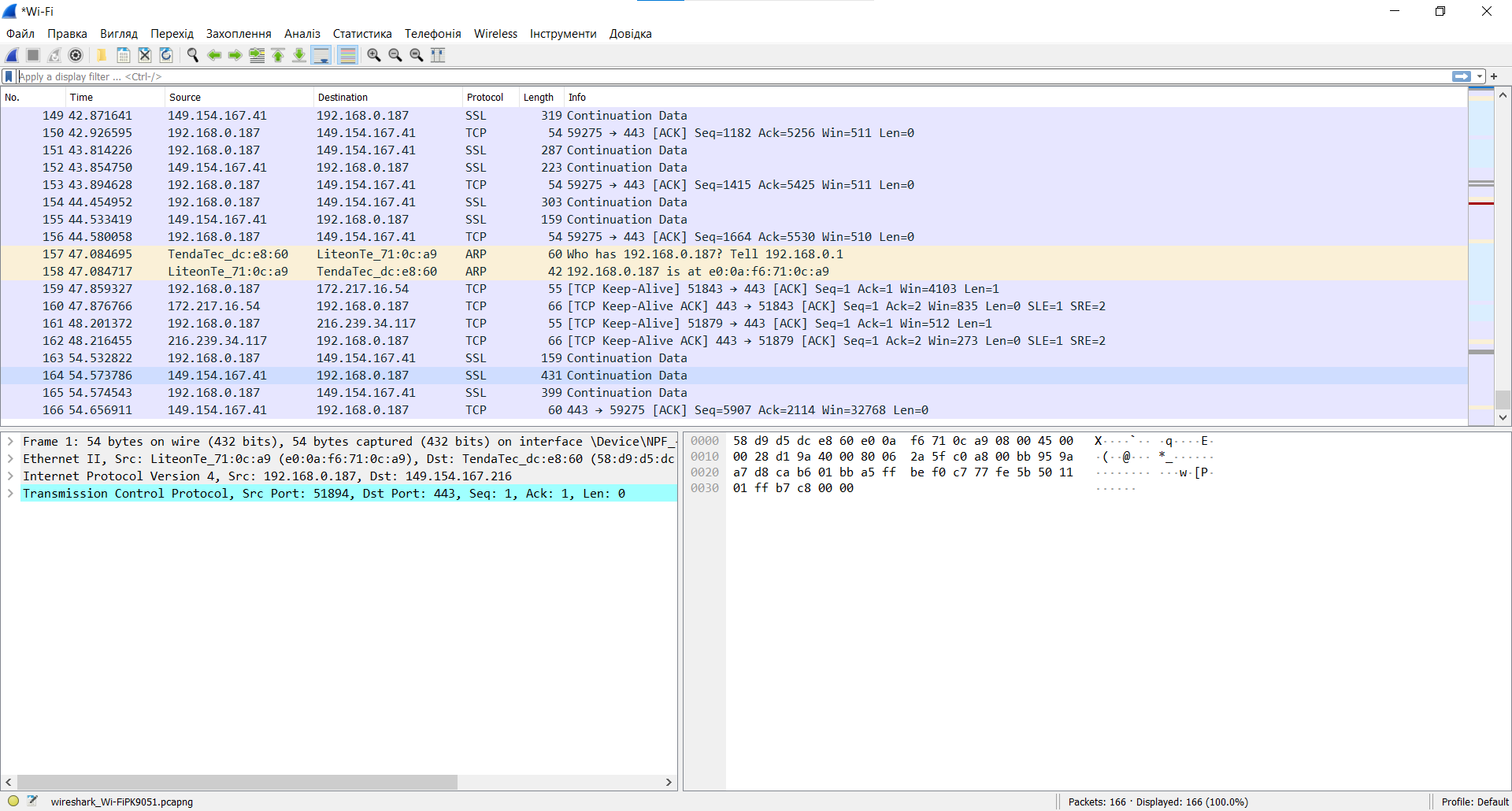
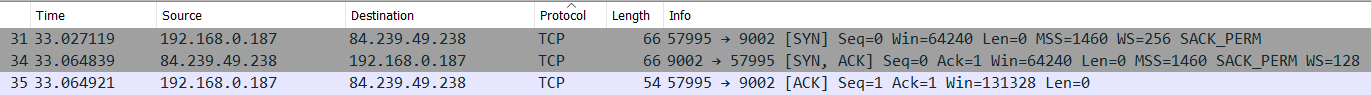
Інтерфейс Wireshark:



поділений на три частини: пакети які приходять і виходять – верхня частина, пакети у такому вигляді, як передаються по мережі – нижня права частина, що саме означають цифри пакетів – нижня ліва частина.

Для розбору TCP будемо брати наступне:



Для встановлення з'єднання в TCP використовується процедура трикратного рукостискання. Відправник та одержувач пересилають один одному три сегменти TCP із спеціально встановленими прапорами, які говорять про намір встановити з'єднання:

1. Відправник передає сегмент із встановленим прапором SYN.

2. Одержувач відповідає сегментом із встановленими прапорами SYN + ACK.

3. Відправник передає сегмент із прапором ACK.

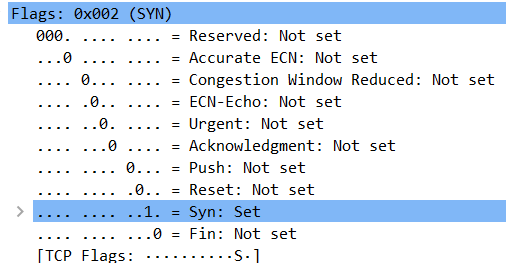
Після цих трьох кроків з'єднання вважається встановленим і можна передавати дані.

Розберемо перший пакет:



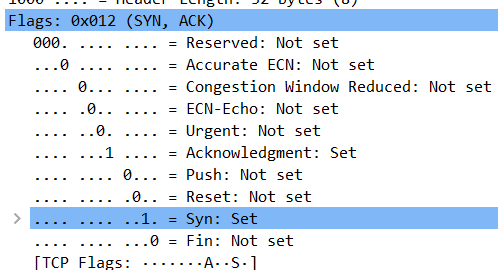
Порт відправника: 57995

Порт призначення: 9002

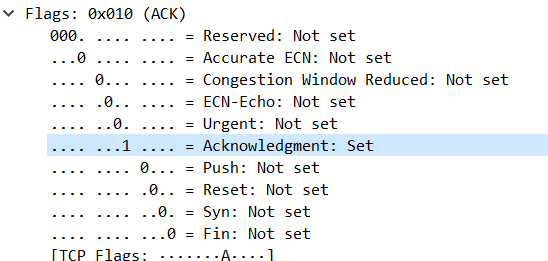


Всі флаги скинуті, окрім одного – Syn – запит на з’єднання.

У наступного будуть порти обернені, а флаги будуть наступні:



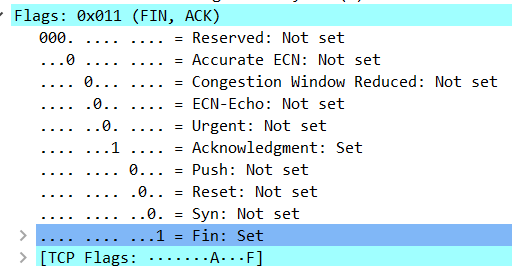
У останього флаг лише один – відповідь – тобто з’єднання дозволено:



Також бачимо що у кінцевого буде наступний номер цей номер дає можливість не перепутати пакети з різних мереж.

Після встановлення передаються дані та йде розрив. Для розриву з'єднання використовуються сегменти із встановленими прапорами FIN та ACK.



