Sieci komputerowe – podstawowe pojęcia

Definicja **sieci komputerowej** mówi, że jest to połączenie przynajmniej dwóch komputerów w celu wymiany danych. Urządzenia komunikują się ze sobą za pomocą mediów transmisyjnych wykorzystując odpowiednie protokoły komunikacyjne.

Obecnie definicję tą trzeba rozszerzyć, ponieważ zwiększyła się liczba urządzeń podłączanych do sieci i korzystających z niej. Dziś do sieci komputerowej podłączamy nie tylko komputery czy serwery, ale również telefony komórkowe, tablety, telewizory, a nawet pralki, lodówki oraz inne urządzenia AGD. Oczywiście podstawa działania sieci komputerowych się nie zmieniła, dalej tworzy się sieci aby udostępniać dane (pliki, strony WWW, usługi sieciowe) oraz zasoby sprzętowe (drukarki, dyski, skanery, napędy optyczne), natomiast różnica polega na tym, że korzysta z niej o wiele więcej urządzeń niż kiedyś.

Pojęcia związane z siecią komputerową:

Host – urządzenie końcowe sieci komputerowej, stanowiące źródło lub celu przesyłania danych w sieci. Hostem jest każde urządzenie w sieci, któremu przypisano adres IP.

Serwer – komputer, na którym zainstalowane jest specjalistyczne oprogramowanie, oferujący usługi innym komputerom (WWW, poczta elektroniczna, zasoby plikowe).

Klient – komputer korzystający z usług udostępnianych przez serwery.

Klient – Serwer – architektura sieci komputerowej, w której występuje komputer udostępniający usługi (serwer) oraz komputery korzystające z tych usług (klienci).

Peer to Peer (P2P) – architektura sieci komputerowej, w której nie występuje jeden komputer udostępniający usługi, wszystkie komputery w tej sieci działają na równorzędnych prawach.

Medium transmisyjne – element sieci komputerowej, za pomocą którego wzajemnie komunikują się urządzenia, może nim być kabel miedziany, światłowodowy, jak również fale radiowe (WiFi).

Protokół komunikacyjny – określony sposób (język) komunikacji, dzięki któremu możliwa jest wymiana danych pomiędzy urządzeniami w sieci.

LAN (ang. Local Area Netwok) – lokalna sieć komputerowa, obejmująca swoim zasięgiem pomieszczenie, piętro, budynek lub zbiór budynków.

MAN (ang. Metropolitan Area Netwok) – miejska sieć komputerowa, obejmująca swoim zasięgiem miasto lub aglomerację miejską.

WAN (ang. Wide Area Netwok) – rozległa sieć komputerowa, komunikująca ze sobą odległe sieci LAN.

Topologia fizyczna sieci – określa sposób połączenia ze sobą komputerów w sieci.

Topologia logiczna sieci – określa sposób komunikowania ze sobą komputerów w sieci.

Karta sieciowa (ang. NIC – Network Interface Card) – adapter instalowany w urządzaniach (komputerach, telefonach, itp.), dzięki któremu możliwe jest podłączenie tego urządzenie do sieci komputerowej.

Ruter (ang. Router) – urządzenie sieciowe, którego głównych zadaniem jest łączenie ze sobą różnych sieci, w celu umożliwienia im wzajemnej komunikacji oraz określanie ścieżki przepływu danych pomiędzy sieciami na podstawie adresu IP (ang. Routing).

Przełącznik (ang. Switch) – urządzenie sieciowe, które łączy urządzenia w sieci lokalnej oraz decyduje o przesyle danych pomiędzy urządzaniami na podstawie adresu MAC.

Adres IP – logiczny adres interfejsu urządzenia w sieci komputerowej.

Adres MAC – fizyczny (sprzętowy) adres karty sieciowej urządzenia, nadawany przez producenta na etapie produkcji.

Internet (Intersieć)– siatka połączonych ze sobą sieci rozległych.

Intranet – prywatna sieć, wykorzystująca w komunikacji standardy sieci Internet, takie jak WWW, FTP czy POP3 i SMTP, do której dostęp mają tylko upoważnieni użytkownicy (np. do Intranetu w firmie XYZ mają dostęp tylko pracownicy tej firmy)

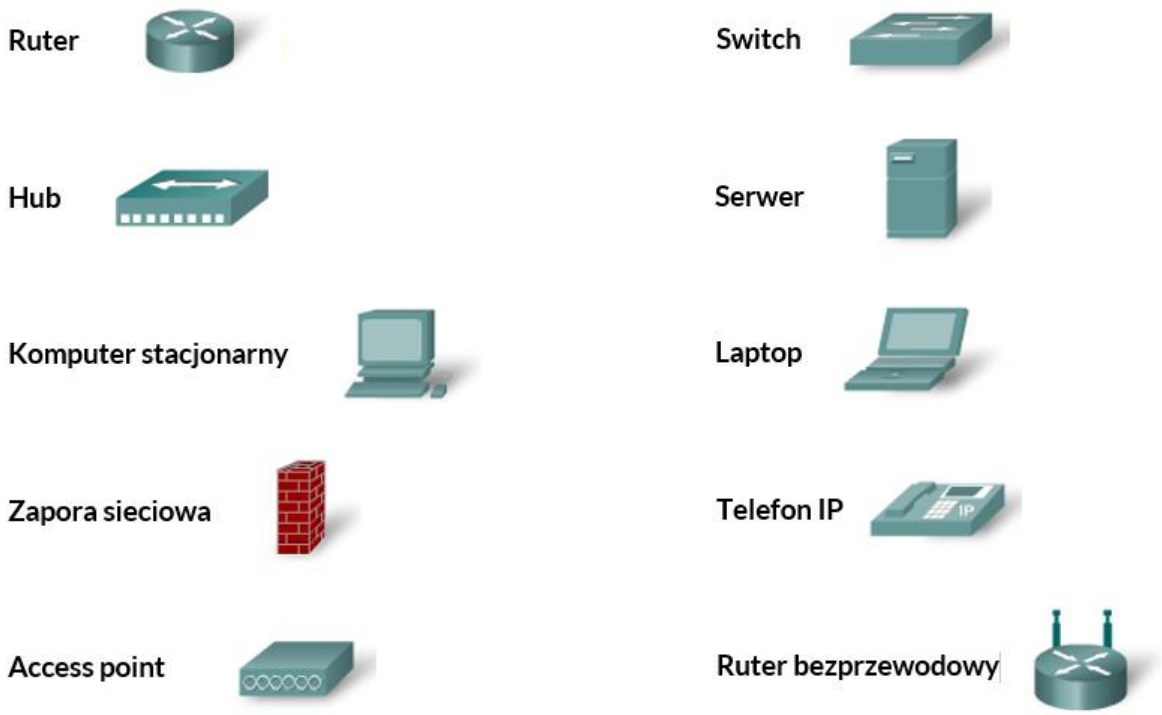
Extranet – rozszerzenie sieci prywatnej (Intranet), umożliwiające dostęp do jej zasobów również dla innych użytkowników.

VPN – prywatna sieć, do zasobów której możemy się dostać przez sieć Internet, wykorzystując tzw. tunelowy kanał transmisji danych.

DNS (ang. Domain Name System) – system zmieniający nazwę mnemoniczną (zrozumiałą dla człowieka), np. onet.pl na odpowiadający jej adres IP w sieci.

Symbole graficzne sieci komputerowej

Projektując sieć komputerową, czy to na papierze czy z wykorzystaniem odpowiedniego oprogramowania, uwzględnić należy rozmieszczenie urządzeń wchodzących w skład naszej sieci. Aby orientować się jakie urządzenie znajduje się w określonym położeniu powinniśmy poznać ich graficzne oznaczenie.
Poniżej przedstawiam Wam zatem symbole graficzne najczęściej wykorzystywanych urządzeń sieciowych.



Znajomość tych symboli będzie Wam pomocna podczas projektowania i symulacji działania sieci w programach komputerowych, takich, jak np. Cisco Pacet Tracert.

Jednostki przesyłu danych w sieciach komputerowych

Sieć komputerowa jest to połączenie ze sobą komputerów (oraz innych urządzeń) w celu wymiany informacji. Wymiana ta odbywa się z daną prędkością, która określa ilość przesłanych danych w jednostce czasu. Podstawową jednostką służącą w informatyce do zapisu danych jest **1 bit[b]**, zatem w zapisie prędkości, z jaką dane w sieci są przesyłane, stosuję się określenie **bity na sekundę** (**b/s** lub też **bps** – bit peer second).

