



Транспортные протоколы стека TCP/IP

Подготовил Темирбулатов Альберт 09-231

План занятия

Введение в транспортные протоколы

- Определение
- Роль
- Адресация

01



Протокол TCP

- Определение
- Сегменты
- Формат пакета
- Соединение и передача

02



Протокол UDP

- Определение
- Дейтаграммы
- Формат пакета
- Передача данных

03



04

Сравнение TCP и UDP

- Ключевые различия:
- Таблица сравнений.



05

Проблемы протоколов

- Адаптация
- Масштабирование
- Безопасность



06

Вопросы для студентов



1. Введение в транспортные протоколы

Что такое транспортные протоколы?

- Транспортные протоколы — это набор правил и стандартов, используемых для обмена данными между приложениями, работающими на разных компьютерах.
- Работают на транспортном уровне модели TCP/IP (4-й уровень модели OSI).
- Основные функции:
 - Обеспечение целостности данных.
 - Устранение ошибок передачи.
 - Разделение данных на пакеты и их сборка.
 - Контроль последовательности доставки.



Роль транспортных протоколов

1) Передача данных между устройствами

Транспортные протоколы обеспечивают доставку данных между приложениями, работающими на разных устройствах.

2) Обеспечение надёжности

Данные доставляются корректно, без потерь, в правильной последовательности.

3) Разделение данных

Протоколы разбивают большие сообщения на небольшие части (пакеты) и собирают их обратно при получении.

4) Управление соединением

Организация связи между отправителем и получателем: установление, поддержание и завершение соединения.

5) Контроль передачи данных

Регулирование скорости передачи данных для предотвращения перегрузки сети.

Адресация. Порты

Для адресации на транспортном уровне используются специальные числовые идентификаторы, которые называются портами. Это число от 1 до 65535.

Каждое сетевое приложение имеет свой порт.

Номера у портов не повторяются.

Для передачи данных между устройствами в сети используются два ключевых элемента:

1. **IP-адрес:**

- Указывает, к какому устройству должны быть доставлены данные. Работает на сетевом уровне (Internet Layer).

2. **Номер порта:**

- Определяет приложение на устройстве, которому предназначены данные. Работает на транспортном уровне (Transport Layer).

Вместе IP-адрес и порт образуют сокет (socket):

- Формат: <IP-адрес>:<номер порта>
- Пример: 192.168.1.10:80 (устройство с IP-адресом 192.168.1.10, использующее порт 80 для HTTP).

Адресация. Пакеты

На транспортном уровне данные передаются между устройствами в сети в виде **пакетов**. Пакет представляет собой структурированную единицу данных, которая используется для организации передачи информации через сеть.

Пакет содержит передаваемые данные и адреса отправителя и получателя.

Адрес отправителя (Source address) :

IP: 82.146.49.55

Port: 2049

Адрес получателя (Destination address) :

IP: 195.34.32.116

Port: 53

Данные пакета:

...



2. Протокол TCP

Протокол надежной доставки TCP

TCP — это протокол транспортного уровня, предназначенный для надёжной доставки данных между устройствами в сети.

Разработан в 1970-х годах для сети ARPANET, предшественника интернета.

Его основные задачи включают:

- **Надёжность передачи**
- **Контроль порядка**
- **Коррекция ошибок**
- **Повторная отправка**
- **Управление соединением**

Сегмент.

Сегмент TCP — это основная единица данных, передаваемая протоколом TCP. Он состоит из двух частей:

- **Заголовок TCP:** Служебная информация, необходимая для управления передачей.
- **Данные:** Полезная нагрузка, передаваемая от приложения.

Почему нужны сегменты?

Транспортная подсистема получает от приложения данные в виде потока байт, длинные сообщения (например, файл или видео) нельзя отправить целиком, так как они могут превышать допустимый размер пакета в сети. Поэтому TCP разбивает сообщение(поток байтов) на небольшие части — сегменты.

