Programowanie klient – serwer

Protokoły

Protokół

Protokół – sformalizowane zasady postępowania.

W przypadku komunikacji między dwoma systemami musi być zdefiniowany **zestaw reguł** rządzących wymianą informacji między nimi.

TCP/IP

Nazwa TCP/IP odnosi się do całego zestawu protokołów komunikacyjnych. Zestaw wziął swoją nazwę od dwóch najważniejszych z nich:

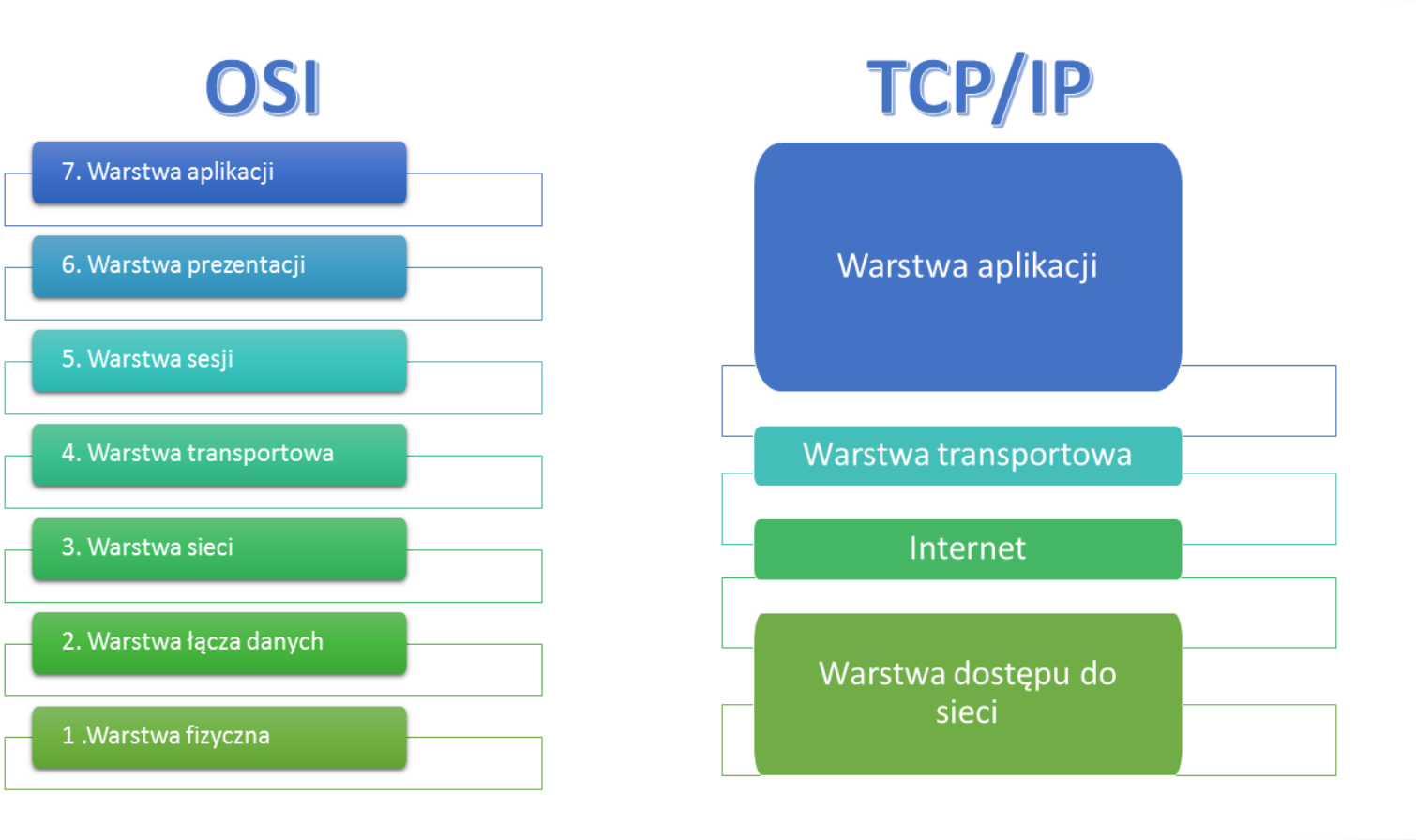
- protokołu kontroli transmisji (*Transmission Control Protocol*, TCP)
- protokołu internetowego (*Internet Protocol*, IP)

TCP/IP udostępnia metody transmisji informacji pomiędzy poszczególnymi hostami w sieci, obsługując pojawiające się błędy oraz tworząc wymagane do transmisji informacje dodatkowe.

