Czytaj lub słuchaj artykuł online

Zeskanuj kod QR, aby otworzyć artykuł na XO.PL

Źródło artykułu:

https://xo.pl/it-na-wesolo-net-tcp-udp

SIECI KOMPUTEROWE

TCP i UDP, czyli przesyłka polecona kontra dostawa bez potwierdzenia

Witamy ponownie na naszej Autostradzie Pakietów, zwanej potocznie Internetem!

Na osiedlu Spokojna Sieć sytuacja jest niemal opanowana. Urząd **DHCP** rozdaje tablice rejestracyjne. Bramkarz od **NAT-u** odnalazł swój zeszyt, ale przez pomyłkę wpisał go do rejestru rzeczy zaginionych. Lodówka nadal nie wie, gdzie jest mleko.

Tymczasem na osiedlu otwarto dwie firmy kurierskie. Obie obiecują dostarczać informacje. Obie mają samochody. Obie korzystają z tej samej autostrady.

I na tym podobieństwa właściwie się kończą.

Dwie firmy, dwa światy

Pierwsza firma nazywa się **TCP**.

Na jej drzwiach widnieje napis:

„Przesyłki dostarczamy w odpowiedniej kolejności. Braki uzupełniamy. Potwierdzenia obowiązkowe”.

W recepcji siedzi elegancki pan w garniturze. Na biurku ma segregator, pieczątkę i telefon, którym potwierdza, że odebrał poprzedni telefon.

Druga firma to **UDP**.

Na drzwiach napis:

„Szybko. Prosto. Bez zbędnych pytań”.

Za biurkiem nikogo nie ma. Na ścianie wisi kartka:

„JEŚLI CZYTASZ TEN NAPIS, KURIER JUŻ WYJECHAŁ”.

Pewien klient postanawia sprawdzić obie firmy. Najpierw idzie do TCP.

— Dzień dobry. Chciałbym wysłać wiadomość.

— Oczywiście. Najpierw musimy nawiązać połączenie.

— Ale przecież rozmawiamy!

— Prywatnie tak. Służbowo jeszcze nie.

— TO CO MAM ZROBIĆ?!

— Proszę rozpocząć procedurę powitania.

— DZIEŃ DOBRY!

— Dzień dobry. Potwierdzam pańskie powitanie i przesyłam własne.

— POTWIERDZAM PAŃSKIE POWITANIE!

— Doskonale. Połączenie ustanowione.

— TO TERAZ MOGĘ WYSŁAĆ PACZKĘ?!

— Oczywiście.

— NA LITOŚĆ BOSKĄ!

To właśnie **TCP (Transmission Control Protocol)** — protokół zapewniający niezawodny, uporządkowany strumień bajtów pomiędzy dwiema stronami połączenia.

A cała ceremonia powitalna to **three-way handshake**, czyli trójetapowe uzgadnianie połączenia.

SYN, SYN-ACK i ACK, czyli trzy razy dzień dobry

W centrali TCP każdy etap powitania ma własny formularz.

Trójetapowe powitanie TCP



Uproszczony schemat nawiązania połączenia TCP.

— SYN: Chciałbym rozpocząć rozmowę!

— SYN-ACK: Słyszę pana i również jestem gotowy!

— ACK: Potwierdzam, że usłyszałem pańskie potwierdzenie!

— Możemy zaczynać?

— Teraz tak.

— A NIE MOGLIŚMY OD RAZU?!

— Wtedy nie wiedzielibyśmy, czy druga strona jest gotowa.

— PRZECIEŻ ODEBRAŁA TELEFON!

— Proszę nie mieszać procedur prywatnych ze służbowymi.

TCP używa przy tym numerów sekwencyjnych, uzgadnia parametry komunikacji i tworzy stan połączenia po obu stronach. Nie chodzi więc tylko o uprzejmość. Obie strony muszą ustalić, że mogą rozpocząć wymianę danych.

UDP, czyli człowiek, którego nie sposób zatrzymać

Klient, nieco zmęczony ceremoniami TCP, idzie do konkurencji.

— Dzień dobry, chciałbym wysłać paczkę do Krakowa.

— JUŻ WYSŁANA!

— Ale ja jeszcze jej panu nie dałem!

— TO PROSZĘ DAĆ NASTĘPNĄ!

— A adres?!

— MA PAN ADRES?!

— Mam!

— TO DAWAĆ!

— A potwierdzenie odbioru?

— NIE PROWADZIMY!

— A JEŚLI PACZKA ZGINIE?!