Oczywiście 1 bit informacji to bardzo mało, dlatego też w określaniu prędkości stosuje się jego wielokrotności. Powinny one być już Wam znane z określania np. pojemności dysków czy pamięci operacyjnych, są nimi **Kilo**, **Mega, Giga**, **Tera**, **Peta.** Pierwsze 3 przeliczniki przedstawiają się w liczbach następująco:

1 Kilobit (1 Kb/s lub 1 Kbps) – **1024 bity/s** (2^{10})

1 Megabit (1 Mb/s lub 1 Mbps) – 1024 Kb/s – **1 048 576 bity/s** ($2^{10}*2^{10}$)

1 Gigabit (1 Gb/s lub 1 Gbps) – 1024 Mb/s – 1 048 576 Kb/s – **1073741824 bity/s** ($2^{10}*2^{10}*2^{10}$)

Zamiana jednostek

Jak już wiemy, prędkość przesyłu danych w sieciach komputerowych określa się w bitach na sekundę. Inaczej niż w przypadku wielkości plików czy pojemności dysków, gdzie zamiast **bitów [b]** stosuję się **bajty [B]**. **1 bajt** [B] to **8 bitów** [b] dlatego też, jeśli chcielibyśmy obliczyć jaką ilość danych jesteśmy w stanie przesłać w jednostce czasu musimy dokonać zamiany jednostek. Proces ten wyjaśnię Wam na przykładach.

<u>Zadanie 1</u>

Obliczymy ile danych pobierzemy z Internetu w czasie jednej godziny przy założeniu, że przepustowość naszego łącza jest stała i wynosi 60 Mb/s.

Dane:

Czas: 1 godzina

Przepustowość łącza: 60 Mb/s

Obliczenia:

1. Zamieniamy jednostkę czasu z godziny na sekundy: **1 godzina to 3600 sekund** (60 sekund*60 minut).
2. Zamieniamy jednostkę przesyłu danych z megabitów na megabajty na sekundę: **60 megabitów to 7,5 megabajtów na sekundę** (60 Mb dzielimy przez 8, ponieważ 1 bajt to 8 bitów).
3. Mnożymy przepustowość przez czas: **7,5 MB/s * 3600 sekund, co daje 27000 MB** (ok. 26,3 GB danych)

Odpowiedź: ok. 26,3 GB.

<u>Zadanie 2</u>

Obliczmy, w jakim czasie pobierzemy plik w wielkości 1 GB przy założeniu, że przepustowość naszego łącza jest stała i wynosi 10 Mb/s.

Dane:

wielkość pliku: 1 GB
prędkość łącza: 10 Mb/s

Obliczenia:

1. Zamieniamy jednostkę przesyłu danych z megabitów na megabajty na sekundę: **10 megabitów to 1,25 megabajtów na sekundę** (10 Mb dzielimy przez 8, ponieważ 1 bajt to 8 bitów).
2. Zamieniamy jednostkę zapisu pliku z gigabajta na megabajty: **1 GB to 1024 MB**.
3. Obliczamy czas przesłania pliku: **1024 MB / 1,25 Mb/s = 819,2 sekundy, co daje ok. 13 minut 39 sekund**.

Odpowiedź: 13 minut 39 sekund.

Zadanie do samodzielnego rozwiązania:

Oblicz prędkość przesłania 10 plików o łącznej wielkości 16,2 GB przez łącze o stałej prędkości 12 Mb/s.

Standardy sieci komputerowych

IEEE 802 jest to grupa standardów stosowanych w sieciach komputerowych. Najpopularniejsze standardy to:

IEEE 802.1D – standard opisujący pracę protokołu drzewa rozpinającego STP (ang. Spanning Tree Protocol), który implementowany jest w przełącznikach sieciowych (ang. Switch). Jego podstawową funkcją jest zapobieganie powstawaniu pętli w lokalnych sieciach komputerowych.

IEEE 802.p – standard opisujący działanie usługi QoS na przełącznikach sieciowych.

IEEE 802.s – standard opisujący mechanizm działania protokołu MSTP (ang. Multiple Spanning Tree Protocol), stosowanego w wirtualnych sieciach lokalnych (VLAN).

IEEE 802.1w – standard opisujący pracę rozszerzonego protokołu drzewa rozpinającego RSTP (ang. Rapid Spanning Tree Protocol).

IEEE 802.1Q – standard opisujący działanie wirtualnych sieci LAN (VLAN).

IEEE 802.2 – standard opisujący działanie podwarstwy LLC w warstwie łącza danych (2 warstwa modelu ISO/OSI).

IEEE 802.3 – 10BASE5 – **ETHERNET** – standard lokalnych sieci komputerowych.

Poszczególne wersje standardu **ETHERNET**:

-
  - **IEEE 802.3a** – **10BASE2** – standard wykorzystujący cienki kabel koncentryczny jako nośnik danych;
  - **IEEE 802.3d** – **Fiber Optic** – standard wykorzystujący światłowód jako nośnik danych;
  - **IEEE 802.3i** – **10BASE-T** – standard **Ethernet** wykorzystujący skrętkę jako nośnik danych;
  - **IEEE 802.3j** – **10BASE-F** – standard **Ethernet** wykorzystujący światłowód jako nośnik danych;
  - **IEEE 802.3u** – **100BASE-TX**, **100BASE-FX** – standard **Fast Ethernet** oparty na skrętce (Tx), oraz na światłowodzie (FX);
  - **IEEE 802.3z** – **1000BASE-X** – standard **Gigabit Ethernet** oparty na światłowodzie;
  - **IEEE 802.3ab** – **1000BASE-T** – standard **Gigabit Ethernet** oparty na skrętce;
  - **IEEE 802.3ac** – standard zwiększający maksymalną wielkość ramki **Ethernetowej do 1522 B**, pozwalający obsługiwać „tagowane” ramki z sieci VLAN;
  - **IEEE 802.3ad** – **Link Aggregation** – standard umożliwiający agregację portów przełącznika;
  - **IEEE 802.3ae** – **10 Gigabit Ethernet** – Standard oparty na kablu światłowodowym;
  - **IEEE 802.3af** – **Power over Ethernet** (PoE) – Standard umożliwiający korzystanie z urządzeń sieciowych takich jak **Access Point’y** bez konieczności podłączenia go do zewnętrznego zasilania. Urządzenie zasilane jest poprzez port przełącznika. Maksymalnie do **15,4W**;
  - **IEEE 802.3an** – **10 Gigabit Ethernet** – standard oparty na skrętce;
  - **IEEE 802.3aq** – **10 Gigabit Ethernet** – standard oparty na kablu światłowodowym wielomodowym;

- **IEEE 802.3at** – **Power over Ethernet** (PoE) – Standard umożliwiający korzystanie z urządzeń sieciowych takich jak **Access Point'y** bez konieczności podłączenia go do zewnętrznego zasilania. Urządzenie zasilane jest poprzez port przełącznika. Zapewnia moc powyżej **25,5W,** dla skrętki cat. 5.

IEEE 802.5 – Token Ring – standard sieci LAN, alternatywny dla standardu Ethernet.

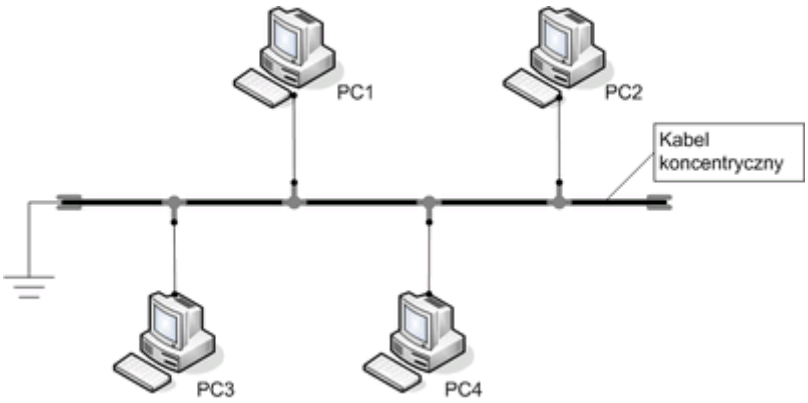
IEEE 802.11 – WiFi – standard sieci bezprzewodowych.

Topologie sieci komputerowych

Topologia sieci komputerowej określa relację pomiędzy urządzeniami w sieci, połączenia między nimi oraz sposób przepływu danych. Topologię należy rozpatrywać w dwojaki sposób, w sposób fizyczny (zwany topologią fizyczną) oraz logiczny (zwany topologią logiczną).

