# **Model OSI**

**Model OSI** (Open System Interconnection) powstał pod kontrolą Międzynarodowej Organizacji Normalizacyjnej (ISO – International Organization for Standarization). Został opracowany w celu ułatwienia realizacji otwartych (obsługiwane w środowiskach wielosystemowych) połączeń systemów komputerowych. Model ten umożliwiał współdziałanie urządzeń i oprogramowania od różnych producentów.

Model OSIzbiera w logicznie ułożone warstwy funkcje potrzebne do zdefiniowania, nawiązania i utrzymania komunikacji między dwoma (lub więcej) urządzeniami niezależnie od ich producenta i architektury. Procesy zachodzące podczas komunikacji dzielone są na 7 warstw funkcjonalnych, które zbudowane są w oparciu o naturalny ciąg zdarzeń zachodzących w czasie sesji komunikacyjnej.

**Warstwy te to:**

1. **Warstwa fizyczna** – najniższa warstwa odpowiedzialna za transmisję strumienia bitów,
2. **Warstwa łącza danych** – odpowiedzialna za poprawność przesyłanych danych,
3. **Warstwa sieciowa** – umożliwia współpracę pomiędzy sieciami, wybiera optymalną trasę transmisji danych,
4. **Warstwa transportu** – pełni funkcję zbliżoną do warstwy drugiej z tą różnicą, że odpowiada za końcową integralność transmisji także poza lokalną siecią,
5. **Warstwa sesyjna** – zarządza przebiegiem komunikacji podczas połączenia między dwoma komputerami (określa czy komunikacja może zachodzić w jednym czy dwóch kierunkach),
6. **Warstwa prezentacji** – odpowiedzialna za sposób kodowania danych,
7. **Warstwa aplikacji** – pełni rolę interfejsu pomiędzy aplikacjami użytkownika a usługami sieci.

**Warstwy wyższe:**

* Aplikacji
* Prezentacji
* Sesji

**Warstwy niższe:**

* Transportowa
* Sieciowa
* Łącza danych
* Fizyczna

**Protokoły komunikacyjne poszczególnych warstw grupuje się w kategorie:**

* **Protokoły aplikacji -** obejmują warstwy: aplikacji, prezentacji i sesji. Zapewniają wzajemne oddziaływanie aplikacji i wymianę danych. Przykłady: FTP, Telnet, SMTP, SNMP, Net BIOS, ...
* **Protokoły transportowe -** realizują połączeniowe usługi przesyłania danych, zapewniają wymianę danych pomiędzy systemami końcowymi w sposób sekwencyjny Przykłady: TCP, SPX, NetBIOS, NetBEUI.
* **Protokoły sieciowe -** zapewniają usługi łączy dla systemów komunikacyjnych, obsługują adresowanie i informację routingu, weryfikację błędów oraz żądania retransmisji. Obejmują również procedury dostępu do sieci określone przez wykorzystywany rodzaj sieci. Przykłady: IP, IPX.

**Protokół** – zdefiniowany sposób komunikowanie się z innym systemem

**Pakiet** – rodzaj „paczki danych” przesyłanej między urządzeniami przy wykorzystaniu łącza komunikacyjnego

**Ramka** – strumień danych przesyłany przez połączenie fizyczne

# **TCP/IP**

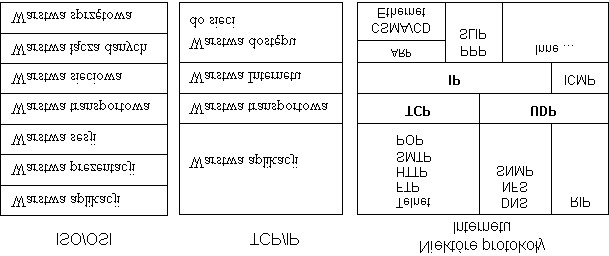
Urządzenia pracujące w jednej sieci mają możliwość komunikacji tylko między sobą.

Aby połączyć je z inną siecią, wymagany jest router.

W obrębie TCP/IP działa dużo innych protokołów, za pomocą których dane podróżują po sieci. Są one podzielone na warstwy.

**Podsumowując, protokół TCP/IP dzieli się na 4 warstwy od najważniejszej:**

* Warstwa fizyczna dostępu do sieci
* Warstwa Internet (sieciowa, międzysieciowa, warstwa IP)
* Warstwa transportowa
* Warstwa aplikacji (zastosowań)



**Warstwa dostępu do sieci** odpowiada za dostarczanie danych do innych urządzeń bezpośrednio dołączonych do sieci. Współpracuje ona bezpośrednio ze sprzętem i sterownikami odpowiedzialnymi za współpracę z siecią Ethernet. W przypadku innych sieci mogą to być protokoły PPP, SLIP lub inne. Warstwa ta współpracuje więc z interfejsem sieciowym (kartą sieciową), modemem lub innym urządzeniem pozwalającym na bezpośrednie połączenie dwóch lub więcej komputerów i separuje resztę warstw od zastosowanych rozwiązań fizycznych (niskopoziomowych). Świadczy ona usługę warstwie wyższej polegającą na wysyłaniu i odbieraniu porcji danych (zwanych ramkami) z komputerów w danej sieci fizycznej.

