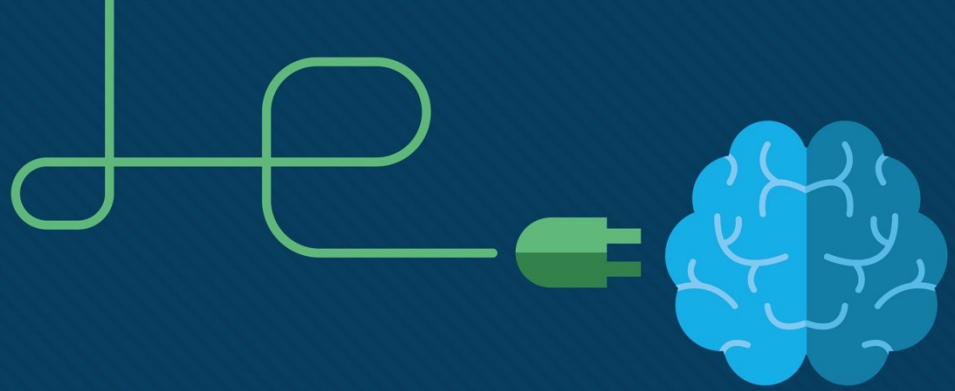


Модуль 14: Транспортный уровень

Введение в сетевые технологии v7.0 (ITN)

Задачи модуля

Название модуля: Транспортный уровень

Цель модуля: Сравнить операции протоколов транспортного уровня при поддержке сквозного канала связи.

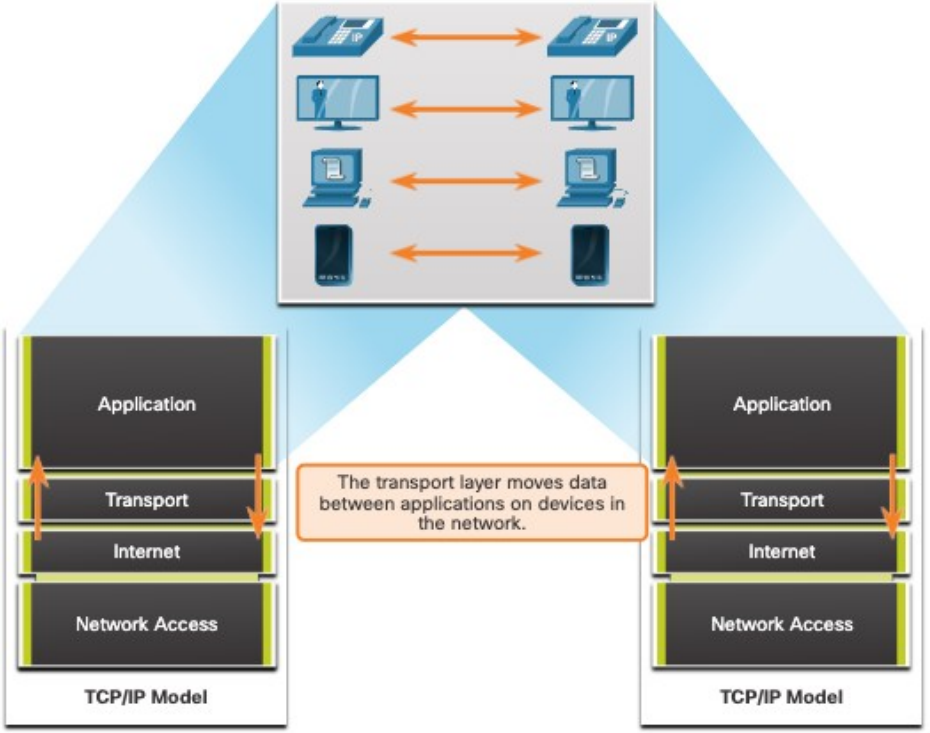
Заголовок темы	Цель темы
Передача данных	Объяснить назначение транспортного уровня в процессе передачи данных по сквозному каналу.
Обзор протокола TCP	Объяснить характеристики TCP.
Обзор протокола UDP	Объясните характеристики UDP.
Номера портов	Объясните, как TCP и UDP используют номера портов.
Обмен данными по протоколу TCP	Объяснить, каким образом процессы установления и завершения сеанса TCP обеспечивают надежный обмен данными.
Надежность и управление потоком передачи данных	Объяснить, каким образом передаются блоки данных протокола TCP и как подтверждается их гарантированная доставка.
Обмен данными по протоколу UDP	Сравнить операции протоколов транспортного уровня при поддержке сквозного канала связи.

14.1 Передача данных

Роль транспортного уровня

Транспортный уровень

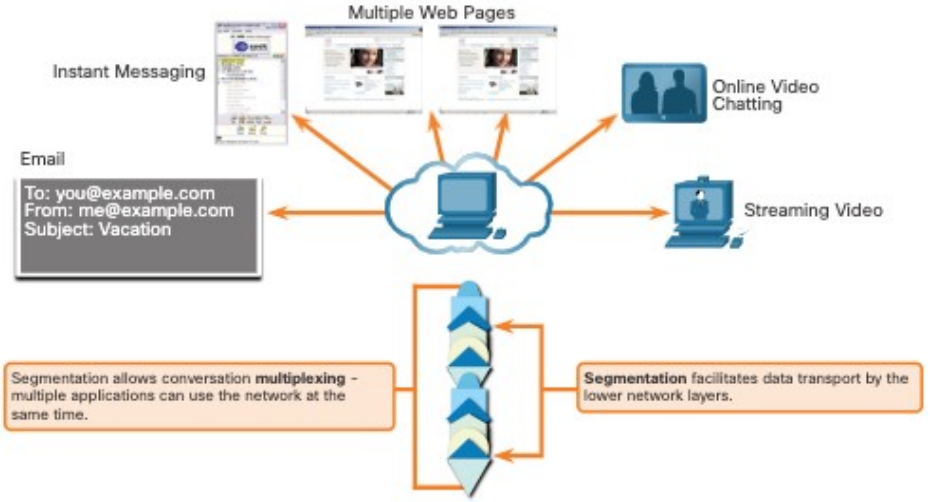
- Отвечает за логические связи между приложениями, работающими на разных хостах.
- Связь между уровнем приложений и нижними уровнями, которые отвечают за передачу данных по сети.



Функции транспортного уровня

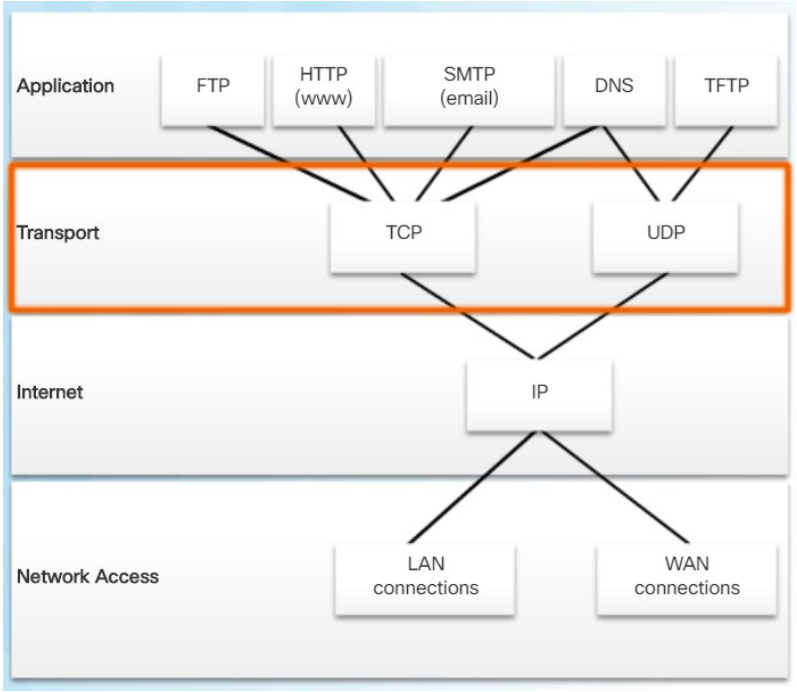
Транспортный уровень имеет несколько функций.