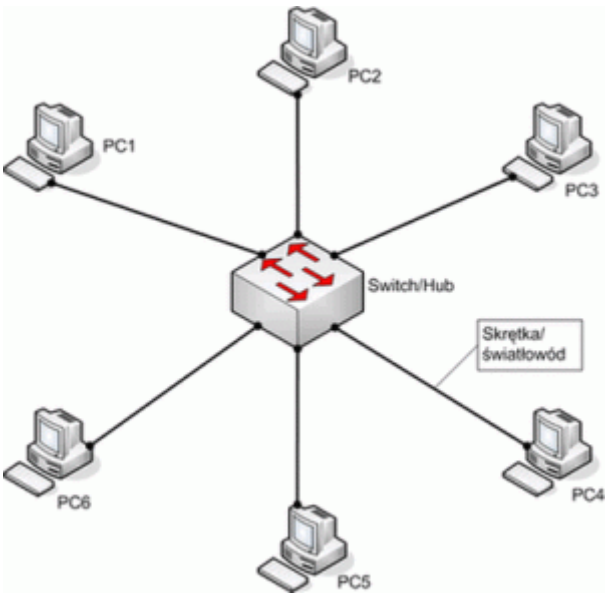
Topologie fizyczne odzwierciedlają fizyczne rozmieszczenie urządzeń w sieci oraz połączenia między nimi:

Topologia magistrali



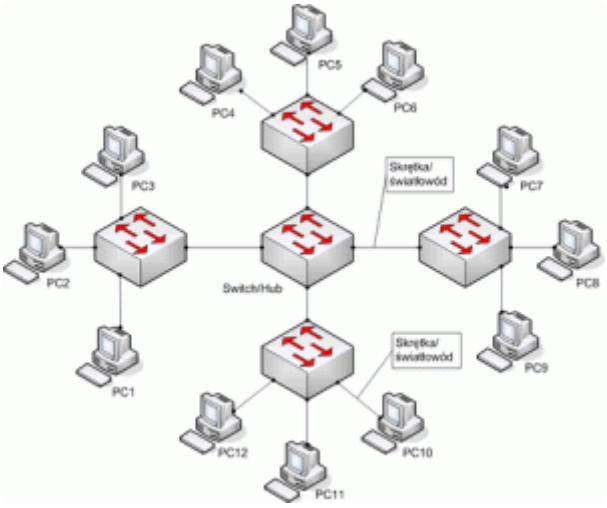
W tej topologii wszystkie urządzenia podłącza się do wspólnego medium transmisyjnego. Powszechnie stosowanym w tej topologii medium był kabel koncentryczny, zarówno cienki, jak i gruby. Zastosowanie „koncentryka" jako fizyczne medium stanowiło jedną z wad tej topologii, a mianowicie niewielką przepustowość (maksymalnie do 10 Mb/s). Topologia ta stosowana była do budowy lokalny sieci komputerowych, „była" jest tutaj słowem właściwym ponieważ nie jest już powszechnie stosowana. Poza niską przepustowością, charakteryzowała ją również duża podatność na awarię sieci. W momencie przerwania kabla koncentrycznego cała sieć przestawała działać. Niewątpliwą zaletą w zastosowaniu tej topologii był niewielki koszt jej wdrożenia, ponieważ nie trzeba było stosować setek metrów kabla ani żadnych urządzeń pośredniczących.

Topologia gwiazdy



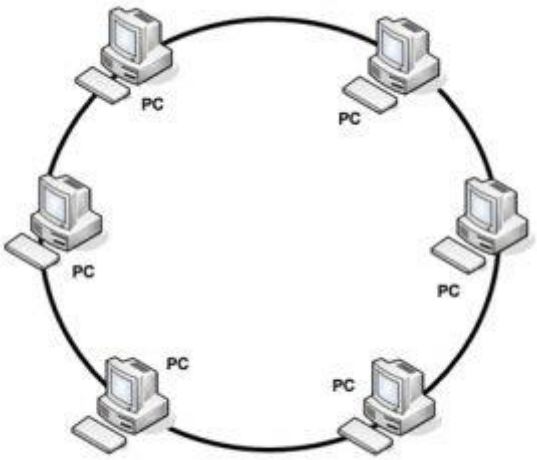
W tej topologii wszystkie urządzenia podłączone są do centralnego punktu, stanowiącego punkt dostępu do sieci. Dawniej punkt ten stanowiły koncentratory (ang. hub), obecnie natomiast stosuje się przełączniki (ang. switch). W lokalnych sieciach jest to najczęściej spotykana topologia, ponieważ jest prosta w zaprojektowaniu, budowie oraz rozbudowie, odporna na awaria i łatwo zarządzalna. Dodatkowym plusem jest fakt, iż można przy jej budowie wykorzystać różne media transmisyjne, takie jak miedziana skrętka, kabel światłowodowy czy fale radiowe (WLAN). Istotną wadę stanowić może natomiast koszt budowy, ponieważ wymagane jest zastosowanie dodatkowych urządzeń (switch'y) oraz wiele metrów okablowania.

Topologia gwiazdy rozszerzonej



Rozwinięcie topologii gwiazdy stosowane w dużych sieciach LAN oraz w sieciach kampusowych.

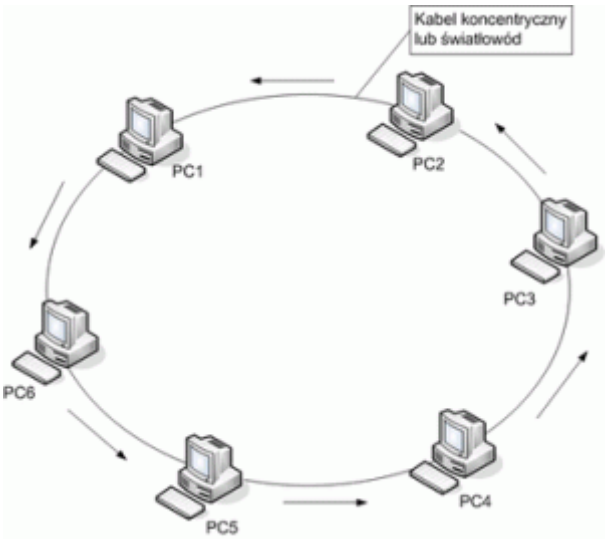
Topologia pierścienia



W tej topologii każde urządzenie podłączone jest z dwoma sąsiadami, tworząc zamknięty krąg. Podobnie jak w przypadku topologii magistrali, przy budowie nie stosuję się dużej ilości okablowania oraz brak jest konieczności stosowania dodatkowych urządzeń.

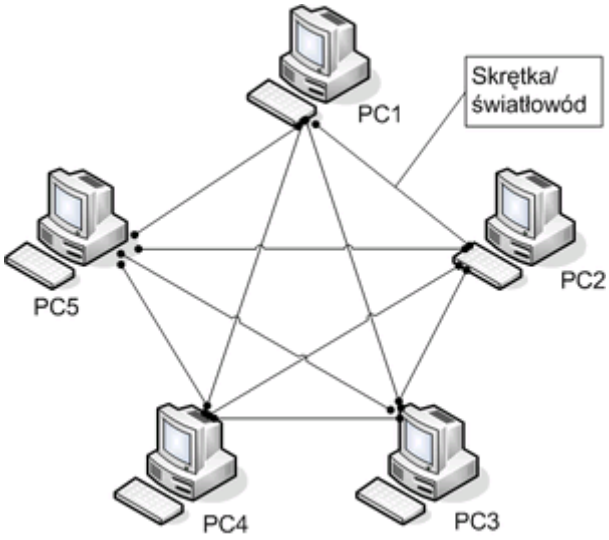
Ponadto można wykorzystać różne media transmisyjne, począwszy od kabla koncentrycznego, po skrętkę miedzianą, aż do kabli światłowodowych. Wadą tego rodzaju topologii jest fakt, iż przerwanie medium lub awaria jednego z komputerów powoduje przerwę w działaniu całej sieci.

Topologia podwójnego pierścienia



Bliźniacza do topologii pierścienia, różnicą się od pierwowzoru zastosowaniem podwójnego połączenia urządzeń, tak aby zabezpieczyć prawidłowe działanie sieci, w momencie awarii jednego łącza. Stosowana w budowie dużych sieci szkieletowych.