Каждый сегмент передаётся отдельно и нумеруется для восстановления на стороне получателя.

Формат заголовка TCP

Формат TCP-сегмента

Порт источника		Порт приемника	
Порядковый номер			
Порядковый номер (подтверждение)			
Размер заголовка	Флаги	Размер приемного окна	
Контрольная сумма		Доп. параметры	
Дополнительные параметры			
Данные			

32 бита



Установление соединения в TCP (3-way handshake)

TCP использует процедуру трёхэтапного рукопожатия для установления соединения:

1. **Клиент отправляет SYN:**
 - Клиент инициирует соединение, отправляя сегмент с установленным флагом SYN и порядковым номером x .
2. **Сервер отвечает SYN-ACK:**
 - Сервер подтверждает получение SYN клиента, отправляя сегмент с флагами SYN и ACK.
 - Сервер указывает свой порядковый номер y и подтверждает получение номера клиента $x + 1$.
3. **Клиент отправляет ACK:**
 - Клиент подтверждает соединение, отправляя сегмент с флагом ACK и подтверждением номера сервера $y + 1$.

После этого соединение установлено, и данные могут передаваться.

Передача данных

Если какой-то сегмент потерялся. Значит получатель не отправляет флаг ACK. Для такого случая существует специальный таймер по истечению которого(если подтверждение не получено) этот сегмент отправляется еще раз. Тайм-аут динамически регулируется (используется алгоритм RTT — Round-Trip Time, чтобы вычислить среднее время доставки).

Если контрольная сумма не совпала, то получатель не отправляет ACK, и тогда отправитель присылает его еще раз.

Получатель упорядочивает данные сразу по мере их поступления. Если приходит сегмент с номером выше ожидаемого, то он сохраняется в буфере и получатель ждет нужного сегмента.

Скользящее окно

Скользящее окно протокола TCP

Скользящее окно (Sliding Window) — это механизм управления потоком данных, который:

- Позволяет передавать несколько сегментов до получения подтверждения.
- Регулируется параметром "Размер окна" (Window Size), задаваемым получателем.

Принцип работы:

Отправитель может передавать данные, пока не исчерпает размер окна. После получения подтверждения (ACK) окно "скользит" вперёд, освобождая место для новых данных. Это позволяет отправителю передавать новые сегменты. Получатель может увеличивать или уменьшать размер окна в зависимости от своих возможностей или состояния сети.

Преимущества:

- Оптимизация пропускной способности сети.
- Предотвращение перегрузки.

Скользящее окно

Зачем нужен механизм скользящего окна?

1. **Управление потоком:**
 - Предотвращает перегрузку у получателя, передавая данные только в объёме, который он может обработать.
2. **Эффективное использование сети:**
 - Позволяет отправителю передавать данные, не дожидаясь подтверждения каждого сегмента.
3. **Динамическая адаптация:**
 - Получатель может уменьшить окно при высокой нагрузке или увеличить его при освобождении ресурсов.



3. Протокол UDP

Протокол пользовательских дейтаграмм UDP

UDP (User Datagram Protocol) — это транспортный протокол, предоставляющий простой способ передачи данных между приложениями. Он работает на транспортном уровне модели ТСП/IP и отличается высокой скоростью, но отсутствием механизмов надёжности. Разработан в 1980-м году Дэвидом Ридом.

Основные особенности:

- **Простота передачи**
- **Отсутствие гарантии доставки**
- **Независимость от порядка**
- **Минимальный контроль ошибок**
- **Простота и скорость**

Дейтаграмма.

Дейтаграмма UDP — это основная единица данных, передаваемая протоколом UDP. Она состоит из двух частей:

- **Заголовок UDP:** Служебная информация, необходимая для управления передачей.
- **Данные:** Полезная нагрузка, передаваемая от приложения.

Почему нужны дейтаграммы?