- Отслеживание отдельных сеансов связи
- Сегментация данных и последующая сборка сегментов
- Добавление информации заголовка
- Определение, разделение и управление несколькими сеансами связи
- Использует сегментацию и мультиплексирование для того, чтобы различные сеансы связи чередовались в одной сети



Протоколы транспортного уровня

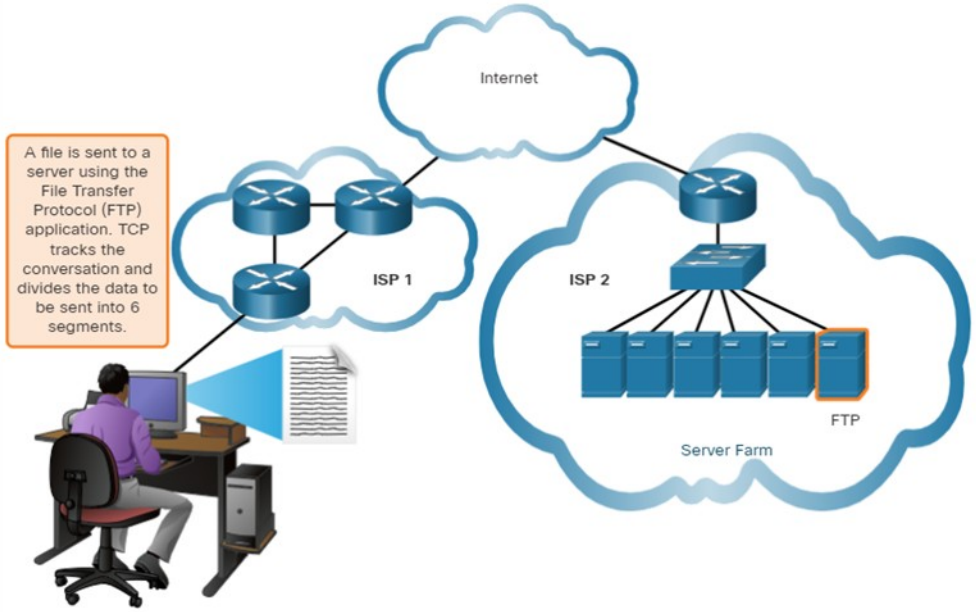
- Он не определяет способ доставки или передачи пакетов.
- Протоколы транспортного уровня определяют способ передачи сообщений между узлами и отвечают за управление требованиями надежности разговора.
- На транспортном уровне действуют два протокола — TCP и UDP.



Протокол управления передачей

TCP обеспечивает надежность и управление потоком. Основные операции TCP:

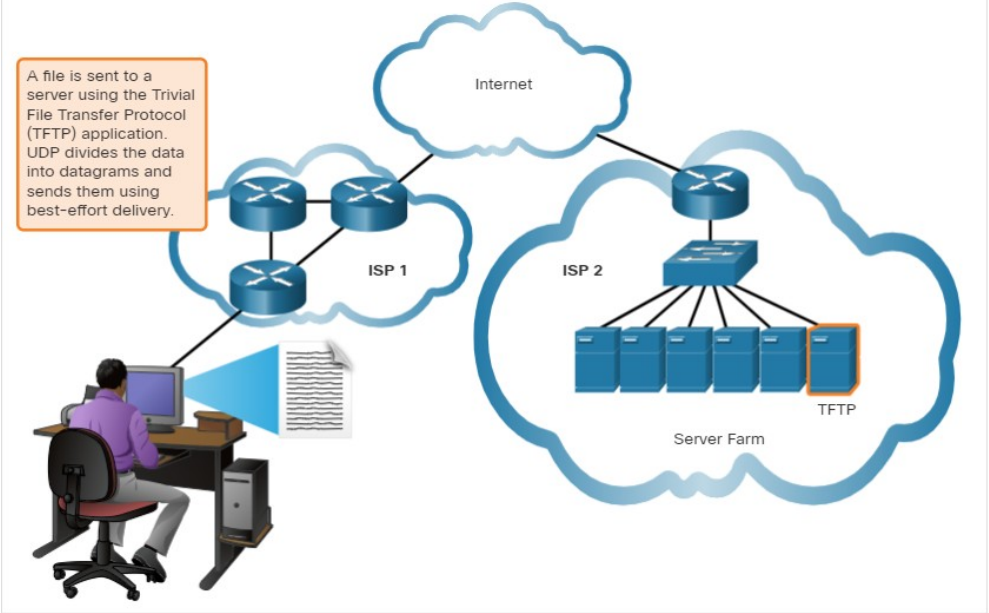
- Отслеживание количества сегментов, отправленных на хост приложением
- Подтверждение полученных данных
- Повторная передача сегментов с неподтвержденными данными по истечении времени ожидания.
- Восстановление последовательности данных, которые могут поступать в неправильном порядке
- Отправка данных с эффективной скоростью, приемлемой получателем



Протокол Пользовательских датаграмм (UDP)

Он обеспечивает только основные функции для обмена сегментами данных между приложениями, при этом данный протокол отличается незначительными накладными расходами и практически отсутствием проверки данных.

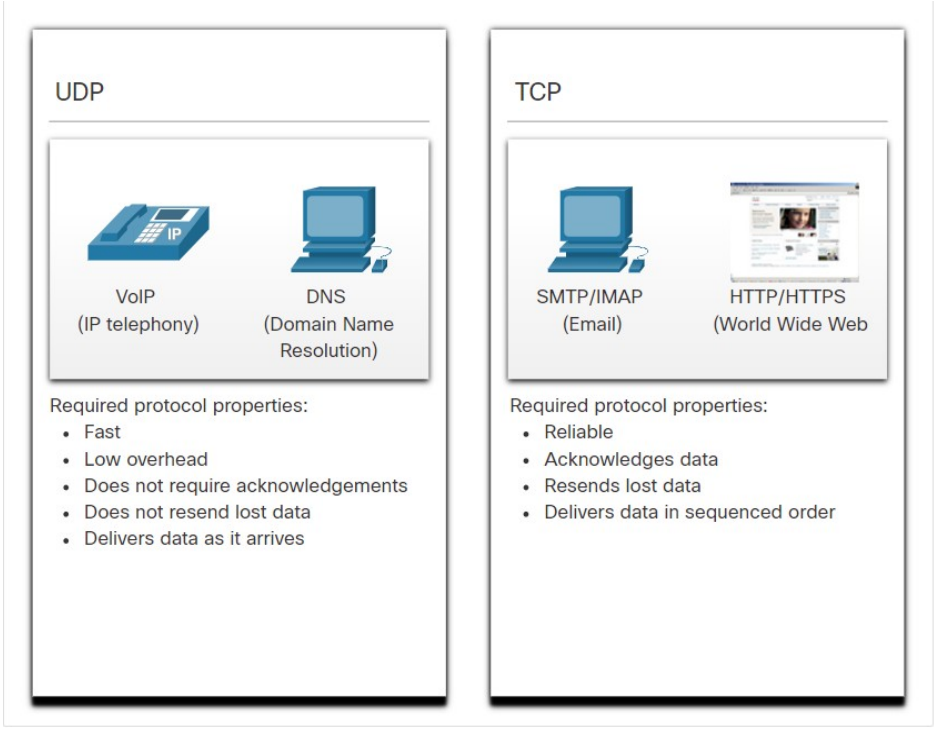
- UDP — протокол транспортного уровня без установки соединения.
- В UDP нет подтверждения того, что данные получены в месте назначения.



Соответствующий протокол транспортного уровня для соответствующего приложения

UDP также используется приложениями запросов и ответов, где данные минимальны, и повторная передача может быть выполнена быстро.

Если важно, чтобы все данные поступали и чтобы они могли быть обработаны в правильной последовательности, TCP используется в качестве транспортного протокола.



