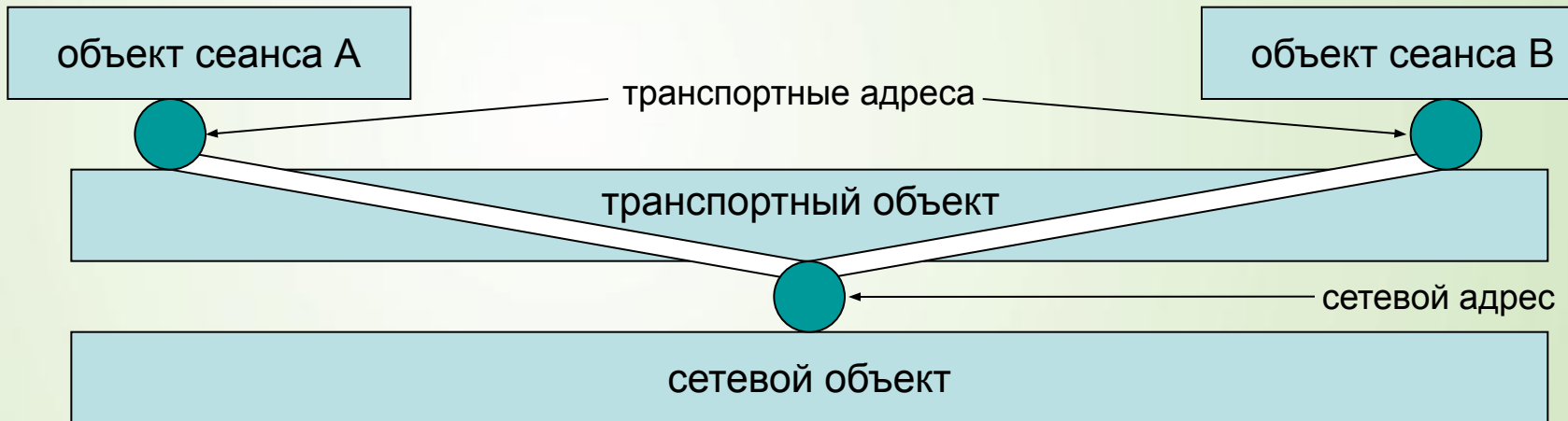


Рисунок 5.2 – Связь одного сетевого адреса с двумя транспортными адресами

2. Мультиплексирование (из конца в конец) транспортных соединений в сетевые соединения.

Чтобы оптимизировать использование сетевых соединений,



отображение транспортных соединений на сетевые соединения не должно основываться на взаимнооднозначном соответствии. Поэтому выполняется и мультиплексирование и расщепление (splitting). Под мультиплексированием понимается отображение нескольких транспортных соединений на одно сетевое или другими словами – одно сетевое соединение обслуживает несколько транспортных.

3. Установление и разрушение транспортных соединений.

3.1 Фаза установления соединения.

В течение этой фазы, транспортный уровень устанавливает транспортное соединение между двумя объектами сеанса. Функции, выполняемые транспортным уровнем на протяжении этой фазы соответствуют запрошенному классу обслуживания с услугами, обеспеченными сетевым уровнем. Следующие функции могут быть выполнены транспортным уровнем в



течение фазы установления соединения:

3.1.1 Обеспечение сетевого соединения, которое соответствует требованиям объекта сеанса, принимая во внимание затраты на обслуживание и качество обслуживания.

3.1.2 Принятие решения о необходимости мультиплексирования или расщепления с целью оптимизирования использования сетевых соединений.

3.1.3 Установка оптимального размера транспортного протокольного блока данных.

3.1.4 Выбор функций, которые будут выполняться после перехода в фазу переноса данных.

3.1.5 Отображение транспортных адресов на сетевые.

3.1.6 Идентификация соединения.

Заключается в идентификации различных транспортных



соединений между парой точек доступа транспортного обслуживания.

Например, в протоколе управления передачей TCP (Transmission Control Protocol), который относится к транспортному уровню, соединение идентифицируется парой полных адресов обоих взаимодействующих процессов – **сокетов**.

Каждый компьютер (или какой-либо другой терминал пользователя) может выполнять несколько процессов, более того, прикладной процесс может тоже иметь несколько точек входа, выступающих в качестве адреса назначения для пакетов данных.

Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных процессов. В терминологии TCP такие системные очереди называются **портами**. Таким образом, адресом назначения, который используется



протоколом TCP, является идентификатор порта прикладной службы. Номер порта в совокупности с номером сети и номером конечного узла однозначно определяют прикладной процесс в сети. Этот набор идентифицирующих параметров имеет название **сокет** (socket).

3.1.7 Перенос данных.

3.2 Фаза переноса данных.

Цель фазы переноса данных состоит в передаче транспортного сервисного блока данных между двумя объектами сеанса, которые связаны транспортным соединением. Это обеспечивается транспортированием транспортного протокольного блока данных и следующими функциями, каждая из которых используется или не используется в зависимости от класса обслуживания, выбранного в фазе установления соединения:



3.2.1 Упорядочивание.

Эта функция обеспечивает упорядочивание сообщений, посылаемых одним транспортным объектом другому. То есть, в какой последовательности сообщения посылались передающим транспортным объектом, в такой же последовательности они будут обрабатываться принимающим транспортным объектом.

3.2.2 Сегментация.

При обработке транспортного сервисного блока данных транспортный уровень выполняет **сегментацию**. **Сегментация** – это преобразование одного сервисного блока в множество протокольных блоков данных (PDU). При сегментации, один сервисный блок данных разбивается на множество сегментов, затем к каждому такому сегменту добавляется заголовок (того уровня на котором выполняется сегментация), в результате получается множество PDU. Причиной сегментации является различие в размерах PDU



уровня N и уровня (N+1).

Сегментация осуществляется при передаче.

В качестве примера протокола, выполняющего эту функцию, можно привести протокол TCP.

PDU протокола TCP является сегмент (понятие сегмента в модели OSI похоже на понятие сегмента TCP, но их нужно различать). Информация, поступающая к протоколу TCP в рамках логического соединения от протоколов более высокого уровня, рассматривается протоколом TCP как неструктурированный поток байтов. Поступающие данные буферизируются средствами TCP. Для передачи на сетевой уровень из буфера «вырезается» некоторая непрерывная часть данных, которая и называется сегментом.



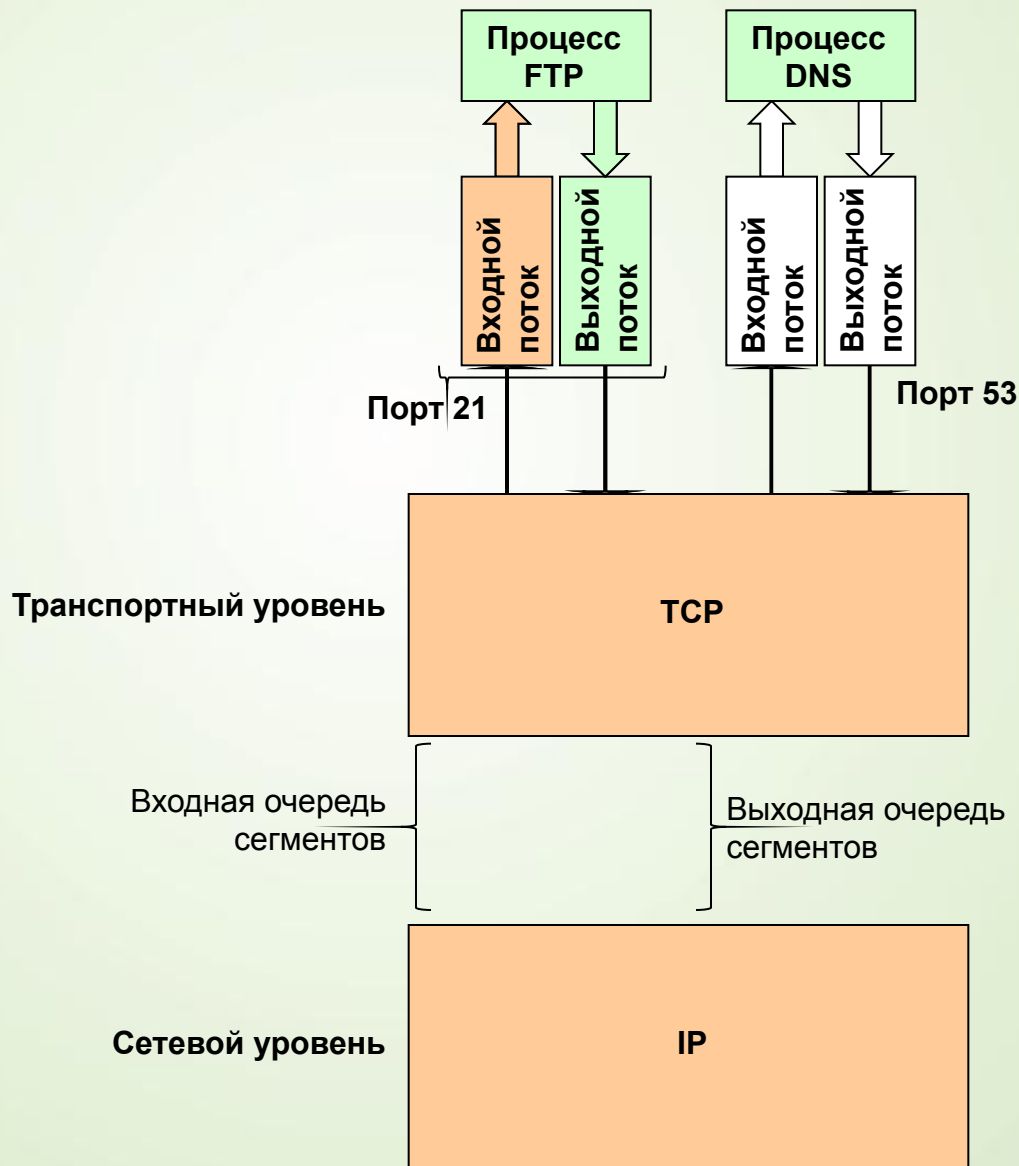


Рисунок 5.3 – Иллюстрация процесса сегментации и реассемблирования



3.2.3 Реассемблирование.

После того, как передающим транспортным объектом, была выполнена сегментация, принимающему транспортному объекту нужно выполнить **реассемблирование**. **Реассемблирование** – преобразование множества PDU в один сервисный блок данных.

При реассемблировании транспортный объект должен собрать сервисный транспортный блок данных, в соответствии с той последовательностью, которая была при сегментации. Транспортный объект удаляет заголовок из каждого PDU, который участвует в реассемблировании сервисного блока данных, а затем образовавшиеся сегменты объединяются в сервисный блок данных.

3.2.4 Мультиплексирование и расщепление (splitting).

В зависимости от того, какое решение было принято относительно мультиплексирования и расщепления в фазе



установления соединения, будет выполняться или мультиплексирование или расщепление.

Например, мультиплексированием в протоколе TCP называется процедура обслуживания запросов, поступающих от нескольких различных прикладных служб. Обратная процедура распределения протоколом TCP поступающих от сетевого уровня пакетов между набором высокоуровневых служб, идентифицированных номерами портов, называется демультиплексированием.

Демультиплексирование не нужно путать с расщеплением. **Расщепление** – функция уровня N, которая заключается в поддержке одного N соединения более чем одним (N-1) соединением. Для транспортного уровня это означает, что при использовании этой функции, множество сетевых соединений будет поддерживать одно транспортное соединение.



3.2.5 Управление потоком данных.

Эта функция обеспечивает управление скоростью передачи транспортных PDU, управление сегментацией, управление мультиплексированием.

Например, управление потоком данных в протоколе TCP реализуется с помощью скользящего окна.

Особенность использования алгоритма скользящего окна в протоколе TCP состоит в том, что, хотя единицей передаваемых данных является сегмент, окно определено на множестве нумерованных байтов неструктурированного потока данных, поступающих с верхнего уровня и буферизируемых протоколом TCP. Получающий модуль TCP отправляет «окно» посылающему модулю TCP (значения «окна» передаётся в поле заголовка сообщения TCP, которое так и называется - Window). Данное окно задаёт количество байтов (начиная с номера байта, о котором уже была выслана квитанция (подтверждение)), которое принимающий модуль TCP готов в



настоящий момент принять.