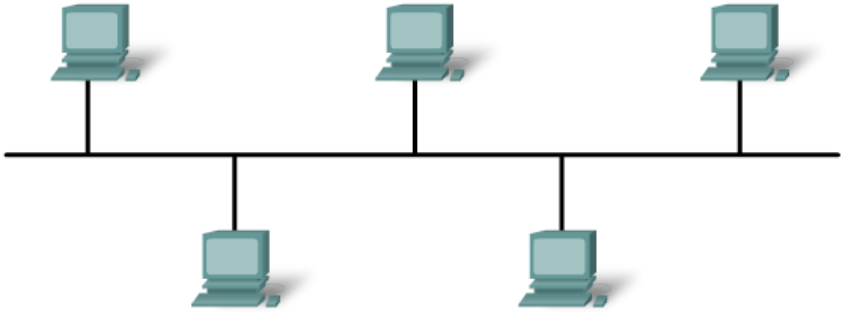
Topologia siatki



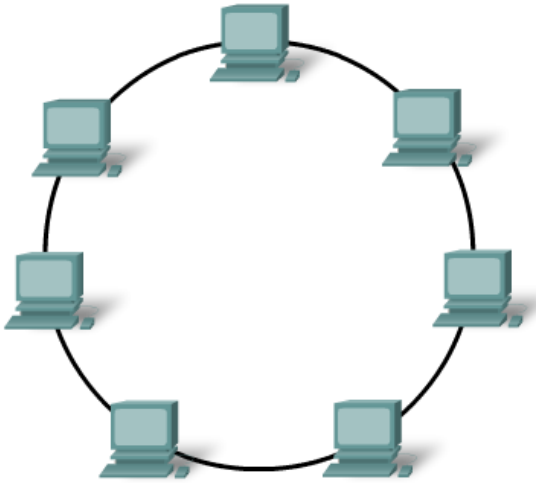
Topologia stosowana przy budowie rozległych sieci, polegająca stosowaniu nadmiarowych połączeń pomiędzy urządzeniami w celu zapewnienia najwyższego stopnia bezpieczeństwa i niezawodności działania sieci.

Topologie logiczne określają sposób przepływu danych w sieci:

Topologia punkt-punkt – w tej topologii przesył danych odbywa się tylko pomiędzy dwoma urządzeniami w sieci, które mogą być do siebie podłączone bezpośrednio, jak również z wykorzystaniem urządzeń pośredniczących (np. switch'y).



Topologia wielodostępowa – w tej topologii przesył danych odbywa się poprzez jedno medium transmisyjne (np. kabel koncentryczny), które współdzielone jest przez wiele urządzeń. Urządzenie transmitujące dane wysyła je do wszystkich (ang. broadcasting), ale tylko urządzenie, do którego dane były adresowane odczytuje je, pozostałe urządzenia je ignorują.



Topologia pierścienia – podobnie jak w topologii wielodostępowej również tutaj dane wysyłane są do wszystkich urządzeń, a tylko urządzenie, do którego były adresowane je odczytuje. W pierścieniu umieszczony jest „żeton" (ang. token), który „krąży" pomiędzy urządzeniami. To urządzenie, które chce nadawać „przechwytuje" go i rozpoczyna transmisję.

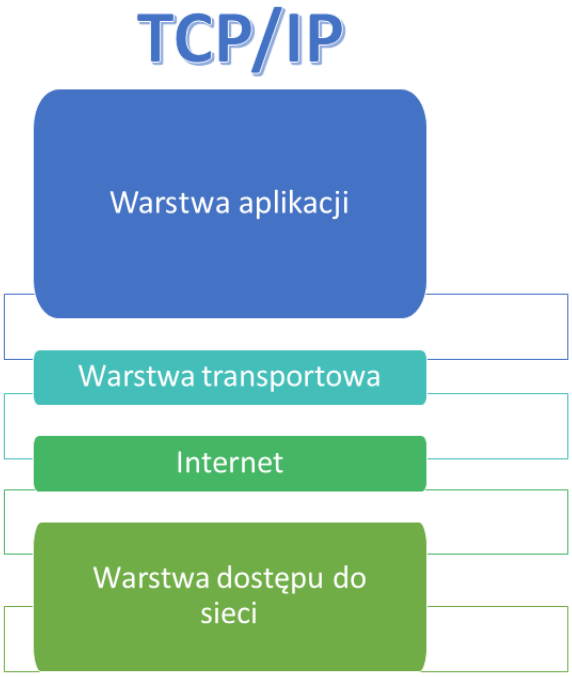
Warstwy sieci komputerowych – model OSI i TCP/IP

Komunikacja pomiędzy urządzeniami w sieciach komputerowych, jak każdy inny typ komunikacji rządzi się określonymi prawami i regułami. Reguły te określa się mianem **protokołów komunikacyjnych**. Różnią się one pomiędzy sobą w zależności od rodzaju komunikacji. Dokładnie tak samo jak w życiu codziennym, inny rodzaj zasad rządzi przy połączeniach telefonicznych, inny przy przekazywaniu informacji poprzez popularne komunikatory.

Aby w sieciach komputerowych komunikacja przebiegała we właściwy sposób wymagane jest współdziałanie wielu protokołów, które określa się mianem **zestawu protokołów**. Na zestaw taki składają się zarówno składniki programowe, jak również urządzenia sieciowe. Najbardziej efektywną metodą wizualizacji współdziałania protokołów jest postrzeganie ich jako nałożone na siebie warstwy stanowiące swojego rodzaju stos.

Stosowanie takiej metody pozwala na rozbicie skomplikowanego zagadnienia jakim jest komunikacja w sieci na mniejsze elementy, w celu prostszego ich opisania i zrozumienia. Ponadto stosowanie modelu warstwowego niesie za sobą szereg innych korzyści, takich jak łatwiejsze projektowanie protokołów czy możliwość współdziałania urządzeń sieciowych rożnych producentów.

Istnieją dwa podstawowe modele odniesienia: model **OSI** oraz model **TCP/IP**.



Model **TCP/IP** powstał w latach siedemdziesiątych XX wieku wieku i określany jest mianem modelu sieci Internet. Składa się z czterech warstw:

- **warstwy aplikacji**, zajmującej się reprezentacją danych dla użytkownika oraz ich kodowaniem,
- **warstwy transportowej**, zapewniającej komunikację pomiędzy różnymi urządzeniami w sieci,
- **internet**, zapewniający najlepszą trasę dla przepływu pakietów,
- **warstwy dostępu do sieci** kontrolującej urządzenia fizyczne i media.

Model **OSI** jest modelem przygotowanym z myślą o tworzeniu tzw. „otwartego systemu", który nie będzie należał do żadnej zamkniętej sieci. Szybki rozwój sieci Internet opartej na modelu TCP/IP spowolnił jednak znacznie jego wdrażanie. Mimo tego udało się stosując model **OSI** wdrożyć kilka istotnych protokołów. W przeciwieństwie do modelu **TCP/IP**, składa się on z siedmiu warstw:

- **warstwy aplikacji**, umożliwiającej komunikację z użytkownikiem (strony WWW, poczta elektroniczna),
- **warstwy prezentacji,** zajmującej się konwersją danych i definiowaniem formatu (dzięki temu dane pochodzące z urządzenia źródłowego mogą być interpretowane przez odpowiednie aplikacje na urządzeniu docelowym)
- **warstwy sesji,** zarządzającej przebiegiem komunikacji,
- **warstwy transportowej,** odpowiedzialnej za integralność (czyli spójność) transmisji danych,
- **warstwy sieci,** zajmującej się określaniem trasy przepływu danych,
- **warstwy łącza danych,** opisującej metody wymiany ramek pomiędzy urządzeniami podłączonymi do tego samego medium (np. do przełącznika),
- **warstwy fizycznej,** sterującej przepływem bitów poprzez media sieciowe.

Pomimo różnic w każdym z tych modeli pojawiają się te same protokoły oraz urządzenia sieciowe co widać na zamieszczonej poniżej tabeli.

Model OSI	Model TCP/IP	PROTOKÓŁ / urządzenie sieciowe
Warstwa aplikacji	Warstwa aplikacji	HTTP, TELENT SSH, FTP POP3, SMTP IMAP, DNS
Warstwa prezentacji		
Warstwa sesji		
Warstwa transportowa	Warstwa transportowa	TCP, UDP
Warstwa sieci	Internet	IP, RIP OSPF, EIGRP /Router, Switch
Warstwa łącza danych	Warstwa dostępu do sieci	ETHERNET / Switch, Most, Hub
Warstwa fizyczna		Media transmisyjne (skrętka, światłowód), kodowanie

Warstwy sieci komputerowych – model OSI i TCP/IP – enkapsulacja

W modelu warstwowym komunikacja polega na przekazywaniu danych pomiędzy warstwami. Proces „przechodzenia" danych z warstwy wyższej do niższej nazwany jest **enkapsulacją**. Dodatkowo, w procesie tym dodawane są do pakietów informację niezbędne do właściwego przesyłu, i tak:

- w **warstwie transportowej** dane dzielone są na **segmenty** (jednostki **PDU**) i opatrywane nagłówkiem zawierającym **numer portu aplikacji**,
- w **warstwie sieciowej** tworzony jest **pakiet** poprzez dodawanie nagłówka zawierającego **adresy IP nadawcy i odbiorcy**,
- w **warstwie łącza danych** tworzona jest **ramka** poprzez dodanie nagłówka zawierającego **adresy fizyczne (MAC) nadawcy i odbiorcy**,
- w **warstwie fizycznej** następuje kodowanie ramki i przesłanie jej do odbiorcy.