— TRUDNO!

— A JEŚLI DOTRZE PÓŹNIEJ NIŻ NASTĘPNA?!

— TO BĘDZIE PÓŹNIEJ!

— A JEŚLI NIE DOTRZE WCALE?!

— TO NIE DOTRZE!

— PAN JEST JAKIMŚ WARIATEM?!

— NIE, PROSZĘ PANA. **UDP!** 🤣

UDP (User Datagram Protocol) jest protokołem bezpołączeniowym. Wysyła datagramy bez wcześniejszego ustanawiania połączenia i sam nie zapewnia potwierdzeń, retransmisji ani kolejności dostarczenia.

Ale uwaga! UDP nie jest zepsutym TCP. To po prostu firma, która ma inny zakres obowiązków. Nie każde zastosowanie potrzebuje całej ceremonii.

Paczka numer 3 przyjechała przed paczką numer 2

TCP przyjmuje zlecenie dostarczenia większego dokumentu. Dzieli go na części i wysyła na autostradę.

Pierwsza przesyłka dociera. Druga utknęła w korku. Trzecia dojechała wcześniej.

Odbiorca otwiera drzwi.

— Dzień dobry, paczka numer trzy!

— A gdzie numer dwa?

— Nie wiem.

— To proszę poczekać.

— ALE PRZECIEŻ JESTEM JUŻ NA MIEJSCU!

— Nie mogę jeszcze przekazać zawartości dalej w odpowiedniej kolejności.

— TO NIECH PAN ZADZWONI DO NADAWCY!

— Już to sygnalizujemy odpowiednimi potwierdzeniami.

Tymczasem w centrali:

— Szefie! Odbiorca dostał dane z dalszej części, ale brakuje wcześniejszej!

— To sprawdzić potwierdzenia!

— WCIĄŻ POTWIERDZA WCZEŚNIEJSZY NUMER!

— To uruchomić retransmisję, gdy spełnione będą warunki!

— A TRZECIA PACZKA?!

— Może poczekać w buforze!

W TCP numery sekwencyjne dotyczą **pozycji bajtów w strumieniu danych**, a nie dosłownie ponumerowanych paczek kurierskich.

TCP potrafi rozpoznawać brakujące fragmenty i retransmitować dane, których dostarczenia nie potwierdzono. Odbiorca może przechowywać dane, które przyszły poza kolejnością, ale udostępnia aplikacji uporządkowany strumień.

— Czyli paczka numer trzy może już być u mnie, ale mam czekać na numer dwa?!

— Tak.

— A MOGĘ CHOCIAŻ ZOBACZYĆ, CO JEST W ŚRODKU?!

— Nie w ramach normalnego strumienia dostarczanego aplikacji.

— JA TYLKO CHCIAŁEM PRZECZYTAĆ PRZEPIS NA NALEŚNIKI!

— Proszę zaczekać na brakujący fragment dotyczący jajek.

— ALE JA JUŻ MAM FRAGMENT DOTYCZĄCY PATELNI!

— Bez jajek procedura przewiduje dalsze oczekiwanie.

UDP i transmisja na żywo

Tymczasem UDP obsługuje transmisję meczu.

Komentator krzyczy:

— ZAWODNIK PODAJE PIŁKĘ!

Datagram odjeżdża.

— ZAWODNIK STRZELA!

Kolejny odjeżdża.

— GOOOOOOL!

Trzeci odjeżdża. Pierwszy dociera. Trzeci dociera. Drugi ginie w korku.

Odbiorca słyszy:

— ZAWODNIK PODAJE PIŁKĘ!

— GOOOOOOL!

— ALE KIEDY ON STRZELIŁ?!

Kierownik UDP rozkłada ręce.

— Nie wiem. Nie mamy tego fragmentu.

— TO PROSZĘ GO DOSŁAĆ!

— Zanim dojedzie, mecz będzie już w innej fazie.

— ALE JA NIE WIDZIAŁEM STRZAŁU!

— Za chwilę może pan zobaczyć następny.

I właśnie dlatego w wielu zastosowaniach czasu rzeczywistego bardziej opłaca się tolerować drobną utratę danych niż czekać na ich ponowne dostarczenie.

Nie oznacza to, że każda transmisja audio lub wideo zawsze używa UDP. Ani że aplikacja nie może stosować własnych mechanizmów naprawiania strat.

UDP daje aplikacji swobodę wyboru — co zrobić, kiedy coś nie dotrze.

— Czyli pan wcale nie jest nieodpowiedzialny?!