Дейтаграммы UDP нужны для быстрой и простой передачи данных без установки соединения, с минимальной задержкой и низкой нагрузкой на сеть. Они подходят для приложений реального времени (видео, аудио), одноразовых запросов (DNS) и рассылок нескольким получателям (мультикаст, бродкаст), где надёжность доставки не критична.

Каждая дейтаграмма передаётся отдельно, не нумеруется и обрабатывается независимо, без гарантии доставки или восстановления порядка.

Формат UDP-сегмента

Порт источника	Порт приемника
Длина	Контрольная сумма
Данные	

← 32 бита →

Передача данных

Так как никаких проверок получения не проводится, то в случае потери дейтаграммы UDP **не обеспечивает повторную передачу**. Если пакет теряется, то он просто пропадает, и протокол не предпринимает никаких действий.

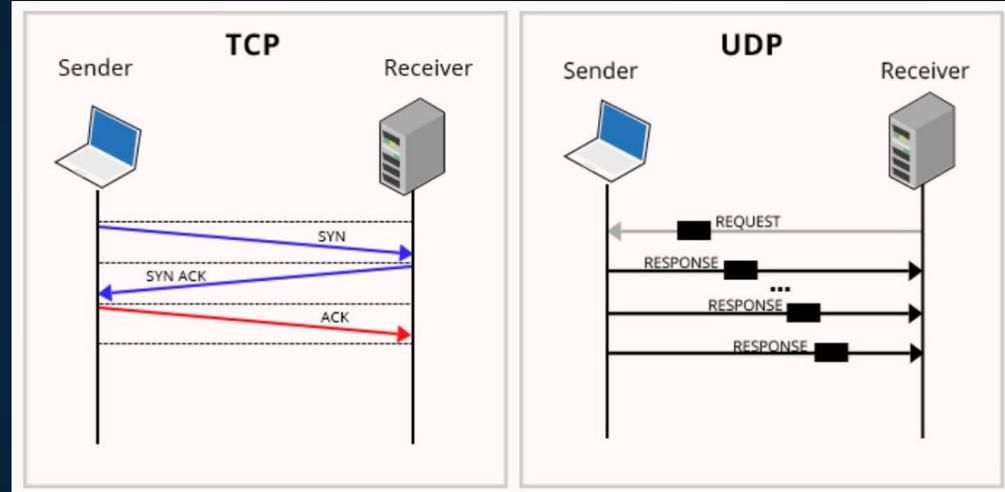
Если контрольная сумма неверна, дейтаграмма **отбрасывается**. Протокол не уведомляет отправителя об ошибке.

UDP **не обеспечивает упорядочивания данных**. Дейтаграммы могут приходить в любом порядке. Порядок доставки зависит от маршрута в сети, задержек и других факторов. О порядке должно заботиться само приложение(если это важно).



4. Сравнение TCP и UDP

Характеристика	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Тип соединения	Протокол с установлением соединения (требуется 3-way handshake).	Протокол без установления соединения (отправка сразу).
Гарантия доставки	Обеспечивает доставку всех данных (повторная отправка при потере).	Не гарантирует доставку данных (потерянные пакеты игнорируются).
Контроль порядка	Данные упорядочиваются перед передачей приложению.	Порядок не гарантируется, пакеты могут приходить вразнобой.
Коррекция ошибок	Использует контрольную сумму, подтверждения (ACK) и повторную отправку.	Только контрольная сумма, повреждённые пакеты отбрасываются.
Скорость	Медленнее из-за дополнительных процедур (установление соединения, подтверждения).	Быстрее, так как отсутствуют подтверждения и восстановление.
Надёжность	Надёжный (гарантия доставки, порядок, контроль ошибок).	Ненадёжный (без гарантий доставки и упорядочивания).
Размер заголовка	20–60 байт (более сложный, содержит много служебной информации).	8 байт (минимальный заголовок для скорости).
Использование ресурсов	Высокое из-за сложной структуры и дополнительных проверок.	Низкое благодаря простоте реализации.
Применение	Критичные к потере данных приложения: веб-сайты, электронная почта, файлообмен.	Приложения реального времени: видеопоток, аудио, игры, DNS.
Механизм управления потоком	Регулирует скорость передачи с помощью скользящего окна и размера буфера.	Отсутствует (весь контроль на уровне приложения).
Мультикаст и бродкаст	Не поддерживается.	Поддерживаются (отправка данных сразу нескольким получателям).
Тип данных	Передача в виде потока байтов.	Передача в виде отдельных дейтаграмм.