14.2 Обзор протокола TCP

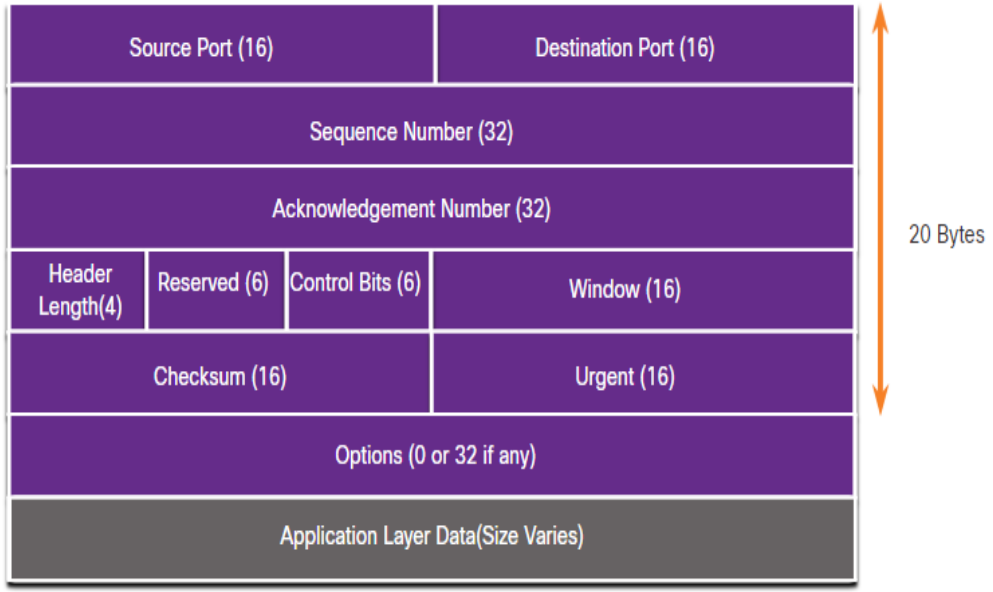
Функции протокола TCP

- **Установление сессии** - Перед пересылкой любого трафика протокол с установлением соединения согласовывает и настраивает постоянное соединение (или сеанс) между устройством источника и устройством назначения.
- **Гарантия надежной доставки** – При передаче по сети один из сегментов может быть поврежден или полностью утрачен. TCP - Обеспечивает гарантированную доставку на узел назначения всех без исключения сегментов данных, отправленных источником
- **Обеспечение доставки в нужном порядке** - Поскольку в сетях могут использоваться несколько маршрутов с разными скоростями передачи информации, в процессе доставки данных их порядок может измениться.
- **Управление потоком передачи данных** - Ресурсы сетевых узлов, такие как память или вычислительные мощности, ограничены. Когда протокол TCP получает информацию о том, что эти ресурсы используются слишком активно, он может потребовать от отправляющего приложения снизить скорость потока данных.

Заголовок протокола TCP

TCP - Протокол, который отслеживает состояние сеанса передачи данных.

Для отслеживания состояния сеанса связи протокол TCP фиксирует, какую информацию он отправил и какая информация была подтверждена.

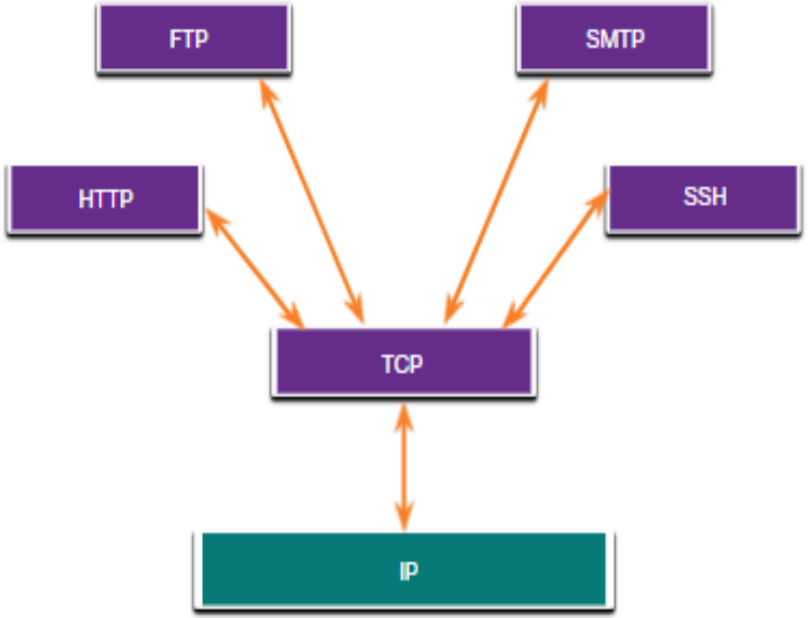


Поля заголовка TCP

Поле заголовка TCP	Описание
Порт источника	16-битное поле, используемое для идентификации исходного приложения по номеру порта.
Порт назначения	16-битное поле, используемое для идентификации приложения назначения по номеру порта.
Порядковый номер	32-битное поле, используемое для пересборки данных.
Номер подтверждения	32-битное поле, используемое для указания того, что данные получены и ожидается следующий байт от источника.
Длина заголовка	4-битное поле, известное как «смещение данных», которое указывает длину заголовка сегмента TCP.
Резерв	6-битное поле зарезервировано для использования в будущем.
Управляющие биты	Биты управления (6 бит) — включает двоичные коды или флаги, которые указывают назначение и функцию сегмента TCP.
Размер окна	16-битное поле, используемое для указания количества байтов, которые могут быть приняты
Контрольная сумма	16-битное поле, используемое для проверки ошибок заголовка и данных датаграммы.
Срочно	16-битное поле, используемое для указания срочности содержащихся данных.

Приложения, использующие протокол TCP

TCP сам выполняет все задачи, связанные с разбиением потока данных на сегменты, обеспечением надежности их передачи, управлением потоком и переупорядочением сегментов.



14.3 Обзор UDP

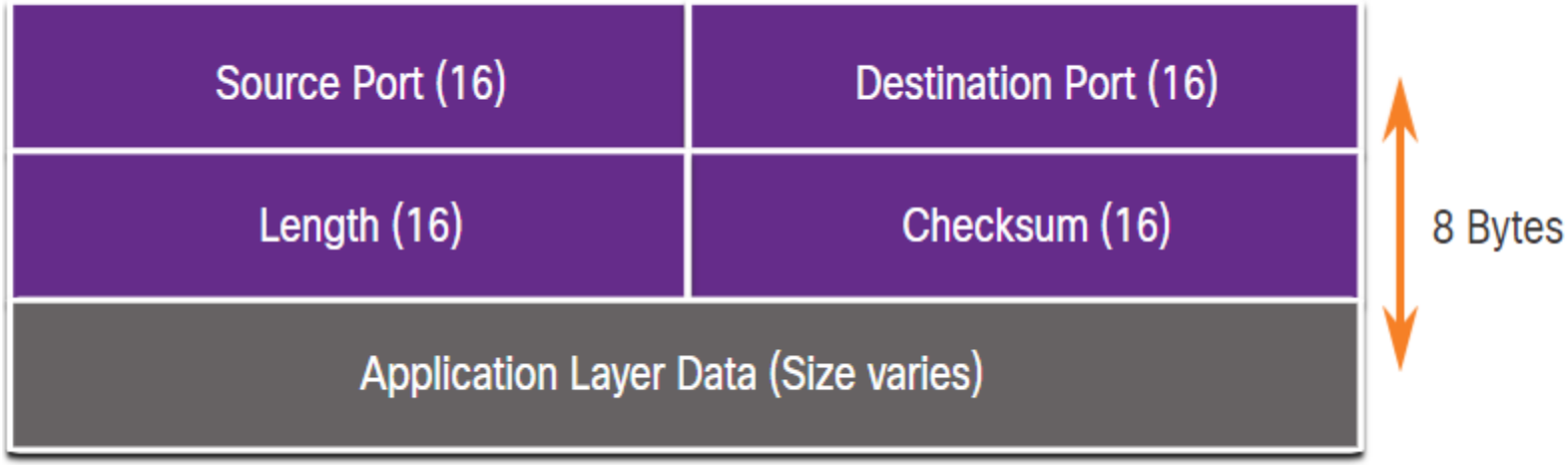
Функции протокола UDP

UDP имеет следующие функции:

- Данные восстанавливаются в том порядке, в котором получены.
- Потерянные сегменты повторно не отправляются.
- Без установления сеанса связи.
- Без уведомления отправителя о доступности ресурса.

Заголовок протокола UDP

Заголовок UDP намного проще, чем заголовок TCP, потому что он имеет только четыре поля и требует 8 байт (т.е. 64 бит).



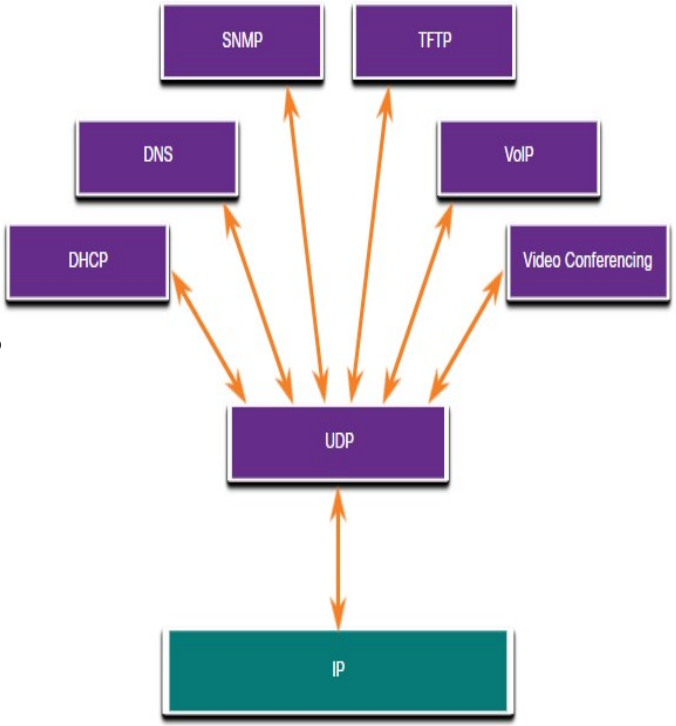
Заголовок протокола UDP

Таблица идентифицирует и описывает четыре поля в заголовке UDP.