TCP/IP

- Otwarty standard, dostępny bezpłatnie, niezależny od platformy sprzętowej czy programowej. Idealny do łączenia wielu różnych komputerów i systemów operacyjnych,
- Niezależność od fizycznej, sprzętowej warstwy sieci, integracja różnych sieci,
- Jednolity system adresowania, pozwalający w identyczny sposób, zaadresować każde urządzenie w sieci, nawet tak dużej jak Internet,
- Standaryzowany protokół wysokiego poziomu, implementujący wspólne i ogólnodostępne usługi sieciowe.

OSI – TCP/IP


OSI
7. Warstwa aplikacji
6. Warstwa prezentacji
5. Warstwa sesji
4. Warstwa transportowa
3. Warstwa sieci
2. Warstwa łącza danych
1 .Warstwa fizyczna
TCP/IP
Warstwa aplikacji
Warstwa transportowa
Internet
Warstwa dostępu do sieci

Warstwa dostępu do sieci

- Najniższa warstwa,
- Protokoły umożliwiające przekazanie danych innym urządzeniom dołączonym do sieci,
- Definiują sposób użycia sieci w celu wysłania lub odebrania datagramu IP,
- Znają cechy sieci fizycznej, do której dołączony jest system (struktura ramek, sposób adresowania),
- Odpowiednik warstwy łącza danych i fizycznej z modeli OSI,
- Opakowanie datagramów IP w ramki akceptowane przez sieć oraz przekształcanie adresów IP na fizyczne adresy urządzeń w sieci, RFC 826, RFC 894.

Warstwa sieciowa

- Definiuje format datagramu,
- Definiuje schemat adresowania w Internecie,
- Przesyła dane pomiędzy warstwą dostępu do sieci a warstwą transportową,
- Routuje datagramy,
- Dokonuje fragmentacji i ponownej defragmentacji datagramów.

Protokół IP

Najważniejszy protokół komunikacyjny warstwy sieciowej.

- Protokół bezpołączeniowy – nie wysyła i nie przyjmuje żadnych informacji kontrolnych, które przed wysłaniem danych ustanawiałyby połączenie między odbiorcą a nadawcą,
- Nie ma korekcji błędów,
- Nie sprawdza czy dane zostały prawidłowo odebrane,
- Zajmuje się **adresowaniem** datagramu,
- Niezawodność transmisji danych jest zapewniania przez protokoły warstw wyższych.

Datagram IP

+	Bity 0 - 3	4 - 7	8 - 15	16 - 18	19 - 31
0	Wersja	Długość nagłówka	Typ usługi	Całkowita długość	
32	Numer identyfikacyjny			Flagi	Kontrola przesunięcia
64	Czas życia pakietu (TTL)		Protokół warstwy wyższej	Suma kontrolna nagłówka	
96	Adres źródłowy IP				
128	Adres docelowy IP				
160	Opcje IP				Uzupełnienie
192	Dane				

Warstwa transportowa

- Udostępnia interfejs pomiędzy warstwami niższymi a warstwą aplikacji,
- Dostarcza dane od nadawcy do odbiorcy,
- Ma możliwość zapewnienia wiarygodnych usług połączeniowych,
- Dzieli duże komunikaty warstwy aplikacji na segmenty lub paczki (fragmentacja).

