Популярные протоколы, используемые в WEB-разработке

Транспортный уровень — важнейший компонент связи между устройствами в сети. Он отвечает за обеспечение надежной и эффективной доставки данных между приложениями, работающими на разных устройствах. Неважно, с какого устройства было отправлено сообщение и какое устройство будет его принимать, транспортный уровень должен обеспечить должную скорость и надежность.

Цель урока — познакомить вас с протоколами транспортного уровня и сферами их применения.

В результате обучения:

- вы познакомитесь с основными протоколами транспортного уровня;
- поймете, по каким критериям стоит выбирать тот или иной протокол в своем проекте;
- узнаете, какие протоколы используются в популярных приложениях;
- изучите основные протоколы, используемые в веб-девелопменте.

План урока

1. Введение

Транспортный уровень играет одну из центральных ролей в обмене данными между устройствами в сети, обеспечивая надежную и эффективную доставку данных между приложениями.

Организовать транспортный уровень помогают протоколы, которые делятся на два типа:

- Физические отвечают за форму — как именно оборудование будет подавать сигналы.
- Логические отвечают за содержание — как в этих сигналах будут организованы данные.

В веб-разработке понимание транспортного уровня и его протоколов важно для создания эффективных веб-приложений, которые не будут терять или портить отправляемые данные. Необязательно, чтобы это были текстовые сообщения — обмен мемами в мессенджере тоже является обменом данных, пусть и выраженным картинкой. Другой пример обмена данными — сессия в онлайн-игре: отправка и получение данных здесь тоже играет важнейшую роль.

Наиболее часто используемые протоколы транспортного уровня:

- TCP — обеспечивает надежную доставку данных, но проигрывает по скорости.
- UDP — обеспечивает более быструю, но менее надежную доставку.

Выбор между двумя протоколами зависит от конкретных требований разрабатываемого приложения.

Наиболее часто используемые протоколы в веб-разработке включают:

- HTTP и HTTPS (просмотр веб-страниц и загрузка файлов);
- FTP (передача файлов);
- SMTP (e-mail).

2. Существующие протоколы

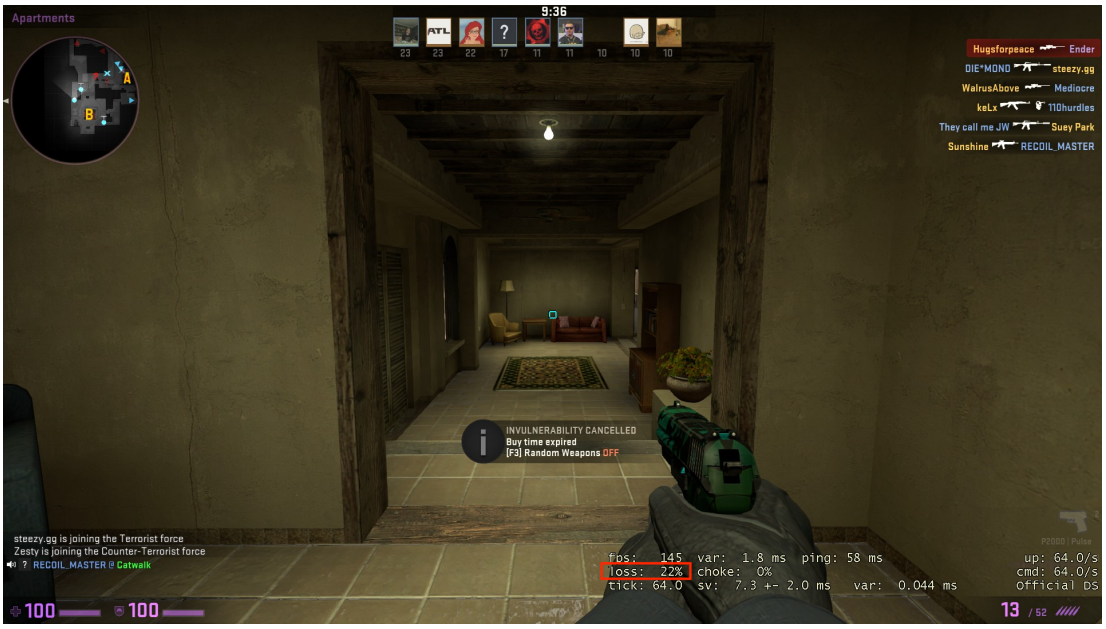
TCP (Transmission Control Protocol)

TCP — один из двух самых популярных протоколов. Чаще всего он употребляется в связке с другим протоколом, Internet Protocol. Отсюда получается уже встречавшаяся в курсе аббревиатура TCP/IP. Важно помнить, что несмотря на частое совместное применение, это два отдельных протокола — у каждого свои правила и цели.

