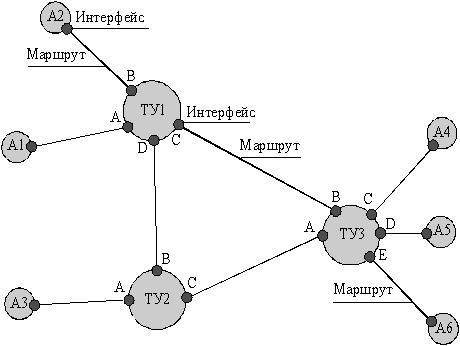
# Основы сетевых технологий

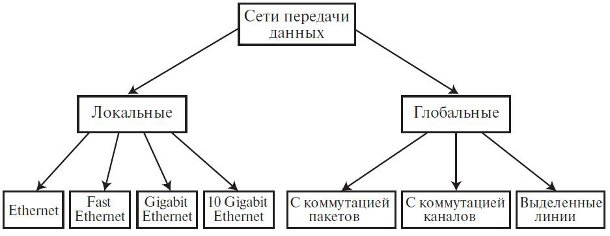
**Телекоммуникационные сети** представляют собой комплекс аппаратных и программных средств, обеспечивающих передачу *информационных сообщений* между абонентами с заданными параметрами качества.

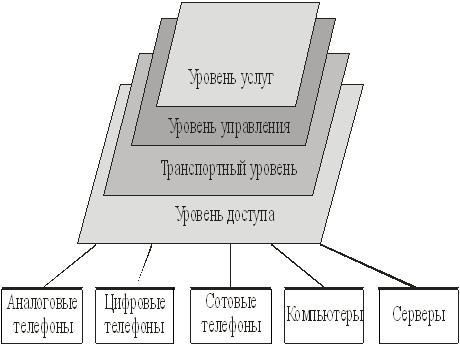


**Сообщение** – форма представления информации, удобная для передачи на *расстояние*. Сообщение отображается изменением какого-либо параметра информационного **сигнала** (электромагнитные сигналы в сетях).

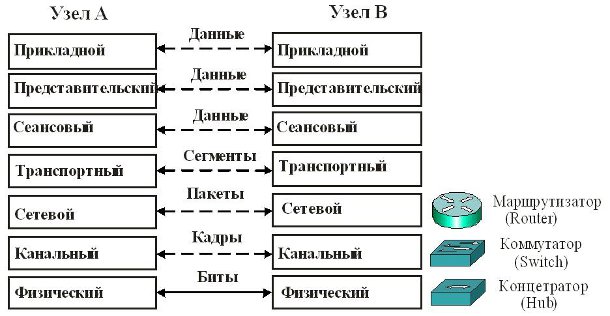
**Коммутацией** также называют передачу (продвижение) сообщения с входного интерфейса на выходной.

Процесс выбора оптимального маршрута получил название **маршрутизация**, а устройство, ее реализующее, – **маршрутизатор**. Выбор оптимального маршрута узлы производят на основе **таблиц маршрутизации** (или коммутации) с использованием определенного критерия – **метрики**.

В настоящее время создаются *сети нового (следующего) поколения* (*Next* *Generation* *Network* – *NGN*), в которых все виды трафика передаются по единой сети связи в цифровой форме. В сетях *NGN* обеспечивается слияние ( *конвергенция* ) всех существующих сетей в единую информационную *сеть* для передачи *мультимедийной* информации. 







Самый верхний, **Прикладной уровень (Application Layer) 7** оперирует наиболее общей единицей данных – сообщением. На этом уровне реализуется управление общим доступом к сети, потоком данных, сетевыми службами, такими, как **FTP, TFTP, HTTP, SMTP, SNMP** и др.

**Представительский уровень (Presentation Layer) 6** изменяет форму представления данных. Например, передаваемые с уровня 7 данные преобразуются в общепринятый формат *ASCII*. При приеме данных происходит *обратный* процесс. На уровне 6 также происходит шифрация и сжатие данных.

