Wojciech Myszka

TCP/IP: Adresy, trasowanie, protokoły, gniazda

wer. 41 z drobnymi modyfikacjami!

2024-03-05 15:59:07 +0100

1. Standaryzacja

Bałagan

1. Jakoś tak w latach siedemdziesiątych powstało wiele działających rozwiązań sieciowych.
2. Łączenie komputerów było bardzo „atrakcyjne" i wiele instytucji angażowało się w takie działania.
3. Z drugiej strony — komputer ciągle był dobrem cennym i niechętnie był udostępniany do „zabawy".
4. Wielość rozwiązań hamowała zaangażowanie (zwłaszcza) przemysłu, a co za tym idzie również dostęp do funduszy.
5. Doprowadzenie, w takiej sytuacji, do współpracy różnych podmiotów było bardzo trudne.
6. Rozpoczęto wiec prace normalizacyjne.
7. Przede wszystkim prace takie podejmowano w przemyśle, który dążył do komercjalizacji rozwiązań.

W pierwszym okresie do sieci podłączane były najsłabsze (nikomu nie potrzebne) maszyny.

Firmy komercyjne widziały potencjał powstających rozwiązań i dążyły do ich komercjalizacji, żeby zarabiać. W tym kontekście warto wspomnieć o dwu przypadkach (wybiegając nieco w przyszłość):

1. IBM Token Ring — bardzo efektywny standard transmisji w kablu koncentrycznym powstał w latach 70. Główną zaletę tego standardu było duża odporność na nasycenie medium[1]. Gdy wymuszono otwarcie standardu (żeby firmy inne niż IBM mogły dostarczać rozwiązania sprzętowe), twórca zaczął wprowadzać do swojego sprzętu (komputery i sprzęt sieciowy) rozwiązania uniemożliwiające współpracę ze sprzętem sieciowym podchodzącym od obcych dostawców. nieuchronną klęską standardu.

[1] Standard pozwalał na sprawną pracę aż do około 80% pojemności medium, gdy standardowy Ethernet zaczynał kiepsko pracować w okolicach 50–60% pojemności.

Skończyło się to
2. 100BaseVG (zwany również 100VG–AnyLan) standard wprowadzony przez firmę Hewlett-Packard w połowie lat 90, pozwalający na łączenie w jednym kablu ramek Ethernet i Tokem Ring, mimo ogromnej promocji ze strony twórców (sprzedawali karty sieciowe obsługujące dwa starndardy: Ethernet 10 Mbit i !00VG w cenie karty sieciowej 10 Mbit) upadł po jakichś trzech latach wyparty przez standard FastEthernet.
   Dodatkową zaletą standardu miało być to, że pozwalał na wykorzystanie **standardowego**, używanego w tamtych czasach okablowania telefonicznego, co maiło prowadzić do zmniejszenia kosztów.

Główni aktorzy

1. Firma Xerox (XNS)
2. Konsorcjum DEC–Intel–Xerox (DIX)
3. Rząd (czy agenda rządowa): ARPANet (TCP/IP)
4. Standard ISO: OSI
5. Standard IEEE (IEEE 802)

Twórcy standardu DIX (głównie Bob Metcalfe i David Boggs) w laboratorium badawczym firmy Xerox w Palo Alto (przy współpracy firm DEC[2] oraz Intel) otrzymali ochronę patentową, która nigdy nie byłą „aktywna" — wszyscy korzystali z rozwiązań, które po niewielkich modyfikacjach stały się standardem ISO.

2. ISO/OSI

Norma

— Pod koniec roku 1979 opracowany został **Open System Interconnection Reference Model** zaakceptowany przez ISO jako *Working Draft*.
— Jest to model warstwowy czerpiący bardzo wiele z doświadczeń ARPANet-u i wiedzy zdobytej podczas implementacji oprogramowania sieciowego w systemach operacyjnych.

Open System Interconnection Reference Model

model odniesienia łączenia systemów otwartych

Model warstwowy

1. Model warstwowy wprowadzony został po to, aby zachęcić producentów do tworzenia aplikacji mogących współpracować z aplikacjami innych firm oraz aby tworzyć oprogramowanie (w jakimś sensie) niezależne od sprzętu, wersji oprogramowania czy systemu operacyjnego.
2. Co prawda model **nigdy** nie został powszechnie zaakceptowany i zaimplementowany (w całości), ale ciągle używany jest z jednej strony jako model referencyjny, a struktura warstwowa używana jest do opisu istniejących rozwiązań.