TCP делает ставку на **надежность** передачи данных, жертвуя при этом скоростью передачи. На отправляющем устройстве TCP разбивает на пакеты данные, которые нужно отправить. А на устройстве-получателе собирает эти пакеты обратно в том порядке, в каком они были разбиты.

Пакеты, на которые TCP разбивает сообщение, состоят из адреса отправителя, адреса получателя, служебной информации, самих данных и номера пакета. Когда очередной пакет достигает принимающего устройства, оно отправляет сигнал о получении этого пакета. Так отправляющее устройство убеждается в корректной доставке. Если такого сигнала не приходит, устройство-отправитель отправляет потерявшийся пакет еще раз.

Потеря пакетов (packet loss) — метрика, которая является важным показателем качества соединения. Она определяется как отношение потерянных пакетов ко всем отправленным пакетам.



В онлайн-играх типа Counter-Strike высокий packet loss заметно затруднит игру. То же относится к просмотру стримов или другого потокового видео.

Именно подтверждение получения добавляет времени соединению по протоколу TCP. Устройства подтверждают или получение диапазона пакетов, или выборочно некоторых. Подтверждение получения каждого пакета заняло бы слишком много времени и сильно снизило бы пропускную способность соединения.

Обратите внимание, что для протокола TCP абсолютно неважно, куда направляются эти данные (за это отвечает протокол IP, формирующий и понимающий сетевые адреса). Главная его задача — обеспечить целостность передачи данных. Именно за ставку на целостность и надежность этот протокол обрел свою популярность.

UDP (User Datagram Protocol)

Чтобы разобраться с этим протоколом, в первую очередь нужно понять, что такое **датаграмма**, упомянутая в названии протокола. Этот термин появился как комбинация двух слов — data (данные) и telegram (телеграмма). Словом «датаграмма» обозначается блок информации, который будет отправляться по протоколу.

У датаграммы нет такого строго регламентированного состава, как у пакета. Единственное, что определено и зафиксировано, — размер датаграммы в байтах. Также у датаграмм нет порядковых номеров.

При приеме датаграммы устройство-получатель не отправляет сигнал о получении. И как следствие, датаграмма отправляется «в один конец» — устройство-отправитель не узнает, чем завершилась отправка.

Единственная проверка, существующая в протоколе UDP — это проверка на целостность полученной датаграммы. Для этого в ней вычисляется контрольная сумма. Если длина сообщения в байтах нечетна, устройство-отправитель может прибавить байт со значением 0 к концу датаграммы.

Устройство-получатель, обрабатывая датаграмму, сравнивает последний байт с общей длиной сообщения и из этого делает вывод о целостности полученного сообщения:

- если количество байт нечетно, то при передаче произошла ошибка;
- если же оно четно, то ошибки не было.

Протокол UDP во многом проще, чем TCP. В нем нет подтверждения получения пакетов, и для передачи не выделяется специальный канал. Благодаря этому передача данных через UDP быстрее (порой даже существенно быстрее), но за эту скорость придется заплатить меньшей надежностью соединения.

Однако не стоит считать, что протокол UDP — аналог «русской рулетки» для компьютерных сетей. Несмотря на то что он менее надежен (нежели TCP), он достаточно надежен, чтобы быстро доставлять данные.

Типичные области, в которых применяется UDP — телефония через Интернет (протокол Voice over IP, VoIP), система получения информации о доменах DNS (чем быстрее получим ответ, тем быстрее поймем, по какому IP-адресу обращаться), голосовой и видеотрафик в программах связи типа Zoom.

Что выбрать: TCP или UDP

Выбор между TCP и UDP зависит от целей, которые разработчик ставит перед своим приложением.

TCP — протокол, который обеспечивает такие качества соединения, как надежность, упорядоченность и последовательность. Вы можете быть полностью уверены в том, что сообщения дойдут до адресата в том порядке, в котором вы их отправили, и получающее устройство сможет собрать их воедино.

За доказательства надежности протокол расплачивается скоростью. Каждая проверка доставки отнимает время и место в канале связи, которое можно было бы занять отправлением других данных.