Proces enkapsulacji widoczny jest poniżej.

1. Warstwa aplikacji				Dane	1. Warstwa aplikacji
2. Warstwa prezentacji				Dane	2. Warstwa prezentacji
3. Warstwa sesji				Dane	3. Warstwa sesji
4. Warstwa transportowa			Nagłówek segmentu	Dane	4. Warstwa transportowa
5. Warstwa sieci		Nagłówek sieciowy	Nagłówek segmentu	Dane	5. Warstwa sieci
6. Warstwa łącza danych	Nagłówek ramki	Nagłówek sieciowy	Nagłówek segmentu	Dane	6. Warstwa łącza danych
7. Warstwa fizyczna	00010101010100101010101010101				7. Warstwa fizyczna

Protokoły warstwy aplikacji

Warstwa aplikacji jest najwyższą warstwą stosowaną w modelach OSI oraz TCP/IP. Można powiedzieć, że jest to warstwa „najbliższa” użytkownikowi sieci ponieważ udostępnia ona interfejs komunikacyjny. Mówiąc jeszcze prościej, to własnie warstwa aplikacji pozwala użytkownikowi korzystać z sieci ponieważ pozwala na przeglądanie stron WWW, przysyłanie plików czy poczty elektronicznej.

Z wielu ogólnie znanych protokołów komunikacyjnych stosowanych w warstwie aplikacji poniżej przedstawiam te najważniejsze.

- **DNS** (ang. Domain Name System, **port: 53**) – protokół używany do odwzorowywania nazw w sieci Internet na adresy IP,
- **HTTP** (ang. Hypertext Transfer Protocol, **port: 80**) – protokół używany do przesyłania plików tworzących strony WWW,
- **HTTPS** (ang. Hypertext Transfer Protocol Secure, **port: 443**) – szyfrowana wersja protokołu HTTP, wykorzystująca szyfrowanie TLS,
- **SMTP** (ang. Simple Mail Transfer Protocol, **port: 25**) – protokół używany do przesyłania wiadomości poczty elektronicznej wraz z załącznikami,
- **POP3** (ang. Post Office Protocol, , **port: 110**) — protokół używany do odbioru poczty elektronicznej,
- **IMAP** (ang. Internet Message Access Protocol, **port: 143**) – protokół używany do odbioru poczty elektronicznej, oprócz funkcji POP3, pozwala również na zarządzanie zdalnymi folderami znajdującymi się na serwerze,
- **Telnet** (ang. Telecommunication Network, **port: 23**) – protokół używany do emulacji terminala umożliwiający komunikację ze zdalnym urządzeniem,
- **FTP** (ang. File Transfer Protocol, **porty: 20, 21**) – protokół używany do interaktywnego przesyłania plików pomiędzy systemami,
- **DHCP** (ang. Dynamic Host Configuration Protocol, **port: 67**) – protokół używany do dynamicznej konfiguracji urządzeń, a dokładniej odpowiedzialny jest za przydzielanie adresów IP, adresu domyślnej bramy i adresów serwerów DNS,
- **SSH** (ang. Secure Shell, **port: 22**) — protokół używany do emulacji terminala sieciowego zapewniający szyfrowanie połączenia,
- **NFS** (ang. Network File System, port: **2049**) — protokół używany do udostępniania systemów plików (dysków sieciowych),
- **SNMP** (ang. Simple Network Management Protocol, **port: 161**) —protokół zarządzania siecią, używany do konfiguracji urządzeń sieciowych i gromadzenie informacji na ich temat,
- **SMB** (ang. Server Message Block, **porty: 137, 138, 139**) protokół używany do do udostępniania plików.

Jak pewnie zauważyliście obok angielskiej nazwy protokołu pojawił się się również element jakim jest numer portu aplikacji. Numery portów stosowane są po to, aby warstwa transportowa potrafiła identyfikować usługi oraz aplikacje warstwy aplikacji, które są źródłem i celem danych. Podczas analizy różnych protokołów i usług warstwy aplikacji modelu OSI oraz TCP/IP często będziemy odnosić się do numerów portów, które są związane z konkretnymi usługami.

Protokoły warstwy transportowej

Warstwa transportowa pełni w procesie przesyłu danych bardzo ważne funkcje. Przede wszystkim odpowiedzialna jest za prawidłowy przebieg komunikacji oraz jej niezawodność. Określa również sposób segmentacji danych (zasady dzielenia ich na możliwe do zarządzania części) oraz późniejszego ich scalania, a także identyfikuje dane pochodzące z różnych aplikacji (poprzez numery portów). Dwa najpopularniejsze protokoły warstwy transportowej to **TCP** (ang. Transmission Control Protocol) i **UDP** (ang. User Datagram Protocol). Oba te protokoły są w stanie zarządzać wieloma równoczesnymi transmisjami. Różnią się zestawem funkcji jakie mogą dostarczyć aplikacjom.

Protokół UDP

UDP jest prostym, bezpołączeniowym protokołem. Jego najważniejszą zaletą jest niewielki narzut dodawany do dostarczanych danych. Porcje danych UDP są nazywane **datagramami**. Nie zapewnia on jednak mechanizmów ochronnych takich jak potwierdzenie dostarczenia datagramu czy retransmisji.

Aplikacje, które używają protokołu UDP to m.in.:

- system nazw domenowych **DNS** (ang. Domain Name System),
- aplikacje przesyłające strumienie Video,
- transmisja głosu przez sieć **IP** (VoIP).

Poniższy rysunek przedstawia jak wygląda typowy datagram UDP.

Bit (0) … Bit (15)	Bit (16) … Bit (31)
Port źródłowy (16)	Port docelowy (16)
Długość (16)	Suma kontrolna (16)
Dane warstwy aplikacji (rozmiar zmienny)	

8 bajtów

Protokół TCP

TCP jest protokołem połączeniowym, tworzącym tzw. **segmenty**, który wprowadza pewien dodatkowy narzut, ze względu na większą liczbę realizowanych funkcji, takich jak dostarczanie ich we właściwej kolejności czy niezawodne dostarczanie i kontrola przepływu. Każdy segment TCP dodaje aż 20 dodatkowych bajtów w nagłówku, gdzie datagram UDP dodaje tylko 8 dodatkowych bajtów. Główną zaletą tego protokołu jest niezawodność, którą gwarantuje konieczność wysłania potwierdzenia przez hosta odbierającego dane zanim dostanie przesłany kolejny segment.

Aplikacje wykorzystujące protokół TCP to:

- przeglądarki stron WWW,
- e-mail,
- programy do przesyłania plików.

Poniższy rysunek przedstawia jak wygląda typowy segment **TCP**.

Bit (0) … Bit (15)	Bit (16) … Bit (31)
Port źródłowy (16)	Port docelowy (16)
Numer sekwencyjny (32)	
Numer potwierdzenia (32)	
Długość nagłówka (4) Zarezerwowane (6) Bity kodu (6)	Okno (16)
Suma kontrolna (16)	Wskaźnik pilności (16)
Opcje (0 lub 32, jeśli istnieją)	
Dane warstwy aplikacji (rozmiar zmienny)	

20 bajtów

Porty aplikacji

Z poprzedniego artykułu wiecie już, że aplikacje stosują różną numerację portów, tak aby warstwa transportowa potrafiła je rozróżnić i we właściwy sposób pokierować przepływem danych. Istnieją 3 typy numerów portów aplikacji:

- **dobrze znane porty** (numery od 0 do 1023) – te numery są zarezerwowane dla usług i aplikacji. Są one powszechnie używane dla aplikacji takich jak serwery WWW (HTTP) czy serwery poczty elektronicznej (POP3/SMTP),
- **zarejestrowane porty** (numery od 1024 do 49151) – te numery są zarezerwowane dla aplikacji i procesów użytkownika. Są to przede wszystkim porty używane przez aplikacje i usługi tworzone na małą skalę,
- **dynamiczne lub prywatne numery portów** (od 49152 do 65535) – to numery portów, które są dynamicznie losowane przez aplikacje klienckie podczas inicjowania połączeń.

Protokoły warstwy sieci

Warstwa sieci (3 warstwa modelu OSI) udostępnia mechanizmy pozwalające na wymianę danych pomiędzy dwoma określonymi urządzeniami końcowymi sieci. W działaniu warstwy sieci można wyróżnić 4 elementy:

- **adresowanie**, określające nadawcę i odbiorcę danych,
- **enkapsulacja**, podczas której dodawane są adresy hosta źródłowego i docelowego,
- **ruting**, określający drogę przesyłania danych pomiędzy sieciami
- **dekapsulacja**, podczas której następuje potwierdzenie czy dane zostały przesłane do właściwego hosta.