Квитанция (подтверждение), в протоколе ТСР, посылается только в случае правильного приёма сегмента, отрицательные квитанции (говорящие о неправильном приёме) не посылаются. В качестве квитанции получатель сегмента отсылает ответное сообщение (сегмент), в которое помещает число, на единицу превышающее максимальный номер байта в полученном сегменте. Это число называют **номером очереди**.

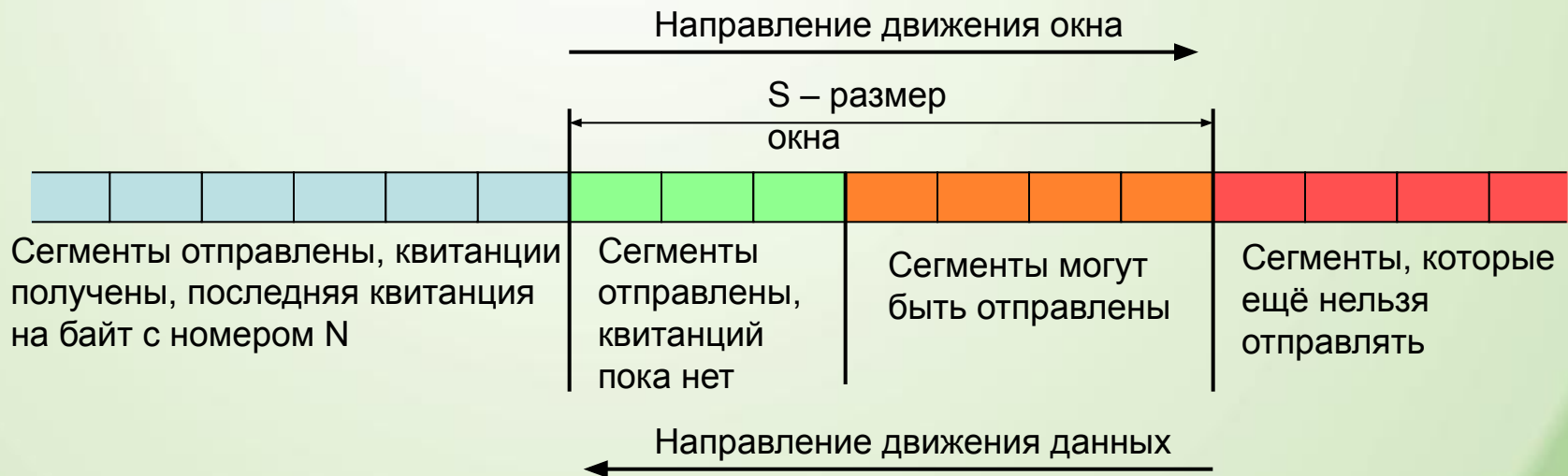


Рисунок 5.4 – Структура скользящего окна



На рисунке 5.4 (предыдущий слайд) показан поток байтов, поступающих на вход протокола TCP. Из потока байтов модуль TCP нарезает последовательность сегментов. В этом потоке можно указать несколько логических границ. Первая граница отделяет сегменты, которые уже были отправлены и на которые уже пришли квитанции. Следующую часть потока составляют сегменты, которые также уже отправлены, так как входят в границы, определённые окном, но квитанции на них пока не получены. Третья часть потока – это сегменты, которые пока не отправлены, но могут быть отправлены, так как входят в пределы окна. И наконец, последняя граница указывает на начало последовательности сегментов, ни один из которых не может быть отправлен до тех пор, пока не придёт очередная квитанция и окно не будет сдвинуто вправо.

Если размер окна равен S , а последняя по времени квитанция содержала значение N , то отправитель может посылать новые сегменты до тех пор, пока в очередной сегмент



не попадёт байт с номером $N+S$. Этот сегмент выходит за рамки окна, и передачу в таком случае необходимо приостановить до прихода следующей квитанции.

Варьируя величину окна, можно влиять на загрузку сети. Чем больше окно, тем большую порцию неподтверждённых данных можно послать в сеть. Но если пришло большее количество данных, чем может быть принято протоколом ТСР, данные будут отброшены. Это приведёт к излишним пересылкам информации и ненужному увеличению нагрузки на сеть и протокол ТСР.

При переполнении приёмного буфера конечного узла «перегруженный» протокол ТСР, отправляя квитанцию, помещает в неё новый, уменьшенный размер окна. Если он совсем отказывается от приёма, то в квитанции указывается окно нулевого размера.



3.2.6 Обнаружение ошибок и любой необходимый контроль качества обслуживания.

Например, в протоколе TCP, обнаружение ошибок обеспечивается с помощью специального поля в заголовке, которое называется checksum (контрольная сумма). Контрольная сумма вычисляется передающей программой TCP по всему сегменту TCP и помещается в поле checksum. Принимающая программа TCP снова рассчитывает контрольную сумму, если значение вновь вычисленной контрольной суммы будет совпадать со значением контрольной суммы, рассчитанной на передаче, то принимающая программа TCP считает, что сегмент принят правильно (без искажений). Если же значения контрольных сумм не совпадают, то значит сегмент искажён; таким образом обнаружена ошибка. Об обнаружении ошибки принимающая программа TCP сообщает передающей программе TCP, не передавая квитанции подтверждения. А квитанция не получена принимающей



программой TCP, значит произошла ошибка.

Если квитанция не пришла к отправителю, то это означает, что либо сегмент искажён, либо сегмент потерян (если сегмент потерян, то и квитанция на него не придёт), либо потеряна квитанция.

Для того, чтобы обеспечить квитирование, каждый сегмент должен нумероваться. В протоколе TCP каждому байту данных присваивается номер очереди. Номер очереди для первого байта данных в сегменте передаётся вместе с этим сегментом и называется номером очереди для сегмента. Сегменты также несут номер подтверждения, который является номером для следующего ожидаемого байта данных, передаваемого в обратном направлении.

3.2.7 Исправление ошибок.

Например, в протоколе TCP исправление ошибок обеспечивается за счёт повторной передачи тех сегментов,



которые не были подтверждены квитанциями. Что бы реализовать повторную передачу, протокол TCP передаёт сегмент, поместив его копию в буфер повторной передачи. Когда сегмент отправлен, протокол TCP сразу же запускает таймер. Когда приходит подтверждение для сегмента, он удаляется из буфера повторной передачи. Если подтверждение не приходит до истечения таймера, то сегмент посылается повторно.

3.2.8 Перенос срочных данных.

Эта функция имеет реализацию в протоколе TCP. В формате поля заголовка TCP есть поле под названием URG, если это поле установлено в 1, то сегмент с таким значением поля URG будет обрабатываться вне очереди. Например, если приёмный буфер получателя будет полностью заполнен и при этом придёт срочный сегмент, то он будет принят, за счёт вытеснения (удаления) других не срочных сегментов.



Примером такого срочного сегмента может быть сегмент, содержащий новый размер окна, который как можно быстрее должен быть доставлен получателю.

3.3 Фаза разъединения соединения.

Запрос на разъединение транспортного соединения может быть инициирован любым из объектов сеанса, участвующем в соединении.

Разъединение соединения может включать следующие функции:

3.3.1 Уведомление о причине разъединения.

3.3.2 Идентификация разъединившегося транспортного соединения.

3.3.3 Перенос данных.

4. Приостановка и продолжение транспортного соединения.

С помощью функции приостановки соединения происходит



разъединение сетевого соединения, а транспортное соединение сохраняется. Запрос на выполнение такой функции может быть инициирован верхними уровнями.

При выполнении функции приостановки транспортные объекты не могут обмениваться данными.

Функция приостановки соединения используется, например, в том случае, когда какой-то из вышележащих уровней принимает решение о том, что выгодно на время прервать передачу данных, а потом через некоторое время возобновить передачу.

С помощью функции продолжить, транспортный уровень вновь устанавливает сетевое соединение и обмен данными может быть продолжен.



Функции транспортного уровня в режиме без установления соединения:

1. Отображение транспортного адреса на сетевой адрес.

2. Отображение передачи данных в режиме без установления соединения транспортного уровня на передачу в режиме без установления соединения сетевого уровня.

То есть, транспортный уровень для того, чтобы передавать свои PDU по сети в режиме без установления соединения, обращается с запросом о обеспечении передачи без установления соединения к сетевому уровню.

3. Обнаружение ошибок и контроль качества обслуживания.



Примером транспортного протокола, работающего в режиме без установления соединения и выполняющего эту функцию, может служить протокол дейтаграмм пользователя UDP (User Datagram Protocol).

В отличие от протокола TCP, в протоколе UDP отсутствует подтверждение приёма блоков данных, что делает UDP намного проще, чем TCP, но относительно менее надёжным. Но протокол UDP может обнаруживать ошибки в дейтаграммах, делает он это с помощью поля checksum, назначение которого такое же как и в протоколе TCP. Если протокол UDP обнаруживает ошибку в дейтаграмме, то он сообщает об этом отправителю и ничего более. Этот механизм является единственным средством для простого контроля качества обслуживания. Если, например получатель регистрирует слишком большое количество ошибок, то он может отказаться от передачи данных на данный момент и возобновить её позже.

Несмотря на то, что протокол UDP не обеспечивает высокую



надёжность, он часто применяется для передачи речи и видео в реальном времени. Это объясняется тем, что протокол UDP прост, а следовательно работает быстро, так как всё, что требуется - это передать принятые блоки данных соответствующему приложению, идентифицируемому номером порта. А функции по исправлению ошибок, в данном случае, возлагаются на протоколы других уровней.

5.1 КЛАССЫ И ТИПЫ ТРАНСПОРТНОГО ОБСЛУЖИВАНИЯ

Определены два типа транспортного обслуживания:

1. Обслуживание в режиме с установлением соединения.
2. Обслуживание в режиме без установления соединения.

При обращении к транспортным услугам пользователь должен заявить о том, какой тип обслуживания он предполагает использовать.

В модели OSI нет никаких определённых классов



транспортного обслуживания.

5.2 КАЧЕСТВО ТРАНСПОРТНОГО ОБСЛУЖИВАНИЯ В РЕЖИМЕ С УСТАНОВЛЕНИЕМ СОЕДИНЕНИЯ

Термин качество обслуживания относится к некоторым характеристикам транспортного соединения.

С помощью параметров качества обслуживания пользователь **транспортных услуг TS** (Transport Service) задаёт свои требования к тем услугам, которыми он будет пользоваться, то есть исходя из этих требований транспортный уровень будет знать какое качество обслуживания нужно обеспечить для того или иного пользователя TS.

Пользователь TS договаривается с транспортным уровнем о качестве обслуживания с помощью специального запроса. Пользователь TS либо сам транспортный уровень может понизить установленное запросом качество обслуживания в течение транспортного соединения. Относительно параметров



качества обслуживания это может означать, что:

1. Задержка увеличивается;
2. Пропускная способность уменьшается;
3. Коэффициент ошибок повышается;
4. Приоритет становится ниже;
5. Вероятность отказа становится выше;
6. Уровень защиты транспортного соединения понижается.

Транспортный уровень не гарантирует поддержку первоначального договора о качестве обслуживания на протяжении всего времени существования транспортного соединения.

Значения параметров качества обслуживания могут назначаться по умолчанию самим транспортным уровнем, а могут быть определены в результате переговоров пользователя TS и транспортного уровня.



Таблица 5.1 – Классификация параметров качества транспортного обслуживания

Фаза	Критерий качества	
	Скорость	Правильность/Надёжность
Установление транспортного соединения	Задержка установления транспортного соединения	Вероятность неудачи установления транспортного соединения (неправильное соединение/отказ)
Перенос данных	Пропускная способность	Остаточный коэффициент ошибки (искажение, дублирование/потеря)
	Транзитная задержка	Вероятность неудачи передачи данных
Разрушение транспортного соединения	Задержка разрушения транспортного соединения	Вероятность неудачного разрушения транспортного соединения

Вероятность неудачи установления транспортного соединения – отношение общего количества неудачных



попыток установления соединения к общему количеству попыток установления соединения.

Неудача в установлении соединения имеет место, когда транспортное соединение не установлено в пределах указанной максимальной задержки установления соединения в результате неправильного соединения или отказа в установлении соединения.