**Warstwa Internet** (IP) odpowiada za dostarczanie danych do urządzeń nie tylko w danej sieci fizycznej. Organizuje ona ruch tzw. pakietów IP między poszczególnymi sieciami fizycznymi połączonymi w intersieć. Korzysta z usług warstwy dostępu do sieci, sama zaś świadczy usługi dostarczania pakietu do dowolnego komputera w Internecie.

**Warstwa transportowa** odpowiedzialna jest za niezawodną wymianę danych z dowolnym komputerem w Internecie. Organizuje też i utrzymuje tzw. sesje, czyli wirtualne połączenia między komputerami. Korzysta z warstwy IP, sama zaś dostarcza usług niezawodnego transportu danych.

**Warstwa zastosowań** jest najwyżej położona. Tej warstwie odpowiadają wszelkie programy (aplikacje) internetowe korzystające z warstwy transportowej. Tu znajdują się wszelkie konkretne zastosowania Internetu - przesyłanie plików (FTP), poczty (SMTP) i inne.

**Składa się ona z dwóch protokołów:**

* **IP** - **I**nternet **P**rotocol - jest protokołem warstwy sieciowej, oddziela on wyższe warstwy od znajdującej się poniżej sieci i obsługuje adresowanie i dostarczanie danych.
* **TCP** - **T**ransmission **C**ontrol **P**rotocol - jest protokołem warstwy transportowej, gwarantuje, że odbiorca otrzyma dane dokładnie w tej samej postaci, w jakiej zostały wysłane.

**Dodatkowe pojęcia:**

* **DTE** (data terminal equipment) - urządzenie terminalowe danych lub inaczej stacja, jest unikalnym, zaadresowanym urządzeniem w sieci.
* **Urządzenie nadawczo-odbiorcze** (transceiver) – urządzenie, które umożliwia stacji transmisje „do” i „z” któregoś ze standardowych mediów normy IEEE 802.3. Dodatkowo transceiver Ethernetowy zapewnia izolację elektryczną pomiędzy stacjami oraz wykrywa i reaguje na kolizje.
* **MAU** (Medium Attachement Unit) moduł dołączania medium jest jednym z określeń IEEE na transceiver. Karta sieciowa najczęściej ma zintegrowany wewnątrz transceiver.
* **AUI** (Attachment Unit Interface) - połączenie pomiędzy kontrolerem i transceiverem. Aktualnie prawie nie występuje, był to rodzaj kabla i gniazdek, do komunikowania się karty sieciowej z dołączanymi do niej transceiverami. Dopiero transceiver mógł zostać podłączony do medium transmisyjnego (np.: koncentryk, skrętka)
* **Segment** – część okablowania sieci ograniczona przez mosty (bridge), przełączniki (switche), rutery, wzmacniaki lub terminatory. Najczęściej połączenie między dwoma komputerami lub koncentratorem i komputerem (dla skrętki i światłowodu), lub jeden odcinek kabla koncentrycznego łączącego wiele urządzeń.

# **TCP/UDP i usługi sieciowe**

Protokoły w. łącza danych (np. Ethernet) oraz w. sieciowej (IP) pozwalają tylko na zaadresowanie komputera (interfejsu sieciowego), a komunikacja zachodzi pomiędzy procesami (aplikacjami)

Do adresowania wybranej aplikacji stosuje się numer portu – adres warstwy transportowej (TCP)

Numer portu jest dwubajtowy (0-65535)

Według specyfikacji URL numer portu dodajemy po nazwie hosta i dwukropku

Pominięcie numeru portu oznacza, że zostanie użyty numer portu domyślny dla danej usługi

Łącze wymaga numerów portów po obu stronach

Najpopularniejsze usługi mają przydzielone specjalne porty, na których standardowo nasłuchują ich procesy serwerów

Usługi w. transportowej są ograniczane możliwościami w. sieci (przepustowość, opóźnienie), ale mogą (nie muszą) oferować jakościowo nową funkcjonalność

Niezawodność połączenia (izoluje proces komunikacji od zawodności w. sieci)

Oferowanie połączenia logicznego (mimo iż komunikacja w w. sieci jest pakietowa)

Oferowanie bezpieczeństwa komunikacji, np. poprzez szyfrowanie danych, mimo iż pakiety podróżują przez nieznane systemy

Kontrola przeciążenia łącza

Multipleksowanie/demultipleksowanie

**Protokół UDP (User Datagram Protocol):**

* Usługa: bezpołączeniowe dostarczenie datagramów
* Nie kontroluje poprawności i prędkości przesyłu
* Minimalizuje przesył dodatkowych danych

**Protokół TCP (Transfer Control Protocol):**

* Usługa: połączeniowe, niezawodne przesyłanie danych (strumień)
* Odpowiada za segmentowanie i powtórne scalanie danych, kontroluje prędkość przesyłu
* Wykorzystuje potwierdzenie z retransmisją (wysyła dane aż do otrzymania potwierdzenia)

**Połączenie pomiędzy aplikacjami odbywa się zwykle w architekturze klient-serwer (TCP/UDP)**

* Jedna aplikacja oczekuje na połączenie (serwer), druga je inicjuje (klient)
* Zakłada się, że transfer danych będzie w większości przebiegał od serwera do klienta, klienci nie komunikują się ze sobą

**Możliwa jest inna architektura: symetryczne połączenie pomiędzy równorzędnymi hostami (często UDP, ale możliwe też TCP)**