Protokołami, które można wyróżnić w warstwie sieci to miedzy innymi:

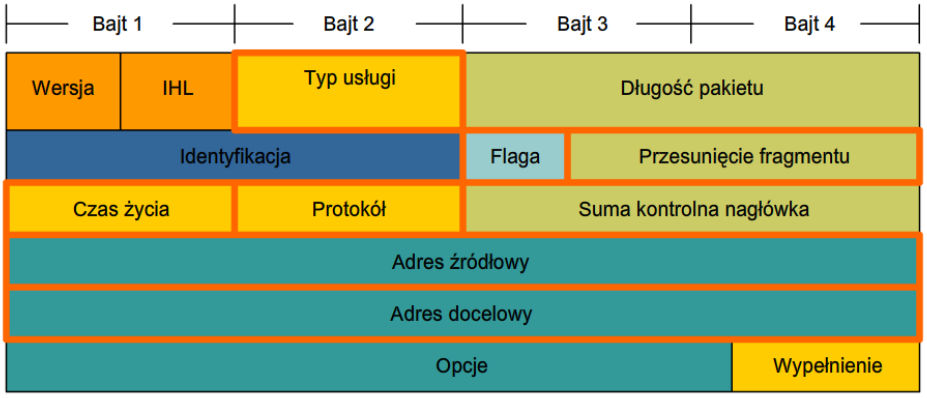
- **IPv4** (Internet Protocol version 4)
- **IPv6** (Internet Protocol version 6)
- **Novell IPX** (Internetwork Packet Exchange)
- **AppleTalk**
- **CLNS/DECNet**

Protokołem stosowanym najczęściej jest protokół **IPv4**. Został on zaprojektowany jako protokół z niedużym narzutem (dodatkowymi informacjami dodawanymi do pakietu). Zapewnia on tylko niezbędne funkcje umożliwiające dostarczanie pakietu ze źródła do celu przez połączone ze sobą sieci.

Ogólna charakterystyka protokołu IPv4:

- bezpołączeniowy – przed wysyłaniem danych nie jest zestawiane żadne połączenie,
- nie posiada mechanizmów zapewniających gwarancję dostarczenia pakietu,
- niezależny od używanego medium transmisyjnego.

Budowa nagłówka pakietu IPv4:



Docelowy adres IP – pole docelowego adresu IP, zapisane w postaci 32-bitowej.

Źródłowy adres IP – pole docelowego adresu IP, zapisane w postaci 32-bitowej.

Czas życia (TTL) – 8 bitowe pole, które określa pozostały czas życia pakietu. Wartość TTL jest zmniejszana o co najmniej 1 za każdym razem, gdy pakiet przechodzi przez router (tj. za każdym przeskokiem). Kiedy wartość osiąga 0, router porzuca pakiet i jest on usunięty ze strumienia danych w sieci. Mechanizm ten chroni pakiety, które nie mogą osiągnąć celu przed przekazywaniem w nieskończoność pomiędzy routerami w tzw. pętli routingu. Jeżeli pętle routingu byłyby dozwolone, sieć byłaby przeciążona pakietami danych, które nigdy nie osiągną swojego celu. Zmniejszanie wartości TTL przy każdym przeskoku zapewnia, że w końcu osiągnie wartość 0 i pakiet, którego pole TTL równe jest 0 będzie odrzucone.

Protokół – ta 8-bitowa wartość wskazuje typ danych, które pakiet przenosi. Pole protokołu umożliwia warstwie sieci przekazać dane do odpowiedniego protokołu warstwy wyższej. Przykładowymi wartościami są: 01 ICMP, 06 TCP, 17 UDP.

Typ usługi (ToS) – zawiera 8-bitową wartość binarną, która używana jest do określenia priorytetu każdego pakietu. Ta wartość umożliwia zastosowanie mechanizmów QoS w stosunku do pakietów o wysokim priorytecie, np. przenoszących dane głosowe.

Przesunięcie fragmentu – pole stosowane podczas rekonstrukcji podzielonego prze ruter pakietu. Wskazuje porządek w jakim ma być ustawiony każdy z pakietów podczas rekonstrukcji.

Flaga MF (ang. More Fragments) – pojedynczy bit używanym z polem przesunięcia fragmentu do podziału i rekonstrukcji pakietów. Gdy bit **flagi MF** jest ustawiony, oznacza to, że dany fragment nie jest ostatnim fragmentem pakietu. Kiedy host odbierający zauważy przybywający pakiet z ustawioną wartością MF=1, sprawdza pole przesunięcia fragmentu, gdzie należy umieścić ten fragment podczas rekonstrukcji pakietu. Kiedy host odbierający zauważy przybywający pakiet z ustawioną wartością MF=0 i niezerową wartość w polu przesunięcia fragmentu, umieszcza ten fragment jako ostatni kawałek rekonstruowanego pakietu.

Flaga DF (ang. Don’t Fragment) – pojedynczy bit, który jeśli jest ustawiony wskazuje, że fragmentowanie pakietu jest niedozwolone. Jeżeli **flaga DF** jest ustawiona, wtedy fragmentacja tego pakietu nie jest dozwolona.

Wersja -zawiera numer wersji protokołu IP (4).

Długość nagłówka (IHL) – określa rozmiar nagłówka pakietu.

Długość pakietu – to pole podaje w bajtach całkowitą wielkość pakietu, zawierającą nagłówek oraz dane.

Identyfikacja – to pole jest używane do jednoznacznego identyfikowania fragmentów podzielonego pakietu IP.

Suma kontrolna nagłówka – pole używane jest do sprawdzenia błędów nagłówka pakietu.

Opcje – jest to miejsce na dodatkowe pola w nagłówku IPv4 do obsługi innych usług. Jest ono jednak rzadko używane.

Protokoły warstwy sieci – protokoły rutingu

Bardzo ważnym elementem w procesie przesyłania danych pomiędzy urządzeniami w sieci jest ruting, który ogólnie rzecz ujmując służy do określania trasy przesyłania pakietów pomiędzy sieciami. Dlatego też omawiając protokoły warstwy sieci, nie można nie wspomnieć o protokołach rutingu, które również zwierają się w tej warstwie. Zanim jedenak o wybranych protokołach rutingu, kilka słów wstępu o samym procesie rutingu.

Hosty znajdujące się w tej samej sieci komputerowej wymieniają między sobą dane bez konieczności stosowania urządzenia pośredniczącego w warstwie sieci (rutera), wystarczy im urządzenie warstwy łącza danych, np. przełącznik. Co się jednak dzieje kiedy pakiet musi zostać dostarczony do urządzenia znajdującego się w innej sieci? Otóż wówczas do gry wkracza ruter, który stanowi bramę pomiędzy siecią, z której pakiet został wysłany, a innymi sieciami. Ruter, który znajduje się w naszej sieci i łączy ją z innymi sieciami określamy mianem **bramy domyślnej** (adres bramy konfiguruje się w ustawieniach sieci komputera). Brama posiada tzw. **tablicę rutingu**, która zawiera adresy ruterów, na drodze pakietu. Jeśli pakiet wysyłany jest do sieci oddalonej o wiele kilometrów, wówczas, zanim dotrze do celu, przesyłany jest przez wiele ruterów. Tablica rutingu przechowuje informacje o przyłączonych i odległych sieciach. Trasy do tych sieci mogą być zaprogramowane przez administratora (tzn. **ruting statyczny**) lub też mogą być aktualizowane automatycznie poprzez protokoły rutingu (tzn. **ruting dynamiczny**). Ruting statyczny jest metodą bardziej żmudną ponieważ wymaga każdorazowej aktualizacji tablic kiedy tylko zmieni się topologia sieci. Ruting dynamiczny ma tą przewagę, że samodzielnie potrafi dokonać aktualizacji w tablicy rutingu ruterów.

Powszechnie stosowane protokoły rutingu to:

- protokół **RIP** (ang. Routing Information Protocol) – protokół działający na podstawie wektora odległości, określający najkrótszą ścieżkę przesyłu pakietu,
- protokół **EIGRP** (ang. Enhanced Interior Gateway Routing Protocol), protokół, działający zarówno na podstawie wektora odległości, ale również na podstawie stanu łącza,
- protokół **OSPF** (ang. Open Shortest Path First) – protokół, działający zarówno na podstawie wektora odległości, ale również na podstawie stanu łącza, stosujący tzw. obszary przy obliczaniu trasy.