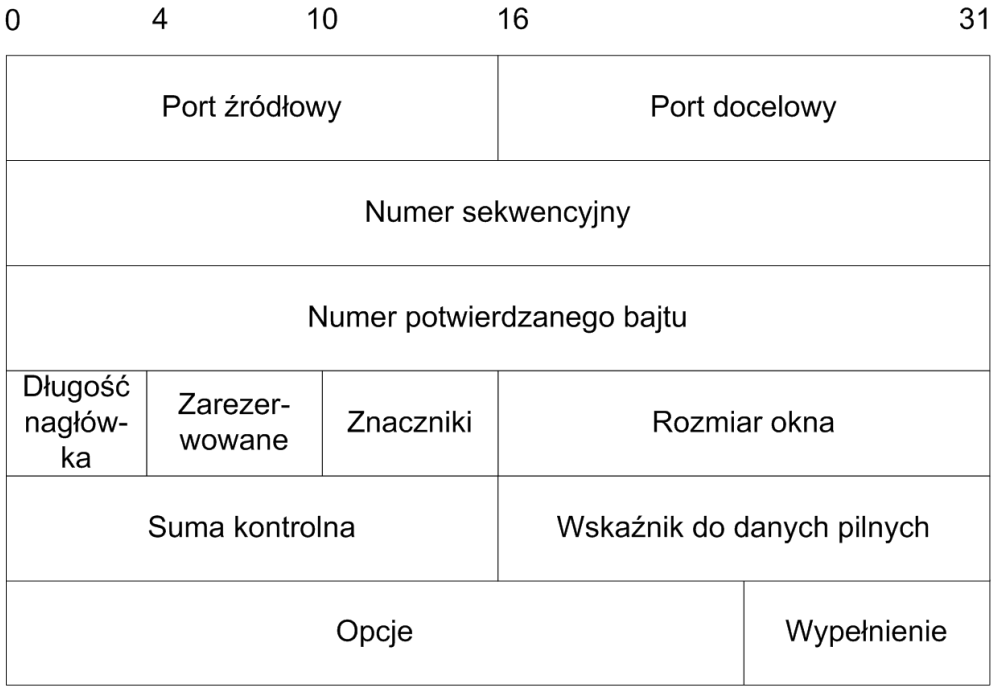
Dwa ważne protokoły:

- protokół kontroli transmisji (*Transmission Control Protocol*, TCP)
- protokół pakietów użytkownika (*User Datagram Protocol*, UDP).

Protokół TCP

- Protokół połączeniowy,
- Niezawodne i wiarygodne przesyłanie danych (mechanizm potwierdzenia z retransmisją, wysyłanie danych dopóki nie zostanie otrzymana wiadomość, że dane przeszły bezbłędnie),
- Kontrola przepływu, korekcja błędów,
- Skierowany na strumieniową (jednolitą) transmisję danych (przesył dużych ilości ciągłych informacji – np. pliki)

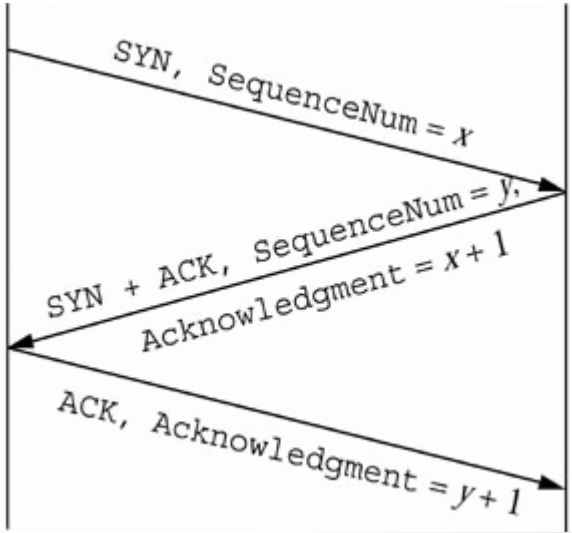
Nagłówek w protokole TCP



Ustanawianie połączenia

„three-way handshake”