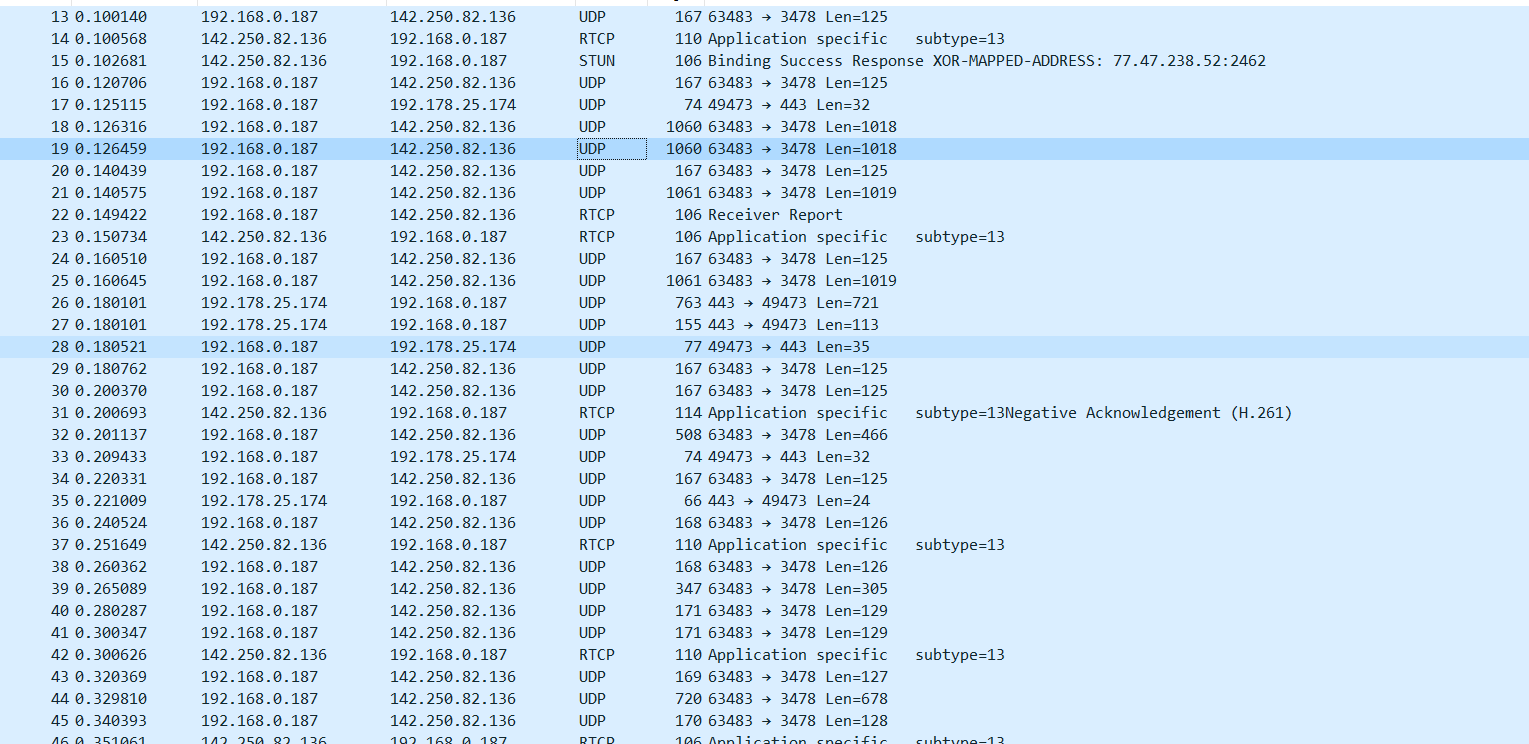


Розбір UDP:

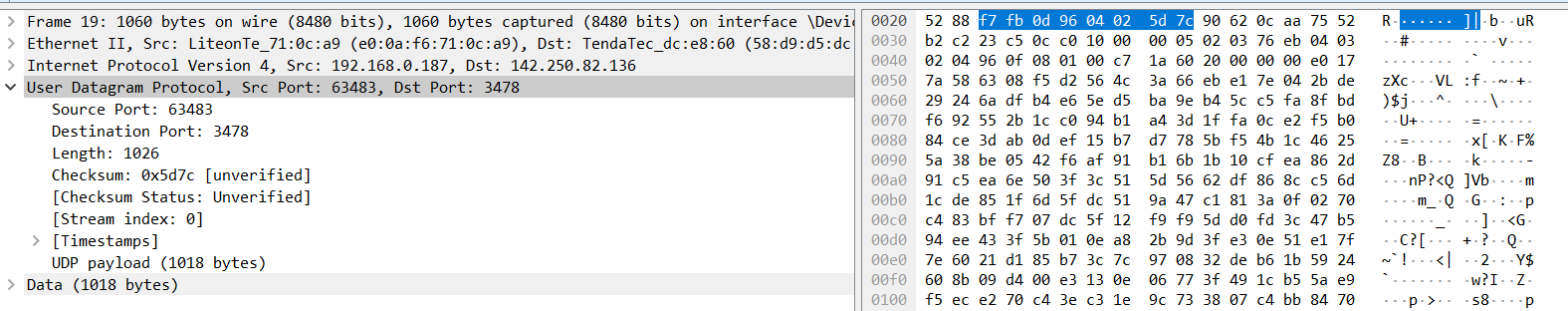
Цей протокол легший, порівняно із TCP.

Запустимо відеоконференцію у Google Meet та подивимося які пакети передаються.

Почали передаватися пакети:



Він має лише такі пункти:



Порт призначення та отримання, довжину та контрольну суму. Він не гарантує доставку і працює швидко.

Теорія:

**TCP - протокол транспортного рівня**

Першим із розглянутих протоколів буде TCP, або Transmission Control Protocol, який використовується для транспортування повідомлень між пристроями мережі.

У мережі файли не передаються повністю, а дробляться і передаються як відносно невеликих повідомлень. Далі вони передаються іншому пристрою – одержувачу, де повторно збираються у файл.

Наприклад, людина хоче завантажити картинку. Сервер обробляє запит і надсилає у відповідь потрібне зображення. Йому, у свою чергу, необхідний шлях або канал, яким він передаватиме інформацію. Тому сервер звертається до мережного сокету для встановлення необхідного з'єднання та надсилання картинки. Сервер дробить дані, інкапсулює їх у блоки, що передаються на рівень TCP одержувача за допомогою IP-протоколу. Далі отримувач підтверджує факт передачі.

У протоколу TCP є кілька особливостей:

* **Система нумерації сегментів.**TCP відстежує сегменти, що передаються і приймаються, присвоюючи номери кожному з них. Байтам даних, які мають бути передані, надається певний номер байта, тоді як сегментам присвоюються порядкові номери.
* **Управління потоком.**Функція обмежує швидкість, на яку відправник передає дані. Це робиться для забезпечення надійності доставки, у тому числі, щоб комп'ютер не генерував пакетів більше, ніж може прийняти інший пристрій. Якщо говорити простою мовою, то одержувач постійно повідомляє відправнику про те, який обсяг даних можна отримати.
* **Контроль помилок.**Функція реалізується підвищення надійності шляхом перевірки байтів на цілісність.
* **Контролює навантаження мережі.**Протокол TCP враховує рівень навантаження у мережі, який визначається обсягом даних, відправлених вузлом.