* Wykorzystanie: sieci P2P, telefonia internetowa
* Połączenie takie można traktować jako złożenie dwóch połączeń klient-serwer

**Protokoły TCP i UDP uzupełniają się, wybieramy jeden z nich w zależności od potrzeb**

**TCP**

* Gdy zależy nam na pewności połączenia i nie tolerujemy utraty danych
* Cena: mniejsza wydajność wykorzystania łącza
* Połączenie jest automatycznie ograniczane, gdy rośnie zajętość łącza
* Odpowiednie dla: poczta, zdalne logowanie, WWW, transfer plików

**UDP**

* Gdy zależy nam na maksymalnej wydajności i tolerujemy utratę danych (albo sami implementujemy niezawodność w warstwie wyższej!)
* Cena: bardziej skomplikowana implementacja
* Nie podlegamy automatycznemu ograniczaniu prędkości (dobrze dla nas, źle dla Internetu ...)
* Odpowiednie dla: strumieniowe multimedia, telefonia VOIP, routing, DNS, sieci P2P

|  |  |  |
| --- | --- | --- |
|  | TCP | UDP |
| Kiedy wykorzystywany: | Gdy zależy nam na pewności połączenia i nie tolerujemy utraty danych | Gdy zależy nam na maksymalnej wydajności i tolerujemy utratę danych |
| Wada: | Mniejsza wydajność wykorzystywanego łącza  Połączenie jest automatycznie ograniczane, gdy rośnie zajętość łącza | Bardziej skomplikowana implementacja  Nie podlegamy automatycznemu ograniczaniu prędkości |
| Odpowiednie dla: | Poczta, zdalne logowanie, WWW, transfer plików | Strumieniowe multimedia, telefonia VOIP, routing, DND, sieci P2P |

**Firewall** to urządzenie działające w warstwie transportowej, kolejne w hierarchii urządzeń sieciowych

* Musi rozpoznawać nie tylko numery IP, ale także numery portów TCP/UDP
* Może sterować ruchem (przepuszczać/blokować lub sterować prędkością) na podstawie numerów portów (w domyśle – usług)
* Realizacja: filtrowanie pakietów

**Gateway** to urządzenie działające w warstwie aplikacji

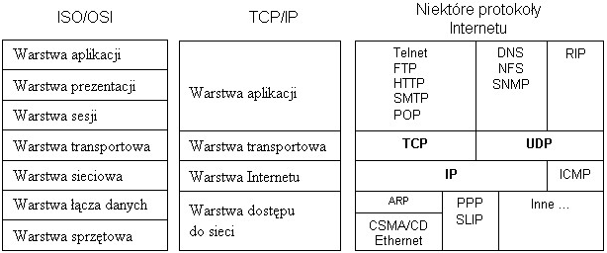
* Do przepuszczenia połączenia wymagana jest autentykacja na urządzeniu
* Zwykle granulacja jest na poziomie usług (kolejna usługa wymaga nowego logowania)

**Serwer proxy** to kolejny sposób dostępu do Internetu z jednoczesną izolacją sieci lokalnej

* Dla każdej usługi na serwerze proxy uruchamiany jest oddzielny program, który przekazuje i jednocześnie buforuje komunikaty przekazywane pomiędzy sieciami
* Może być tradycyjny (wymaga logowania) lub przezroczysty („transparent proxy”)
* Najczęściej łączony z technologią NAT – adresy w sieci lokalnej są nieroutowalne
* Pakiety nigdy nie przechodzą pomiędzy sieciami, są zawsze buforowane i ponownie wysyłane przez serwer proxy

# **Protokoły:**

## **Ogólnie:**



**Podział protokołów:**

* Statyczne
* Dynamiczne
* Wewnętrzne
* Zewnętrzne
* Protokoły TCP/IP

**Protokoły wchodzące w skład TCP/IP**

* **Protokoły poziomu sieciowego**
  + Address Resolution Protocol (ARP)
  + Internet Control Message Protocol (ICMP)
  + Internet Protocol (IP)
  + Transmission Control Protocol (TCP)
* **Protokoły aplikacyjne**
  + Simple Mail Transfer Protocol (SMTP)
  + Domain Name System (DNS)
  + Hypertext Transfer Protocol (HTTP)
  + Telnet
  + Gopher
  + File Transfer Protocol (FTP)
  + Network News Transport Protocol (NNTP)
  + I inne…… (MIME, Finger itp.)

## **Skróty:**

|  |  |  |  |
| --- | --- | --- | --- |
| Skrót: | Nazwa: | Działa w warstwie: | |
| **TCP/IP:** | **ISO/OSI:** |
| ARP | Adress Resolution Protocol | Warstwa dostępu do sieci | Łącza danych |
| ICMP | Internet Control Protocol | Warstwa Internetu | Warstwa sieciowa |
| IP | Internet Protocol | Warstwa Internetu | Warstwa sieciowa |
| TCP | Transmission Control Protocol | Warstwa transportowa | Warstwa transportowa |
| SMTP | Simple Mail Transfer Protocol | Warstwa aplikacji | Warstwa aplikacji |
| DNS | Domain Name System | Warstwa aplikacji | Warstwa aplikacji |
| HTTP | Hypertext Transfer Protocol | Warstwa aplikacji | Warstwa aplikacji |
| Telnet | --- | Warstwa aplikacji | Warstwa aplikacji |
| Gopher | --- | Warstwa aplikacji | Warstwa aplikacji |
| FTP | File Transfer Protocol | Warstwa aplikacji | Warstwa aplikacji |
| UDP | User Datagram Protocol | Warstwa transportowa | Warstwa transportowa |
| NNTP | Network News Transfer Protocol | Warstwa aplikacji | Warstwa aplikacji |
| SNMP | Simple Network Management Protocol | Warstwa aplikacji | Warstwa aplikacji |

## **ARP – Adress Resolution Protocol**

Protokół komunikacyjny przekształcania adresów IP na fizyczne, 48-bitowe adresy MAC w komputerowych sieciach lokalnych typu Ethernet.