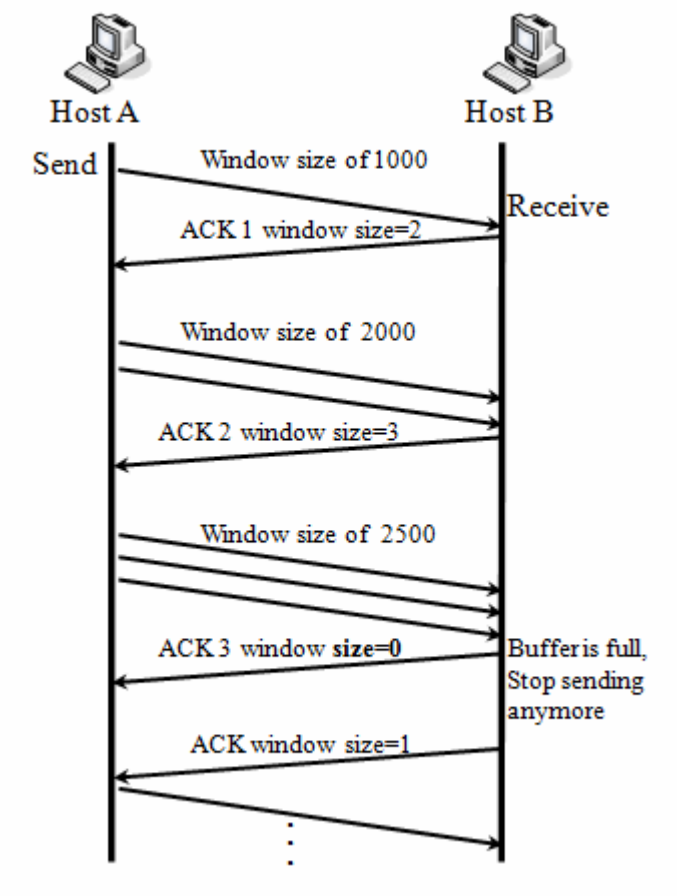
1. Nawiązujący połączenie host A wysyła do hosta B segment z ustawionym bitem SYN (synchronizacji). Segment ten informuje host B, że host A chce nawiązać połączenie oraz jaki będzie początkowy numer sekwencji przesyłania danych.
2. Host B odpowiada hostowi A segmentem z ustawionymi bitami ACK (potwierdzenie) i SYN, potwierdzając odbiór segmentu od A (pole numer potwierdzenia) i informując go, od jakiego numeru sekwencyjnego będzie odliczał wysyłane przez siebie dane.
3. Host A wysyła segment potwierdzający odbiór pakietu od B, zawierający pierwsze przesyłane dane.



Transmisja danych w TCP

- TCP interpretuje wysyłane dane jako ciągły strumień bajtów, a nie jako niezależne pakiety. Z tego powodu istotne jest zachowanie kolejności (pola numer sekwencyjny i numer potwierdzenia).
- Segmentom nadawane są kolejne numery sekwencyjne.
- Nadawca czeka na potwierdzenie otrzymania przez odbiorcę segmentów. Jeśli nie otrzyma potwierdzenia, wysyła je ponownie.
- Odbiorca informuje nadawcę o otrzymaniu segmentu poprzez wysłanie segmentu z ustawionym bitem ACK i odpowiednim numerem sekwencji segmentu na polu potwierdzenia. W polu „okno” wysyła też liczbę segmentów, które jeszcze może przyjąć.
- Standard nie nakazuje potwierdzania każdego segmentu. Jeśli odbiorca wyśle potwierdzenie N, oznacza to, że otrzymał wszystkie segmenty do N-1 włącznie.
- Nadawca może wysyłać segmenty tak długo, dopóki sumaryczna wartość wysłanych danych nie przekroczy rozmiaru okna.

Transmisja danych TCP



Zakończenie połączenia

1. Host A wysyła segment z ustawionym bitem FIN (koniec danych)
2. Host B wysyła potwierdzenie odebrania tego segmentu
3. Host B wysyła segment z ustawionym bitem FIN
4. Host A wysyła potwierdzenie odebrania tego segmentu

Sygnały 2 i 3 mogą być wysłane jednym segmentem.

Protokół UDP

- Protokół bezpołączeniowy,
- Skierowany na przesyłanie wiadomości, gdzie pojedynczy pakiet stanowi integralną informację.
- Nie ma narzutu na nawiązywanie połączenia i śledzenie sesji (w przeciwieństwie do TCP),
- Nie ma mechanizmów kontroli przepływu i retransmisji - zawodny
- Korzyści – większa szybkość transmisji danych i brak dodatkowych zadań, którymi musi zajmować się host posługujący się tym protokołem,
- Często używany w takich zastosowaniach jak wideokonferencje, strumienie dźwięku w Internecie i gry sieciowe, gdzie dane muszą być przesyłane możliwie szybko, a poprawianiem błędów zajmują się inne warstwy modelu OSI.