**Сеансовый уровень (Session Layer) 5** устанавливает *сеанс* связи двух конечных узлов (компьютеров), определяет, какой *компьютер* является передатчиком, а какой приемником, задает для передающей стороны время передачи.

**Транспортный уровень (Transport Layer) 4** делит большое сообщение узла источника информации на части, при этом добавляет заголовок и формирует **сегменты** определенного объема, а короткие сообщения может объединять в один сегмент. В узле назначения происходит *обратный* процесс. В заголовке сегмента задаются **номера порта** источника и назначения, которые адресуют службы верхнего прикладного уровня для обработки данного сегмента. Кроме того, *транспортный уровень* обеспечивает надежную доставку пакетов. При обнаружении потерь и ошибок на этом уровне формируется *запрос* повторной передачи, при этом используется протокол **TCP**. Когда необходимость проверки правильности доставленного сообщения отсутствует, то используется более простой и быстрый протокол *дейтаграмм*пользователя (*User* *Datagram* *Protocol* – **UDP** ).

**Сетевой уровень (Network Layer) 3** адресует сообщение, задавая единице передаваемых данных **(пакету) логические сетевые адреса** узла назначения и узла источника ( **IP-адреса** ), определяет **маршрут**, по которому будет отправлен **пакет данных**, транслирует логические *сетевые адреса* в физические, а на приемной стороне – *физические адреса* в логические. Сетевые *логические адреса* принадлежат пользователям.

**Канальный уровень (Data Link) 2** формирует из пакетов **кадры** данных (frames). На этом уровне задаются *физические адреса* устройства-отправителя и устройства-получателя данных. Например, **физический адрес** устройства может быть прописан в *ПЗУ* *сетевой карты* компьютера. На этом же уровне к передаваемым данным добавляется *контрольная сумма*, определяемая с помощью алгоритма *циклического кода*. На приемной стороне по *контрольной сумме* определяют и по возможности исправляют ошибки.

# ВОПРОСЫ

1. Что собой представляют *телекоммуникационные сети*?

**Телекоммуникационные сети** представляют собой комплекс аппаратных и программных средств, обеспечивающих передачу *информационных сообщений* между абонентами с заданными параметрами качества. **Сообщение** – форма представления информации, удобная для передачи на *расстояние*. Сообщение отображается изменением какого-либо параметра информационного **сигнала** (электромагнитные сигналы в сетях).

1. Чем отличаются *сети с коммутацией каналов* от *сетей с коммутацией* сообщений?

Различают сети с **коммутацией каналов**, когда телекоммуникационные узлы выполняют функции *коммутаторов*, и с **коммутацией пакетов** (сообщений), когда телекоммуникационные узлы выполняют функции маршрутизаторов. В *сетях с коммутацией каналов* канал создается до *передачи сообщения*.

Эти два вида сетей используются для передачи двух различных видов трафика. *Сети с коммутацией каналов* обычно передают равномерный (потоковый) трафик – например, *телефонные сети*. В сетях передачи данных с *пульсирующим трафиком* применяется *коммутация пакетов* (сообщений), например, в компьютерных сетях.

Различие *коммутации пакетов* или сообщений состоит в том, что сообщение может быть очень большим. Поэтому если в нем обнаруживается ошибка, то повторно нужно передавать все сообщения большого объема. В *сетях с коммутацией пакетов*большое сообщение предварительно разбивается на сравнительно небольшие пакеты (*сегменты*). Поэтому при потере или искажении части сообщения повторно передается только потерянный пакет (сегмент).

1. Какие функции выполняет маршрутизатор?

Основная функция маршрутизатора — чтение заголовков пакетов сетевых протоколов, принимаемых по каждому порту и принятие решения о дальнейшем маршруте следования пакета по его сетевому адресу.

Функции маршрутизатора могут быть разбиты на 3 группы в соответствии с уровнями модели OSI (рис. 11):

* уровень интерфейсов;
* уровень сетевого протокола;
* уровень протокола маршрутизации.