[2] Firma już nie istnieje, została przejęta przez Compaq.

3. Każda warstwa implementowana jest jako sterownik.
4. Warstwy są implementowane niezależnie jedna od drugiej. Określone są jedynie bardzo precyzyjne zasady komunikowania się między warstwami.
5. Podczas przesyłania danych komunikują się między sobą odpowiednie warstwy na obu komputerach.
6. Warstw jest siedem:
   a) **Warstwa łącza fizycznego** (najniższa). Odpowiada za przesyłanie danych po medium fizycznym, definiuje fizyczne cechy interfejsu (zarówno mechaniczne jak i elektryczne). Przykłady:
      — RS-232C
      — kabel koncentryczny
      — skrętka
      — światłowód

      Pragnąc wprowadzić nowe medium transmisji danych wystarczy — oprócz wprowadzenia odpowiedniej normy określającej sposób transmisji, rodzaje „wtyczek", poziomy napięć elektrycznych i napisać oprogramowanie zgodne z resztą modelu warstwowego. Powoduje to, że z punktu widzenia przeglądarki WWW nie widać różnicy w sposobie przesyłania informacji.
   b) **Warstwa łącza danych.** Warstwa definiuje sposób przesyłania danych po łączu fizycznym. W warstwie tej dokonuje się kodowania danych, tworzy się z nich „paczki" (ramki). Sprawdzana jest poprawność transmisji (detekcja błędów).
      — **Ethernet**
      — Token Ring
      — HDLC

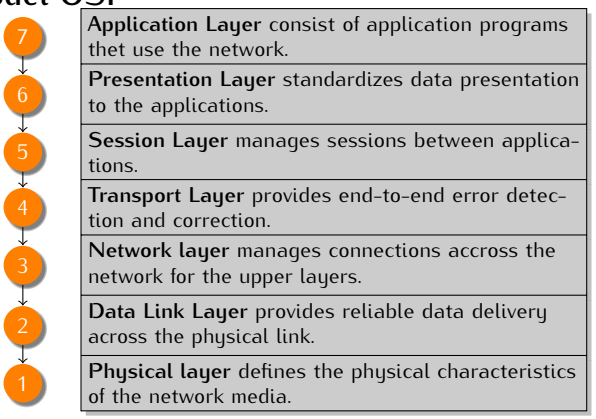
      (W pewnych sytuacjach kontrola poprawności danych wykonywana może być w wyższych warstwach.)
   c) **Warstwa sieci.** Tu podejmowane są decyzje trasie przesyłanych informacji. W szczególności trzeba podejmować decyzje (na podstawie adresu docelowego) czy odbiorca znajduje się w sieci lokalnej czy zdalnej.
      — **IP**
      — IPX
      — X25
   d) **Warstwa transportowa.** Na tym poziomie odbywa się logiczna kontrola poprawności transmisji. „Paczki" danych (pakiety) otrzymują swój numer, i w tej warstwie następuje kontrola czy nadchodzą one we właściwej kolejności, czy nie ma zdublowanych pakietów. W przypadku błędów — następuje wymuszenie powtórnej transmisji. Również w tej warstwie można zestawić „logiczne połączenie" dwu systemów.
      — **TCP**
      — **UDP**
      — NEtBIOS (Microsoft)
7. **Warstwa sesji.** W tej warstwie dokonuje się koordynacja wymiany ważnych informacji pomiędzy systemami zdalnym a lokalnym. Tu ustalane

mogą być, na przykład, formy retransmisji lub wznowienia transmisji po ponownym nawiązaniu przerwanego połączenia.

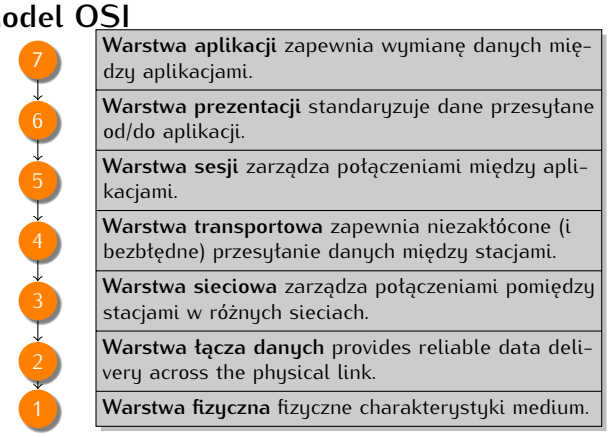
8. **Warstwa prezentacji.** Przeznaczona do konwersji, de/szyfrowania, de/kodowania, translacji danych (w przypadku gdy system zdalny i lokalny używają różnego kodowania znaków).
9. **Warstwa aplikacji.** Na tym poziomie odbywa się logiczna wymiana informacji pomiędzy systemami. Tu pracują wszystkie aplikacje korzystające z sieci:
   — poczta elektroniczna,
   — **przeglądarka WWW**,
   — „dysk" sieciowy.