UDP — более простой по своей сути протокол. Вместо гарантий безопасности он может предложить скорость. Если два человека созваниваются через VoIP, они в разговоре смогут самостоятельно понять, полностью ли пришло сообщение от собеседника или же что-то потерялось.

Свойство UDP, которое может быть как минусом, так и плюсом — неупорядоченность. Мы не можем узнать точно, какая из датаграмм первой достигнет получателя.

Таким образом, если нужна ставка на порядок и уверенность, то чаша весов склонится к TCP. UDP же больше подходит в ситуациях, когда нужно передавать много данных потоком: пусть иногда из потока дойдет не все, зато сохранившиеся датаграммы придут быстро.

3. Применение протоколов в часто используемых приложениях

Рассмотрим несколько примеров различных протоколов в часто используемых приложениях.

Веб-страницы и загрузка файлов

Два самых используемых протокола в Интернете — HTTP и HTTPS (второй — улучшенная вариация первого). Чуть реже используется FTP. Эти три протокола составляют большинство серфинга в Интернете.

HTTP и HTTPS. HTTP (Hypertext Transfer Protocol) — это основной протокол обмена данными в Интернете. В случае с посещением сайтов клиент и сервер обмениваются не потоком сообщений, а единичными конечными документами. HTTP приближен к приложению, а не к соединению, кабелям или портам.

Свое развитие протокол HTTP получил в виде HTTPS (hypertext transfer protocol, secure) — соединения HTTP с шифрованием TLS. Связано это с тем, что HTTP не предоставляет какой-либо возможности зашифровать данные, хотя с течением времени на это естественным образом формировался запрос.

На сегодняшний день HTTPS постепенно становится стандартом веб-разработки.

FTP. Другой, менее используемый на сегодняшний день протокол — это FTP (File Transfer Protocol). Как следует из названия, его основное назначение — обмен файлами: как загрузка на сервер, так и скачивание с сервера.

В настоящее время два популярных браузера (Google Chrome и Firefox) убрали поддержку взаимодействия с FTP-серверами. Для этого существуют и другие специально предназначенные программы — например, FileZilla или WinSCP.

Стриминг и онлайн-игры

Стриминг — один из примеров задач, когда при выборе между TCP и UDP в итоге оказывается задействован второй протокол: ставка делается на скорость передачи. То же часто оказывается верно и для онлайн-игр.

Для этих задач разработали и другие протоколы. Например, WebRTC (Web Real-Time Communication), изначально придуманный под нужды голосовой и видеосвязи. Еще пример — HLS (HTTP Live Streaming), придуманный компанией Apple. Он отличается высокой производительностью, адаптивностью для Apple-устройств и использует в своей работе HTTP-серверы.

VPN

VPN-сервисы нередко предоставляют несколько вариантов подключения, каждый из которых использует различные протоколы. Среди них могут быть и уже названные TCP и UDP.

Другой популярный сейчас протокол для VPN-соединения — IKEv2, работающий вместе с протоколом IPSec. Он был разработан Cisco и Microsoft в 2005 году и постепенно набрал свою популярность. Сейчас он поддерживается и в современных компьютерах, и на устройствах Android и iOS.

При установлении подключения по этому протоколу сервер отправляет сертификат, удостоверяющий его (чтобы клиент был уверен, что запросы не отправляются дальше на какой-то сторонний сервер). Затем клиент и сервер обмениваются сессионными ключами, которые позволяют шифровать и дешифровать данные для обмена. В отличие от TLS, пара IKEv2 и IPSec предоставляет доступ к гораздо большему набору алгоритмов шифрования.

Другие используемые VPN-сервисами протоколы — Wireguard, L2TP и OpenVPN.

4. Другие протоколы, используемые в веб-разработке

HTTP и HTTPS

Протоколы HTTP и HTTPS работают с клиент-серверной архитектурой.

На месте клиента может быть кто угодно — веб-браузер, робот, обходящий все страницы сайта; прокси-сервер, служащий «прокладкой» между истинным клиентом и сервером; или иное устройство. Чтобы унифицировать отправителя запроса, введено понятие user-agent (участника обмена) — описание устройства, отправившего запрос.

User-agent — это строка, которая состоит из названия продукта (чаще всего — веб-браузера), версии этого продукта и других комментариев, описывающих устройства.

Примеры существующих юзер-агентов:

- Браузер Mozilla Firefox, ОС Windows — Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0), Gecko/20100101, Firefox/47.0.
- Браузер Google Chrome, Linux-based ОС — Mozilla/5.0 (X11; Linux x86_64), AppleWebKit/537.36 (KHTML, like Gecko), Chrome/51.0.2704.103, Safari/537.36.