— Jestem odpowiedzialny za to, co obiecałem.

— A CO PAN OBIECAŁ?!

— Że wyślę.

— I TYLE?!

— I tyle.

Numery portów, czyli do którego okienka z przesyłką

Na naszej autostradzie **adres IP** wskazuje, dokąd ma dotrzeć samochód. Ale w jednym budynku może działać wiele biur.

Pod jednym adresem mieszka serwer WWW, serwer pocztowy i jeszcze jakieś dziwne urządzenie, które nikt nie pamięta, po co zostało zainstalowane.

Kurier dojeżdża.

— Dzień dobry, mam przesyłkę!

— Do kogo?

— Pod ten adres.

— To jest cały budynek.

— NO TO PROSZĘ PRZYJĄĆ!

— Ale dla którego biura?!

— NIE WIEM!

— Numer portu?!

— JAKIEGO PORTU?!

— Sieciowego!

— TO NIECH PAN OD RAZU MÓWI, ŻE BIURO MA NUMER!

Przykładowe porty usług:

Port	Z czym zwykle się kojarzy?
TCP 80	**HTTP**
TCP 443	**HTTPS**
TCP 22	SSH
TCP 25	SMTP
UDP 53	**DNS** — część zapytań
UDP 123	NTP

Ważne: **TCP i UDP mają oddzielne przestrzenie numerów portów.** Numer 53 w TCP i numer 53 w UDP nie oznacza jednego wspólnego okienka protokołu.

— Czyli mogę mieć okienko 53 dla TCP i 53 dla UDP?!

— Tak.

— I NIE BĘDĄ SIĘ KŁÓCIĆ?!

— Nie o sam numer.

— TO DLACZEGO TAMTEN URZĘDNIK OD **NAT**-U MIAŁ TYLE PROBLEMÓW?!

— Bo jemu proszę pana zawsze ktoś przyjeżdża bez kompletu dokumentów.

W artykule o **NAT i maskaradzie** poznaliśmy już zarządcę, który zapisuje adresy i porty w swoim słynnym zeszycie. Teraz wiemy również, dlaczego musi rozróżniać protokoły.

Okno transmisyjne, czyli ile paczek można wysłać bez czekania

W firmie TCP pojawia się nowy klient.

— Mam dziesięć tysięcy paczek do wysłania.

— W porządku.

— To niech wszystkie jadą!

— Nie od razu.

— DLACZEGO?!

— Odbiorca musi mieć miejsce, żeby je przyjmować.

— A ILE MA?!

— Zgłosił nam dostępną pojemność.

— TO NIECH PAN WYSYŁA DO TEGO LIMITU!

— Właśnie tak robimy.

TCP wykorzystuje **mechanizm okna odbiorczego (receive window)**, czyli kontrolę przepływu (*flow control*). Odbiorca informuje nadawcę, ile danych może jeszcze przyjąć do swojego bufora.

— A jeśli odbiorca ma pełen magazyn?!

— To informuje o braku miejsca.

— I CO WTEDY?!

— Ograniczamy wysyłanie.

— A JEŚLI JA CHCĘ WYSŁAĆ?!

— To proszę poczekać, aż będzie gdzie rozładować.

— TO JEST FIRMA KURIERSKA CZY PRZECHOWALNIA BAGAŻU?!

— Zależy od aktualnego stanu magazynu.

Ale jest jeszcze drugi limit. Bo nawet jeśli odbiorca ma ogromny magazyn, sama autostrada może być zatkana.

Autostrada jest zatkana, a TCP wyciąga kalkulator

Do centrali TCP dzwoni zarządca węzła.

— Dzień dobry. Mamy korek.

— Jak duży?

— Bardzo.

— Czy gubimy przesyłki?

— Zaczynamy.

Kierownik TCP marszczy brwi.

— To trzeba ograniczyć tempo.

— ALE MAMY JESZCZE TYSIĄC PACZEK!

— Tym bardziej nie możemy wrzucić ich wszystkich naraz.

TCP stosuje mechanizmy **kontroli przeciążenia (congestion control)**.

W praktyce rozmiar danych, które można mieć jednocześnie w drodze, jest ograniczany między innymi przez okno odbiorcy oraz mechanizmy kontroli przeciążenia po stronie nadawcy.

— Czyli odbiorca mówi, ile może pomieścić?

— Tak.

— A autostrada mówi, ile może przewieźć?

— Nie dosłownie. Nadawca ocenia sytuację na podstawie sygnałów z sieci i zachowania połączenia.