5. Общие проблемы TCP и UDP

- 1) **Проблемы адаптации:** Протоколы TCP и UDP были разработаны несколько десятилетий назад и плохо приспособлены к современным требованиям, таким как поддержка IoT-устройств, потоковых сервисов и переход на IPv6. Их архитектура недостаточно гибкая для эффективного реагирования на изменения в нагрузке и условиях работы сети.
- 2) **Проблемы масштабируемости:** Увеличение количества подключённых устройств и объёмов трафика создаёт значительные сложности. Протоколы не справляются с управлением огромными таблицами маршрутизации, что приводит к снижению производительности сети и увеличению задержек.
- 3) **Проблемы безопасности:** TCP и UDP не имеют встроенного шифрования, что делает их уязвимыми для атак, таких как перехват данных, подделка пакетов и DDoS. Это создаёт угрозу целостности и конфиденциальности передаваемых данных.

Решения проблем

- 1) Для адаптации используются современные протоколы, такие как QUIC, которые улучшают производительность и надёжность. Также внедряются алгоритмы управления нагрузкой, например TCP BBR, которые позволяют быстрее реагировать на изменения в сети. Постепенный переход на IPv6 и улучшение совместимости с новыми технологиями также являются ключевыми мерами.
- 2) Для повышения масштабируемости используются программно-определяемые сети (SDN), которые позволяют эффективно управлять маршрутизацией. Применение алгоритмов оптимизации маршрутов и использование сетей доставки контента (CDN) помогает равномерно распределять нагрузку и снижать давление на магистральные каналы.
- 3) Для повышения безопасности используется шифрование данных с помощью протоколов TLS (для TCP) и DTLS (для UDP). Для защиты от DDoS внедряются системы фильтрации трафика и специализированные сервисы защиты. Также современный подход включает переход на протоколы с улучшенной безопасностью, такие как DNS over HTTPS, чтобы предотвратить подделку запросов.



6. Вопросы для студентов

1) Для чего используется механизм трёхэтапного рукопожатия в ТСР?

- a) Для передачи данных
- b) Для установления соединения между устройствами
- c) Для проверки контрольной суммы
- d) Для завершения соединения



2) Как TCP обеспечивает надёжную передачу данных?

- a) Использует контрольную сумму и повторную передачу потерянных данных
- b) Упрощает передачу, избегая подтверждений
- c) Упорядочивает данные без проверки целостности
- d) Пропускает повреждённые сегменты



3) Что происходит, если контрольная сумма в UDP не совпала?

- a) Дейтаграмма переправляется заново
- b) Соединение сбрасывается
- c) Дейтаграмма отбрасывается
- d) Данные передаются в приложение



4. Какая максимальная длина порта?

- a) 16
- b) 65536
- c) 1024
- d) 65535



5) Какая основная функция контрольной суммы?

- a) Установление соединения
- b) Разделение данных
- c) Проверка целостности данных
- d) Определение IP-адреса



6) Как называется механизм управления потоком данных в ТСР?

- a) Буферизация
- b) Скользящее окно
- c) Дейтаграммный контроль
- d) Дублирование



7) Чем управляет размер окна в ТСР?