Poniżej diagramy dotyczące modelu OSI po polsku i po angielsku (aby przyswajać terminologię).

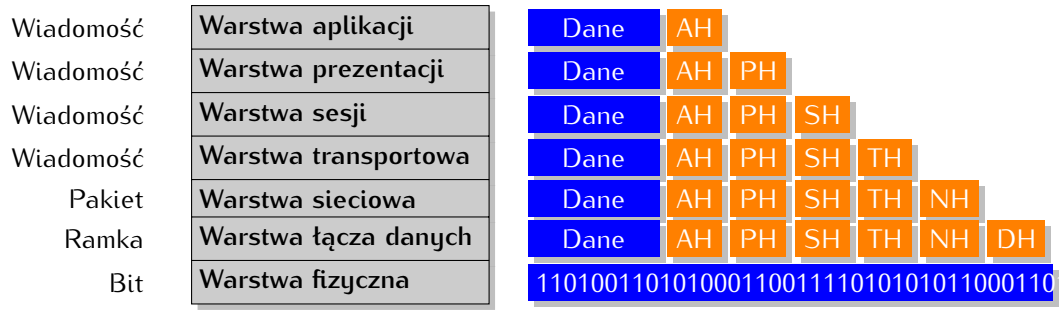
Model OSI



Model OSI



Komunikacja

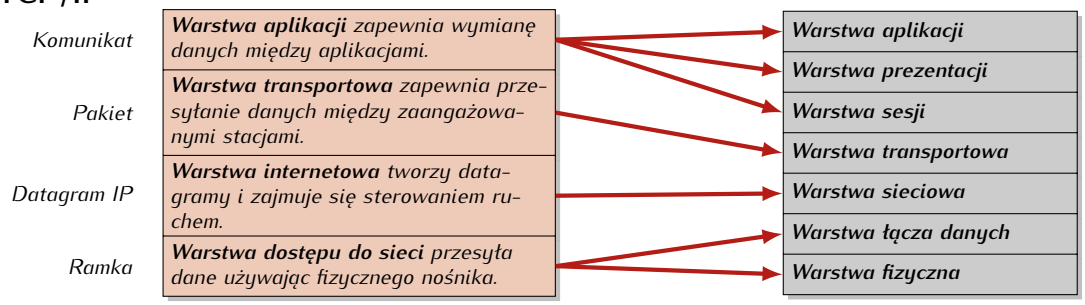


ISO/OSI

— Model ISO/OSI właściwie nie został (prawie) nigdzie zaimplementowany.
— Nazywany jest modelem odniesienia i używany do analizy innych protokołów.
— Najpopularniejszy protokół sieciowy stosowany współcześnie to TCP/IP.
— Ma on mniej warstw, ale realizowane funkcje są zbliżone.

3. TCP/IP

TCP/IP



Ale dodać trzeba, że nie ma zgody co do tego jak dokładnie te warstwy wyglądają ☺

Czasami (po stronie TCP/IP) dodaje się jeszcze piątą warstwę: warstwę sprzętu.

3.1. Historia

TCP/IP — historia

— Początek protokołu TCP/IP to sieć ARPANET.
— Pierwotnie realizowano model „klient–serwer”, później *host-to-host*. W tym układzie zadania rozłożone były asymetrycznie. W układzie host-to-host jest pełna symetria.
— Opracowano protokół NCP *Network Control Protocol*.
— Podczas prób łączenia różnych komputerów pojawiły się poważne problemy: każdy producent oferował **różne** techniki sprzętowe i programowe.