— I MA PAN NA TO JAKIŚ ALGORYTM?!

— Oczywiście.

— A KTÓRY?!

— To zależy od implementacji i konfiguracji.

— ZNOWU „TO ZALEŻY”?!

— Witamy w informatyce.

Mechanizmy takie jak **CUBIC** czy **BBR** mogą w różny sposób sterować wysyłaniem.

Nie będziemy jednak teraz otwierać osobnego wydziału matematyki drogowej. Zarządca już i tak cierpi.

UDP też może mieć zasady!

Do firmy UDP przychodzi klient z reklamacją.

— Słyszałem, że wysyłacie paczki bez kontroli!

— Standardowy UDP nie zapewnia niezawodnego dostarczania.

— CZYLI JESTEŚCIE DO NICZEGO!

— Niezupełnie.

— A CO JEŚLI JA CHCĘ KONTROLOWAĆ DOSTAWY?!

— Może pan zrobić to w swojej aplikacji.

— SAM?!

— Tak.

— TO PO CO MI WASZA FIRMA?!

— Bo zapewniamy prosty sposób przesyłania datagramów, bez obowiązkowego ustanawiania połączenia i całej obsługi TCP.

— ALE JA CHCĘ SZYBKO I NIEZAWODNIE!

— To może pan zbudować odpowiednie mechanizmy ponad UDP.

— NA PRZYKŁAD?!

— QUIC.

— CO TO JEST?!

— Protokół transportowy wykorzystujący UDP, który może zapewniać niezawodność, **szyfrowanie** i niezależne strumienie.

— CZYLI UDP POTRAFI BYĆ NIEZAWODNE?!

— Samo UDP nie. Ale protokoły nad nim mogą.

— TO DLACZEGO PAN MI TEGO NIE POWIEDZIAŁ OD RAZU?!

— Bo nie zapytał pan o transport z dodatkowym wyposażeniem.

— TO JA CHCĘ Z DODATKOWYM!

— Proszę przejść do działu aplikacji.

— A GDZIE ON JEST?!

— W następnym budynku.

— A NIECH TO JASNA CHOLERA!

Nie należy utożsamiać UDP wyłącznie z zawodnymi transmisjami na żywo. To podstawa wielu różnych rozwiązań, które same określają potrzebne mechanizmy transmisji.

Zamykamy połączenie, czyli TCP nie wyjdzie bez pożegnania

Po długim dniu klient firmy TCP kończy rozmowę.

— Dziękuję. To wszystko.

— Przyjąłem. Chce pan zakończyć wysyłanie?

— Tak.

— W takim razie proszę przesłać FIN.

— CO PRZESŁAĆ?!

— Informację o zakończeniu wysyłania.

— Przecież właśnie powiedziałem, że kończę!

— Proszę to zrobić protokołem.

— FIN!

— ACK. Przyjąłem.

— To mogę już iść?!

— Druga strona może jeszcze mieć własne dane do wysłania.

— ALE JA JUŻ NIC NIE CHCĘ!

— Proszę zaczekać na jej zakończenie.

Po chwili:

— FIN.

— ACK.

— I TERAZ?!

— Teraz możemy zakończyć wymianę.

— NARESZCIE!

— Proszę pamiętać o ewentualnym stanie TIME_WAIT.

— O CZYM?!

— To mechanizm, który pomaga uniknąć pomylenia starych segmentów z nowym połączeniem.

— ALE JA JUŻ SIĘ POŻEGNAŁEM!

— Procedury jeszcze trwają.

— JA SIĘ TU ZESTARZEJĘ!

TCP pozwala każdej stronie niezależnie zakończyć swoją część wysyłania, dlatego uporządkowane zamykanie połączenia często obejmuje wymianę komunikatów **FIN i ACK** w obu kierunkach.

Istnieje również **RST**, czyli reset połączenia. I to już zupełnie inny styl zakończenia rozmowy.

— Dzień dobry, chciałem tylko...

— RST!

TRZASK!

— HALO?!

— ...

— NO I CO, ŻE TCP JEST UPRZEJME?! 🤣

A jak kończy UDP?

— Dziękuję, to wszystko!

Cisza.

— Halo?!

Cisza.

— PANIE KURIERZE?!

Z oddali:

— JUŻ JESTEM W KRAKOWIE!

— A POŻEGNANIE?!

— NIE MAMY TAKIEJ PROCEDURY!