- a) Количеством сегментов в соединении
- b) Объёмом данных, которые можно отправить до подтверждения
- c) Скоростью передачи данных
- d) Порядком доставки данных



8) Как TCP определяет, что сегмент был потерян?

- a) Получает флаг RST
- b) Получает повторный запрос
- c) Не получает подтверждение (ACK) в течение заданного времени
- d) Неверная контрольная сумма



9) Что происходит, если АСК не приходит вовремя в ТСР?

- a) Сегмент пропускается
- b) Размер окна увеличивается
- c) Соединение закрывается
- d) Сегмент передаётся повторно



10) Какая основная функция флага SYN в TCP?

- a) Завершение соединения
- b) Передача данных
- c) Запрос на установление соединения
- d) Подтверждение данных



11) Что означает поле "Порт назначения" в заголовке UDP?

- a) Указывает порт приложения, которое должно обработать данные
- b) Указывает на отправителя
- c) Указывает на IP-адрес получателя
- d) Используется для шифрования



12) Чем UDP отличается от TCP?

- a) Использует трёхэтапное рукопожатие
- b) Гарантирует доставку данных
- c) Передаёт данные без установки соединения
- d) Управляет потоками данных



13) Какая особенность протокола UDP делает его ненадёжным?

- a) Отсутствие заголовка
- b) Отсутствие сокетов
- c) Нет подтверждения доставки
- d) Большие задержки



14) Какой протокол используют для ГОЛОСОВЫХ ВЫЗОВОВ И ПОТОКОВОГО видео?

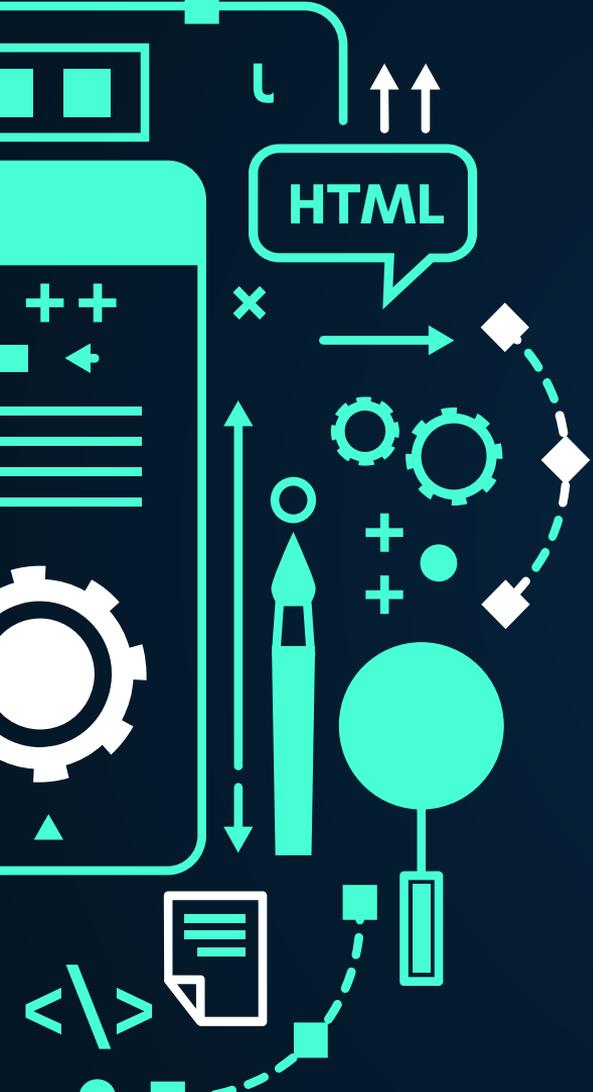
- a) TCP
- b) UDP
- c) FTP
- d) DNS



15) Какое основное преимущество UDP перед TCP?

- a) Надёжность доставки
- b) Контроль порядка
- c) Высокая скорость и минимальная задержка
- d) Шифрование данных





**Спасибо за
внимание!!!**