Format nagłówka UDP

+	Bity 0-15	Bity 16-31
0	Port nadawcy	Port odbiorcy
32	Długość	Suma kontrolna
64	Dane	

TCP vs. UDP

- Protokołu **TCP** używamy, gdy zależy nam na pewnym dostarczaniu wiadomości. Także tam, gdzie kolejność dostarczaniu segmentów ma kluczowe znaczenie – duże pliki dzielone na mniejsze segmenty.

- Protokołu **UDP** używamy, gdy bardziej zależy nam na szybkości niż niezawodności, a zagubienie pojedynczego pakietu nie wpłynie na krytyczny błąd naszej aplikacji.

Warstwa aplikacji

- Najwyższy poziom, w którym pracują użyteczne aplikacje np. serwer WWW czy przeglądarka internetowa,
- Obejmuje zestaw gotowych protokołów, które aplikacje wykorzystują do przesyłania różnego typu informacji w sieci.

Najbardziej znane protokoły aplikacji:

- **telnet** - protokół terminala sieciowego, umożliwiający zdalne zalogowanie się i pracę w odległym systemie,
- **FTP** - protokół transferu plików, który umożliwia zapisanie w odległym systemie lub odczytanie z niego wskazanych plików,
- **SMTP** - podstawowy protokół przesyłania poczty odpowiedzialny za dystrybucję poczty elektronicznej,
- **HTTP** - odpowiedzialny za przesyłanie w sieci stron WWW.

Adresowanie

- Do komunikacji potrzebna jest możliwość zidentyfikowania i odnalezienia się nawzajem.
- Każdemu komputerowi w sieci TCP/IP trzeba przypisać unikatowy identyfikator, czyli **adres IP**. Adres ten należy do warstwy 3 modelu OSI i pozwala jednemu komputerowi w sieci zlokalizować inny
  np. 192.168.5.45, 10.0.15.158, 212.178.45.4
- Wszystkie komputery mają także unikatowy adres fizyczny zwany **adresem MAC**. Adresy te są nadawane przez producentów kart sieciowych i należą do warstwy 2 modelu OSI
  np. 00-50-56-C0-0A-01, 00-50-56-C4-00-01

Protokoły i porty

- Do identyfikacji protokołów warstwy transportowej IP używa numerów protokołów.
- Protokoły transportowe do określania aplikacji wykorzystują numery portów.
- Niektóre numery protokołów i portów są zarezerwowane dla tzw. dobrze znanych usług. Mianem tym określa się standardowe, powszechnie używane protokoły sieciowe, takie jak FTP czy telnet.
- Numery protokołów i portów przyporządkowane dobrze znanym usługom zdefiniowane są w RFC 1700.

Numery portów

- Warstwa transportowa przekazuje dane otrzymane od IP do odpowiednich procesów aplikacji.
- Procesy te (zwane też usługami sieciowymi) identyfikowane są według 16-bitowych numerów portów.
- Port jest adresem wewnętrznym, zapewnia interfejs pomiędzy aplikacją a protokołem transportowym
- Numer portu źródłowego, odpowiadający procesowi, który wysłał dane oraz numer portu docelowego, odpowiadający procesowi, który ma dane otrzymać, są zawarte w pierwszym słowie nagłówka każdego segmentu TCP i pakietu UDP.
- Numery portów mają przydzielone zakresy:
  - numery poniżej 1024 – dobrze znane numery portów,
  - 1024 – 49151 zarejestrowane porty, mogą z nich korzystać programy i
  - procesy zwykłych użytkowników,
- 49152 – 65535 porty dynamiczne i/lub prywatne
- Host źródłowy dynamicznie przydziela numery portów źródła rozpoczynającego transmisję. Numery te są zawsze większe od 1023

Gniazda

- Gniazdo (*socket*) jest mechanizmem komunikacji między procesami, służącym jako punkt końcowy połączenia,
- Stanowi kombinację adresu IP oraz numeru portu,
- Gniazdo jednoznacznie określa każdy proces sieciowy w Internecie.
- Czasem terminy gniazdo i numer portu używane są zamiennie.
- Także dobrze znane usługi sieciowe określane są mianem dobrze znanych gniazd.

Rozróżniamy:

- Gniazda potokowe (TCP)
- Gniazda datagramowe (UDP)