1. Что собой представляет *метрика* *протокола маршрутизации*?
2. В чем различие *коммутации пакетов* или сообщений?

Различие *коммутации пакетов* или сообщений состоит в том, что сообщение может быть очень большим. Поэтому если в нем обнаруживается ошибка, то повторно нужно передавать все сообщения большого объема. В *сетях с коммутацией пакетов*большое сообщение предварительно разбивается на сравнительно небольшие пакеты (*сегменты*). Поэтому при потере или искажении части сообщения повторно передается только потерянный пакет (сегмент).

1. В чем различие между локальными и глобальными сетями передачи данных?

Согласно одной из классификаций *сети передачи данных* подразделяются на **локальные** и **глобальные**. *Сеть* может размещаться на ограниченном пространстве, например, в отдельном здании, в аудитории. При этом она называется *локальной вычислительной сетью* – **ЛВС** (*Local* *Area* *Network* – **LAN** ). Глабольные сети могут быть размещены на достаточно большом расстоянии

1. Каковы основные функции Уровня 1 модели *OSI*?

**Физический уровень (Physical) 1** осуществляет передачу потока битов по соответствующей физической среде (электрический или оптический *кабель*, *радиоканал*) через соответствующий *интерфейс*. На этом уровне производится *кодирование* данных, синхронизация передаваемых битов информации.

1. Каковы основные функции Уровня 2 модели *OSI*?

**Канальный уровень (Data Link) 2** формирует из пакетов **кадры** данных (frames). На этом уровне задаются *физические адреса* устройства-отправителя и устройства-получателя данных. Например, **физический адрес** устройства может быть прописан в *ПЗУ* *сетевой карты* компьютера. На этом же уровне к передаваемым данным добавляется *контрольная сумма*, определяемая с помощью алгоритма *циклического кода*. На приемной стороне по *контрольной сумме* определяют и по возможности исправляют ошибки.

1. Каковы основные функции Уровня 3 модели *OSI*?

**Сетевой уровень (Network Layer) 3** адресует сообщение, задавая единице передаваемых данных **(пакету) логические сетевые адреса** узла назначения и узла источника ( **IP-адреса** ), определяет **маршрут**, по которому будет отправлен **пакет данных**, транслирует логические *сетевые адреса* в физические, а на приемной стороне – *физические адреса* в логические. Сетевые *логические адреса* принадлежат пользователям.

1. Каковы основные функции Уровня 4 модели *OSI*?

**Транспортный уровень (Transport Layer) 4** делит большое сообщение узла источника информации на части, при этом добавляет заголовок и формирует **сегменты** определенного объема, а короткие сообщения может объединять в один сегмент. В узле назначения происходит *обратный* процесс. В заголовке сегмента задаются **номера порта** источника и назначения, которые адресуют службы верхнего прикладного уровня для обработки данного сегмента. Кроме того, *транспортный уровень* обеспечивает надежную доставку пакетов. При обнаружении потерь и ошибок на этом уровне формируется *запрос* повторной передачи, при этом используется протокол **TCP**. Когда необходимость проверки правильности доставленного сообщения отсутствует, то используется более простой и быстрый протокол *дейтаграмм*пользователя (*User* *Datagram* *Protocol* – **UDP**).

1. Каковы основные функции Уровня 5 модели *OSI*?

**Сеансовый уровень (Session Layer) 5** устанавливает *сеанс* связи двух конечных узлов (компьютеров), определяет, какой *компьютер* является передатчиком, а какой приемником, задает для передающей стороны время передачи.

1. Каковы основные функции Уровня 6 модели *OSI*?

**Представительский уровень (Presentation Layer) 6** изменяет форму представления данных. Например, передаваемые с уровня 7 данные преобразуются в общепринятый формат *ASCII*. При приеме данных происходит *обратный* процесс. На уровне 6 также происходит шифрация и сжатие данных.

1. Каковы основные функции Уровня 7 модели *OSI*?