- Робот Google — Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html).

При использовании протокола HTTPS сервер сначала устанавливает соединение с клиентом, который предоставляет список возможных функций шифрования и хэш-функций. Шифрование обеспечивает протокол TLS, который, в свою очередь, является усовершенствованной версией протокола SSL (Secure Sockets Layer).

Сервер выбирает (чаще всего случайно) один из вариантов шифрования и отправляет клиенту сертификат, удостоверяющий, что запросы идут именно с этого сервера. Убедившись в подлинности сертификата, клиент генерирует ключ шифрования — и только после этого начинается обмен данными по протоколу HTTP. Когда сеанс связи заканчивается (например, пользователь закрыл окно браузера), ключи стираются, и при следующем сеансе связи с сервером их нужно сгенерировать заново.

FTP (File Transfer Protocol)

Ссылки на такие сервера начинаются с **ftp://**, как HTTP-ссылки начинаются с **http://** и HTTPS — с **https://**.

При подключении к FTP-серверу задействуется два порта:

- первый используется для передачи сообщений о действиях на сервере (обычно это порт 20);
- второй — для непосредственной передачи файлов (как правило, порт 21).

SMTP (Simple Mail Transfer Protocol), POP3, IMAP

Протокол SMTP был придуман в 1982 году, но 40 лет спустя все еще популярен. Он используется исключительно для отправки почты. А для ее получения используются протоколы POP (Post Office Protocol) и IMAP (Internet Message Access Protocol). Современные почтовые программы поддерживают все три протокола. Некоторые же используют собственные протоколы (например, IBM Domino).

Когда пользователь отправляет e-mail, он сначала попадает к mail user-agent (почтовому клиенту; user-agent здесь имеет такое же значение, как и в HTTP(S) протоколах), который передает его на почтовый сервер. Почтовый сервер, в свою очередь, сверяется с таблицей DNS, чтобы передать его на почтовый обменник получателя. Из почтового обменника email попадает к агенту доставки почты (mail delivery agent), который отправляет письмо на почтовый сервер получателя.

Адресат заберет свое письмо при помощи протокола POP3 или IMAP. Каждому письму, пришедшему на этот адрес, присваивается некоторый численный идентификатор, а самому адресу — другой идентификатор. Если используется протокол POP3, то с почтового сервера будут выбраны все письма с численным ID большим, чем последнее полученное письмо. В случае протокола IMAP с почтового сервера будут получены все письма с персональным ID пользователя.

Отметим, что POP3 — это составное название. Сам протокол называется POP, а число — это название версии протокола. POP2 и POP1 в современной разработке не используются.

5. Заключение

Транспортный уровень и его протоколы играют решающую роль в обмене данными между устройствами в сети. Понимание транспортного уровня и его протоколов важно для создания эффективных и надежных веб-приложений. Выбор между TCP и UDP будет зависеть от конкретных требований разрабатываемого приложения, а использование других протоколов транспортного уровня в веб-разработке широко распространено и необходимо для множества приложений, таких как почта, голосовая связь или отправка/получение потокового видео.

Все протоколы, упомянутые в этом лонгриде, также имеют заранее предназначенные для них порты. Так, для HTTP зарезервирован порт 80, для SSL/TLS — 443 (как следствие, его же использует HTTPS). Почтовые клиенты используют порты 25, 110 и 143 для SMTP, POP3 и IMAP соответственно. Подробнее со списком зарезервированных портов можно ознакомиться на сайте IANA (Internet Assigned Numbers Authority).

Дополнительные материалы

- Что такое TCP/IP и как работает этот протокол: https://timeweb.com/ru/community/articles/chto-takoe-tcp-ip
- Что такое TCP/IP и зачем они нужны: https://thecode.media/tcp-ip/
- TCP и UDP — в чем разница: https://wiki.merionet.ru/seti/23/tcp-i-udp-v-chem-raznica/

- User-Agent:
  https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent
- Why is HTTP not secure:
  https://www.cloudflare.com/en-gb/learning/ssl/why-is-http-not-secure/
- Transport Layer Security:
  https://developer.mozilla.org/en-US/docs/Glossary/TLS
- Протокол пользовательских дейтаграмм:
  https://www.ibm.com/docs/ru/aix/7.1?topic=protocols-user-datagram-protocol
- Протоколы TCP/IP:
  https://www.ibm.com/docs/ru/aix/7.1?topic=protocol-tcpip-protocols