Każdy komputer w sieci powinien posiadać tzw. tablicę ARP. Znajduje się w niej adres IP i przypisany do niego adres MAC. Dzięki temu komputery mogą się ze sobą komunikować za pośrednictwem adresu MAC, ale tylko w obrębie danej sieci LAN. Jeśli jakieś informacje mają być przesłane do innej sieci, to adres MAC musi być zastąpiony adresem IP.

ARP jest protokołem pracującym w warstwie łącza danych, ponieważ pracuje ona na ramkach i może je analizować tzn. np. sprawdzać ich poprawność.

Protokół ARP jest zdefiniowany w dyrektywie RFC 826.

## **ICMP – Internet Control Message Protocol**

Internetowy protokół komunikatów kontrolnych. ICMP jest ściśle związany z protokołem IP, dostarczając nieobecnej w nim funkcji informowania o błędach. Protokół pozwala wysyłać pakiety kontrolne służące do ustalenia bieżącego stanu hosta, w tym:

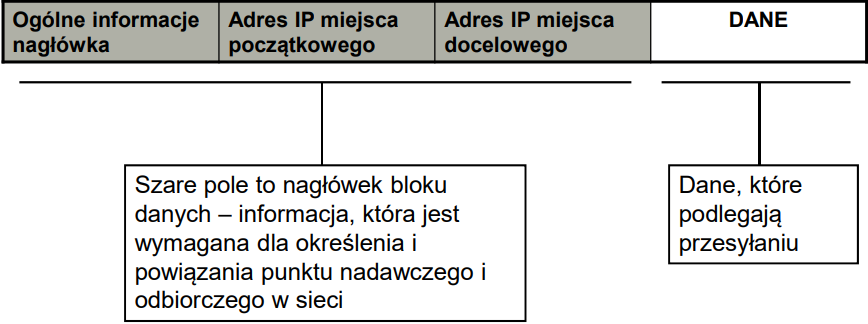
* Brak możliwości dostarczenia pakietu do miejsca przeznaczenia
* Określenia opóźnienia pakietów przesyłanych przez sieć
* Zmiana wcześniej wyznaczonej trasy przez jeden z routerów pośredniczących
* Brak wolnej pamięci buforowej dla zapamiętania pakietu i związane z tym chwilowe wstrzymania nadawania
* Przekroczenie czasu życia (TTL) pakietu

## **IP – Internet Protocol**

Dane w sieciach IP są wysyłane w formie bloków określanych mianem pakietów. W przypadku protokołu IP, przed rozpoczęciem transmisji nie jest zestawiana wirtualna sesja komunikacyjna pomiędzy dwoma hostami, które nie komunikowały się ze sobą wcześniej.

Protokół IP jest protokołem zawodnym **–** nie gwarantuje, że pakiety dotrą do adresata, nie zostaną pofragmentowane, czy też zdublowane, a ponadto mogą dotrzeć do odbiorcy w innej kolejności niż zostały nadane. Niezawodność transmisji danych jest zapewniana przez protokoły warstw wyższych (np. TCP), znajdujących się w hierarchii powyżej warstwy sieciowej.

**Datagram IP**



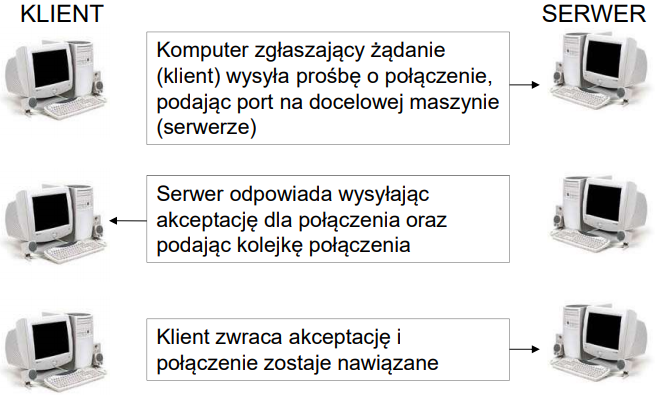
## **TCP – Transmission Control Protocol**

Strumieniowy protokół komunikacji między dwoma komputerami. W modelu OSI TCP odpowiada warstwie Transportowej.

W przeciwieństwie do UDP, TCP zapewnia wiarygodne połączenie dla wyższych warstw komunikacyjnych przy pomocy sum kontrolnych i numerów sekwencyjnych pakietów, w celu weryfikacji wysyłki i odbioru. Brakujące pakiety są obsługiwane przez żądania retransmisji. Host odbierający pakiety TCP porządkuje je według numerów sekwencyjnych tak, by przekazać wyższym warstwom modelu OSI pełen, złożony segment.