Пропускная способность определена для каждого направления передачи в транспортном соединении. Для каждого направления определены максимальное значение пропускной способности и среднее значение пропускной способности. Максимальное значение – это максимальная скорость при которой транспортный уровень непрерывно принимает и отправляет данные при отсутствии управления потоком данных.

Среднее значение пропускной способности – это ожидаемая скорость передачи в транспортном соединении, включая



эффекты, возникающие при управлении потоком данных.

Транзитная задержка – время прошедшее между посылкой транспортного сервисного блока данных и его приёмом. Рассчитывается транзитная задержка только по тем транспортным сервисным блокам данных, которые успешно переданы их получателю. Под успешностью понимается пересылка транспортных сервисных блоков данных без ошибок, в надлежащей последовательности и до разъединения транспортного соединения.

Остаточный коэффициент ошибки – отношение общего количества ошибочных, потерянных и дублированных транспортных сервисных блоков данных к общему количеству транспортных сервисных блоков данных перенесённых транспортной службой в течение периода измерения.

Вероятность неудачи передачи данных – отношение общего количества неудачных передач к общему числу передач, составляющих выборку, наблюдаемых в течение



измерения.

Выборка передач – это те передачи сервисных транспортных блоков данных за которыми было установлено наблюдение.

Вероятность неудачного разрушения транспортного соединения – отношение общего количества неудачных запросов на разрушение транспортного соединения к общему количеству запросов на разрушение транспортного соединения, включенных в выборку измерений. Этот параметр обычно определяется независимо для каждого пользователя TS.

Приоритет транспортного соединения – определяет отношения между транспортными соединениями. Этот параметр определяет относительную значимость транспортного соединения по следующим аспектам:

1. Порядок, в котором транспортные соединения должны ухудшить качество обслуживания в случае необходимости.
2. Порядок, в котором транспортные соединения будут нарушены, в случае необходимости возврата ресурсов,



которые они используют.

5.3 КАЧЕСТВО ТРАНСПОРТНОГО ОБСЛУЖИВАНИЯ В РЕЖИМЕ БЕЗ УСТАНОВЛЕНИЯ СОЕДИНЕНИЯ

Отличительной чертой обслуживания в режиме без установления соединения является отсутствие переговоров о качестве обслуживания во время передачи данных. То есть пользователь TS не может договариваться с транспортным уровнем о значениях параметров качества обслуживания. Единственное, чем обеспечиваются пользователи TS в режиме без установления соединения – это знанием значений параметров качества обслуживания.

Параметры качества обслуживания, определённые для режима передачи без установления соединения: транзитная задержка, защита, остаточная вероятность ошибки, приоритет.

Определение **транзитной задержки** такое же, как и в случае режима с установлением соединения, но с учётом следующей



особенности – транзитная задержка определяется для каждой посылки транспортного сервисного блока данных отдельно, в то время как в режиме с установлением соединения транзитная задержка может быть определена по соединению в целом.

Определение **остаточной вероятности ошибки** тоже соответствует определению, данному для режима с установлением соединения, но с той же поправкой.

Приоритет – этот параметр определяет относительную значимость передач транспортных сервисных блоков данных по следующим аспектам:

1. Порядок, в котором для транспортных сервисных блоков данных, в случае необходимости, понизится качество обслуживания (причём качество обслуживания понижается для всех транспортных сервисных блоков данных, которыми обмениваются пользователи TS).

2. Порядок, в котором транспортные сервисные блоки данных, в случае необходимости, должны быть отброшены, для



того, чтобы вернуть используемые ресурсы.

6. СЕТЕВОЙ УРОВЕНЬ

Сетевой уровень (Network Layer) обеспечивает функциональные и процедурные средства для транспортных объектов, благодаря которым транспортные объекты освобождаются от функций ретрансляции и маршрутизации.

Сетевой уровень обеспечивает установку, поддержку и завершение сетевых соединений между открытыми системами, которые содержат взаимодействующие прикладные объекты.

Сетевой уровень обеспечивает обмен сетевыми сервисными блоками данных (network-service-data-units, далее как сетевые SDU) между транспортными объектами через сетевые соединения.

Сетевой уровень служит для образования единой транспортной системы, объединяющей несколько сетей, причём эти сети могут использовать совершенно различные принципы



передачи сообщений между конечными узлами и обладать произвольной структурой связей.

Услуги, предоставляемые транспортному уровню в режиме с установлением соединения:

Основная услуга сетевого уровня заключается в обеспечении прозрачного переноса данных между транспортными объектами. Благодаря этой услуге содержимое передаваемой информации детально интерпретируется только верхними уровнями (то есть уровнями располагающимися выше сетевого).

1. Сетевые адреса.

Транспортные объекты известны сетевому уровню благодаря сетевым адресам. С помощью сетевых адресов транспортный объект уникально идентифицирует тот транспортный объект, с которым он хочет связаться.



Например, в протоколе TCP отправитель и получатель уникально идентифицируются сокетами. Как уже было отмечено ранее, сокет - это номер порта в совокупности с номером сети и номером конечного узла. Номер сети и номер узла составляют сетевой адрес, а именно IP-адрес.

Протокол межсетевого взаимодействия IP (Internet Protocol), относится к сетевому уровню, он работает в режиме без установления соединения и обеспечивает передачу дейтаграмм от отправителя к получателю через объединённую систему компьютерных сетей.

Например в заголовке протокола IPv4 (четвёртая версия) под IP-адрес источника отведено 32 бита (под IP-адрес назначения также отведено 32 бита). Из этих 32 бит какая-то часть отведена для номера сети (в зависимости от класса IP-адреса), а оставшая часть для номера узла в этой сети. Например IP-адрес 127.220.64.23 относится к классу A. В двоичном виде он будет выглядеть так: 01111111.11011100.01000000.00010111.



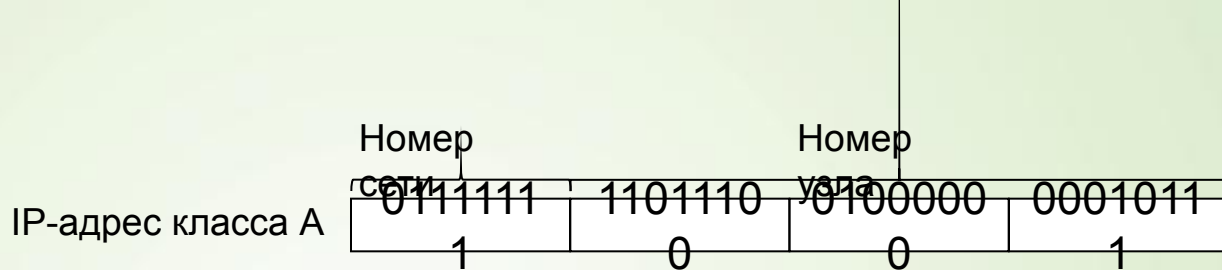


Рисунок 6.1 – Пример сетевого адреса

Например, в протоколе X.25 на сетевом уровне может использоваться адрес любой длины (в пределах формата поля адреса). Максимальная длина поля адреса в пакете X.25 составляет 16 байт. Сетевым уровнем в технологии X.25 является протокол который так и называется X.25. Сеть X.25 – это сеть с коммутацией пакетов, ориентированная на передачу по ненадёжным линиям. Сетевой уровень X.25 может работать в режиме с установлением соединения.

Ещё одним примером сетевой адресации может послужить адресация на уровне 3 МТР стека протоколов ОКС7. Уровень 3 МТР относится к сетевому уровню. На этом уровне отправитель идентифицируется кодом исходящего пункта ОРС (Origination



Point Code), а получатель кодом пункта назначения DPC (Destination Point Code). Каждый из кодов OPC и DPC занимают 14 бит в формате сигнальной единицы (так называется сетевой PDU в ОКС7).

2. Сетевые соединения.

Через сетевое соединение обеспечивается обмен данными между транспортными объектами, идентифицированными сетевыми адресами. Сетевой уровень обеспечивает установление соединения, его поддержку и разъединение.

Сетевое подключение работает по принципу «точка-точка». Дело в том, что между транспортными объектами часто бывает более двух сетевых объектов и при создании сетевого соединения, оно будет создаваться от одного сетевого объекта к другому сетевому объекту, то есть от точки к точке. По этой же причине между одной и той же парой транспортных объектов может существовать более одного сетевого соединения.



3. Перенос сетевых SDU.

Сетевой уровень обеспечивает перенос сетевых SDU, обеспечивая целостность их содержимого.

4. Параметры качества обслуживания.

Сетевой уровень устанавливает и поддерживает выбранное качество обслуживания на протяжении сетевого соединения.

Параметры качества обслуживания включают остаточный коэффициент ошибки, готовность обслуживания, надёжность, пропускную способность, транзитную задержку (включая её колебание) и задержку установления сетевого соединения.

5. Уведомление об ошибках.

Сетевой уровень при обнаружении неисправленных ошибок сообщает о них транспортным объектам. Обнаружение неисправленных ошибок может привести к разъединению сетевого соединения.



6. Перенос срочных сетевых SDU.

Передача срочных сетевых SDU обеспечивает дополнительное средство информационного обмена для сетевого соединения.

Эта услуга является дополнительной и не всегда может быть доступной. Это означает, что пользователь сетевых услуг должен предварительно запрашивать эту услугу, а сетевой уровень рассмотрев этот запрос может удовлетворить его, а может отказать в предоставлении данной услуги.

7. Сброс сетевого соединения.

При выполнении этой услуги сетевой уровень отбрасывает все сетевые SDU, которые в данный момент проходят по сетевому соединению. О сбросе сетевой уровень уведомляет окончательный транспортный объект, который участвовал в данном сетевом соединении.

Эта услуга является дополнительной и не всегда может быть



доступной.

8. Разъединение сетевого соединения.

Запрос на разъединение сетевого соединения может инициироваться транспортным объектом.

Сетевая служба не гарантирует доставку данных, которые были запрошены непосредственно перед разъединением соединения и тех данных, которые проходят по сетевому соединению после запроса на разъединение.

9. Квитанция подтверждения.

Транспортный объект может отправлять другому транспортному объекту, через сетевое соединение, квитанции подтверждения о получении данных. Об использовании этой услуги договариваются между собой пользователи сетевого соединения в фазе установления соединения.

Эта услуга является дополнительной и не всегда может быть доступной.



Услуги, предоставляемые транспортному уровню в режиме без установления соединения:

1. Передача сетевых SDU определённого максимального размера;
2. Параметры качества обслуживания;
3. Уведомление об ошибках.

Функции сетевого уровня:

1. Маршрутизация и ретрансляция.

Сетевые соединения обеспечиваются сетевыми объектами в оконечных открытых системах и промежуточными открытыми системами, которые обеспечивают ретрансляцию.

Чтобы установить окончательную связь, сетевой уровень может использовать услуги уровня звена данных, для того, чтобы управлять взаимосвязью между физическими объектами. Для того, чтобы управлять взаимосвязью между физическими



объектами, сетевой уровень должен обеспечить взаимодействие между сетевым объектом и физическим объектом в одной и той же открытой системе. Но так как модель OSI разрешает прямое взаимодействие только между смежными уровнями, сетевой объект не может взаимодействовать непосредственно с физическим объектом, поэтому взаимодействие между этими объектами осуществляется через уровень звена данных.

С помощью функций маршрутизации определяется подходящий маршрут между сетевыми адресами. В реальных сетях устройства выполняющие функции маршрутизации часто называются маршрутизаторами. Маршрутизаторы соединяют между собой различные сети.