Поле заголовка UDP	Описание
Порт источника	16-битное поле, используемое для идентификации исходного приложения по номеру порта.
Порт назначения	16-битное поле, используемое для идентификации приложения назначения по номеру порта.
Длина	16-битное поле, указывающее длину заголовка датаграммы UDP.
Контрольная сумма	16-битное поле, используемое для проверки ошибок заголовка и данных датаграммы.

Приложения, использующие протокол UDP

- Мультимедийные приложения и передача видео в режиме реального времени. Такие приложения допускают небольшие потери данных, но не допускают задержки (либо минимальные). Например, VoIP и потоковое видео.
- Простые приложения запросов и ответов. Приложения с операциями, где хост отправляет запрос и может получить или не получить ответ. Например, DNS и DHCP.
- Приложения, самостоятельно обеспечивающие надежность передачи данных, — ненаправленный обмен данными, при котором управление потоком, обнаружение ошибок, отправка подтверждений и восстановление после сбоев не требуются или выполняются самим приложением. Например, SNMP и TFTP.



14.4 Номера портов

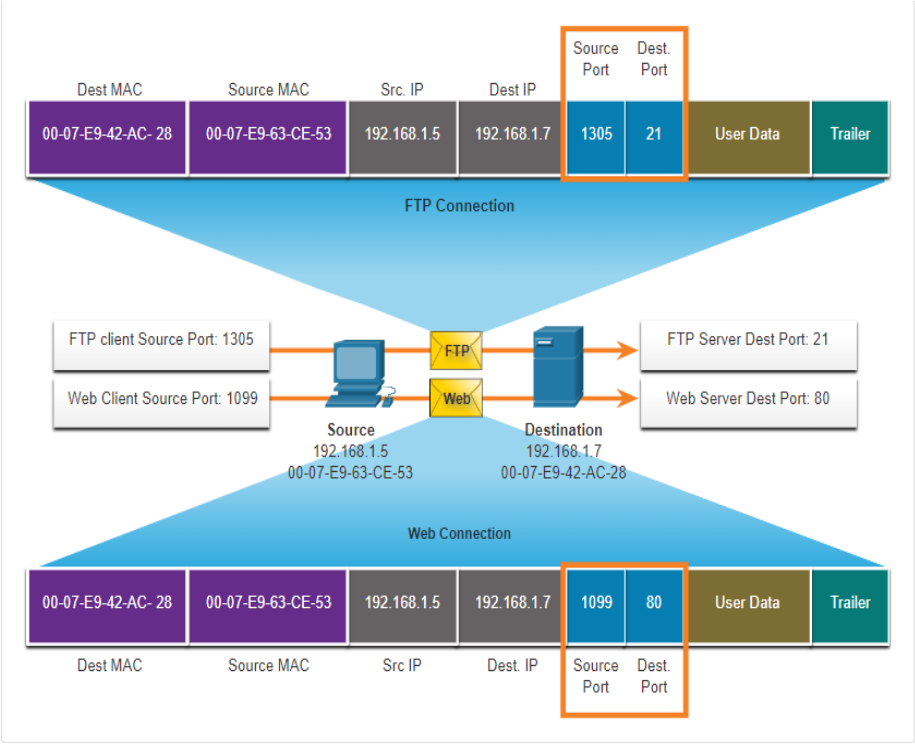
Несколько отдельных сеансов передачи данных

Протоколы транспортного уровня TCP и UDP используют номера портов для управления несколькими одновременными диалогами.

Номер порта источника связан с исходным приложением на локальном узле, тогда как номер порта назначения связан с целевым приложением на удаленном узле.

Пары сокетов

- Номера порта источника и порта назначения записываются в сегмент.
- Затем эти сегменты инкапсулируются в пакете IP.
- Сочетание IP-адреса источника и номера порта источника или IP-адреса назначения и номера порта назначения называется сокетом.
- Сокеты позволяют различать несколько процессов, выполняющихся на клиенте, а также распознавать различные подключения к процессу сервера.



Группы номеров портов

Группа портов	Диапазон номеров портов	Описание
Общеизвестные порты	**От 0 до 1023**	• Они обычно используются приложениями, такими как веб-браузеры и почтовые клиенты, а также клиентами удаленного доступа. • Определенные хорошо известные порты для общих серверных приложений позволяют клиентам легко определить требуемую службу.
Зарегистрированные порты	**От 1024 до 49151**	• IANA по запросу организаций присваивает данные порты для каких-либо специфичных процессов или приложений. • Эти процессы в основном представляют собой отдельные приложения, которые пользователь решил установить, а не широко распространенные приложения, которым обычно присваивают общеизвестные номера портов. • Например, Cisco зарегистрировал порт 1812 для процесса аутентификации сервера RADIUS.
Частные и/или **динамические порты**	**От 49152 до 65535**	• Эти порты также известны как *эфемерные порты*. • Операционная система клиента обычно присваивает номера портов динамически при инициировании подключения к службе. • После чего такой порт используется для определения клиентского приложения во время обмена данными.

Группы номеров портов (продолжение)

Номер порта	Протокол	Применение
20	TCP	File Transfer Protocol (FTP) - Передача данных
21	TCP	File Transfer Protocol (FTP) - Управление передачей
22	TCP	Протокол Secure Shell (SSH)
23	TCP	Программа Telnet
25	TCP	Протокол SMTP
53	UDP, TCP	Служба доменных имен (DNS)
67	UDP	Сервер протокола динамической конфигурации узла (Dynamic Host Configuration Protocol, DHCP)
68	UDP	Dynamic Host Configuration Protocol (Протокол динамической настройки узла) (клиент)
69	UDP	Простейший протокол передачи файлов (TFTP)
80	TCP	Протокол HTTP
110	TCP	Протокол почтового отделения (Post Office Protocol version 3, POP3)
143	TCP	Протокол IMAP
161	UDP	Протокол SNMP
443	TCP	Защищенный протокол передачи гипертекста (HTTPS)

Команда netstat

Неопознанные TCP-соединения могут представлять значительную угрозу безопасности. Netstat является важным инструментом для проверки подключений.

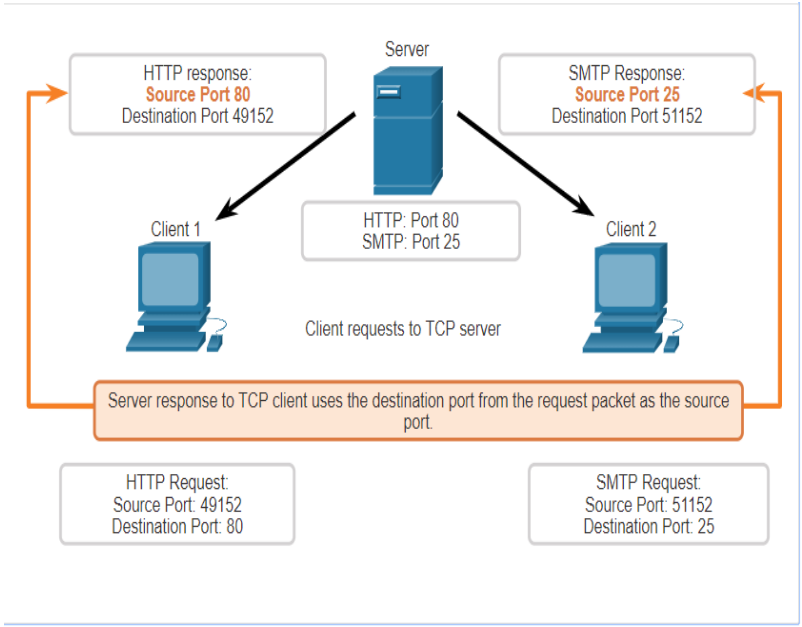
```
 C:\> netstat
Активные соединения
Proto Local Address Foreign Address State
TCP 192.168.1.124:3126 192.168.0.2:netbios-ssn ESTABLISHED
TCP 192.168.1.124:3158 207.138.126.152:http ESTABLISHED
TCP 192.168.1.124:3159 207.138.126.169:http ESTABLISHED
TCP 192.168.1.124:3160 207.138.126.169:http ESTABLISHED
TCP 192.168.1.124:3161 sc.msn.com:http ESTABLISHED
TCP 192.168.1.124:3166 www.cisco.com:http ESTABLISHED
```

14.5. Процесс обмена данными по протоколу TCP

Процесс TCP-сервера

Каждый процесс приложения, работающий на сервере, использует номер порта

- Не допускается использование двумя различными службами на одном и том же сервере одного и того же порта с одинаковым протоколом транспортного уровня.
- Активное серверное приложение, которому присвоен какой-либо определенный порт, считается открытым, что означает, что транспортный уровень может принимать и обрабатывать сегменты, направляемые на этот порт.
- Любой входящий запрос, который адресован правильному сокету, будет принят, а данные будут переданы приложению сервера.