Charakterystyczny dla TCP jest moment nawiązania połączenia, nazywany ang. three-way handshake. Host inicjujący połączenie wysyła pakiet zawierający segment TCP z ustawioną flagą SYN (synchronize). Host odbierający połączenie, jeśli zechce je obsłużyć, odsyła pakiet z ustawionymi flagami SYN i ACK (acknowledge – potwierdzenie). Inicjujący host powinien teraz wysłać pierwszą porcję danych, ustawiając już tylko flagę ACK (gasząc SYN). Jeśli host odbierający połączenie nie chce lub nie może odebrać połączenia, powinien odpowiedzieć pakietem z ustawioną flagą RST (Reset). Prawidłowe zakończenie połączenia polega na wysłaniu flagi FIN.

**Three-Way Handshake**



## **SMTP – Simple Mail Transfer Protocol**

SMTP to względnie prosty, tekstowy protokół, w którym określa się co najmniej jednego odbiorcę wiadomości (w większości przypadków weryfikowane jest jego istnienie), a następnie przekazuje treść wiadomości.

SMTP zaczęło być szeroko używane we wczesnych latach osiemdziesiątych dwudziestego wieku. W tamtym okresie było to uzupełnienie UUCP, który lepiej sprawdzał się przy przekazywaniu poczty między maszynami posiadającymi jedynie okresowe połączenie. SMTP natomiast lepiej działa, gdy zarówno maszyna nadająca jak odbierająca są na stałe przyłączone do sieci.

Początkowo protokół ten nie radził sobie dobrze z plikami binarnymi, ponieważ stworzony był w oparciu o czysty tekst ASCII. W celu kodowania plików binarnych do przesyłu przez SMTP stworzono standardy takie jak MIME. W dzisiejszych czasach większość serwerów SMTP obsługuje rozszerzenie 8BITMIME pozwalające przesyłać pliki binarne równie łatwo jak tekst.

SMTP nie pozwala na pobieranie wiadomości ze zdalnego serwera. Do tego celu służą POP3 lub IMAP.

Jednym z ograniczeń pierwotnego SMTP jest brak mechanizmu weryfikacji nadawcy, co ułatwia rozpowszechnianie niepożądanych treści poprzez pocztę elektroniczną (wirusy, spam). Żeby temu zaradzić stworzono rozszerzenie SMTP-AUTH, które jednak jest tylko częściowym rozwiązaniem problemu - ogranicza wykorzystanie serwera wymagającego autoryzacji do zwielokrotniania poczty. Nadal nie istnieje metoda, dzięki której odbiorca autoryzowałby nadawcę - nadawca może "udawać" serwer i wysłać dowolny komunikat do dowolnego odbiorcy.

## **DNS – Domain Name System**

Jest to system serwerów oraz protokół komunikacyjny zapewniający zamianę adresów znanych użytkownikom Internetu na adresy zrozumiałe dla urządzeń tworzących sieć komputerową. Dzięki wykorzystaniu DNS nazwa mnemoniczna pl.wikipedia.org, może zostać zamieniona na odpowiadający jej adres IP, czyli 145.97.39.135.

Adresy DNS składają się z domen internetowych rozdzielonych kropkami. Dla przykładu w adresie Wikipedii org oznacza domenę funkcjonalną organizacji, Wikipedia domenę należącą do fundacji Wikimedia, a pl polską domenę w sieci tej instytucji. W ten sposób możliwe jest budowanie hierarchii nazw, które porządkują Internet.

DNS to złożony system komputerowy oraz prawny. Zapewnia z jednej strony rejestrację nazw domen internetowych i ich powiązanie z numerami IP. Z drugiej strony realizuje bieżącą obsługę komputerów odnajdujących adresy IP odpowiadające poszczególnym nazwom.

Podstawy protokołu DNS zostały opisane w 1982 roku w dokumencie IETF - RFC 819.

## **HTTP – Hypertext Transfer Protocol**

Protokół sieci WWW (World Wide Web).

Właśnie za pomocą protokołu HTTP przesyła się żądania udostępnienia dokumentów WWW i informacje o kliknięciu odnośnika oraz informacje z formularzy. Zadaniem stron WWW jest publikowanie informacji - natomiast protokół HTTP właśnie to umożliwia.

Protokół HTTP jest tak użyteczny, ponieważ udostępnia znormalizowany sposób komunikowania się komputerów ze sobą. Określa on formę żądań klienta dotyczących danych oraz formę odpowiedzi serwera na te żądania. Jest zaliczany do protokołów stateless (bezstanowy), z racji tego, że nie zachowuje żadnych informacji o poprzednich transakcjach z klientem, po zakończeniu transakcji wszystko "przepada" - z tego powodu tak bardzo spopularyzowały się cookies.

HTTP korzysta z portu nr 80.

Obecną definicję HTTP stanowi RFC 2616.

## **Telnet**

Telnet jest usługą (programem) pozwalającą na zdalne połączenie się komputera (terminala) z oddalonym od niego komputerem (serwerem) przy użyciu sieci, wykorzystując do tego celu protokół TCP-IP oraz standardowo przypisany port 23. Umożliwia on zatem ustanowienie użytkownikowi zdalnej sesji na serwerze tak, jak gdyby siedział tuż przed nim.