Маршрутизатор – это устройство, которое собирает информацию о топологии межсетевых соединений и на её основании пересылает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя,



находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач между сетями, или хопов (от hop - прыжок), каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

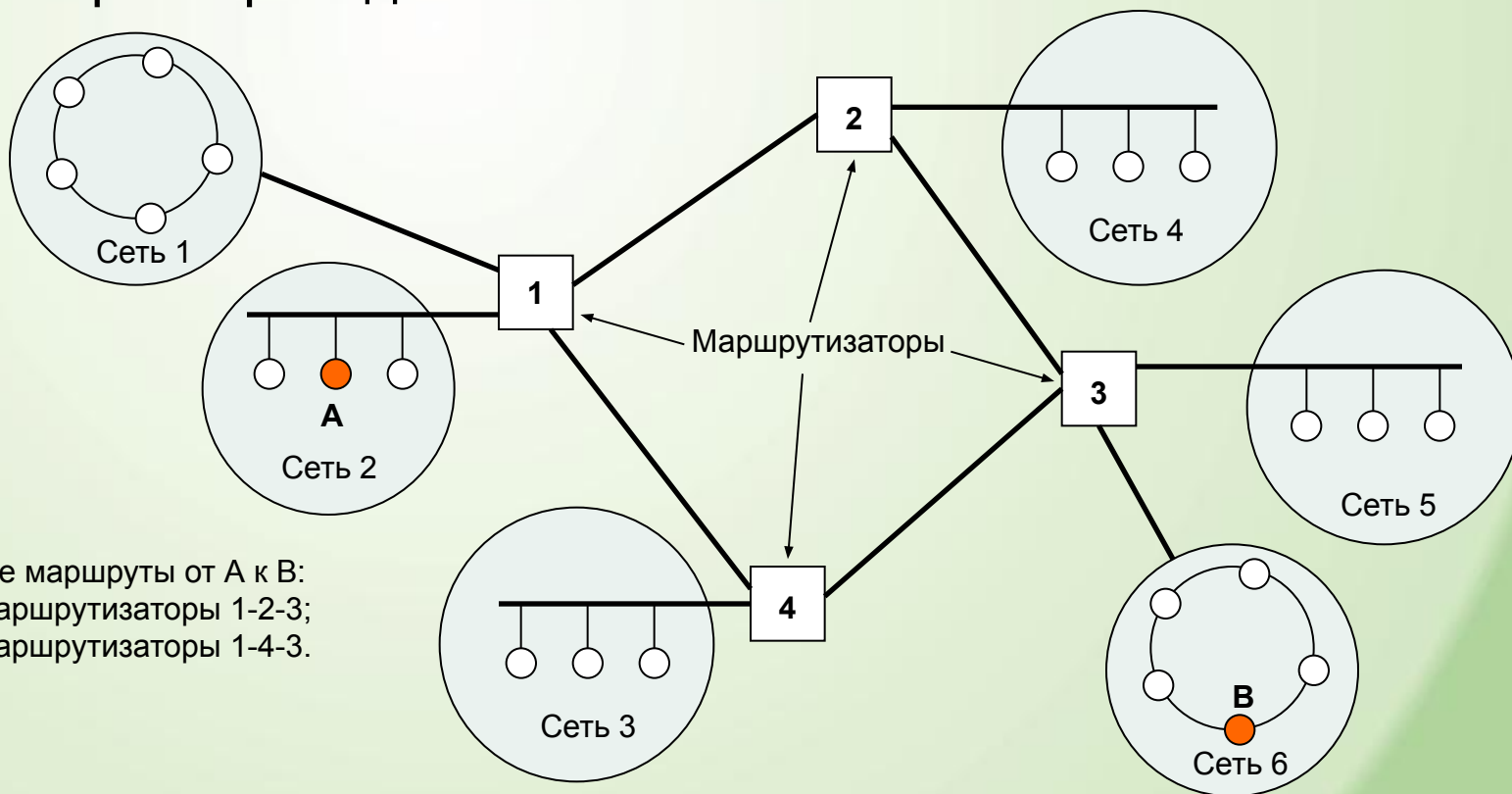


Рисунок 6.2 – Пример составной сети



Проблема выбора наилучшего пути называется маршрутизацией, и её решение является одной из главных задач сетевого уровня. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например надёжности передачи.

Принципы маршрутизации можно рассмотреть на примере составной сети, изображённой на рисунке 6.3 (следующий слайд). В этой сети 12 маршрутизаторов объединяют 11 сетей в общую составную сеть; N_1, N_2, \dots, N_{11} – номера сетей.



N11	-	MA	0
default	M11(2)	MA	-

N1	-	M1(2)	0
N2	-	M1(1)	0
N3	M2(1)	M1(1)	1
N4	M3(2)	M1(1)	1
N5	M3(2)	M1(1)	1
N6	M2(1)	M1(1)	2
default	M12(1)	M1(2)	-

N1	M3(3)	MB	
N2	M3(3)	MB	
N3	M4(1)	MB	
N4	-	MB	
N5	M3(3)	MB	
N6	M4(1)	MB	
default	M4(1)	MB	

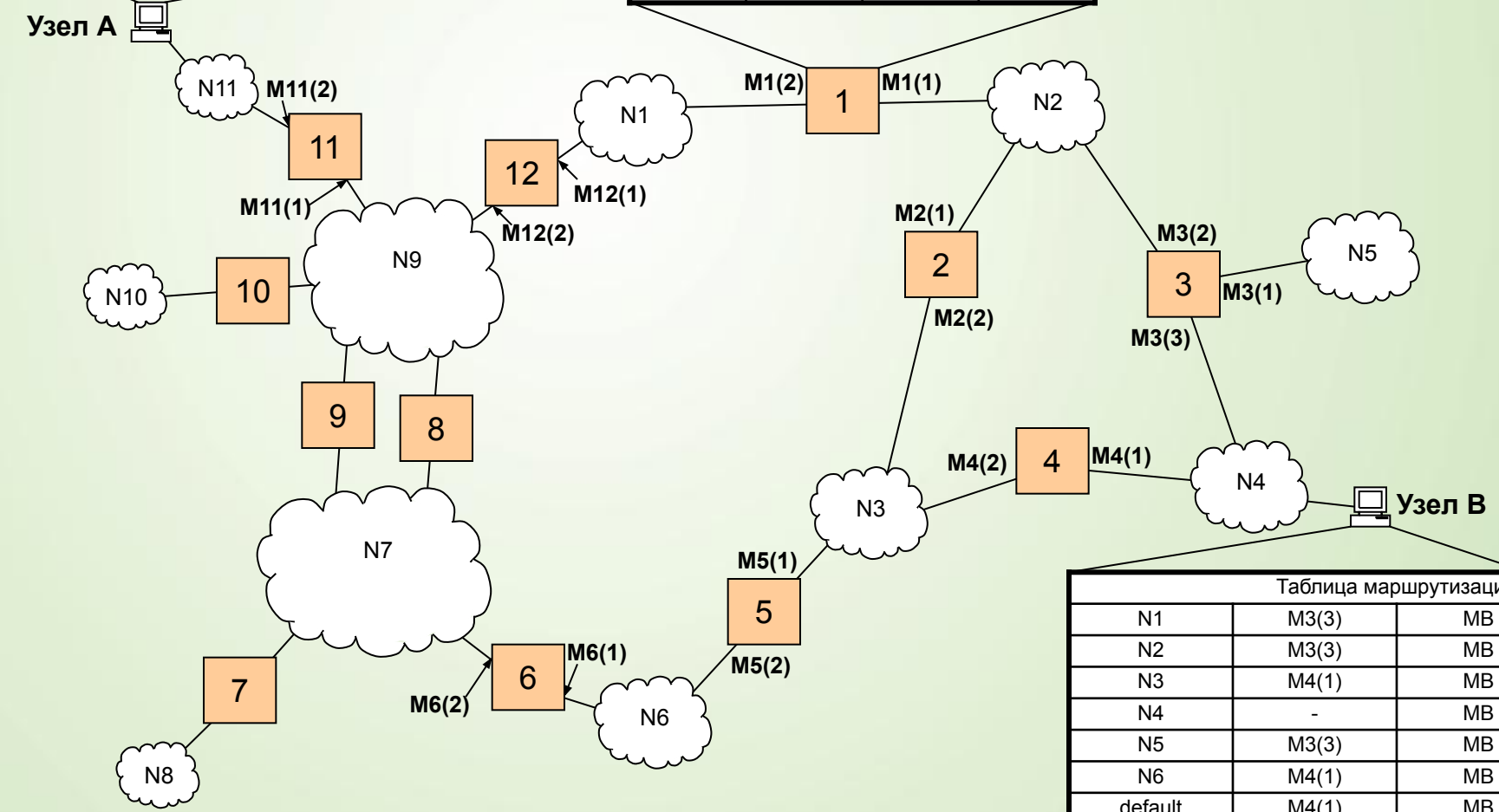


Рисунок 6.3 – Принципы маршрутизации в составной сети



Маршрутизаторы имеют по несколько портов (по крайней мере, по два), к которым присоединяются сети. Каждый порт маршрутизатора можно рассматривать как отдельный узел сети: он имеет собственный сетевой адрес и собственный локальный адрес в той подсети, которая к нему подключена (локальный адрес относится к системе адресации уровня звена данных). Например, маршрутизатор под номером 2 имеет два порта, к которым подключены сети N2 и N3. На рисунке сетевые адреса этих портов обозначены как M2(1) и M2(2). Порт M2(1) имеет локальный адрес в сети N2, порт M2(2) в сети N3. Таким образом, маршрутизатор можно рассматривать как совокупность нескольких узлов, каждый из которых входит в свою сеть. Как единое устройство маршрутизатор не имеет ни отдельного сетевого адреса, ни отдельного локального адреса.

В сложных составных сетях почти всегда существует несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Так, пакет, отправленный из



узла А в узел В, может пройти через маршрутизаторы 11, 12, 1, 3 или маршрутизаторы 11, 8, 6, 5, 4. Существуют и другие пути.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании указанного критерия выбора маршрута. Обычно в качестве критерия выступает задержка прохождения маршрута отдельным пакетом или средняя пропускная способность маршрута для последовательности пакетов. Часто также используется весьма простой критерий, учитывающий только количество пройденных в маршруте промежуточных маршрутизаторов (хопов).

Чтобы по адресу сети назначения можно было бы выбрать рациональный маршрут дальнейшего следования пакета, каждый конечный узел и маршрутизатор анализирует специальную информационную структуру, которая называется таблицей маршрутизации. В таблице 6.2 приведён один из



возможных вариантов таблицы маршрутизации маршрутизатора 1.

Таблица 6.2 – Таблица маршрутизации маршрутизатора 1

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения (в хопах)
N1	-	M1(2)	0
N2	-	M1(1)	0
N3	M2(1)	M1(1)	1
N4	M3(2)	M1(1)	1
N5	M3(2)	M1(1)	1
N6	M2(1)	M1(1)	2
default	M12(1)	M1(2)	-

В каждой строке таблицы следом за номером сети указывается сетевой адрес следующего маршрутизатора (более точно, сетевой адрес соответствующего порта



следующего маршрутизатора), на который надо направить пакет, чтобы тот передвигался по направлению к сети с данным номером по рациональному маршруту.

Когда на маршрутизатор поступает новый пакет, номер сети назначения, извлечённый из поступившего кадра, последовательно сравнивается с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети указывает, на какой ближайший маршрутизатор следует направить пакет. Например, если на какой-либо порт маршрутизатора 1 поступает пакет, адресованный в сеть N5, то из таблицы следует, что адрес следующего маршрутизатора – M3(2), то есть очередным этапом движения данного пакета будет движение к порту 2 маршрутизатора 3.

Поскольку пакет может быть адресован в любую сеть составной сети, может показаться, что каждая таблица маршрутизации должна иметь записи обо всех сетях, входящих в составную сеть. Но при таком подходе в случае крупной сети



объём таблиц маршрутизации может оказаться очень большим, что влияет на время просмотра, потребует много места хранения и т. п. Поэтому на практике число записей в таблице маршрутизации стараются уменьшить за счёт использования специальной записи – «маршрутизатор по умолчанию» (default). Действительно, если принять во внимание топологию составной сети, то в таблицах маршрутизаторов, находящихся на периферии составной сети, достаточно записать номера сетей, непосредственно подсоединённых к данному маршрутизатору или расположенных поблизости, на тупиковых маршрутах. Обо всех же остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется маршрутизатором по умолчанию, а вместо номера сети в соответствующей строке помещается особая запись, например default. Таким маршрутизатором для сети N1 является маршрутизатор 12, точнее его порт M12(1).



Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов он должен поместить данный пакет. Для этого служит третий столбец таблицы маршрутизации.

Задачу маршрутизации решают не только промежуточные узлы-маршрутизаторы, но и конечные узлы. Средства сетевого уровня, установленные на конечном узле, при обработке пакета должны, прежде всего, определить, направляется ли он в другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, то для данного пакета не требуется решать задачу маршрутизации. Если же номера сетей отправления и назначения не совпадают, то маршрутизация нужна. Таблицы маршрутизации конечных узлов полностью аналогичны таблицам маршрутизации, хранящимся на маршрутизаторах.