Самый верхний, **Прикладной уровень (Application Layer) 7** оперирует наиболее общей единицей данных – сообщением. На этом уровне реализуется управление общим доступом к сети, потоком данных, сетевыми службами, такими, как **FTP, TFTP, HTTP, SMTP, SNMP** и др.

1. Что собой представляет *инкапсуляция данных*?

Обрамление единиц информации заголовками со служебной информацией называется *инкапсуляцией*.

Передаваемое сообщение, сформированное приложением, проходит три верхних сетенезависимых уровня и поступает на транспортный уровень, где делится на части и каждая часть инкапсулируется (помещается) в сегмент данных ( рис. 1.7). В заголовке сегмента содержится номер протокола прикладного уровня, с помощью которого подготовлено сообщение, и номер протокола, который будет обрабатывать данный сегмент.

1. Какие устройства функционируют на Уровне 3 модели *OSI*?

См. Определение уровня 3

1. Какие устройства функционируют на Уровне 2 модели *OSI*?
2. См. Определение уровня 2
3. Какие устройства функционируют на Уровне 1 модели *OSI*?

См. Определение уровня 1

1. Перечислите уровни модели TCP/IP.



1. Какие три системы адресации используются в *сетевых технологиях*?

Тройная система адресации (*логические адреса*, *физические адреса*, *номера портов*) позволяет адресовать устройства, пользователей и программное обеспечение приложений.

1. На каком уровне модели *OSI* задаются IP-адреса?

IP-адреса задаются на сетевом уровне (Network Layer) 3 модели OSI

**2.1. Прикладной уровень**

1. Каковы две формы программных средств прикладного уровня?

*Прикладной уровень* представляет собой комплекс программных средств, представленных в двух формах: приложений и служб сервиса.

П**риложения** прикладного уровня обеспечивают *интерфейс* (сопряжение) человека с сетью. **Службы сервиса**используют *программные средства* протоколов, чтобы подготовить информацию для передачи по сети.

1. Где находятся основные ресурсы сети модели "клиент-сервер"?

Согласно **модели "клиент-сервер"** клиент запрашивает информацию, пересылая *запрос* *выделенному серверу* (*upload*), который в ответ на *запрос* посылает (*download*) *файл*, принимаемый клиентом. Следовательно, клиент инициирует процесс обмена информацией в среде "клиент-*сервер*" и получает от сервера требуемую информацию. Главным достоинством модели "клиент-*сервер*" является централизация управления сетью и обеспечение безопасности.

1. Где находятся основные ресурсы сети модели "*peer*-to-*peer*"?

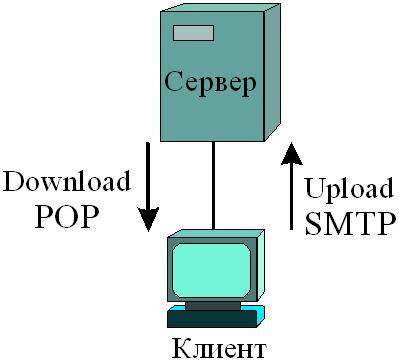
В **сети peer-to-peer** связанные через *сеть* *конечные* узлы разделяют общие ресурсы (принтеры, файлы) без *выделенного сервера*. Каждое конечное устройство (*peer*) может функционировать либо как *сервер*, либо как клиент. *Компьютер* может выполнять роль сервера для одного соединения и роль клиента для другого.

1. Назовите протоколы передачи электронной почты.

Протоколы электронной почты (Simple Mail Transfer Protocol – SMTP, *Post Office Protocol* – *POP*, *Internet Messaging Access Protocol* – *IMAP*);

При передаче электронной почты и взаимодействии *почтовых серверов* между собой используется простой протокол передачи почты (Simple Mail Transfer Protocol – **SMTP** ), у которого номер порта 25. Для получения клиентом сообщения с сервера используется протокол почтового отделения (*Post Office Protocol* – **POP** ) с номером порта 110 или протокол доступа к сообщениям (*Internet Messaging Access Protocol* – **IMAP** ).