Protokół obsługuje tylko terminalne alfanumeryczne, co oznacza, że nie obsługuje myszy ani innych urządzeń wskazujących. Nie obsługuje także graficznych interfejsów użytkownika. Wszystkie polecenia muszą być wprowadzane w trybie znakowym w wierszu poleceń. Polecenia wydawane za pomocą naszego komputera przysłane są poprzez sieć serwera, na którym zainstalowane jest oprogramowanie serwera telnetu. W odpowiedzi serwer odsyła nam komunikaty, które następnie wyświetlane są na naszym ekranie.

Bardzo często usługa telnet implementowana jest do urządzeń aktywnych sieci (switche, routery) w celu ułatwienia konfiguracji tychże urządzeń.

Telnet jest najstarszą i najbardziej elementarną usługą internetową. Został opisany w dokumentach RFC numer RFC 854 i RFC 855.

## **Gopher**

Powstał w kwietniu 1991 roku na University of Minnesota Microcomputer, Workstation, Networks Center w celu dystrybucji informacji wewnątrz wydziałowej.

Był pierwszym rozpowszechnionym systemem informacyjnym w sieci integrującym różne protokoły: FTP, telnet, WAIS własne struktury danych z możliwością dostępu do różnych typów danych, tak czysto tekstowych, jak i grafik i danych czysto binarnych (archiwów wszelkiego rodzaju). Odchodzi powoli w zapomnienie (dogorywają jeszcze resztki dawnej jego świetności) z powodu sztywnej, hierarchicznej struktury (gdzie jednym z elementów ścieżki dostępu był typ pliku), niewygodnych metod tworzenia serwisów, braku pełnej "multimedialności" czy wreszcie dlatego, że WWW zyskał większe wsparcie tak producentów jak i środowisk akademickich.

## **FTP – File Transfer Protocol**

Protokół transmisji plików. Jest protokołem typu klient-serwer, który umożliwia przesyłanie plików z i na serwer poprzez sieć TCP/IP.

FTP jest protokołem 8-bitowym, dlatego nie wymaga specjalnego kodowania danych na postać 7-bitową, tak jak ma to miejsce w przypadku poczty elektronicznej.

Do komunikacji wykorzystywane są dwa połączenia TCP. Jedno z nich jest połączeniem kontrolnym za pomocą którego przesyłane są np. polecenia do serwera, drugie natomiast służy do transmisji danych m.in. plików. FTP działa w dwóch trybach: aktywnym i pasywnym, w zależności od tego, w jakim jest trybie, używa innych portów do komunikacji.

Jeżeli FTP pracuje w trybie aktywnym, korzysta z portów: 21 dla poleceń (połączenie to jest zestawiane przez klienta) oraz 20 do przesyłu danych. Połączenie nawiązywane jest wówczas przez serwer. Jeżeli FTP pracuje w trybie pasywnym wykorzystuje port 21 do poleceń i port o numerze > 1024 do transmisji danych, gdzie obydwa połączenia zestawiane są przez klienta.

Protokół ten jest zdefiniowany przez IETF w RFC 959.

## **UDP – User Datagram Protocol**

Jeden z podstawowych protokołów internetowych. Umieszcza się go w warstwie czwartej (transportu) modelu OSI.

Bezpośrednie dostarczanie protokołów, nie kontroluje poprawności i prędkości przesyłu, minimalizuje przesył dodatkowych danych, nie gwarantuje dostarczenia datagramu.

Jest to protokół bezpołączeniowy, więc nie ma narzutu na nawiązywanie połączenia i śledzenie sesji (w przeciwieństwie do TCP). Nie ma też mechanizmów kontroli przepływu i retransmisji. Korzyścią płynącą z takiego uproszczenia budowy jest szybsza transmisja danych i brak dodatkowych zadań, którymi musi zajmować się host posługujący się tym protokołem. Z tych względów UDP jest 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. Przykładem może być VoIP lub protokół DNS.

## **NNTP – Network News Transport Protocol**

Oparty o TCP/IP protokół polegający na przesyłaniu ciągów tekstowych przez siedmiobitowe kanały ASCII. Jest używany zarówno do przesyłania tekstów między serwerami, jak również do czytania i wysyłania artykułów. NNTP zdefiniowano w RFC 977 w roku 1986. W roku 2000 w RFC 2980 został on rozszerzony o nowe możliwości. Format wiadomości opisuje RFC 1036.

Na protokole tym oparte jest działanie usługi Usenet. NNTP działa standardowo na porcie nr 119.

## **SNMP – Simple Network Management Protocol**

Określa sposób komunikacji między programem - klientem do zarządzania siecią używanym przez administratora a serwerem działającym na komputerze lub ruterze.

Definiuje również zależności administracyjne oraz możliwość uwierzytelniania administratorów.

# **Poszczególne protokoły:**

## **ARP RARP – Detal**

Dwie maszyny mogą się komunikować tylko wtedy, kiedy znają nawzajem swoje adresy fizyczne.  
Zachodzi więc potrzeba przekształcenia adresu IP na adres fizyczny tak aby informacja mogła być poprawnie przesyłana

**Działanie ARP**

1. Porównać numery sieci (adresy IP)
   1. Takie same to wysłać na broadcast
   2. Inne wysłać zapytanie do routera
2. Jeżeli zna adres routera (sztywne ARP-y)
   1. To bezpośrednio do niego
3. Dostaje adres mac routera
4. Uzupełnia i wysyła do routera zapytanie
5. Router analizuje datagram IP
   1. Jeżeli w jego podsieci to wysyła zapytanie o mac
   2. Jeżeli nie to do kolejnego routera

**Tablica ARP**

Odwzorowanie adresów IP i sprzętowych

Tworzy się z zapytań i odpowie ARP

Dynamicznie tworzone i kasowane

Możliwość wpisów statycznych (podnosi bezpieczeństwo sieci)

**Proxy ARP**

Umożliwia ruterowi odpowiadać na zapytanie ARP kierowane z jednej dołączonej do niego sieci umieszczając informacje o hoście pracującym w drugiej sieci.