Таблица 6.3 представляет один из возможных вариантов



таблицы маршрутизации конечного узла В.

Таблица 6.3 – Таблица маршрутизации конечного узла В

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения (в хопах)
N1	M3(3)	MB	2
N2	M3(3)	MB	1
N3	M4(1)	MB	1
N4	-	MB	0
N5	M3(3)	MB	1
N6	M4(1)	MB	2
default	M4(1)	MB	-

Здесь MB – сетевой адрес порта конечного узла В. На основании этой таблицы конечный узел В выбирает, на какой из двух имеющихся в сети N4 маршрутизаторов следует посылать тот или иной пакет.

Конечные узлы в ещё большей степени, чем



маршрутизаторы, пользуются приёмом маршрутизации по умолчанию. Хотя они также в общем случае имеют в своём распоряжении таблицу маршрутизации, её объём обычно незначителен, что объясняется периферийным расположением всех конечных узлов. Конечный узел часто вообще работает без таблицы маршрутизации, имея только сведения об адресе маршрутизатора по умолчанию. При наличии одного маршрутизатора в локальной сети этот вариант – единственно возможный для всех конечных узлов. Но даже при наличии нескольких маршрутизаторов в локальной сети, когда перед конечным узлом стоит проблема их выбора, задание маршрута по умолчанию часто используется в конечных узлах для сокращения объёма их таблицы маршрутизации.

Для автоматического построения таблиц маршрутизации маршрутизаторы обмениваются информацией о топологии составной сети в соответствии со специальным служебным протоколом. Протоколы этого типа называют протоколами



маршрутизации. Протоколы маршрутизации (например, протокол маршрутной информации RIP (Routing Information Protocol)) следует отличать от таких сетевых протоколов, как, например IP. И те и другие выполняют функции сетевого уровня модели OSI – участвуют в доставке пакетов адресату через разнородную составную сеть. Но в то время как первые собирают и передают по сети чисто служебную информацию, вторые предназначены для передачи пользовательских данных.

2. Сетевые соединения.

Эта функция обеспечивает сетевые соединения между транспортными объектами, используя соединения уровня звена данных.

На следующем слайде проиллюстрировано сетевое соединение, на примере подсистемы управления соединениями сигнализации SCCP, которая входит в состав OKC7. Подсистема SCCP относится к сетевому уровню.



После того как отправлено сообщение CR сразу устанавливается таймер T1

Подтверждение соединения пришло до истечения таймера T1, значит T1 сбрасывается.

После сброса T1, сразу устанавливается таймер T2.

Запрос разъединения послан стороной A до истечения T2, значит T2 сбрасывается.

После сброса T2, сразу устанавливается таймер T3. Сторона A ожидает подтверждения

Подтверждение разъединения пришло до истечения таймера T3, значит T3 сбрасывается, разъединение сетевого соединения прошло успешно.

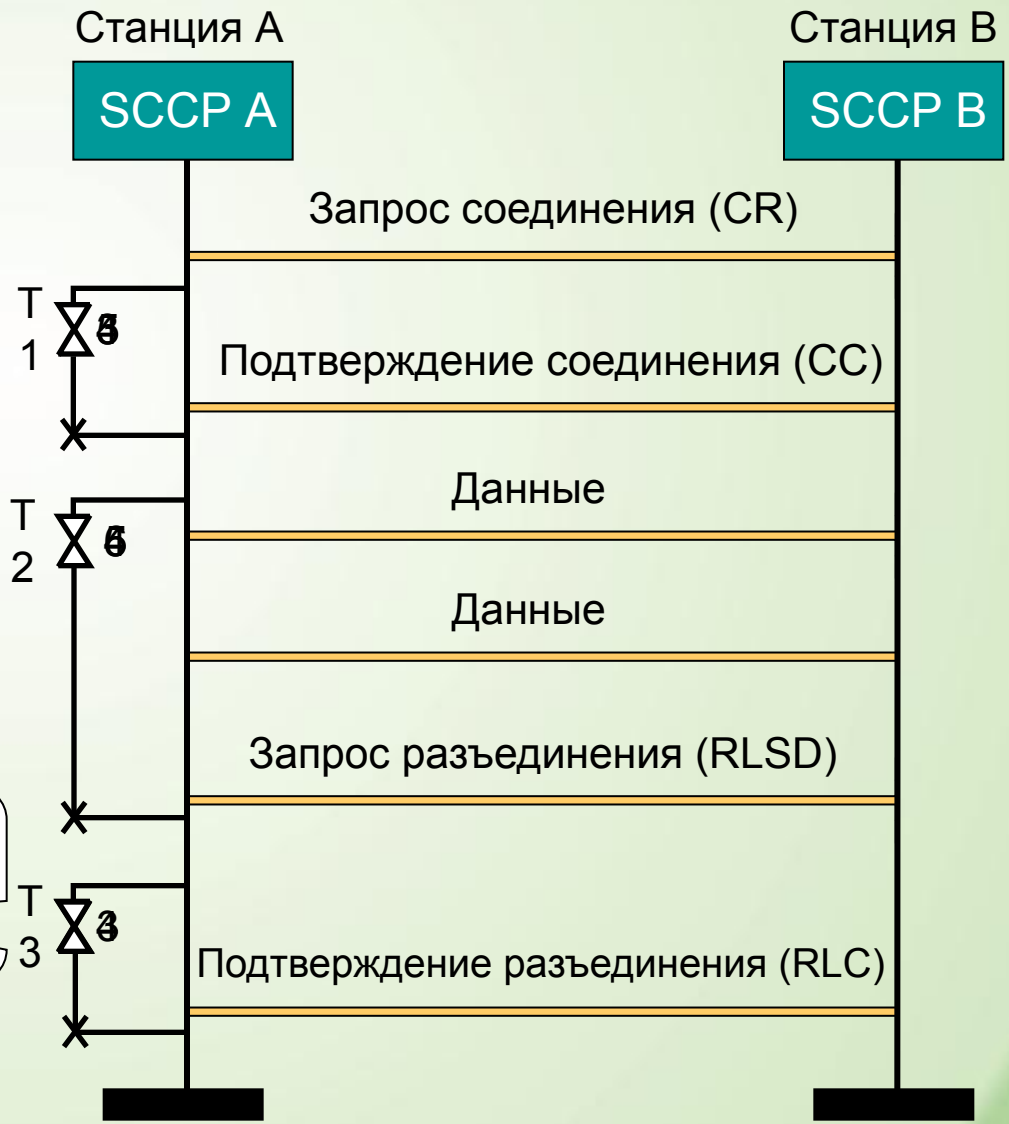


Рисунок 6.4 – Иллюстрация процесса установления и разъединения сетевого соединения



Сетевое соединение может быть обеспечено как соединение **подсетей** в тандеме, то есть использование нескольких отдельных подсетей последовательно.

Подсеть – абстракция **реальной подсети** (здесь специфическое определение подсети дано в рамках модели OSI).

Реальная подсеть – совокупность оборудования и физической среды, которая формирует автономное целое и которая может использоваться для связи реальных систем.

Связанные подсети могут иметь одни и те же или различные сервисные возможности.

Взаимосвязь пары подсетей отличающихся по качеству обслуживания может быть достигнута двумя способами:

1. Эти две подсети связываются без каких-либо изменений. Качество результирующего сетевого соединения будет не выше чем качество той сети, в которой оно наиболее низкое.

2. Та сеть, которая имеет более низкое качество



обслуживания будет усовершенствована до уровня той сети, в которой качество обслуживания более высокое. Качество результирующего сетевого соединения будет приблизительно на уровне той сети, которая изначально имела более высокое качество обслуживания.

Выбор между этими двумя альтернативами зависит от степени различия в качестве подсетей, стоимости решения и других экономических факторов.

3. Мультиплексирование сетевых соединений.

Эта функция может быть выполнена для оптимизации использования соединения уровня звена данных. То есть одно соединение уровня звена данных может поддерживать более одного сетевого соединения.

Если используется функция мультиплексирования, то функция демупльтиплексирования также используется.



4. Сегментация.

Смысл сегментации на сетевом уровне заключается в том, чтобы разделить сетевой SDU на множество сетевых PDU.

Например, протокол IP не выполняет сегментацию потому, что TCP обеспечивает IP протоколу сетевой SDU уже нужного размера.

Другой функцией, которая отличает протокол IP от других сетевых протоколов, является фрагментация, которая чём-то похожа на сегментацию. IP-фрагментация становится необходимой при передаче пакета в последующую сеть, для которой размер пакета является слишком большим. Смысл фрагментации заключается в разбиении слишком длинного для конкретного типа составляющей сети сообщения на более короткие пакеты с созданием соответствующих служебных полей, нужных для последующей сборки фрагментов в исходное сообщение.



5. Обнаружение ошибок.

Например, в протоколе IP ошибки обнаруживаются только в заголовке пакета с помощью 16-битного поля контрольной суммы. Если контрольная сумма неверна, то IP-пакет отбрасывается. О возникающих ошибках протокол IP, с помощью протокола межсетевых управляющих сообщений ICMP (Internet Control Message Protocol), уведомляет источник отправления пакетов.

Например, протокол X.25/3 (так называется сетевой уровень в X.25) обнаруживает ошибки с помощью бита подтверждения доставки D (delivery confirmation), который располагается в заголовке пакета. Обычный механизм подтверждения принятия пакетов с помощью квитанций имеет для протокола X.25 только локальный смысл – приём пакета подтверждает ближайший коммутатор сети, через который конечный узел запросил и установил виртуальное соединение. Если же узел источник запросил подтверждение приёма конечным узлом, то это



подтверждение индицируется установкой бита D в единицу в пакетах, идущих от узла назначения.

6. Исправление ошибок.

Например, в протоколе IP не обеспечивается исправление ошибок, в этом он полагается на протокол TCP (если IP работает в стеке TCP/IP).

Например, уровень 3 МТР, который входит в стек протоколов ОКС7, не обеспечивает обнаружения и исправления ошибок в сигнальных единицах, но уровень 2 МТР, над которым работает уровень 3 МТР, в полной мере обеспечивает обнаружение и исправление ошибок.

Из этих примеров несложно понять, что различные сетевые протоколы не всегда выполняют, в полной мере, тот набор функций сетевого уровня, который описан в модели OSI.



7. Упорядочивание.

Эта функция обеспечивает упорядоченную поставку сетевых SDU в данном сетевом соединении, когда этого требует транспортный объект.

8. Управление потоком данных.

Примером протокола, выполняющего данную функцию может служить уровень 3 МТР, одной из функций которого является управление сигнальным трафиком, включающее в себя реконфигурацию сигнального трафика в ответ на изменение в состоянии сети.

Функция управления сигнальным трафиком обеспечивает процедуры, необходимые для поддержания потока сигнального трафика в случае нарушений в сети сигнализации. К таким нарушениям относятся, в частности, отказ звеньев сигнализации или отказ транзитных пунктов сигнализации. В случае перегрузки пункта сигнализации, функция



используется для снижения нагрузки на этот пункт сигнализации. Для достижения этого определены отдельные процедуры перехода на резервное звено сигнализации, возврата на исходное звено сигнализации, запрещения управления звеном сигнализации и управления потоком сигнального трафика.

Управление потоком сигнального трафика используется для ограничения сигнального трафика в источнике в тех случаях, когда сеть сигнализации не в состоянии передать весь сигнальный трафик, предлагаемый подсистемами пользователей.

9. Перенос срочных данных.

Например, протокол IP выполняет эту функцию с помощью трёх-битного подполя приоритета пакета (Precedence) (это подполе входит в поле Тип сервиса (Type of Service)). Приоритет может иметь значения от самого низкого – 0



(нормальный пакет) до самого высокого – 7 (пакет управляющей информации). Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Например, IP-пакет с маршрутной информацией протокола RIP может относиться к срочным данным, а значит будет иметь высокий приоритет.

10. Сброс сетевого соединения.

Эта функция обеспечивает реализацию услуги сброса сетевого соединения.

11. Сервисный выбор.

Эта функция обеспечивает требуемое качество обслуживания для конечных точек сетевого соединения, даже если это соединение охватывает некоторое количество подсетей с различным уровнем качества.



12. Отображение между сетевыми адресами и адресами звена данных.