UDP nie ma własnego mechanizmu nawiązywania ani zamykania połączenia takiego jak TCP. Jeśli aplikacja potrzebuje powitań, pożegnań albo informacji o zakończeniu sesji, sama musi się tym zająć.

Co naprawdę robią nasi przewoźnicy?

TCP i UDP działają ponad IP. To IP odpowiada za adresowanie i przekazywanie pakietów między sieciami, a TCP i UDP zapewniają różne sposoby komunikacji między aplikacjami.

Cecha	TCP	UDP
Połączenie	Wymaga ustanowienia	Bezpołączeniowy
Potwierdzenia	Tak	Nie w samym UDP
Retransmisje	Tak, gdy potrzebne	Nie w samym UDP
Kolejność danych	Uporządkowany strumień	Brak gwarancji kolejności

Granice wiadomości	Strumień bajtów	Zachowuje granice datagramów
Kontrola przepływu	Tak	Nie w samym UDP
Kontrola przeciążenia	Wbudowane mechanizmy	Zależy od aplikacji lub protokołu wyżej
Typowe przykłady	SSH, SMTP, tradycyjny **HTTP**/1.1 i **HTTP**/2	**DNS**, NTP, wiele transmisji na żywo, QUIC

Nie ma jednego zwycięzcy.

TCP jest dobrym wyborem, gdy aplikacja potrzebuje niezawodnego, uporządkowanego strumienia danych.

UDP sprawdza się tam, gdzie potrzebne są datagramy i aplikacja chce samodzielnie zdecydować, jakie dodatkowe mechanizmy zastosować.

Oczywiście rzeczywiste aplikacje potrafią wykorzystywać oba podejścia.

A kiedy przesyłki utkną w korkach, wrócimy do naszej historii o **routingu i Autostradzie Pakietów**, gdzie poznaliśmy już opóźnienia, kolejki i zaginione samochody.

Jeśli chcesz przejechać dalej po naszej Autostradzie Pakietów, zajrzyj także do tematów: **adresach IP**, **DHCP**, **DNS**.

Tymczasem na osiedlu Spokojna Sieć...

Do punktu odbioru przesyłek podjeżdża lodówka.

— Dzień dobry. Oczekuję dostawy mleka.

Kurier TCP sprawdza dokumenty.

— Numer zamówienia?

— 153.

— Numer przesyłki?

— 154.

— Potwierdzenie nadania?

— Nie mam.

— To nie mogę wydać.

— ALE MLEKO JUŻ PRZYJECHAŁO!

— Tak, ale brakuje nam wcześniejszego potwierdzenia.

— TO PROSZĘ JE ZNALEŹĆ!

— Procedura retransmisji została uruchomiona.

Z drugiej strony podjeżdża kierowca UDP.

— MLEKO?!

— TAK!

— PROSZĘ!

Wyrzuca karton przez okno i odjeżdża.

Lodówka patrzy na karton. Otwiera go. W środku jest kefir.

— HALO?! TO NIE JEST MLEKO!

Kurier UDP z oddali:

— PACZKA DOTARŁA?!

— DOTARŁA, ALE NIE TA!

— TO JUŻ NIE MOJA WARSTWA!

Lodówka spogląda na TCP.

— A pan co robi?

— Czekam na potwierdzenie.

— CZEGO?!

— Że pani odebrała kefir.

— ALE TO BYŁO UDP!

— Aha. To przepraszam.

Chwila ciszy.

— To może mi pan chociaż dostarczy mleko?

— Oczywiście.

— KIEDY?!

— Najpierw ustalimy połączenie.

— ...

— SYN?

— **NIECH PAN SOBIE TEN SYN WSADZI DO SEGREGATORA!!!**

Z pobliskiego urzędu **DHCP** wychyla się urzędnik.

— PROSZĘ NIE KRZYCZEĆ! MY TU PRÓBUJEMY ROZWIESIĆ PRANIE!

A zarządca **NAT-u**, który właśnie przechodził obok, pyta:

— A kto podpisał odbiór?

— NIKT!

— To ja zapiszę w zeszycie.

— **NIECH PAN TEN ZESZYT WRESZCIE ZAMKNIE!!!** 🤣🤣

---

*Na kolejnej wycieczce możemy sprawdzić, dlaczego niektóre samochody krążą w kółko, aż kończy im się licznik przejazdów. Poznamy **TTL** i urzędnika, który ma tylko jedną odpowiedź: „Limit wyczerpany. Następny!”.*

Firma TCP potwierdziła otrzymanie skargi. Firma UDP prawdopodobnie też. Nie pytałyśmy.