* Umieszczony pomiędzy sieciami ukrywa przed sobą istnienie pozwalając na korzystanie w obydwu z tego samego adresu sieci
* Pozwala na komunikacje źle skonfigurowanym hostom
* Pozwala na komunikacje z hostami nieobsługującymi podsieci

**Gratuitous ARP (nadmiarowy/niepotrzebny)**

* Wyklucza istnienie dwóch takich samych hostów o tym samym IP
* Pozwala na odświeżanie informacji po wymianie NIC

## **TCP – Detal**

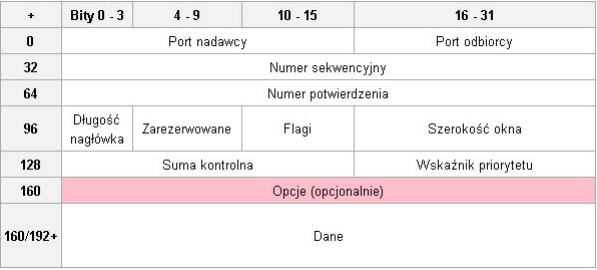
Chociaż protokół definiuje pakiet TCP, to z punktu widzenia wyższej warstwy oprogramowania, dane płynące połączeniem TCP należy traktować jako ciąg oktetów.

W szczególności – jednemu wywołaniu funkcji API (np. send()) nie musi odpowiadawysłanie jednego pakietu.

Dane z jednego wywołania mogą zostać podzielone na kilka pakietów

Dane z kilku wywołać mogą zostać połączone i wysłane jako jeden pakiet (dziki użyciu algorytmu Nagle'a). Równie funkcje odbierające dane (recv()) w praktyce odbierają nie konkretne pakiety, ale zawartość bufora stosu TCP/IP, wypełnianego sukcesywnie danymi z przychodzących pakietów.

**Nagłówek TCP**



**Zawartość pól:**

**Port źródłowy** (16 bitów) i pole port docelowy (16 bitów) zawierają numery portów procesów aplikacyjnych korzystających z usług TCP. Kombinacja tych numerów z adresami sieciowymi określa par gniazd tworzących połączenie protokołu TCP.

**Numer sekwencyjny** (32 bity) zawiera numer sekwencyjny pierwszego bajtu danych w segmencie. Ta wartość określa pozycję segmentu w strumieniu bajtów. Podczas ustanawiania połączenia, i jeśli bit syn w polu znaczniki jest ustawiony na 1, to w tym polu zawarty jest inicjujący numer sekwencyjny ins, od którego rozpoczyna się numeracja bajtów w połączeniu. Zatem pierwszy wysłany bajt ma numer ins + 1.

**numer potwierdzenia** (32 bity) zawiera numer sekwencyjny następnego oczekiwanego bajtu po stronie odbiorczej. Jednocześnie jest to potwierdzenie poprawnego odbioru bajtów o numerach sekwencyjnych mniejszych od zawartego w tym polu. Potwierdzenia mówią nadawcy, ile bajtów danych zostało już poprawnie odebranych.

**Długość nagłówka** (4 bity) określa liczb 32 - bitowych słów w nagłówku segmentu TCP. Tym samym określone zostaje miejsce, w którym rozpoczynają się dane. Pole to ma tak określone znaczenie tylko wtedy, gdy bit ack równy jest 1.

**Rezerwa** (6 bitów) jest przeznaczone dla przyszłych zastosowań. Zawiera same zera.

**Znaczniki** składają się z sześciu bitów sterujących, które ustawione na 2 mają następujące znaczenie:

* **UGR** - wskazuje na ważność pola wskaźnik pilności,
* **ACK** - wskazuje na ważność pola numer potwierdzania,
* **PSH** - wskazuje na działanie funkcji wymuszającej wysyłanie segmentu,
* **RST** - wyzerowanie połączenia,
* **SYN** - wskazuje, że w polu numer sekwencyjny umieszczony jest inicjujący numer sekwencyjny INS. Jest on przeznaczony do synchronizacji numerów sekwencyjnych w fazie ustanowienia połączenia.
* **FIN** - wskazuje, że nadawca nie ma nic więcej do nadania - sygnał końca danych.

**Okno** (16 bitów) określa liczbę bajtów jaką może jeszcze zaakceptować odbiorczy moduł TCP.

**Suma kontrolna** jest 16 - bitowym jedynkowym uzupełnieniem jedynkowo uzupełnionej sumy wszystkich 16 - bitowych słów w segmencie. Ta suma obejmuje zarówno nagłówek jak i dane segmentu.

**Wskaźnik pilności** (16 bitów) jest interpretowane tylko wtedy, gdy bit UGR jest równy 1. Pole to zawiera numer sekwencyjny bajtu następującego po pilnych danych.