Примером протокола, относящегося к сетевому уровню и выполняющего данную функцию может служить протокол разрешения адресов ARP (Address Resolution Protocol). С помощью ARP по сетевому адресу определяется локальный адрес (который относится к уровню звена данных). Для этого в ARP протоколе определены специальные ARP-таблицы соответствия сетевых и локальных адресов. Например, такая ARP-таблица может содержать следующую строку:

```
192.55.05.123 00 FA 18 4C AA 04
```

В данной строке IP-адресу соответствует MAC-адрес.



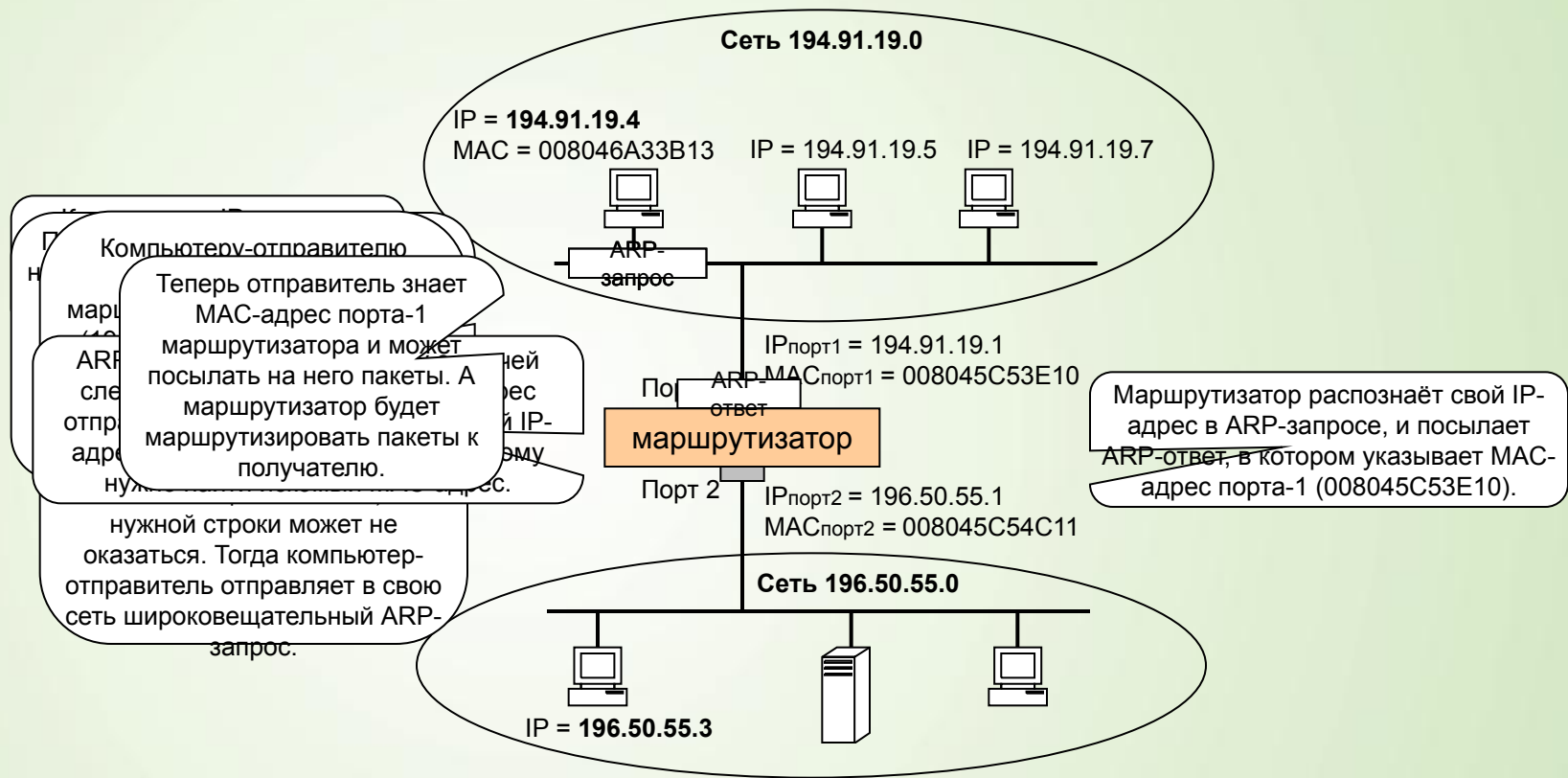


Рисунок 6.5 – Пример установления соответствия между сетевым и локальным адресами

13. Отображение передачи данных в режиме без установления соединения сетевого уровня на передачу в режиме без установления соединения уровня звена данных.



6.1 КЛАССЫ И ТИПЫ СЕТЕВОГО ОБСЛУЖИВАНИЯ

Определены два типа сетевого обслуживания:

1. Обслуживание в режиме с установлением соединения.
2. Обслуживание в режиме без установления соединения.

При обращении к сетевым услугам пользователь этих услуг должен заявить о том, какой тип обслуживания он предполагает использовать.

В модели OSI нет никаких определённых классов сетевого обслуживания.

6.2 КАЧЕСТВО СЕТЕВОГО ОБСЛУЖИВАНИЯ В РЕЖИМЕ С УСТАНОВЛЕНИЕМ СОЕДИНЕНИЯ И БЕЗ УСТАНОВЛЕНИЯ СОЕДИНЕНИЯ

Термин качество обслуживания относится к некоторым характеристикам сетевого соединения.

Параметры качества сетевого обслуживания могут быть разделены на две категории следующим образом:



1. Те значения, которые передаются между пользователями сетевых услуг NS (Network Service) во время установления сетевого соединения. О таких значениях могут договариваться пользователь NS и сетевой уровень.

2. Те значения, которые не передаются между пользователями NS или о которых не договариваются пользователи NS. Пользователи могут только получать информацию о таких значениях.

Как только сетевое соединение установлено, пользователи NS уже не могут повторно договариваться о параметрах сетевого обслуживания первой категории.

Сетевой уровень не гарантирует поддержку первоначального договора о качестве обслуживания на протяжении всего времени существования сетевого соединения.



Таблица 6.4 – Классификация параметров качества сетевого обслуживания, о которых договариваются между собой пользователи NS во время установления соединения

Фаза	Критерий качества	
	Скорость	Правильность/Надёжность
Установление сетевого соединения	Задержка установления сетевого соединения	Вероятность неудачи установления сетевого соединения (неправильное соединение/отказ)
Перенос данных	Пропускная способность	Остаточный коэффициент ошибки (искажение, дублирование/потеря)
	Транзитная задержка	Вероятность неудачи передачи данных
Разрушение сетевого соединения	Задержка разрушения сетевого соединения	Вероятность неудачного разрушения сетевого соединения

Определения понятий для параметров из таблицы 6.4 аналогичны определениям, данным ранее для параметров качества транспортного обслуживания.



Некоторые отличия есть в определении **пропускной способности**. Так для пропускной способности на сетевом уровне определяются её желательное значение (то есть то, которое запланировал использовать пользователь NS) и минимальное значение (это то значение при котором сетевое обслуживание сохраняет свою приемлемость).

Таблица 6.5 – параметры качества сетевого обслуживания, о которых не договариваются между собой пользователи NS во время установления соединения.

Защита сетевого соединения
Приоритет сетевого соединения
Максимальная приемлемая стоимость

Защита сетевого соединения – степень с которой сетевой уровень пытается противостоять угрозам защиты, используя службы безопасности сетевого уровня, уровня звена данных и физического уровня.

Приоритет сетевого соединения – определяет



относительную значимость сетевого соединения, независимо по следующим аспектам:

1. Приоритет получения сетевого соединения;
2. Приоритет удержания сетевого соединения;
3. Приоритет данных сетевого соединения.

1 и 2 пункты вместе определяют порядок, в котором сетевые соединения должны быть нарушены с целью возвращения ресурсов (в случае необходимости), которые они занимают.

3 пункт определяет порядок, в котором сетевые соединения должны ухудшить качество обслуживания.

Максимальная приемлемая стоимость – стоимость определяемая в абсолютных или относительных единицах затрат.

Качество сетевого обслуживания в режиме без установления соединения описывается аналогично качеству транспортного обслуживания в режиме без установления соединения (но есть



отличие, а именно качество сетевого обслуживания обладает параметром максимальной приемлемой стоимости, в отличие от качества транспортного обслуживания).



7. УРОВЕНЬ ЗВЕНА ДАННЫХ

Уровень звена данных обеспечивает связь между сетевыми объектами в режиме без установления соединения и в режиме с установлением соединения. Для связи сетевых объектов в режиме с установлением соединения уровень звена данных обеспечивает установление, поддержку и разъединение соединения уровня звена данных.

Услуги, предоставляемые сетевому уровню в режиме с установлением соединения:

1. Адреса уровня звена данных.

Примером адресации, используемой на уровне звена данных может служить MAC-адресация. MAC-адреса назначаются сетевым адаптерам и сетевым интерфейсам маршрутизаторов (портам маршрутизаторов). MAC-адреса назначаются производителями оборудования и являются уникальными, так



как управляются централизованно. Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байт, например 00 B3 7C 56 78 0A.

Например, в технологии Ethernet, которая соответствует физическому уровню и уровню звена данных для адресации используются MAC-адреса. Например, каждый конечный узел (компьютер) в сети Ethernet имеет свой MAC-адрес. Один сетевой узел может иметь и несколько MAC-адресов, например маршрутизатор имеет несколько MAC-адресов, так как каждому порту маршрутизатора назначается MAC-адрес. Компьютер воспринимает порт маршрутизатора как отдельный сетевой узел.

2. Соединение уровня звена данных.

Соединение уровня звена данных обеспечивает средства передачи данных между сетевыми объектами, идентифицированными адресами звена данных.



3. Обмен SDU звена данных.

Уровень звена данных обеспечивает обмен SDU звена данных через соединение уровня звена данных.

4. Идентификация конечных точек соединения уровня звена данных.

Если необходимо, уровень звена данных обеспечивает данную услугу, которая может использоваться сетевым объектом, чтобы идентифицировать тот сетевой объект с которым он взаимодействует.

5. Уведомление об ошибке.

Об обнаружении ошибки, которая не была исправлена, уровень звена данных уведомляет сетевой объект.

6. Параметры качества обслуживания.

Параметры качества обслуживания могут произвольно выбираться. Уровень звена данных устанавливает и



обеспечивает выбранное качество обслуживания на протяжении соединения уровня звена данных.

7. Сброс соединения уровня звена данных.

Сетевой объект может использовать эту услугу, обратившись к объекту звена данных, который выполнит сброс соединения уровня звена данных.

Услуги, предоставляемые сетевому уровню в режиме без установления соединения:

1. Адреса уровня звена данных.

2. Передача SDU уровня звена данных определённого максимального размера.

3. Параметры качества обслуживания.



Функции уровня звена данных в режиме с установлением соединения:

1. Установление и разъединение соединения уровня звена данных.

Соединения уровня звена данных устанавливаются и разъединяются на активированных физических соединениях. Физическое соединение может иметь множество конечных точек (например, многоточечное соединение – соединение, в котором участвует более двух конечных узлов). В таком случае уровень звена данных выполняет функцию идентификации соединений уровня звена данных, которые используют физические соединения.

Примерами протоколов, работающих в режиме с установлением соединения могут служить протокол LAP-B (работающий на втором уровне технологии X.25) и протокол LAP-F (работающий на втором уровне технологии frame relay).



2. Передача данных.

3. Расщепление звена данных.

Эта функция обеспечивает расщепление одного соединения уровня звена данных на множество физических соединений.

4. Управление порядком следования SDU уровня звена данных.

Эта функция обеспечивает передачу SDU уровня звена данных от источника до получателя в нужной последовательности.

5. Синхронизация.

Например, в Ethernet эта функция выполняется с помощью преамбулы (preamble). Такая преамбула всегда сопровождает кадр Ethernet (PDU в Ethernet называется **кадром**), она состоит из 8 байт. Преамбула нужна для вхождения приёмника в синхронизм с передатчиком.



6. Обнаружение ошибок.