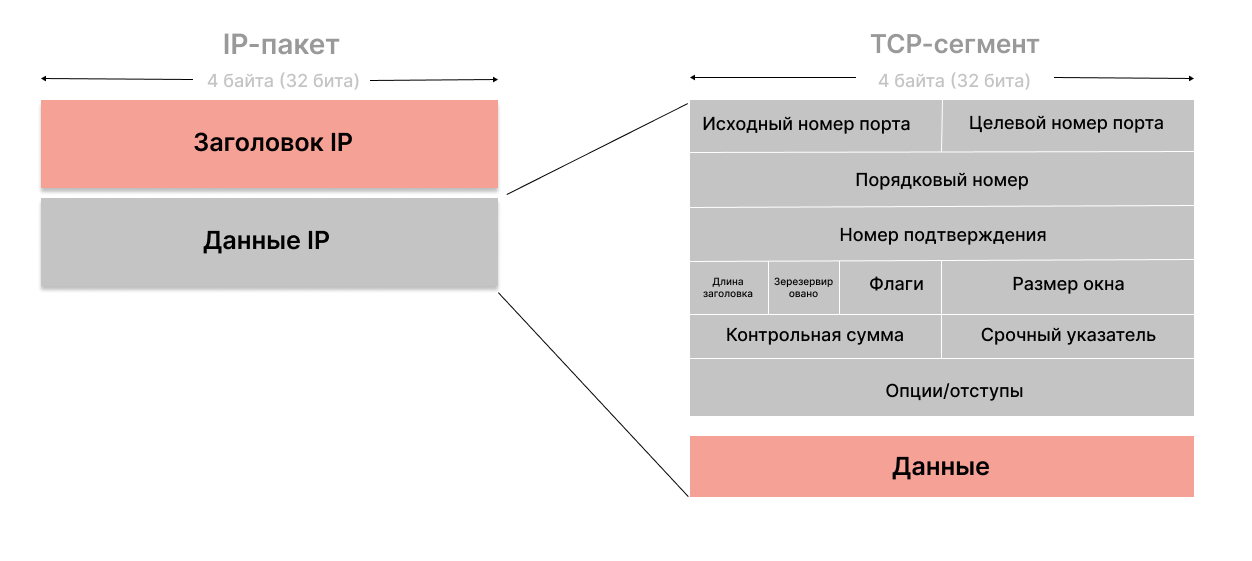
**Приклади застосування мережного протоколу TCP**

Протокол TCP гарантує доставку, а також забезпечує цілісність даних, що передаються в мережі. Тому він застосовується передачі даних, які чутливі до порушення цілісності, — наприклад, текстів, файлів тощо. Ось кілька протоколів, які працюють за TCP:

* SSH, FTP, Telnet: у даних протоколах TCP використовується обмінюватись файлами.
* SMTP, POP, IMAP: протоколи, де TCP відповідає за надсилання електронних листів.
* HTTP/HTTPS: протоколи, де TCP відповідає за завантаження сторінок з Інтернету.

Ці приклади працюють лише на рівні додатків стека TCP/IP і передають дані до TCP, на транспортний рівень.

**Будова протоколу TCP**



У кожен пакет даних TCP додає заголовок загальним обсягом 20 байт (або октетів), в якому містяться 10 обов'язкових полів:

* **Порт джерела** – порт пристрою-відправника.
* **Порт призначення** — порт пристрою, що приймає.
* **Порядковий номер.**Пристрій, який ініціює TCP-з'єднання, повинен вибрати випадковий початковий порядковий номер, який потім збільшується відповідно до кількості переданих байтів.
* **Номер підтвердження.**Пристрій збільшує цей номер з нуля відповідно до кількості отриманих байтів.
* **Зсув даних TCP.**Цей параметр визначає розмір заголовка, щоб система могла зрозуміти, де починаються дані.
* **Зарезервовані дані** – зарезервоване поле, значення якого завжди дорівнює нулю.
* **Прапори керування.**TCP використовує дев'ять прапорів керувати потоком даних у певних ситуаціях — наприклад, під час ініціювання скидання сесії.
* **Розмір вікна** – найважливіша частина заголовка TCP. Це поле використовується одержувачем для вказівки відправнику обсягу даних, які може прийняти.
* **Контрольна сума** . Відправник генерує контрольну суму і передає в заголовку кожного пакета. Пристрій може використовувати контрольну суму для перевірки помилок в отриманому файлі.
* **Терміновий покажчик** — це пропонована протоколом можливість позначати деякі байти даних тегом «Терміново» для їх пересилання та обробки позачергово.
* **Поле опції.**Може використовуватися для розширення протоколу або тестування.