**Opcje** ma długość zmienną będącą wielokrotnością 8 bitów. Zawiera ono numery opcji - każdy numer zapisany w jednym bajcie. Dla protokołu TCP zdefiniowano trzy opcje:

* 0 - koniec listy opcji
* 1 - brak działania
* 2 - maksymalna długość segmentu

**Wypełnienie** uzupełnia nagłówek do wielokrotności 32 bitów

## **UDP – Detal**

DOKOŃCZYĆ

# **Routing**

**Routing** – wyznaczanie trasy i wysłanie nią pakietu danych w sieci komputerowej.

## **Routing statyczny**

Polega on na trwałej konfiguracji routerów w ten sposób, że pakiety o podanych adresach wysyłane są na z góry ustalony interfejs. Samo przydzielenie adresów IP sieciom, interfejsom routerów i wszystkich komputerów wymaga trochę pracy. Administrator musi jednak jeszcze ustalić zasady routingu między wszystkimi sieciami. Dokonuje tego przez budowę tak zwanych tablic routingu na wszystkich routerach.

## **Routing dynamiczny**

Realizacja routingu statycznego wymaga od administratora trochę pracy. Wprawdzie konfiguracja routerów wykonywana jest tyko raz i potem sieć po prostu działa, ale w razie zmiany topologii sieci (na przykład dołożenie lub usunięcia podsieci) procedura musi być ponownie przeprowadzona. W przypadku awarii któregoś routera lub jednego z łączy konieczna jest szybka interwencja administratora i przekonfigurowanie routerów, bo bez tego cześć lub całość sieci nie działa.

## **Routing wewnętrzny**

Jest to routing wykonywany wewnątrz pojedynczego systemu autonomicznego.

Wyróżniamy następujące protokoły routingu wewnętrznego:

* RIP
* OSPF
* IGRP, EIGRP
* OSI

## **Routing zewnętrzny**

Jest routingiem wykonywanym między wieloma systemami autonomicznymi.

Wyróżniamy następujące protokoły routingu zewnętrznego:

* EGP – Exterior Gateway Protocol
* BGP – Border Gateway Protocol
* CIDR – Classless Inter-Domain Routing

## **Protokoły routingu**

Routery budują tablice routingu na podstawie informacji wymienionych z innymi routerami. Wymiana ta opiera się na protokołach routingu. Mają one za zadanie poinformować inne węzły sieci o sieciach, do których dany router ma dostęp. Takie rozwiązanie pozwala na dynamiczne budowanie struktury. Dołączenie kolejnej sieci do jednego z routerów nie wymaga rekonfiguracji pozostałych węzłów sieci. Zostaną one automatycznie „poinformowane” o zaistniałych zmianach.

**Ze względu na sposoby działania rozróżnia się następujące protokoły routingu:**

* **Protokoły wektora odległości** (ang. distance vector protocols) — wysyłają w określonych interwałach czasowych do sąsiednich routerów zawartość tablicy routingu wraz z metrykami. Jeśli dana trasa nie jest dostępna (nie dotarła informacja od sąsiedniego routera), wówczas wpis dotyczący trasy i sieci, które były osiągalne, zostaje usunięty z tablicy routingu i — o ile to możliwe — jest zastępowany innym wpisem (np. wcześniej odrzuconym jako mniej korzystny).
* **Protokoły stanu łącza** (ang. link state protocols) wysyłają do wszystkich routerów informację, która zawiera jedynie dane o podsieciach podłączonych do routera. Aktualizacje informacji są wysyłane okresowo lub wywoływane zmianami zachodzącymi w sieci.

### **Router**

**Routery** to urządzenia brzegowe, które znajdują się na styku dwóch sieci działających w innych klasach adresacji IP. Mają za zadanie przesyłać pakiety do adresata, a dokładnie do sieci, w której znajduje się jego adres IP. Pakiet zaadresowany do komputera znajdującego się w naszej sieci jest kierowany bezpośrednio do niego. Jeśli ma zostać wysłany poza sieć, trafia do routera, który sprawdza, czy pakiet ten jest kierowany do sieci bezpośrednio podłączonej do tego routera, czy ma być przesłany do urządzenia znajdującego się poza podłączonymi do niego sieciami. Pakiety wędrują od jednego węzła (routera) do drugiego poprzez wiele węzłów pośredniczących, często mogą też być transmitowane różnymi trasami. Zadaniem routera jest wybrać najlepszą dostępną drogę pomiędzy jednym a drugim węzłem. Decyzja o wyborze trasy jest podejmowana

na podstawie wpisów znajdujących się w tablicy routingu — spisie sieci podłączonych bezpośrednio do routera oraz sieci dostępnych na routerach sąsiadujących.

## **Numery portów**

**Numery portów** są dodawane do segmentów na poziomie warstwy czwartej (przez protokoły TCP i UDP). Numery portów zapewniają, że dane zostaną przetworzone przez konkretną aplikację. Na przykład podczas pobierania stron WWW zapytanie ze strony przeglądarki jest wysyłane na port 80 wybranego serwera WWW. Portem nadającym jest pierwszy wolny port powyżej 1023. Dane trafiają do serwera WWW na port 80 — jest to port, za którego obsługę odpowiada serwer HTTP. Serwer WWW wysyła dane (wybraną stronę) do klienta, kierując odpowiedź na port, z którego przyszło zapytanie.