Примером протокола выполняющего данную функцию может послужить протокол доступа к звену (или к каналу) данных LAP-B (Link Access Protocol). Ошибки обнаруживаются с помощью 16-битового поля контрольной последовательности кадров FCS

(Frame Check Sequence), это поле выполняет ту же функцию, что и контрольная сумма. Протокол LAP-B используется на втором уровне модели OSI, который относится к уровню звена данных и выполняет данную функцию является протокол LAP-F, который работает в сети frame relay. В этом протоколе ошибки обнаруживаются с помощью поля CRC (Cyclic Redundancy Check), которое выполняет ту же функцию, что и контрольная сумма.

Протоколы LAP-B и LAP-F произошли от одного протокола под названием HDLC (High-level Data Link Control Procedure).



7. Исправление ошибок.

В X.25, после того как в кадре обнаружена ошибка с помощью протокола LAP-B, этот кадр будет повторно передан, тем самым ошибка будет исправлена (в сети X.25 на каждый пользовательский кадр передаётся квитанция подтверждения).

8. Управление потоком данных.

В качестве примера, выполняющего эту функцию можно привести протокол LAP-F, использующийся в сети frame relay. Каждый абонент сети frame relay заключает соглашение при установлении виртуального соединения, основным параметром этого договора является согласованная информационная скорость CIR (Committed Information Rate). В общем случае пользователь не должен передавать в сеть данные со средней скоростью, превосходящей CIR. Если же он нарушает соглашение, то сеть не только не гарантирует доставку кадра, но помечает этот кадр признаком DE (Discard Eligibility), равным



1, то есть кадр, подлежащий удалению. Однако кадры, отмеченные таким признаком, удаляются из сети только в том случае, если коммутаторы сети испытывают перегрузки. Если же перегрузок нет, то кадры с признаком DE=1 доставляются адресату.

В технологии frame relay определён ещё и дополнительный (необязательный) механизм управления кадрами. Этот механизм оповещения конечных пользователей о том, что в коммутаторах сети возникли перегрузки (переполнение необработанными кадрами). Бит FECN (Forward Explicit Congestion Bit) кадра извещает об этом принимающую сторону. На основании значения этого бита принимающая сторона должна с помощью протоколов более высокого уровня (например TCP/IP) известить передающую сторону о том, что та должна снизить интенсивность отправки пакетов в сеть.

Бит BECN (Backward Explicit Congestion Bit) извещает о переполнении в сети передающую сторону и является



рекомендацией немедленно снизить темп передачи.

9. Отображение SDU звена данных в PDU звена данных.

Эта функция заключается в том, что SDU уровня звена данных помещается в поле данных PDU уровня звена данных.

10. Сброс соединения звена данных.

Эта функция заключается в выполнении услуги сброса соединения уровня звена данных.

11. Коммутация и ретрансляция.

Функция коммутации на втором уровне выполняется с помощью коммутаторов. Коммутатор имеет сетевые интерфейсы – порты, через которые он получает и отправляет кадры. Каждый порт имеет приёмную часть и передающую часть. Функция коммутатора заключается в следующем: когда порт коммутатора принимает кадр, коммутатор должен проанализировать адресную информацию, содержащуюся в



этом кадре и на основе этого принять решение на какой выходной порт отправить данный кадр.

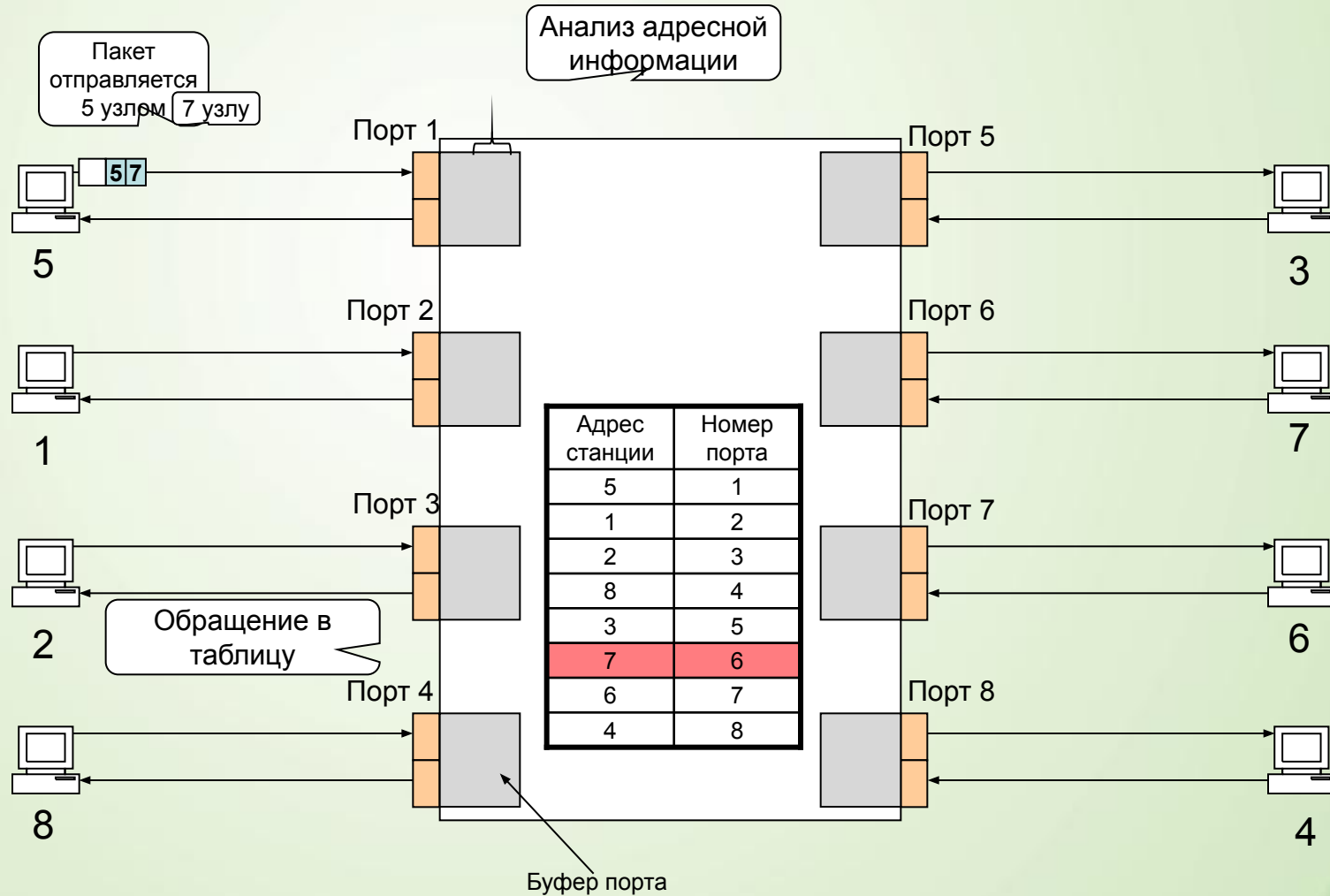


Рисунок 7.1 – Иллюстрация принципа работы коммутатора на втором уровне



Для упрощения на рисунке 7.1 вместо реальных адресов используются короткие адреса. В реальности адресами станций на втором уровне являются локальные адреса, например MAC-адреса.

Функции уровня звена данных в режиме без установления соединения:

- 1. Отображение SDU звена данных на PDU звена данных.**
- 2. Обнаружение ошибок.**
- 3. Коммутация и ретрансляция.**

Кроме функций, которые были перечислены выше можно выделить ещё функции, которые характерны как для режима с установлением так и для режима без установления соединения.



Поддержка топологии сети (функция характерная для локальных вычислительных сетей).

В протоколах уровня звена данных, используемых в локальных сетях, заложена определённая структура связей между компьютерами и способы их адресации. Хотя уровень звена данных и **обеспечивает доставку кадра между любыми двумя узлами локальной сети**, он это делает только в сети с совершенно определённой топологией связи, именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда, а также структуры, полученные из них с помощью коммутаторов.

Обеспечение обмена информацией между смежными узлами сети (функция характерная для глобальных сетей).

В глобальных сетях, которые редко обладают регулярной топологией, уровень звена данных часто обеспечивает обмен



сообщениями только между двумя соседними узлами сети, соединёнными индивидуальной линией связи. Примером глобальной сети, в которой уровень звена данных выполняет рассматриваемую функцию, является телефонная сеть. Примерами протоколов «точка-точка» (как часто называют такие протоколы) могут служить широко распространённые протоколы PPP и LAR-V.

7.1 КЛАССЫ И ТИПЫ ОБСЛУЖИВАНИЯ УРОВНЯ ЗВЕНА ДАННЫХ

Определены два типа обслуживания уровня звена данных:

1. Обслуживание в режиме с установлением соединения.
2. Обслуживание в режиме без установления соединения.

При обращении к услугам звена данных пользователь должен заявить о том, какой тип обслуживания он предполагает использовать.

Нет никаких определённых классов обслуживания уровня звена данных.



7.2 КАЧЕСТВО ОБСЛУЖИВАНИЯ УРОВНЯ ЗВЕНА ДАННЫХ В РЕЖИМЕ С УСТАНОВЛЕНИЕМ СОЕДИНЕНИЯ

Термин качество обслуживания относится к некоторым характеристикам соединения уровня звена данных.

Параметры качества обслуживания уровня звена данных могут быть разделены на два типа:

1. Параметры качества обслуживания, которые определяются в течение установления соединения уровня звена данных.

2. Параметры качества обслуживания, которые не определяются в течение установления соединения уровня звена данных, но значения которых известны.



Таблица 7.1 – Классификация параметров качества обслуживания уровня звена данных, которые относятся к первому типу

Критерий качества	
Скорость	Правильность/Надёжность
Пропускная способность	Остаточный коэффициент ошибки (искажение, дублирование/потеря)
Транзитная задержка	

Определение понятия пропускной способности аналогично определению, которое было дано для сетевого соединения.

Определения понятий транзитной задержки и остаточного коэффициента ошибки аналогичны определениям, данным для транспортного соединения.

Ко второму типу параметров качества обслуживания относятся **защита** и **приоритет**.

Защита – это та степень, с которой уровень звена данных противостоит неправоначальному контролю или манипуляции пользовательскими данными.

Возможны три разновидности защиты:



1. Отсутствие возможностей защиты.
2. Защита против скрытого наблюдения.
3. Защита против модификации, добавления и удаления данных.

Определение понятия приоритета аналогично определению, которое было дано для транспортного соединения.

7.3 КАЧЕСТВО ОБСЛУЖИВАНИЯ УРОВНЯ ЗВЕНА ДАННЫХ В РЕЖИМЕ БЕЗ УСТАНОВЛЕНИЯ СОЕДИНЕНИЯ

Таблица 7.2 – Классификация параметров качества обслуживания уровня звена данных, которые относятся к первому типу.

Критерий качества	
Скорость	Правильность/Надёжность
Транзитная задержка	Остаточный коэффициент ошибки (искажение, дублирование/потеря)

Ко второму типу параметров качества обслуживания относятся **защита** и **приоритет**.



8. ФИЗИЧЕСКИЙ УРОВЕНЬ

Физический уровень (Physical Layer) обеспечивает механические, электрические, функциональные и процедурные средства активации и деактивации физических соединений для побитовой передачи данных между объектами уровня звена данных. Физическое соединение может включать промежуточные открытые системы. Физические объекты связаны посредством физической среды.

Физический уровень имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищённость, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих



дискретную информацию, например, крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизируются типы разъёмов и назначение каждого контакта.

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Например, со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъём RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.



Например, в ОКС7 на физическом уровне обычно используются каналы 64 Кбит/с тракта ИКМ.

Услуги, предоставляемые физическим уровнем:

Услуги физического уровня не классифицированы по режимам соединения, так как эти услуги, определены в соответствии с характеристиками базовой среды и слишком разнообразны.

1. Физические соединения.

Физическое соединение может быть обеспечено посредством взаимодействия **цепей данных**, при использовании функции ретрансляции на физическом уровне (рисунок 8.1, на следующем слайде).

Цепь данных – путь в физической среде OSI между двумя или более физическими объектами. При этом, на протяжении этого пути обеспечивается побитная передача.