**UDP – протокол транспортного рівня**

Якщо нам дуже важлива швидкість передачі, а ось втрата пакетів не така критична (як, наприклад, у голосовому або відеотрафіку), то краще використовувати UDP, або User Datagram Protocol. На відміну від TCP, він забезпечує передачу даних без отримання підтвердження від користувача. Простіше кажучи, просто відправляє пакети і не чекає нічого у відповідь. Через це досягається висока швидкість на шкоду надійності.

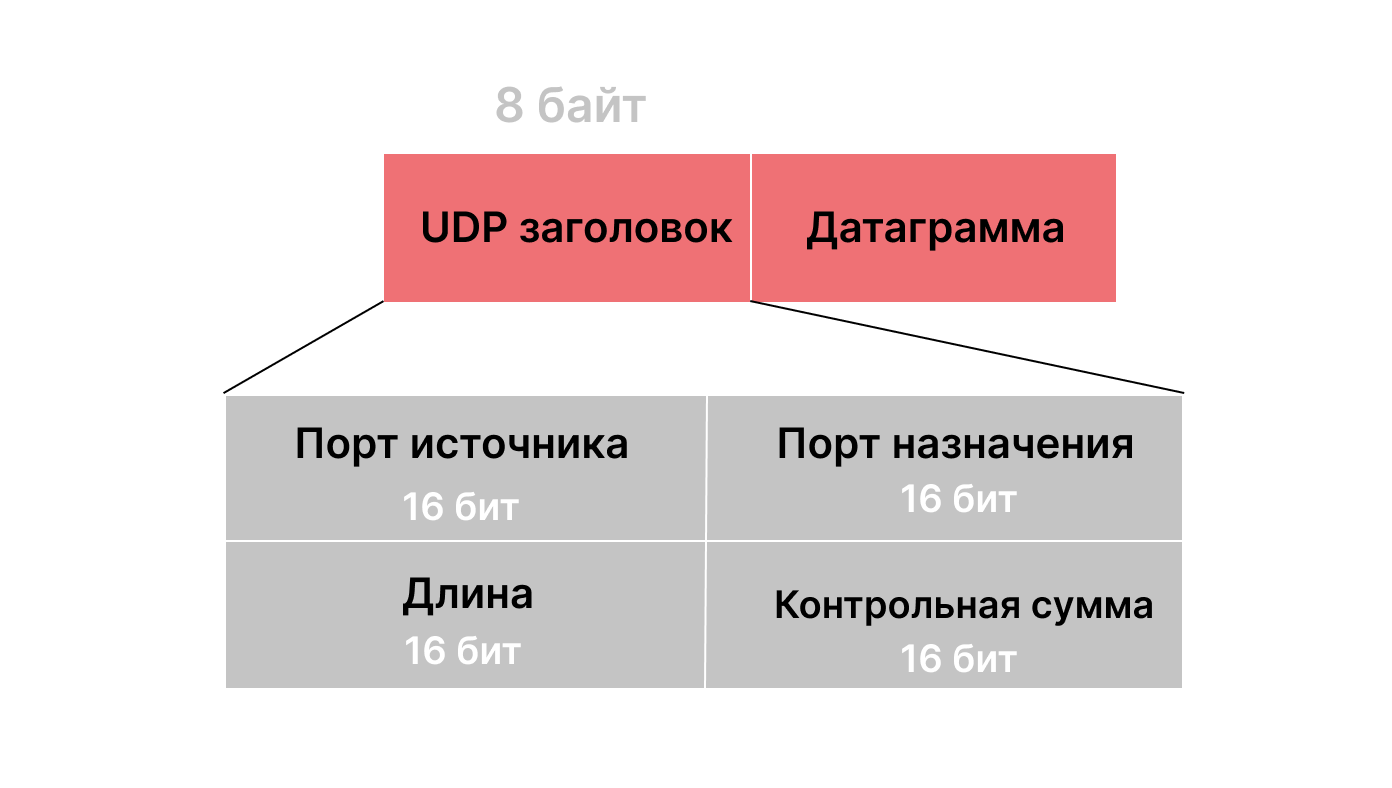
Найчастіше UDP застосовується в чутливих на час службах, де втратити пакети краще, ніж чекати. Дзвінки в Skype або Google Meet, стрімінг відео, онлайн-трансляції використовують цей протокол через те, що вони є чутливими до часу і розраховані на певний рівень втрат. Весь голосовий зв'язок через інтернет працює за протоколом UDP. Також UDP часто використовується в онлайн-іграх. Аналогічна історія з DNS-серверами, оскільки вони мають бути швидкими та ефективними.