На рисунке приведена модель 'клиент-сервер' в службе электронной почты. При пересылке почты от клиента на сервер используется протокол SMTP, при этом происходит процесс *upload*.



1. Какие функции выполняет протокол HTTP?

Его основным приложением является веб-браузер, который отображает данные на веб-страницах, используя текст, графику, звук и видео. Веб- страницы создаются с применением *языка разметки* *гипертекста* *Hypertext* *Markup Language* (HTML), который определяет местоположения для размещения текста, файлов и объектов, которые должны быть переданы от сервера по сети до веб-браузера. *Номер порта протокола HTTP* – 80, функционирует совместно с протоколом транспортного уровня TCP.

1. В чем различие между протоколами HTTP и HTTPS?

Протокол HTTP характеризуется сравнительно невысоким уровнем безопасности, поскольку передаваемые по сети сообщения не зашифрованы. Для повышения уровня безопасности передачи сообщений через Интернет был разработан протокол HTTP Secure ( **HTTPS** ). В этом протоколе используется процесс криптографирования данных ( *encryption* ) и аутентификации ( *authentication* ), что существенно повышает уровень безопасности. *Номер порта протокола HTTPS* – 443.

1. В чем различие между протоколом FTP и HTTP?

http создан для передачи гипертекста, а ftp для передачи файлов

1. Для чего используется система доменных имен DNS?

**Система доменных имен (Domain Name System – DNS)** используется в Интернете для того, чтобы переводить имена сайтов или доменов в числовые значения IP-адреса. Людям легче запомнить *доменное имя*, например, [http://www.cisco.com](http://www.cisco.com/), чем числовой адрес 198.133.219.25. Кроме того, числовые адреса могут со временем меняться. Например, в настоящее время указанный выше числовой адрес сайта [http://www.cisco.com](http://www.cisco.com/) изменен на 72.163.4.161. Поскольку в ряде случаев требуется знание числового адреса, хост может обратиться к DNS-серверу и по имени получить соответствующий адрес. DNS использует распределенный набор серверов разного уровня иерархии, чтобы получить соответствие между именем и числовым адресом

1. По какой команде можно получить адрес *DNS-сервера*?

Операционные системы компьютеров содержат утилиту **nslookup**, которая позволяет пользователю вручную запрашивать имя сервера и идентифицировать название хоста. Когда клиент делает запрос, локальный сервер сначала проверяет собственные записи. Если соответствующих пар "имя-адрес" у него нет, то он связывается с другими серверами DNS более высокого уровня иерархии.

Служба прикладного уровня DNS характеризуется *номером порта* 53 и взаимодействует как с протоколом транспортного уровня TCP, так и с *протоколом UDP*.

1. Какие протоколы обеспечивают виртуальное соединение пользователя с удаленными сетевыми устройствами?

**Протокол Telnet** обеспечивает виртуальное соединение пользователя с удаленными сетевыми устройствами: компьютерами, маршрутизаторами, коммутаторами

Чтобы осуществить подключение клиента по протоколу *Telnet*, обычно задают имя удаленного хоста. В качестве имени хоста используется IP-адрес или имя доменной системы DNS удаленного устройства. Вся обработка информации и использование памяти производится на процессоре удаленного устройства, а отображение результатов конфигурирования протокол *Telnet* транслирует на монитор пользователя. *Telnet* работает на прикладном уровне модели TCP/IP, поэтому охватывает все уровни модели *OSI*. *Номер порта* – 23.

Протокол *Telnet*  поддерживает аутентификацию, поэтому на удаленном устройстве задается пароль, который должен знать пользователь. Однако *Telnet* не поддерживает криптографирование данных, которые передаются по сети как простой текст. Это означает, что данные могут быть перехвачены. Для защиты передаваемой информации разработан протокол *SSH* (Secure Shell). Он обеспечивает криптографирование данных и более надежную аутентификацию, *номер порта* – 22. Протокол *SSH* заменяет *Telnet*.