— W 1973 roku Kahn (DARPA) i Cerf (Stanford University) rozpoczęli prace nad TCP/IP.
— Były pomysły aby jakoś zintegrować TCP/IP z ISO/OSI, ale nic z tego nie wyszło.
— Opracowany (ze środków publicznych) standard TCP/IP dostępny był za darmo. Miało to wpływ na jego rozpowszechnianie się.
— Wszystkie prace rozwojowe protokołu TCP/IP były nadzorowane przez rząd USA za pomocą testów i certyfikacji — zapobiegło to fragmentacji rynku i wprowadzaniu niekompatybilnych rozwiązań komercyjnych.

3.2. TCP/IP: Zasady

TCP/IP: Zasady

1. Protokół TCP zestawia dwustronne (w trybie duplex) połączenie między dwoma systemami.
2. Każdy z systemów ma swój własny adres. (O adresach będzie później.)
3. Połączenie realizowane jest między dwoma gniazdami (*socket*) na obu systemach. Z gniazdem związana jest aplikacja użytkowa wykorzystywana do komunikacji. (O gniazdach też będzie później.)
4. Połączenie umożliwia:
   — **sterowanie przepływem** (*flow control*); pozwala to na dostosowanie szybkości transmisji do możliwości systemu,
   — **potwierdzanie** odbioru pakietów,
   — **zachowanie kolejności** przesyłanych pakietów,
   — **sprawdzanie sumy kontrolnej**,
   — **retransmisję** uszkodzonych/utraconych pakietów.

Niestety, nawiązywanie i utrzymywanie połączenia jest operacją zajmującą sporo czasu. W przypadku przesyłania krótkich komunikatów może trwać dłużej niż sama transmisja danych.

Opracowano więc *User Datagram Protocol* (UDP) i wydzielono *Internet Protocol* (IP) zajmujący się wyłącznie ruchem datagramów.

1. **Protokół TCP** zapewnia niezawodny transport danych z jednego węzła do drugiego.
2. **Protokół UDP** zapewnia usługi datagramowe (przesyłanie danych bez kontroli). Sprawdzenie poprawności dokonuje aplikacja we własnym zakresie.
3. **Protokół IP** pracuje w trybie bezpołączeniowym i zapewnia sprawny ruch datagramów.

3.3. Warstwa dostępu do sieci

— Najniższy poziom.
— Protokół posiada narzędzia pozwalająca komunikować się z innymi węzłami bezpośrednio połączonymi do sieci.
— Zazwyczaj nie przyciąga uwagi zwykłych użytkowników.
— Bardzo istotny gdy pojawia się nowy sprzęt: wszystkie wyższe warstwy potrzebują go.

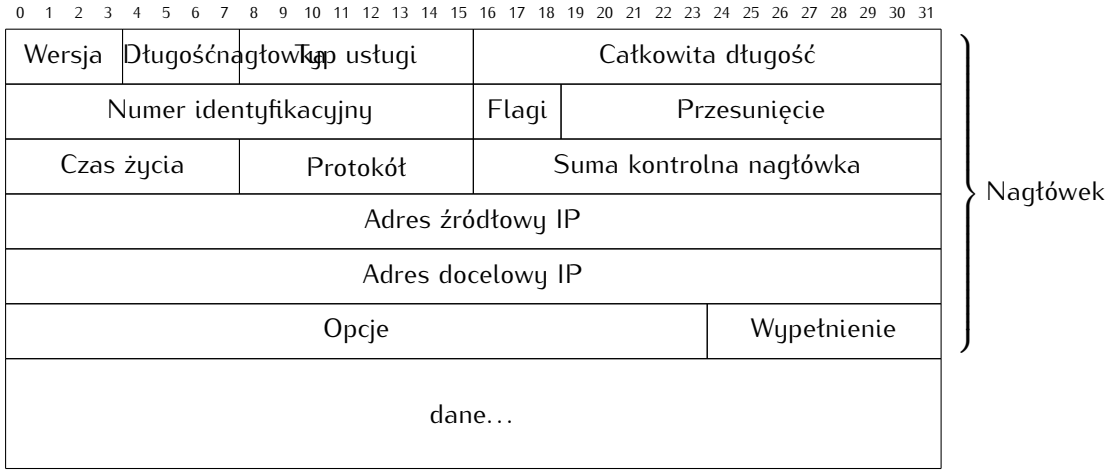
— Najważniejszą funkcją protokołu jest enkapsulacja datagramów IP w ramki (*frames*) oraz tłumaczenie adresów IP na adresy fizyczne używane w sieci lokalnej.
— *Na tym poziomie możliwa jest jedynie komunikacja między węzłami podłączonymi do sieci lokalnej!*