Adresacja IP

Adresacja stanowi niezwykle ważny element w działaniu sieci komputerowych ponieważ na jej podstawie odbywa się dostarczanie informacji. Kiedy listonosz dostarcza nam list, dokonuje tego na podstawie naszego adresu zamieszania, w sieciach komputerowych natomiast, aby dostarczyć pakiety do odpowiedniego hosta potrzebny jest inny adres, nazywamy **adresem IP**. Stanowi on kluczowy element w funkcjonowaniu protokołów warstwy sieciowej. W tym artykule omówimy sobie, w jaki sposób działa **adresowanie IP w wersji 4**.

Adres IP jest adresem logicznym interfejsu sieciowego (hosta) to znaczy, że nie musi to być tylko adres konkretnego komputera, może to być również adres interfejsu (interfejsów) rutera czy access pointa.

Każdy adres IPv4 składa się z **32-bitowego** ciągu zer i jedynek. Wynika to z faktu, że działanie urządzeń w sieciach komputerowych oparte jest na logice cyfrowej, co oznacza, że adresy te interpretowane są jako **liczby binarne**. Dla prostszego stosowania adresacji i zapamiętywania adresów, na co dzień stosuje się jednak zapis dziesiętny. Zapis ten składa się z **4 części** (każda z nich to **1 bajt** czyli **8 bitów**), zwanych **oktetami**.

<u>Dla lepszego zrozumienia posłużę się przykładem:</u>

adres IP w zapisie dziesiętnym **192.112.20.101** będzie miał postać binarną
równą **11000000.01110000.00010100.01100101**

W każdym adresie IP, pewna część bitów (liczona od lewej strony), reprezentuje adres sieci, reszta natomiast stanowi adres konkretnego hosta. Jest to logiczne ponieważ pakiet najpierw musi trafić do właściwej sieci, dopiero potem trafia do konkretnego hosta. Jak w życiu, list najpierw trafia do miasta, dopiero potem pod wskazany numer domu na konkretnej ulicy. Jaka część bitów przeznaczona jest na adres sieci, a jaka na adres hosta określone jest przez tzw. **maskę podsieci**, która to określaja wielkość sieci oraz liczbę hostów w niej funkcjonujących.

Zanim przejdziemy do omówienia maski, musimy jeszcze określić typy adresów, które związane są z każdą siecią, są nimi:

- **adres sieciowy** (ang. network address) -adres, który określa całą sieć,
- **adres rozgłoszeniowy** (ang. broadcast address) – specjalny adres używany w celu wysyłania danych do wszystkich hostów w określonej sieci,
- **adres hosta** (interfejsu urządzenia końcowego) – adres przyporządkowany urządzeniu końcowemu pracującemu w sieci.

Podział ten będzie nam potrzebny aby we właściwy sposób móc obliczać adresy IP.

Wróćmy teraz do naszej maski podsieci, która to określa, jaka część adresu IP określa sieć, a jaka hosty. Podobnie jak sam adres IP, również maska zapisana jest w postaci 32-bitowej i przez urządzenia sieciowe interpretowana jest jako liczba binarna, natomiast w zapisie oraz konfiguracji urządzeń stosuje się zapis dziesiętny, podzielony na 4 oktety. Przykładowa maska podsieci może mieć postać **255.255.255.0** dziesiętne czyli **11111111111111111111111100000000** binarnie. Czasami też możemy spotkać się z tzw. **skróconym zapisem maski** (np. /24), określającym, ile jest jedynek w binarnym zapisie maski. My będziemy stosować oba te zapisy.

W jaki sposób interpretować zapis binarny, a co za tym idzie jak określić, która część adresu jest adresem sieci, a która hosta? Bardzo prosto: <u>jedynki określają sieć, a zera hosty należące do tej sieci.</u>

Teraz omówimy sobie, w jaki sposób należy obliczać adresy IP. Jest to przydatne w momencie kiedy będziemy chcieli na podstawie podanego adresu i maski określić **adres sieci**, do której należy host, **maksymalną** liczbę hostów należących do tej sieci oraz **adres rozgłoszeniowy.** Aby wykonać tego typu zadanie potrzebna będzie umiejętność konwersji liczb dziesiętnych na binarne (i odwrotnie) oraz wiedza z zakresu wykonywania operacji logicznych, a konkretnie operacji **AND** i **NOT**. Zaczynamy!

Podany jest adres ip w postaci: **192.168.1.145** oraz maska **255.255.255.128**. Na podstawie tych danych należy obliczyć:

- **adres sieci,**
- **adres rozgłoszeniowy,**
- **maksymalną liczbę hostów,**

oraz wskazać **adres pierwszego i ostatniego hosta** w sieci.

<u>Rozwiązanie:</u>

1. Obliczamy adres sieci

Zamieniamy **adres IP** oraz **maskę** na postać binarną

<u>192.168.1.145</u>: **11000000101010000000000110010001**
<u>255.255.255.128</u>: **11111111111111111111111110000000**

Na otrzymanych liczbach binarnych wykonujemy operację **AND** (czyli mnożymy liczby w kolumnach)

1 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0

1 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0

Otrzymaną postać binarną konwertujemy na liczbę dziesiętną

1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
192								168								1								128							

192.168.1.128 – to jest adres naszej sieci

2. Obliczamy adres rozgłoszeniowy

Na postaci binarnej **maski** wykonujemy operację logiczną **NOT** (jedynki zamieniamy na zera, a zera na jedynki)

	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
NOT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Zamieniamy otrzymaną liczbę binarną na postać dziesiętną

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0								0								0								127							

Otrzymaną liczbę dziesiętną dodajemy do **adresu sieci**

	0	0	0	127
+	192	168	1	128
	192	168	1	255

192.168.1.255 – to jest adres rozgłoszeniowy

3. Obliczamy maksymalną liczbę hostów w sieci

Obliczając maksymalną liczbę hostów w sieci korzystamy ze wzoru

$$2^{(\text{liczba bitów adresu IP – skrócony zapis maski})} - 2$$

przypominam, że skrócony zapis maski to liczba jedynek w jej postaci binarnej

Co w naszym przypadku daje

$$2^{(32-25)} - 2 = 2^{(7)} - 2 = 128 - 2 = 126 \text{ hostów}$$

Na koniec wskazujemy adres pierwszego i ostatniego hosta w sieci

Jeśli adres sieci ma postać **192.168.1.128** to adres pierwszego hosta będzie miał postać

192.168.1.129

Jeśli natomiast adres rozgłoszeniowy ma postać **192.168.1.255** to adres ostatniego hosta będzie miał postać

192.168.1.254

Podsumowując:

Adres sieci: **192.168.1.128**.

Adres rozgłoszeniowy: **192.168.1.255**.

Liczba hostów: **126**.

Pierwszy host: **192.168.1.129**.

Ostatni host: **192.168.1.254**.

Powyższy sposób przeliczania sprawdzi przy adresie z każdej klasy adresowej. Zwróćcie uwagę na to, że wszelkie operacje (AND, NOT, dodawanie czy odejmowanie) wykonywaliśmy w kolumnach, oktet pod oktetem. Jeśli obliczalibyśmy adres z innej klasy, np 153.15.102.120 /18 i przykładowo, wykonując operację NOT na postaci binarnej maski i konwertując ją na liczbę dziesiętną otrzymalibyśmy postać **0.0.63.255**, to aby uzyskać adres rozgłoszeniowy **63** dodajemy do **trzeciego oktetu**, natomiast **255** do **czwartego oktetu adresu sieci**. Stosując te zasady, będziecie w stanie przeliczać każdy adres IP.

Zadanie do samodzielnego wykonania

Dany jest adres IP w postaci: **172.16.34.200** oraz maska **255.255.192.0**. Na podstawie tych danych oblicz:

- adres sieci,
- adres rozgłoszeniowy,
- liczbę hostów,
- adres pierwszego i ostatniego hosta.