1. Какой протокол обеспечивает динамическое назначение адресов узлов?

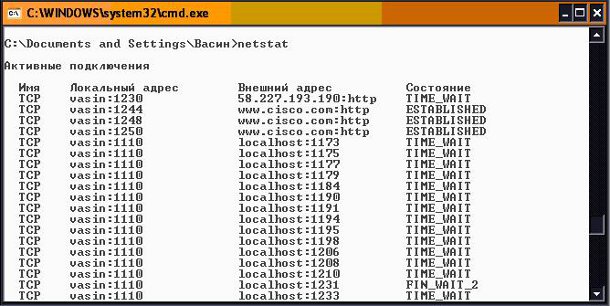
**Протокол динамического назначения адресов узлов (Dynamic Host Configuration Protocol – DHCP)** позволяет автоматизировать процесс назначения IP-адресов рабочим станциям из диапазона, предоставленного администратору провайдером. Динамическое назначение адресов протоколом *DHCP* производится по запросу клиента на определенный промежуток времени, для продления которого пользователь должен периодически обращаться к серверу. При освобождении IP-адреса возвращаются *DHCP*-серверу, который перераспределяет их. При повторном запросе клиента, освободившего IP-адрес, сервер пытается назначить ранее использовавшийся адрес. Помимо IP-адреса протокол *DHCP* предоставляет пользователю еще целый ряд параметров (*маску подсети*, шлюз по умолчанию, IP-адрес *сервера DNS* и др.)

1. В чем различие между протоколами TCP и *UDP*?

Наиболее известными протоколами транспортного уровня являются протокол контроля передачи (*Transmission* *Control* *Protocol* – **TCP** ) и протокол *дейтаграмм* пользователя (*User* *Datagram* *Protocol* – **UDP** ). Протокол контроля передачи *TCP* является ориентированным на предварительное соединение типа *connection-oriented*. Помимо деления сообщения на *сегменты* и идентификации приложений *TCP*обеспечивает *контроль потока и надежность*. Он взаимодействует с протоколами прикладного уровня: *HTTP*, *SMTP*, *FTP*, *Telnet* и другими. *Протокол UDP* является протоколом дейтаграммного типа *connectionless*, взаимодействует с такими протоколами прикладного уровня, как *система доменных имен* – *DNS*, передачи потока видеоданных – *Video Steaming*, голос поверх *IP* – *Voice over* *IP* и рядом других. Следует отметить, что система *DNS* взаимодействует как с *TCP*, так и с *UDP*.

1. По какой команде можно узнать, какие TCP-соединения активны на сетевом конечном узле?

Если необходимо узнать, какие *TCP*-соединения активны на сетевом конечном узле, то можно использовать команду netstat в режиме командной строки. В распечатке команды указаны: *протокол (TCP*), локальный *адрес* узла с динамически назначенным номером порта, внешний *адрес* (или *имя) узла* назначения с номером порта, а также состояние связи.



Результат выполнения команды netstat

В данном примере *номер порта* *локального адреса* является динамически назначаемым зарегистрированным портом источника с номером больше 1023. Для адреса [http://www.cisco.com](http://www.cisco.com/) внешний *порт* задан символически: *http*. Состояние связи может быть с установленным соединением (ESTABLISHED) или с ожиданием окончания соединения (TIME\_WAIT), когда был послан *запрос* окончания соединения (FIN).

1. Какую функцию в формате заголовка сегмента TCP выполняет *номер последовательности*?

**Номер последовательности (Sequence Number)** – 32 бита номера первого байта в сегменте, используемого, чтобы гарантировать объединение частей (порций) данных в корректном порядке в устройстве назначения;

Поскольку *UDP* является протоколом дейтаграммного типа, то в заголовке его сегмента отсутствуют такие параметры, как Номер последовательности, Номер подтверждения, Размер окна.