3.4. Warstwa internetowa

3.4.1. IP

— IP od Internet Protocol.
— Protokół bezpołączeniowy (nie wymienia dodatkowych informacji przed wysłaniem danych).
— Protokół IP nie przejmuje się ewentualnymi błędami ani przesyłaniem pakietów we właściwej kolejności. Tym zajmują się warstwy wyższe.
— Gdy pojawia się przeciążenia — protokół IP **porzuca** (gubi) pakiety.
— Datagramy mogą być adresowane do pojedynczych węzłów lub do wielu odbiorców.
— Datagramy mogą korzystać z różnych dróg (jeżeli takie istnieją).

3.5. Datagram IPv4



Datagram IP

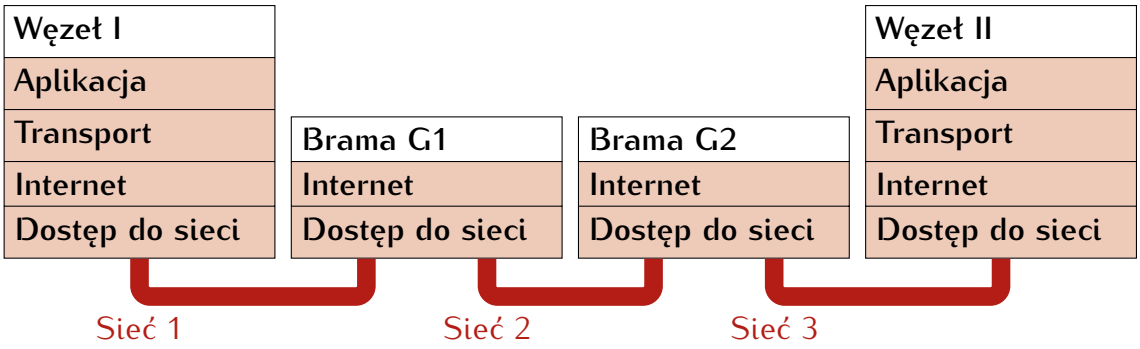
— **Wersja** (4 bity) – (ang. *Version*) pole opisujące wersję protokołu, jednoznacznie definiujące format nagłówka.
— **Długość nagłówka** (4 bity) – (ang. *Internet Header Length*) długość nagłówka IP wyrażona w 32-bitowych słowach; minimalny, poprawny nagłówek to co najmniej 5 słów.
— **Typ usługi** (8 bitów) – (ang. *Type of Services*) pole wskazujące jaka jest pożądana wartość QoS dla danych przesyłanych w pakiecie. Na podstawie tego pola, routery ustawiają odpowiednie parametry transmisji.
— **Całkowita długość** pakietu (16 bitów) – (ang. *Total Length*) długość całego datagramu IP (nagłówek oraz dane); maksymalna długość datagramu wynosi $2^{16} - 1 = 65535$ bajtów. Minimalna wielkość datagramu

jaką musi obsłużyć każdy host wynosi 576 bajtów, dłuższe pakiety mogą być dzielone na mniejsze (fragmentacja).
— **Numer identyfikacyjny** (16 bitów) – (ang. *Identification*) numer identyfikacyjny, wykorzystywany podczas fragmentacji do określenia przynależności pofragmentowanych datagramów
— **Flagi** (3 bity) – (ang. *Flag*) flagi wykorzystywane podczas fragmentacji datagramów. Zawierają dwa używane pola: DF, które wskazuje, czy pakiet może być fragmentowany oraz MF, które wskazuje, czy za danym datagramem znajdują się kolejne fragmenty.
— **Przesunięcie** (13 bitów) – (ang. *Fragment Offset*) w przypadku fragmentu większego datagramu pole to określa miejsce danych w oryginalnym datagramie; wyrażone w jednostkach ośmiooktetowych.
— **Czas życia** (8 bitów) – (ang. *Time to live*) czas życia datagramu. Zgodnie ze standardem liczba przeskoków przez jaką datagram znajduje się w obiegu. Jest zmniejszana za każdym razem, gdy datagram jest przetwarzany w routerze – jeżeli czas życia osiąga wartość 0, datagram jest usuwany z sieci (nie przekazywany dalej) o czym nadawca usuniętego pakietu jest informowany zwrotnie z wykorzystaniem protokołu ICMP. Istnienie tej wartości jest konieczne, zapobiega krążeniu pakietów w sieci.
— **Protokół** warstwy wyższej (8 bitów) – (ang. *Protocol*) informacja o protokole warstwy wyższej, który jest przenoszony w polu danych datagramu IP.
— **Suma kontrolna nagłówka** (16 bitów) – (ang. *Header Checksum*) suma kontrolna nagłówka pakietu, pozwalająca stwierdzić czy został on poprawnie przesłany, sprawdzana i aktualizowana przy każdym przetwarzaniu nagłówka.
— **Adres źródłowy** (32 bity) i **adres docelowy** (32 bity) – (ang. *Source/Destination IP Address*) pola adresów nadawcy i odbiorcy datagramu IP.
— **Opcje** (32 bity) – (ang. *Options*) niewymagane pole opcji, opisujące dodatkowe zachowanie pakietów IP
— **Wypełnienie** – (ang. *Padding*) – opcjonalne pole wypełniające nagłówek tak, aby jego wielkość była wielokrotnością 32, wypełnione zerami.