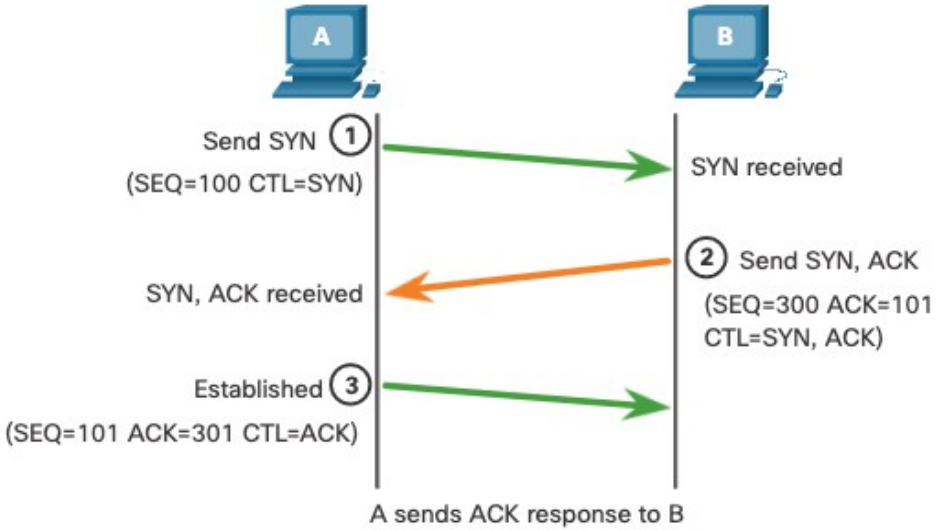


Установление TCP-соединения

Шаг 1. Инициирующий клиент запрашивает сеанс связи «клиент-сервер» с сервером.

Этап 2. Сервер подтверждает сеанс обмена данными «клиент-сервер» и запрашивает сеанс обмена данными «сервер-клиент».

Шаг 3. Инициирующий клиент подтверждает сеанс связи «сервер-клиент».



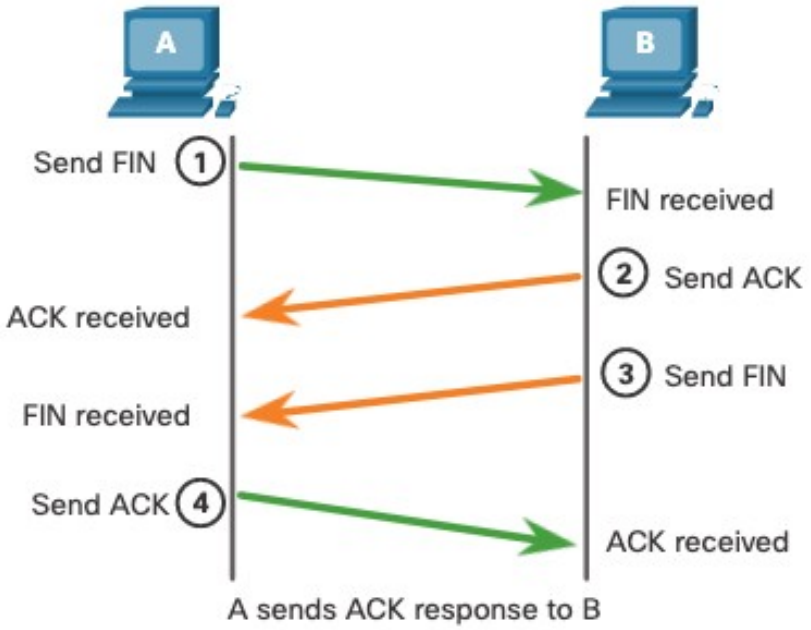
Прекращение TCP-соединения

Шаг 1. Когда у клиента больше нет данных для отправки в потоке, он отправляет сегмент с установленным флагом FIN.

Шаг 2. Сервер отправляет подтверждение ACK, чтобы подтвердить получение FIN для завершения сеанса связи «клиент-сервер».

Шаг 3. Сервер отправляет FIN клиенту, чтобы завершить сеанс «сервер-клиент».

Шаг 4. Клиент отправляет в ответ сегмент ACK для подтверждения получения сегмента FIN от сервера.



Анализ трехстороннего квитирования TCP

Таковы функции трехстороннего рукопожатия:

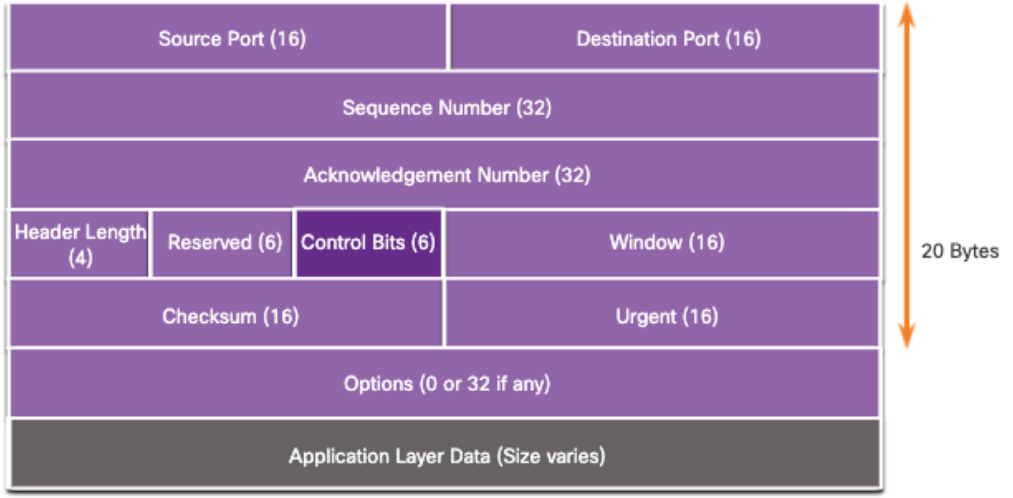
- Определяет, присутствует ли в сети устройство назначения.
- Проверяет, имеется ли на устройстве назначения активная служба и принимает ли она запросы на номер порта назначения, который инициирующий клиент планирует использовать.
- Информирует устройство назначения, что клиент источника планирует установить сеанс связи на этом номере порта.

По завершении обмена данными все сеансы закрываются, а соединение прерывается. Механизмы подключения и осуществления сеанса связи включают в себя функции TCP, обеспечивающие надежность.

Анализ трехстороннего квитирования TCP (продолжение)

Шесть контрольных битовых флагов выглядят следующим образом:

- **URG** - флаг «Указатель важности»
- **ACK** - Флаг подтверждения, используемый при установке соединения и завершении сеанса
- **PSH** - флаг "Push"
- **RST-** Флаг RST используется для сброса соединения при возникновении ошибки или в случае превышения времени ожидания.
- **SYN** - Синхронизировать порядковые номера, используемые при установке соединения
- **FIN** - больше нет данных от отправителя и используется при завершении сеанса



Видео. Трехстороннее квитирование TCP

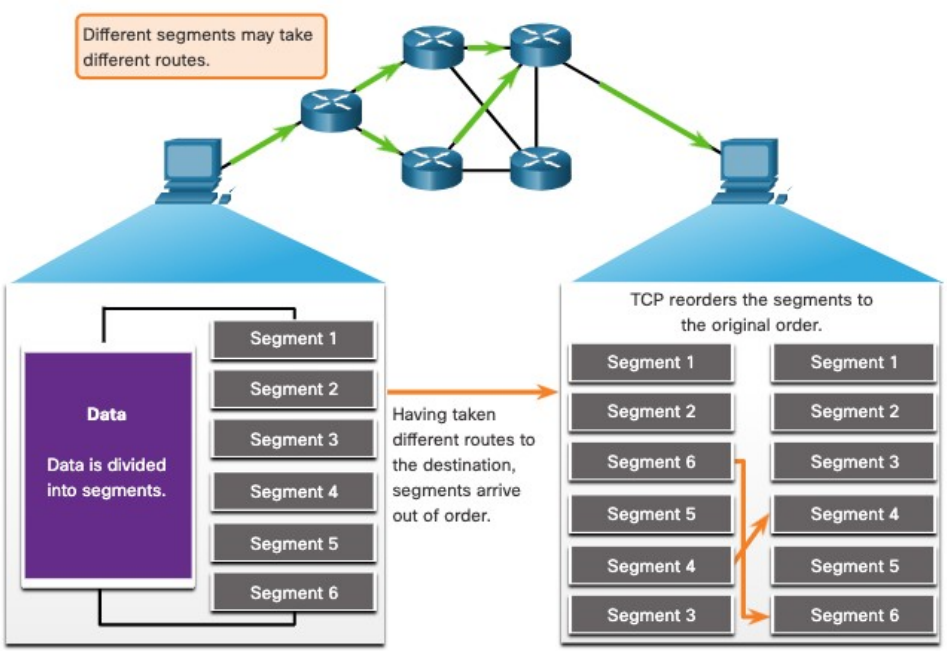
Это видео будет охватывать следующее:

- Трехэтапное квитирование TCP
- и завершения сеанса TCP.