**Приклади використання протоколу UDP**

Прикладами протоколів, які використовують UDP-протокол, є:

* DNS — протокол, що перетворює домени на IP-адреси, щоб зробити можливим завантаження інтернет-ресурсу через браузер.
* SNMP — протокол, який дозволяє системному адміністратору проводити моніторинг, контролювати продуктивність мережі та змінювати конфігурацію підключених пристроїв.
* DHCP - протокол, який відповідає за автоматичне призначення IP-адреси клієнту.

**Структура пакета UDP.**



**Різниця між TCP та UDP**

Ключовою відмінністю між TCP і UDP є швидкість, оскільки TCP порівняно повільніше за UDP. В цілому, UDP є швидким, простим і ефективним протоколом, проте повторна передача втрачених пакетів даних можлива лише TCP.

Ще одна помітна відмінність між TCP та UDP полягає в тому, що перший забезпечує впорядковану доставку даних від користувача до сервера (і навпаки). UDP, своєю чергою, не перевіряє готовність одержувача і може доставляти пакети вроздріб.

Розглянемо різницю параметрів протоколів TCP і UDP.

|  |  |  |
| --- | --- | --- |
|  | TCP | UDP |
| **Стан з'єднання** | Потрібне з'єднання для передачі даних (з'єднання має бути закрите після завершення передачі) | Протокол без з'єднання, без вимог до відкриття, підтримки чи переривання з'єднання |
| **Гарантія доставки** | Може гарантувати доставку даних одержувачу | Не гарантує доставку даних одержувачу |
| **Повторна передача даних** | Повторна передача кількох кадрів у разі втрати одного з них | Відсутність повторної передачі втрачених пакетів |
| **Перевірка помилок** | Повна перевірка помилок | Базовий механізм перевірки помилок. Використовує протоколи для перевірки цілісності. |
| **Метод передачі** | Дані зчитуються як потік байтів; повідомлення передаються за межами сегментів | UDP-пакети з певними межами; відправляються окремо та перевіряються на цілісність після прибуття |
| **Сфера застосування** | Використовується для надсилання повідомлень електронної пошти, HTML-сторінок браузерів | Відеоконференції, потокове мовлення, DNS, VoIP, IPTV |
|  |  |  |

**Відмінності TCP та UDP при використанні у VPN**

Також нерідко виникає питання щодо використання даних протоколів при VPN-з'єднаннях. Наприклад, в OpenVPN існує можливість вибору між TCP-і UDP-протоколами.

Умовимося, що VPN загортає дані в ще один протокол (насправді все набагато складніше). Якщо ваш VPN-тунель використовує як транспортний протокол TCP, то передача даних UDP-протоколу втрачає свої переваги. Як мінімум на певній ділянці колії. Тому для VPN-тунелю радять використовувати UDP-протокол, адже TCP штатно працюватиме всередині UDP-тунелю.

**Підсумок порівняння протоколів TCP та UDP**

Кожен протокол хороший під свої завдання, недарма вони є одними із найпоширеніших в інтернеті. На завершення порівняння TCP і UDP можна назвати, що TCP застосовується там, де важливо доставити всі дані у порядку. Зона застосування UDP, у свою чергу, - це голосовий та відеотрафік, де доставка всіх пакетів не є обов'язковою.

Також серйозною відмінністю TCP від ​​UDP є розмір заголовків. У TCP він становить 20-60 байт, а UDP — всього 8 байт. Це показує, наскільки складніше влаштований протокол TCP, адже він пріорітізує трафік та перевіряє блоки даних на наявність помилок.