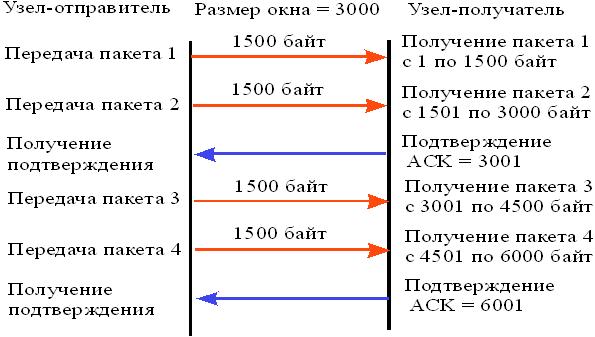
1. Какую функцию в формате заголовка сегмента TCP выполняет подтверждение?

После получения каждой порции данных узел назначения посылает источнику квитанцию **подтверждения (acknowledgment)**. Подтверждение (квитирование) обеспечивает **надежность** *сети передачи данных*. Если подтверждение не получено, то неподтвержденная порция данных передается узлом-источником повторно.

1. Какую функцию в формате заголовка сегмента TCP выполняет размер *скользящего окна*?

**размер скользящего окна (Window)** – определяет число байт, передаваемых за одну порцию;

**Размер скользящего окна (Window)** заголовка сегмента TCP определяет, сколько байт данных передается в одной порции неподтвержденных данных. Последовательность сегментов передаваемых данных представляет собой последовательность байт. Поэтому и размер окна в заголовке сегмента задается в количестве передаваемых байт. Узелполучатель передает отправителю подтверждение *ACK*, когда примет указанное в окне количество байт данных.



1. Какую функцию в формате заголовка сегмента TCP выполняет номер порта?

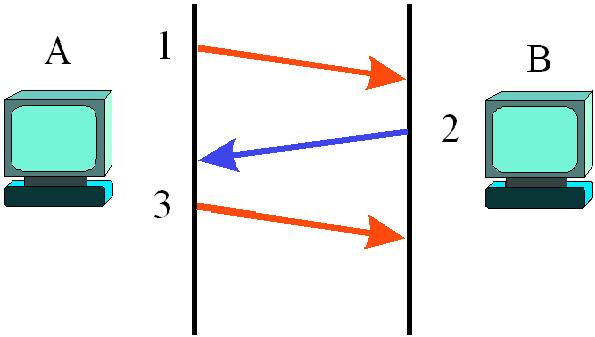
В *протоколах TCP* и *UDP* в качестве **идентификатора приложения используют номер порта**. Номер порта в заголовке сегмента транспортного уровня указывает, какое приложение создало передаваемое сообщение и какое должно обрабатывать полученные данные на приемной стороне. При множестве одновременно протекающих обменов данными каждому из приложений или услуг назначается свой **адрес (номер порта)** так, чтобы транспортный уровень мог определить, с каким конкретно приложением или службой передаваемые данные должны взаимодействовать.

1. За сколько этапов выполняется предварительное установление соединения у *протокола TCP*?

Во-первых, узел-отправитель инициализирует установление связи, посылая узлу-получателю запроса синхронизации *SYN* (1).

Во-вторых, узел-получатель подтверждает запрос синхронизации и задает свои параметры синхронизации *ACK* (2).

В-третьих, узлу-получателю посылается подтверждение, что обе стороны готовы к тому, чтобы соединение было установлено (3).



1. Чем определяется размер поля данных сегмента? (НЕ ТОЧНЫЙ)

Сегменты данных нужно представить пользователю-получателю в том же порядке, в котором они были переданы. Сбой происходит, если какие-то сегменты данных потеряны, повреждены или получены в неверном порядке. Поэтому получатель должен подтвердить получение каждого сегмента. Однако если бы отправитель ждал ответ *ACK* после посылки каждого сегмента, то производительность сети была бы низкой. Поэтому надежный, ориентированный на предварительное соединение протокол, например TCP, позволяет послать несколько сегментов прежде, чем отправитель получит подтверждение *ACK*.