3.6. Przekazywanie pakietów

— Warstwa dostępu do sieci pozwala jedynie na przekazywanie pakietów bezpośrednio wewnątrz sieci lokalnej.
— Jeżeli w sieci jest węzeł, który został podłączony również do innej sieci lokalnej (i jest on w stanie przekazywać pakiety) pojawia się możliwość komunikacji między sieciami.
— Węzeł taki nazywany bywa bramą (*gateway*) albo routerem.
— Do tego (między innymi) wykorzystywane są protokoły warstwy internetowej.

Routing



3.7. Fragmentacja datagramów

Fragmentacja datagramów

— Gdy datagram podróżuje przez różne sieci (o różnej architekturze) pojawia się problem maksymalnej jego długości.
— Musi być wówczas podzielony na mniejsze fragmenty.
— Dla każdej sieci określona jest MTU (*maximum transmission unit*) czyli rozmiar największego pakietu, który może być przesłany.
— Format każdego fragmentu jest identyczny jak format każdego innego datagramu. Jedynie pole *Numer identyfikacyjny* zawiera informacje pozwalające złożyć później pakiet w całość. Pomaga w tym pole *Przesunięcie* mówiące w „którym miejscu" pakietu mają się znaleźć dane. Dodatkowo w polu *Flagi* jest informacja czy będą występowały kolejne fragmenty.

3.8. Przekazywanie informacji „wyżej"

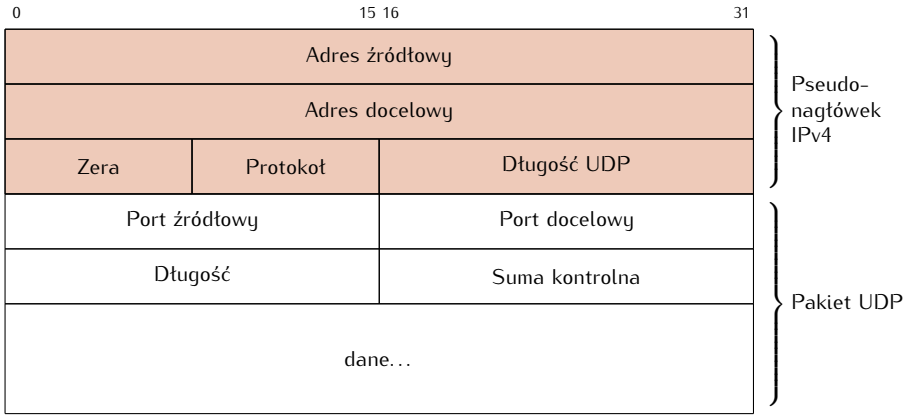
Przekazanie informacji do warstwy transportowej

— Gdy warstwa Internetowa otrzyma pakiet, dla którego węzeł jest węzłem docelowym, musi przekazać go do warstwy wyższej.
— Decyzja podejmowana jest na podstawie zawartości pola *Protokół* (w trzecim słowie nagłówka).

3.9. ICMP

Internet Control Message Protocol

— Protokół korzysta z warstwy internetowej do przesyłania komunikatów związanych z przebiegiem transmisji:
  — **Sterowaniem ruchem pakietów** (*Flow Control*) — gdy dane napływają zbyt szybko, prośba o spowolnienie.
  — **Informacja o niedostępności węzła** (*Detecting unreachable destination*) — gdy węzeł/sieć przeznaczenia jest niedostępny, (ostatnia) brama wysyła odpowiednią informację do nadawcy.