14.6. Надежность и управление потоком

Надежность TCP: упорядоченная доставка

- TCP также может помочь поддерживать поток пакетов, чтобы устройства не перегружались.
- Могут быть случаи, когда сегменты TCP не приходят к месту назначения.
- Все данные должны быть получены и данные в этих сегментах должны быть собраны в исходном порядке.
- Для этого в заголовке каждого пакета указываются порядковые номера.



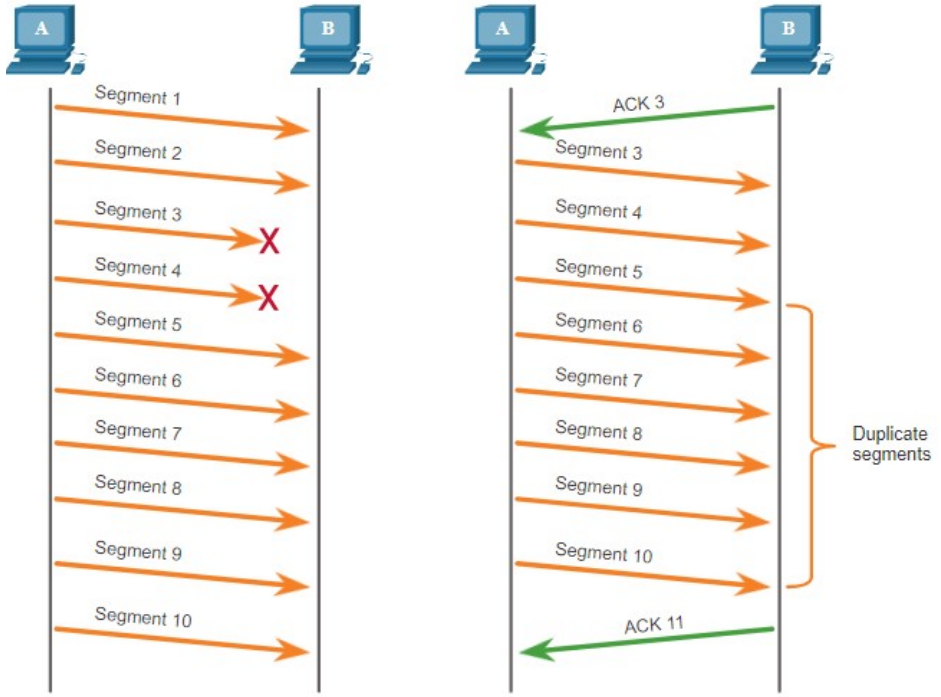
Видео. Надежность TCP: порядковые номера и подтверждения

В этом видеоролике показан упрощенный пример работы протокола TCP.

Надежность TCP: потеря данных и повторная передача

Независимо от того, насколько хорошо разработана сеть, иногда происходит потеря данных.

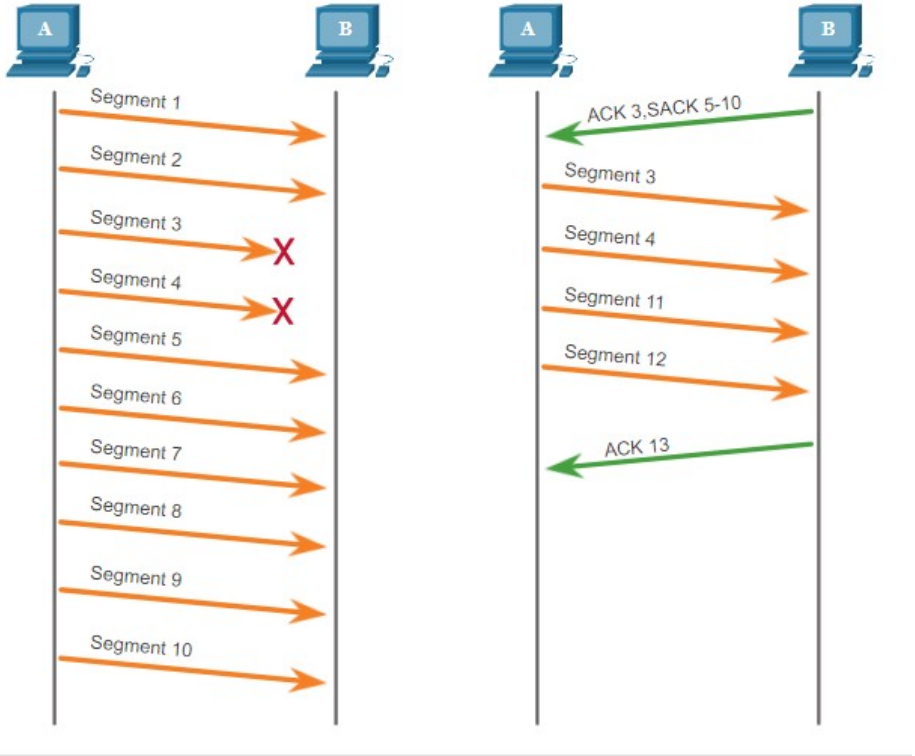
Протокол TCP обеспечивает возможности для управления потерянными сегментами. Среди них — механизм повторной передачи сегментов с данными, получение которых не было подтверждено.



Надежность TCP: потеря данных и повторная передача

В настоящее время серверные операционные системы обычно используют опциональную функцию TCP, называемую выборочным подтверждением (SACK), согласованную во время трехстороннего рукопожатия.

Если оба узла поддерживают SACK, приемник может явно определить, какие сегменты (байты) были получены, включая любые сегменты прерывания.



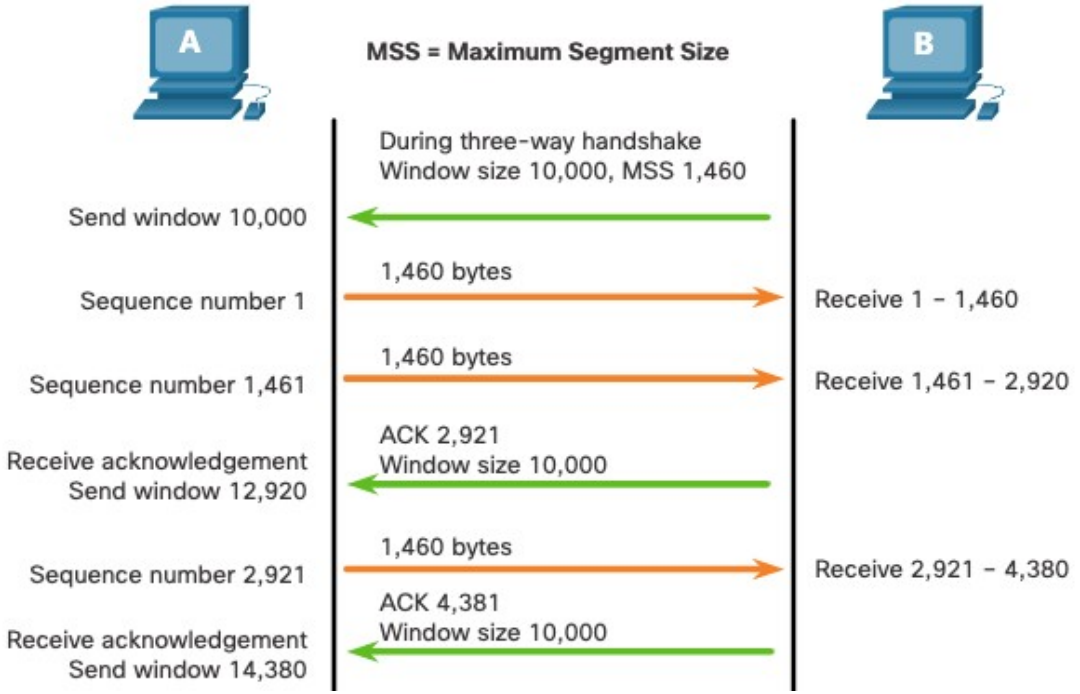
Видео. Потеря данных и повторная передача

В этом видео показан процесс повторной обработки сегментов, которые изначально не были получены конечным назначенным.