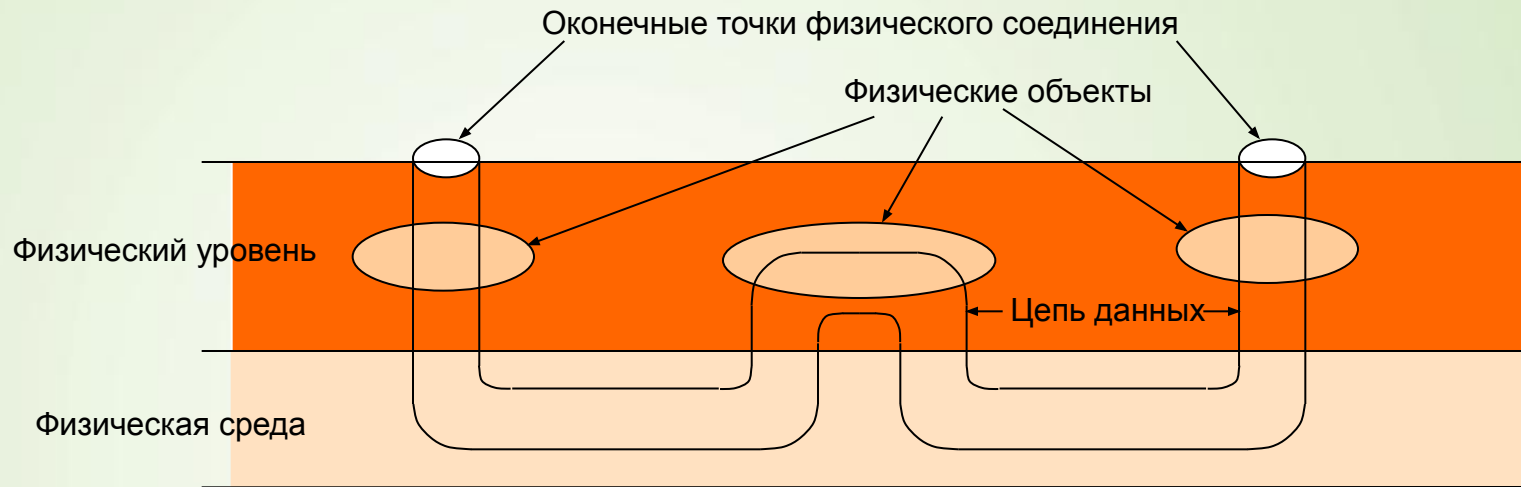


Рисунок 8.1 – Взаимодействие цепей данных на физическом уровне

2. Физические SDU.

Физический SDU состоит из одного бита или последовательности бит.

Физическое соединение может предоставлять дуплексную (передача данных в обоих направлениях одновременно), полудуплексную (передача данных в обоих направлениях, но попеременно во времени) или симплексную (только в одном направлении) передачу потоков битов.



3. Физические оконечные точки соединения.

На физическом уровне оконечные точки соединения имеют идентификаторы, которые могут использоваться объектом уровня звена данных для идентификации этих оконечных точек.

Физическое соединение может иметь тип «точка-точка» либо многоточечный тип (рисунок 8.2 и 8.3 соответственно).

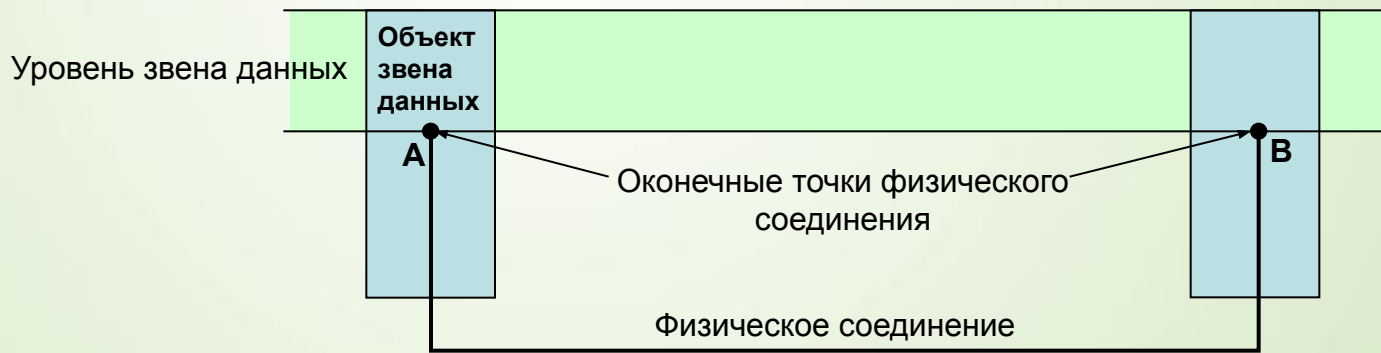


Рисунок 8.2 – Пример физического соединения «точка-точка»



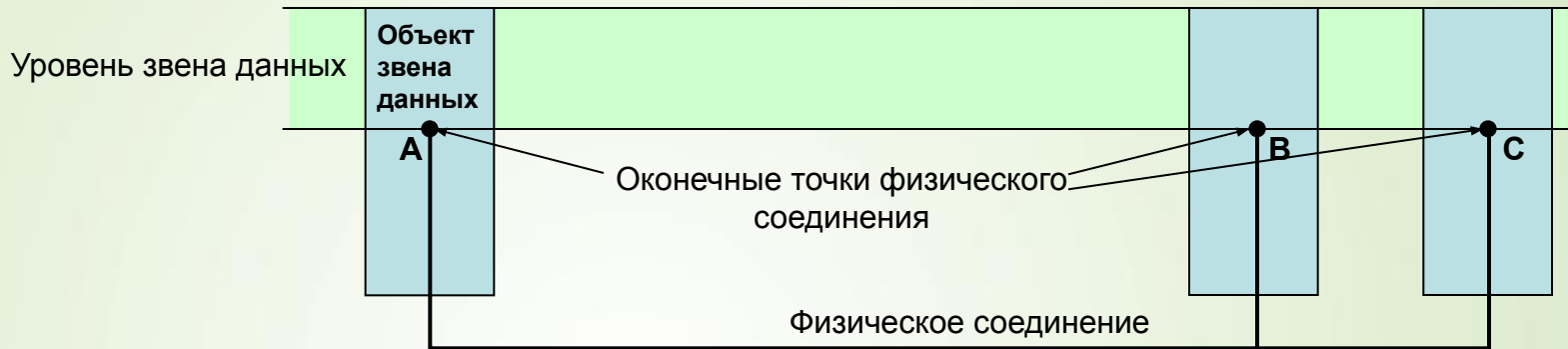


Рисунок 8.3 – Пример многоточечного физического соединения

4. Идентификация цепей данных.

Физический уровень идентифицирует цепи данных между двумя смежными открытыми системами.

5. Упорядочивание.

Физический уровень доставляет биты в том же самом порядке, в котором они отправлялись.

6. Уведомление об аварийном состоянии.

Физический уровень уведомляет объекты уровня звена



данных об авариях, обнаруженных в его пределах.

7. Параметры качества обслуживания.

Качество обслуживания может характеризоваться:

1. Коэффициентом ошибок, которые могут возникать в результате чередования, потери, дублирования и других причин;
2. Скоростью передачи;
3. Транзитной задержкой.

Функции физического уровня:

1. Активация и деактивация физического соединения.

Эта функция обеспечивает активацию и деактивацию физического соединения между двумя объектами уровня звена данных по запросу уровня звена данных. При этом выполняется функция ретрансляции, которая обеспечивает взаимодействие



цепей данных.

2. Передача физических SDU.

Передача физических SDU (то есть битов) может быть синхронной и асинхронной.

3. Мультиплексирование.

Эта функция обеспечивает поддержку двух или более физических соединений одной цепью данных.

Функция мультиплексирования является дополнительной.

8.1 КЛАССЫ ФИЗИЧЕСКОГО ОБСЛУЖИВАНИЯ

На физическом уровне не выделены какие-то определённые классы обслуживания. На физическом уровне определено чем могут различаться классы физического обслуживания. Классы могут различаться по следующим аспектам:



1. Тип передачи: синхронный и асинхронный;
2. Режим работы: дуплексный, полудуплексный, симплексный;
3. Топология: двухточечная или многоточечная;
4. Постоянная или переменная битовая скорость передачи информации.



9. ПРИМЕР СООТВЕТСТВИЯ СТЕКА HTTP/TCP/IP/Ethernet УРОВНЯМ МОДЕЛИ OSI

На следующем слайде проиллюстрировано соответствие стека протоколов HTTP/TCP/IP/Ethernet уровням модели OSI.

HTTP (HyperText Transfer Protocol) – протокол переноса гипертекста.



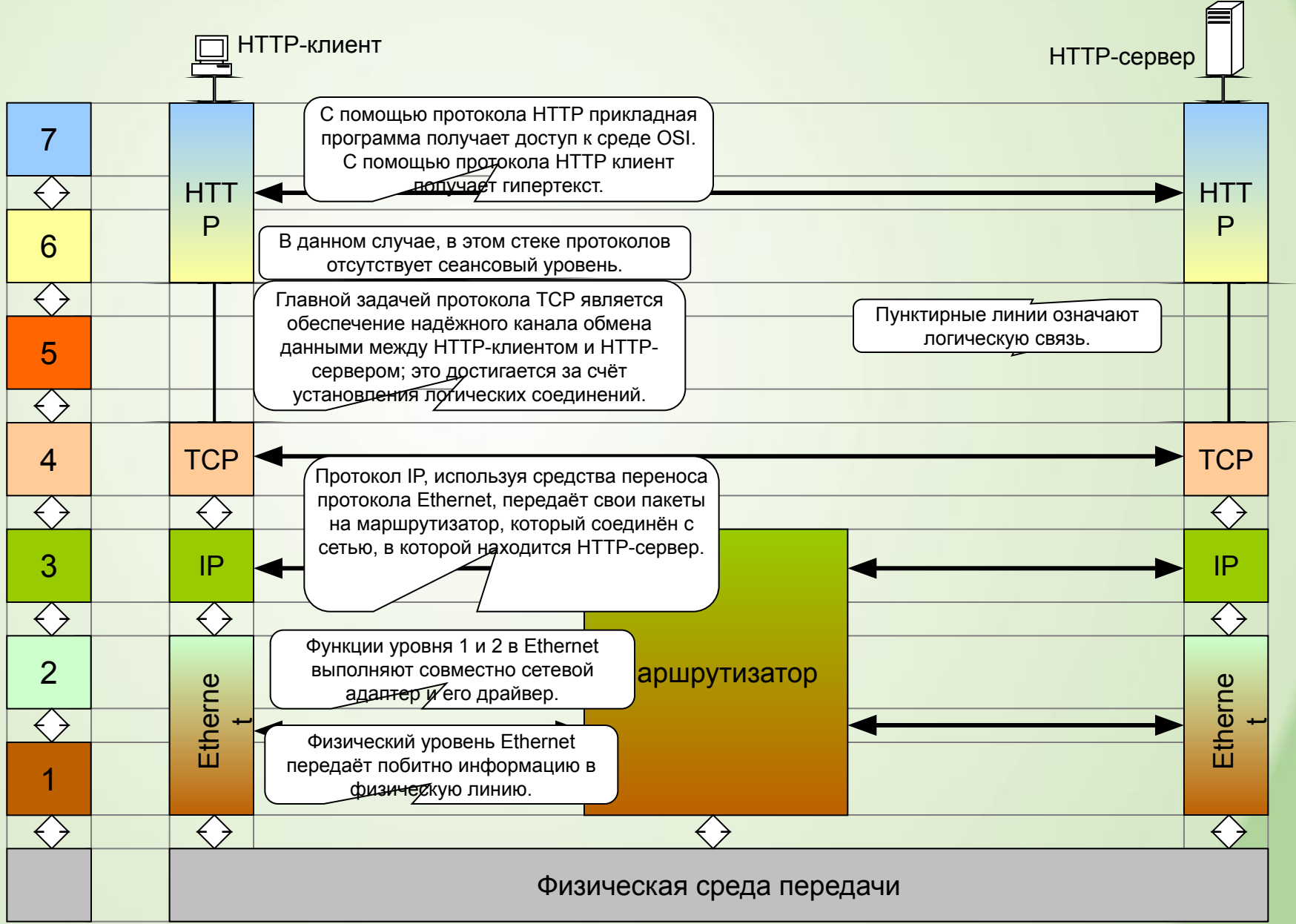


Рисунок 8.4 – Соответствие протоколов стека HTTP/TCP/IP/Ethernet модели OSI