Rysunek 1. Format pakietu UDP (w środowisku IPv4)

- **Lepsza trasa** (*Redirecting Routes*) — gdy brama posiada informację, że istnieje trasa lepsza — przekazuje o tym informację do węzła wysyłającego.
- **Sprawdzanie zdalnego węzła** (*ICMP Echo Message*) do węzła docelowego wysyłany jest specjalny pakiet (*ping*), który powinien zostać odesłany. Pozwala to stwierdzić, czy zdalny węzeł „żyje" (i jak daleko się znajduje).

— Szczegółowa lista informacji jest znacznie dłuższa.

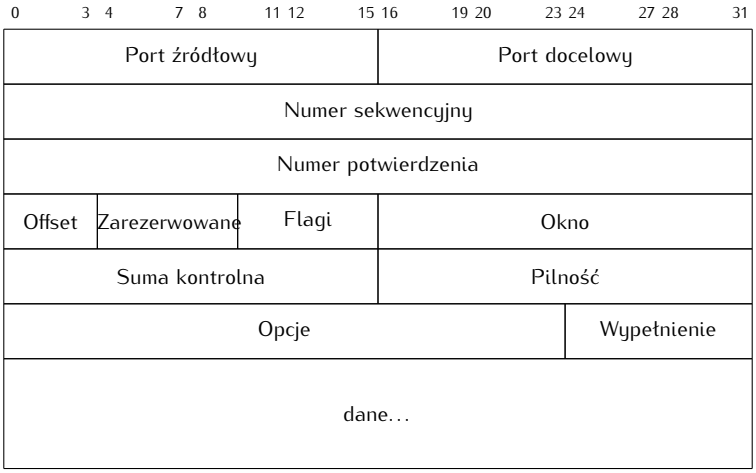
3.10. Warstwa transportowa

Do czynienia mamy z dwoma protokołami:

1. User Datagram Protocol:
   - Prosty,
   - Nie-niezawodny (bo to nie jest odwrotność od niezawodności, tylko podkreślenie, że na poziomie transportowym nie ma mechanizmów sprawdzania poprawności; są/mogą być na poziomie wyższym)
   - Transaction Oriented (zapytanie–odpowiedź) szczególnie dobry w takich sytuacjach.
   - Nadaje się do transmisji dźwięku albo video (drobne zniekształcenia spowodowane błędami transmisji są lepsze niż oczekiwanie na retransmisję dużej ilości danych).
   - Zwłaszcza nadaje się do wysyłania informacji do wielu odbiorców
2. Transmisson Control Protocol:
   - Niezawodny — zawiera mechanizmy pozwalające panować nad poprawnością transmisji.
   - Nawiązanie transmisji wymaga wymiany trzech pakietów.
   - Wiadomości docierają w takiej kolejności w jakiej zostały wysłane.
   - Dane traktowane są jako strumień.

TCP

Nawiązanie połączenia (*handshake*):



Rysunek 2. Format segmentu TCP

1. Nadawca rozpoczyna połączenie wysyłając specjalny pakiet (SYN). Informuje w nim jak będzie numerował kolejne pakiety (standardowo numeracja zaczyna się od zera, ale możne to być dowolna liczba).
2. Odbiorca (jeżeli godzi się na połączenie) wysyła (specjalny) pakiet (SYN), który mówi, że przyjmuje połączenia oraz zawiera informacje jak będą numerowane pakiety wysyłane w odpowiedzi.
3. Nadawca potwierdza otrzymanie odpowiedzi od nadawcy i zaczyna wysyłanie danych.