Управление потоком TCP: размер окна и подтверждения

Протокол TCP использует механизмы для управления потоком данных

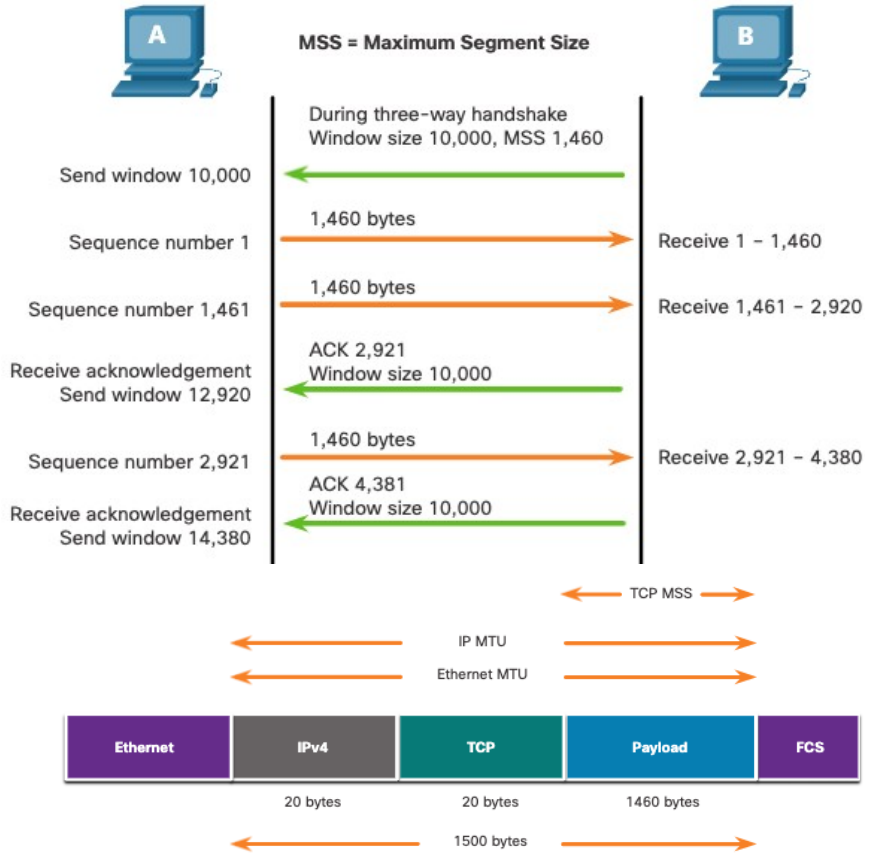
- Управление потоком данных - это объем данных, которые получатель может получить и надежно обработать.
- Управление потоком позволяет поддерживать надежность передачи по протоколу TCP, регулируя скорость потока данных между узлами источника и назначения в течение определенного сеанса.



Управление потоком TCP: размер окна и подтверждения

Максимальный размер сегмента (MSS) — это максимальный объем данных, который может получить конечное устройство.

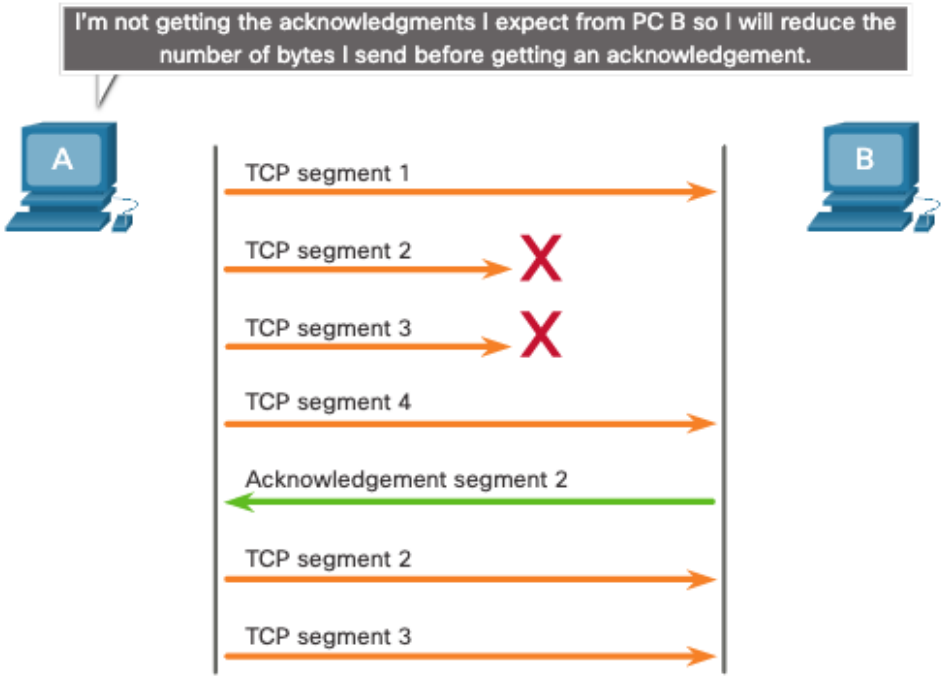
- Обычный MSS составляет 1 460 байт при использовании IPv4.
- Узел определяет значение своего поля MSS путем вычитания размера заголовков IP и TCP из размера MTU для Ethernet.
- 1500 минус 60 (20 байт для заголовка IPv4 и 20 байт для заголовка TCP) оставляет 1460 байт.



Управление потоком TCP: предотвращение перегрузок

В случае возникновения перегрузки в сети перегруженный маршрутизатор перестает обрабатывать пакеты.

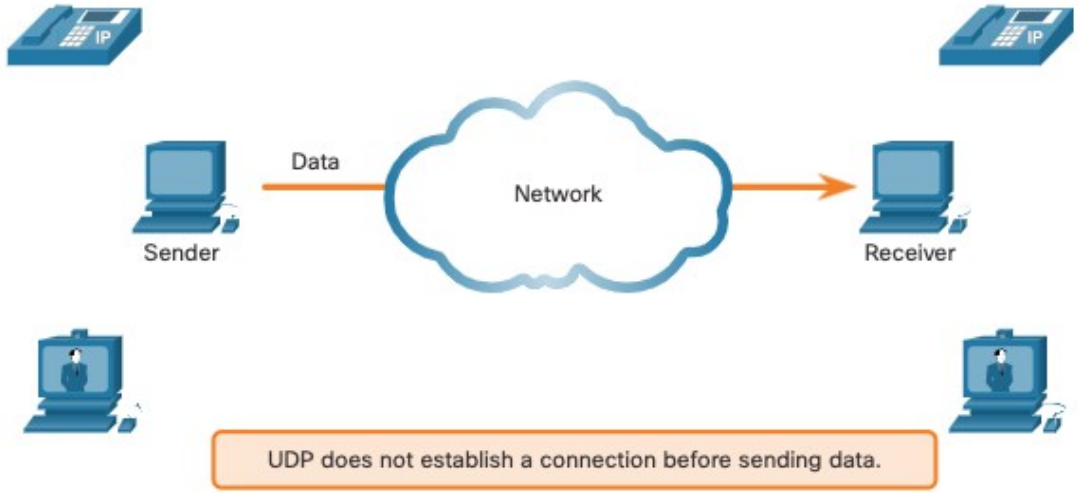
Во избежание таких ситуаций и для предотвращения перегрузок сети в протоколе TCP предусмотрен ряд соответствующих механизмов, таймеров и алгоритмов.