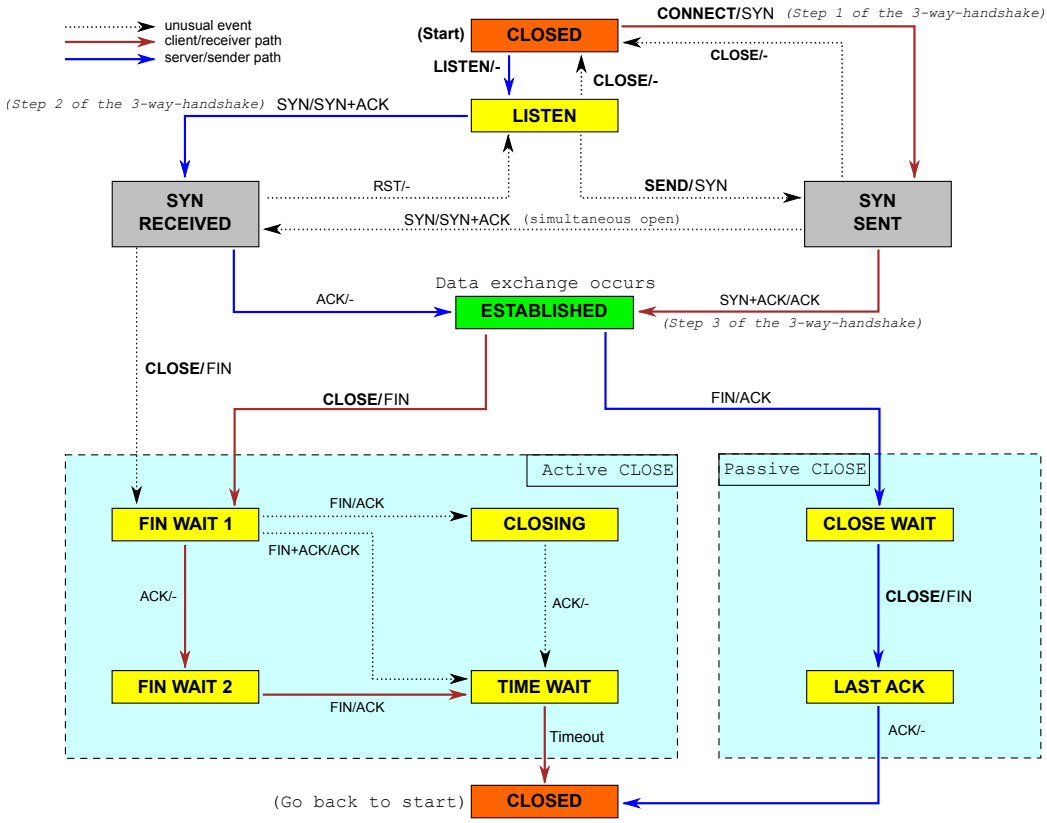
Koniec transmisji również wymaga przesłania trzech pakietów:
— *Koniec transmisji!*
— *Zrozumiałem!*
— *OK!*

Okno

— Protokół TCP zakłada, że przesłanie każdego pakietu **musi** być potwierdzone przez odbiorcę.
— Może to powodować spore opóźnienia w ruchu (na przykład wtedy gdy łącze jest niesymetrycznie obciążone): duży pakiet danych przechodzi szybko, ale potwierdzenie „czeka w kolejce".
— Opracowano więc koncepcję okna: strony umawiają się ile informacji może być przesłane bez otrzymywania potwierdzeń.
— Jeżeli wielkość okna to 10000 bajtów — informacje są wysyłane bez oczekiwania na potwierdzenia pakietów. Potwierdzenia nadchodzą i są odnotowywane przez odbiorcę. Każdy potwierdzony bajt „przesuwa" okno (pozwalając na transmisję kolejnych bajtów) Gdy (po „wyczerpaniu" pojemności okna i odczekaniu określonego czasu) wszystkie wysłane informacje nie zostaną potwierdzone — powtarza się wysyłkę zaczynając od pierwszego niepotwierdzonego bloku informacji.

Przekazanie informacji do warstwy aplikacji...

...odbywa się na podstawie numeru portu (pierwsze słowo nagłówka).



Rysunek 3. Uproszczony diagram stanu TCP

Będzie o tym mowa później.

3.11. Warstwa aplikacji

Warstwa aplikacji

1. Aplikacje to wszystkie programy użytkowe korzystające z sieci:
   - przeglądarka WWW,
     - HTTP
     - HTTPS
   - program pocztowy
     - IMAP (Thunderbird)
     - POP3 (obecnie prawie nieużywany)
     - SMTP (Komunikacja między serwerami pocztowymi)
   - dysk sieciowy
   - czat
   - gry komputerowe
2. Aplikacje mają swoje specyficzne protokoły umożliwiające komunikację. Czasami protokoły te obejmują szyfrowania.

Dziś najchętniej warstwa aplikacji realizowana jest w przeglądarce internetowej.