14.7. Обмен данными по протоколу UDP

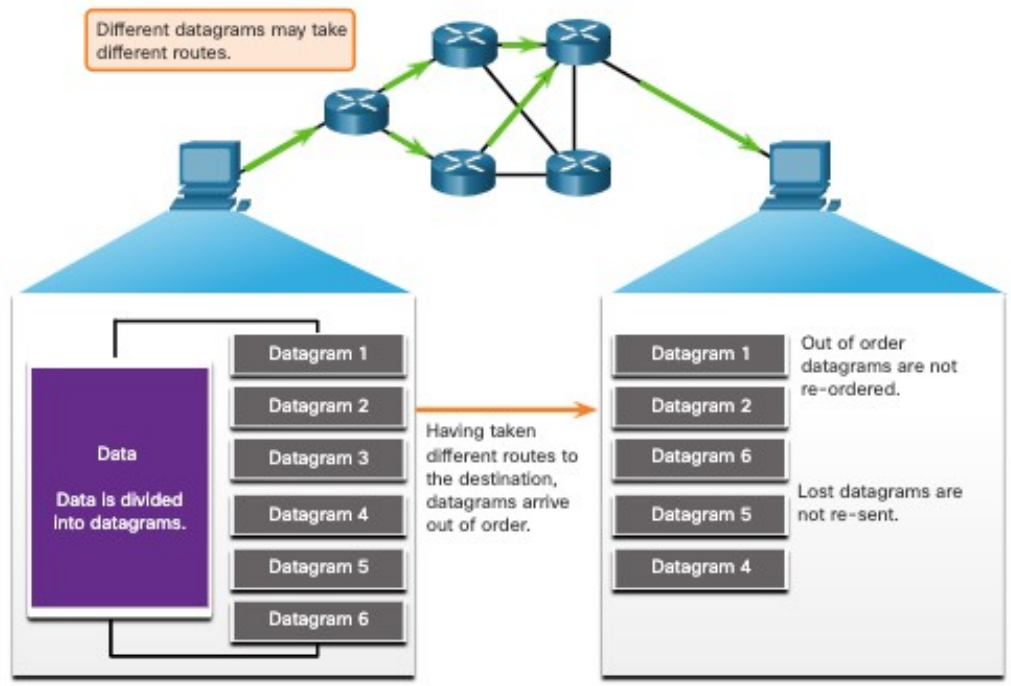
UDP: низкая нагрузка или надежность

UDP не устанавливает соединение перед отправкой данных. Протокол UDP обеспечивает передачу данных с меньшими накладными расходами, поскольку он имеет небольшой заголовок датаграммы и не обменивается управляющим трафиком.



Повторная дефрагментация датаграммы UDP

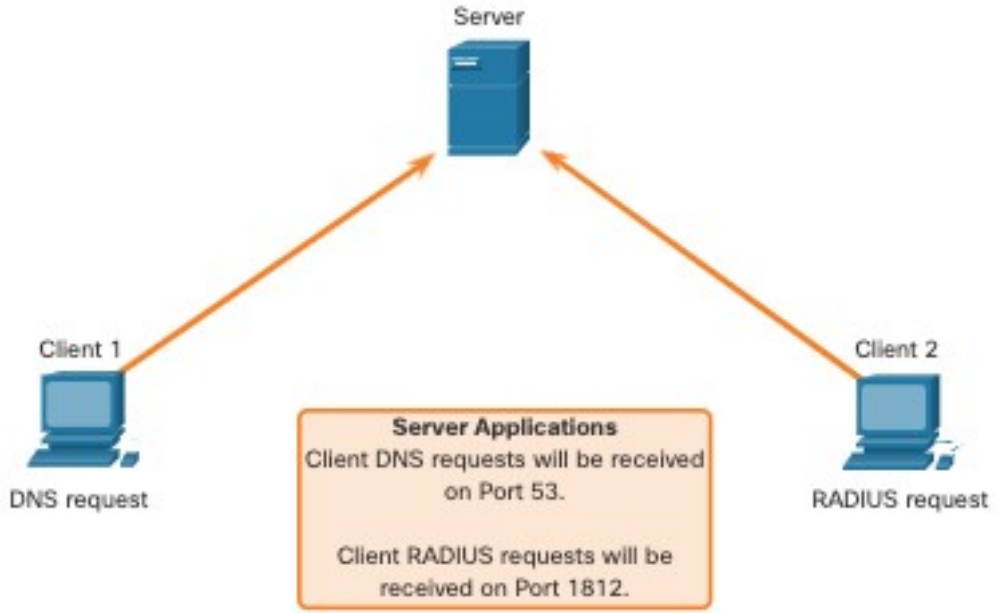
- Протокол UDP не отслеживает порядковые номера, как это делает протокол TCP.
- Протокол UDP не может повторно скомпоновать датаграммы в том порядке, который использовался при их передаче
- Таким образом, протокол UDP просто повторно собирает данные в том порядке, в котором они были приняты, и пересылает их приложению.



Процессы и запросы UDP-сервера

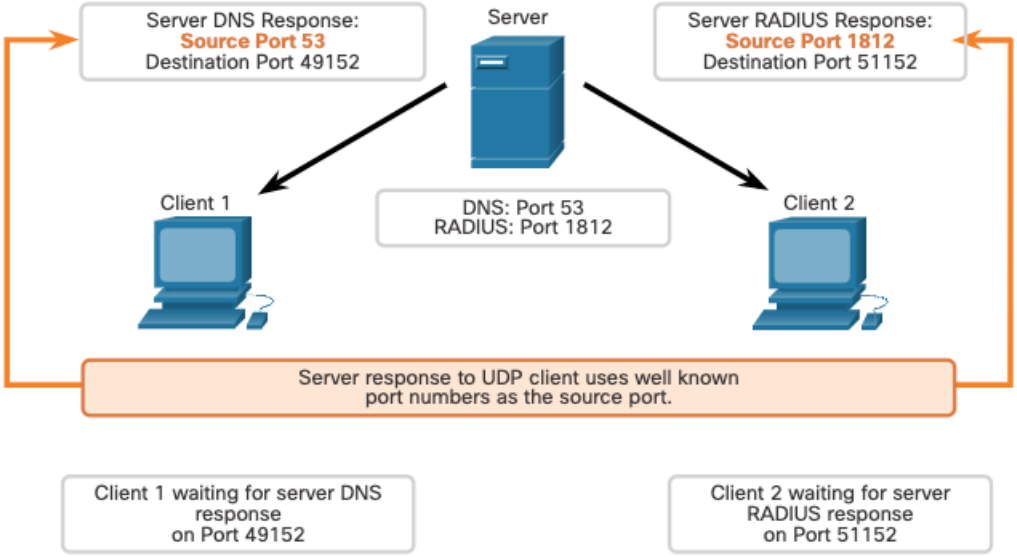
Серверным приложениям на базе UDP назначаются широко известные или зарегистрированные номера портов.

Если UDP получает датаграмму, адресованную одному из этих портов, он пересылает данные приложения соответствующему приложению, исходя из его номера порта.



Процессы UDP-клиента

- Процесс UDP-клиента динамически выбирает номер порта из диапазона номеров портов и использует его в качестве порта источника для сеанса связи.
- Как правило, порт назначения — это общеизвестный или зарегистрированный номер порта, присвоенный процессу сервера.
- После того как клиент выбрал порты источника и назначения, эта же пара портов будет указана в заголовке всех датаграмм, которые используются в процессе пересылки.



14.8 Практика и контрольная работа модуля

Packet Tracer - Обмен данными с использованием TCP и UDP

В этом задании Packet Tracer вы будете делать следующее:

- Генерация сетевого трафика в режиме моделирования
- Изучение функциональных возможностей протоколов TCP и UDP

Что я изучил в этом модуле?

- Транспортный уровень — это канал между уровнем приложений и нижними уровнями, которые отвечают за передачу данных по сети.
- На транспортном уровне действуют два протокола — TCP и UDP.
- TCP устанавливает сеансы, обеспечивает надежность, обеспечивает доставку одного и того же заказа и поддерживает управление потоком.
- UDP — это простой протокол, обеспечивающий работу основных функций транспортного уровня.
- UDP восстанавливает данные в том порядке, в котором они были получены, потерянные сегменты не повторно отправляются, не создаются сеансы, а UPD не сообщает отправителю о доступности ресурсов.
- Протоколы транспортного уровня TCP и UDP используют номера портов для управления несколькими одновременными разговорами.
- Каждый процесс приложения, работающий на сервере, использует номер порта
- Номер порта автоматически назначается или настраивается системным администратором вручную.
- Для того чтобы получатель смог расшифровать изначальное сообщение, данные в этих сегментах повторно собираются в исходном порядке.

Что я изучил в этом модуле? (Продолжение)

- В заголовке каждого пакета указываются порядковые номера.
- Управление потоком позволяет поддерживать надежность передачи по протоколу TCP, регулируя скорость потока данных между узлами источника и назначения в течение определенного сеанса.
- Источник передает 1460 байт данных в каждом сегменте TCP. Это типичный MSS, который может получить устройство назначения.
- Процесс отправки подтверждений узлом назначения по мере обработки полученных байтов и непрерывная регулировка окна отправки источника называются скользящими окнами.
- Во избежание таких ситуаций и для предотвращения перегрузок сети в протоколе TCP предусмотрен ряд соответствующих механизмов, таймеров и алгоритмов


CISCO