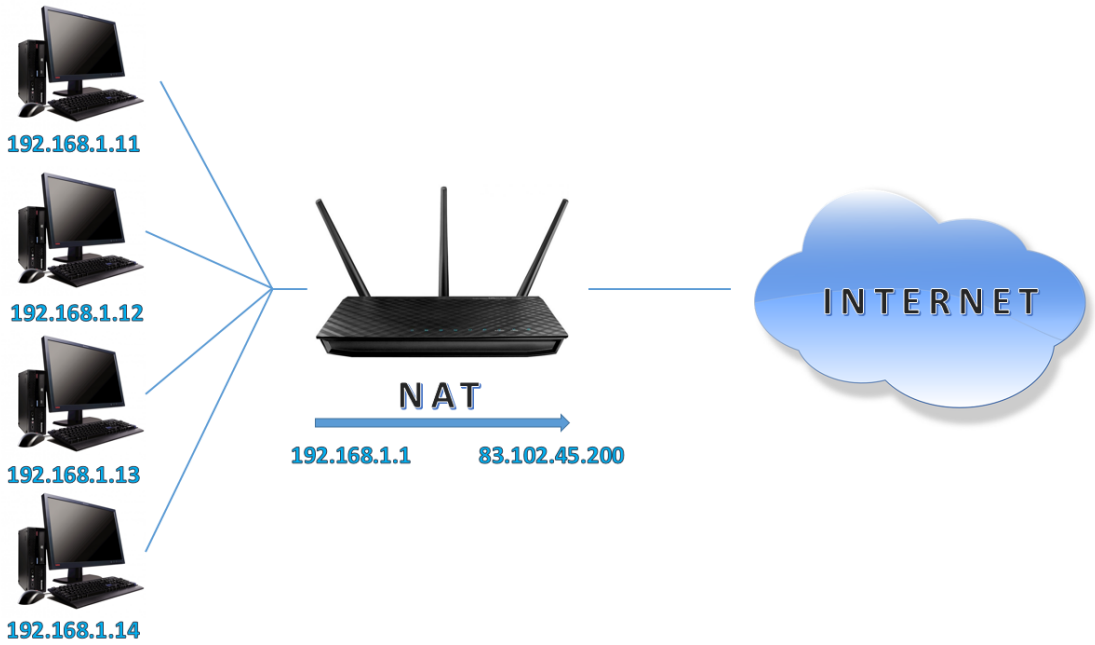
Adresy publiczne i prywatne

Omawiając adresację IPv4 nie można pominąć bardzo ważnego jej elementu jaki jest podział adresów IP na **publiczne** oraz **prywatne**. Adresy publiczne są wykorzystywane w celu umożliwienia hostom dostępu do Internetu. Adresy te są unikatowe, to znaczy, że tylko jeden host w całej globalnej sieci może mieć przypisany dany adres. W sieciach lokalnych (LAN) stosuje się natomiast adresy prywatne, które również są unikatowe, ale tylko w obrębie tej sieci. Oznacza to, że w ramach jednej sieci LAN, każdy komputer ma inny adres IP, ale w już w innej sieci LAN adres taki może się powtórzyć.

Dla lepszego zrozumienia tego stanu rzeczy posłużę się przykładem: komputery w Twojej szkolnej sieci mogą mieć przykładowe adresy **192.168.0.x**, jeśli w swoim domu również masz sieć lokalną (ruter + komputer + laptop) to istnieje spore prawdopodobieństwo, że urządzenia w Twojej sieci również mają adresy z zakresu **192.168.0.x**. Nawet jeśli tak nie jest, to uwierz mi, że na świecie są miliony małych sieci lokalnych o takiej samej adresacji.

Jeśli urządzenia w Twojej sieci szkolnej lub domowej posiadają dostęp do Internetu, a zapewne tak jest to muszą korzystać również z adresu (lub adresów) publicznych. Adres ten przydzielony jest przez tzw. **providera** czyli firmę, która udostępnia połączenie internetowe. W większości przypadków provider udostępnia tylko jeden adres publiczny.

Może tutaj pojawić się pytanie dlaczego w takim razie wszystkie urządzenia w sieci (a może ich być przecież bardzo dużo) mogą korzystać z Internetu? Dlatego, że istnieje usługa, zwana **NAT (**ang. Network Address Translation), która „tłumaczy” adresy prywatne na publiczne. Jest ona skonfigurowana na ruterze lub szkolnym serwerze. Jej zadaniem jest, mówiąc skrótowo umożliwienie urządzeniom w sieci LAN na korzystanie z sieci globalnej. Zobrazowałem Wam działanie usługi NAT na poniższej grafice.



Jeśli zatem korzystacie z usług sieci Internet za pośrednictwem komputera w sieci lokalnej, to dla użytkowników i usług w Internecie jesteście „widoczni" pod adresem publicznym, a każdy komputer znajdujący się w sieci lokalnej, będzie komunikował się z Internetem poprzez ten sam publiczny adres (o ile oczywiście provider przydzielił tylko jeden publiczny adres).

Liczba prywatnych adresów IP jest określona i mieści się w następujących przedziałach:

- od **10.0.0.0** do **10.255.255.255** (10.0.0.0 /8),
- od **172.16.0.0** do **172.31.255.25**5 (172.16.0.0 /12),
- od **192.168.0.0** do **192.168.255.255** (192.168.0.0 /16).

Klasy adresów

Obecnie stosowanie adresacji IPv4 opiera się na określaniu jaka ilość hostów będzie działać w danej sieci, co nazwane zostało **adresowanie bezklasowym**. Natomiast jeszcze kilka, kilkanaście lat temu stosowano podział adresów, który dzielił je na trzy główne grupy, co oznaczane było mianem **adresacji klasowej**.

Klasa A

Blok adresów w **klasie A** został określony do tworzenia bardzo dużych sieci, w których można zaadresować ponad **16 milinów hostów**. Klasa ta domyślnie posiada maskę **255.0.0.0**, co oznacza, że tylko jeden (pierwszy) oktet adresu IP zawiera adres sieci, pozostałe określają adres hosta.

Klasa B

Blok adresów w **klasie B** został określony do tworzenia dużych i średnich sieci, w których można zaadresować ponad **65 tysięcy hostów**. Klasa ta domyślnie posiada maskę **255.255.0.0**, co oznacza, że pierwsze dwa oktety adresu IP zawierają adres sieci, pozostałe dwa określają adres hosta.

Klasa C

Blok adresów w **klasie C** został określony w celu tworzenia i obsługi małych sieci skupiających maksymalnie **254 hosty**. Klasa ta domyślnie posiada maskę **255.255.255.0**, co oznacza, że trzy pierwsze oktety adresu IP zawierają adres sieci, ostatni określa adres hosta.

Istnieją jeszcze klasy **D** i **E**, są tzw. adresy grupowe (klasa D) i eksperymentalne (klasa E) i nie są wykorzystywane do adresowania hostów.

Poniżej przedstawiam Wam tabelę z klasami adresów, która przedstawia ich zakresy oraz liczbę możliwych do zaadresowania sieci i hostów.

Klasa adresu	Zakres pierwszego oktetu	Bity pierwszego oktetu	Części adresu: sieci (S) i hosta (H)	Domyślna maska	Możliwa liczba sieci	Możliwa liczba hostów
A	1 – 127	00000000 – 01111111	S.H.H.H	255.0.0.0	128 (2^7)	16 777 214 ($2^{24} – 2$)
B	128 – 191	100000000 – 10111111	S.S.H.H	255.255.0.0	16 384 (2^{14})	65 384 ($2^{14} – 2$)
C	192 – 223	11000000 – 11101111	S.S.S.H	255.255.255.0	2 097 150 (2^{21})	254 ($2^8 – 2$)

Adresacja IP – podział podsieci

W życiu każdego projektanta oraz administratora sieci następuje czas, w którym z dostępnego bloku adresowego musi utworzyć klika logicznych sieci (podsieci). Czas, który poświęcicie na zapoznanie się z tym artykułem oraz jego analizę, pozwoli Wam w przyszłości zaoszczędzić sporo stresu podczas wykonywania tej czynności.

Sięgnijmy pamięcią do poprzedniego artykułu, w którym omówiliśmy sobie czy jest adres IPv4 oraz maska. Informacje te będą nam potrzebne, aby zrozumieć proces podziału sieci na mniejsze podsieci, co w przyszłości pozwoli we właściwy sposób wykonywać taki podział. Maska jest elementem konfiguracji, który określa, która część adresu IP jest **adresem sieci**, a która **adresem hosta**. Weźmy dla przykładu adres **192.168.1.0** z maską **255.255.255.0**.

W tym przypadku na adres sieci przypadają **3 pierwsze oktety** adres IP, wynika to z faktu, iż w postaci binarnej maski (**11111111.1111111.1111111.00000000**), na **3 pierwszych oktetach** mamy **1**, a przypomnę, że **1** odpowiada części sieciowej, a **0** części hosta. Jeśli chcielibyśmy taki blok podzielić na **2 logiczne podsieci**, to musimy z **części hosta „pożyczyć" 1 bit** do **części sieciowej**. Wówczas nasza maska, dla każdej z tych podsieci, będzie binarnie miała postać **11111111.11111111.11111111.10000000**, czyli dziesiętnie **255.255.255.128**. Analogicznie postąpimy, kiedy będziemy chcieli wykonać podział na **4 logiczne podsieci**, wówczas „pożyczymy" **2 bity** z części hosta i otrzymamy maskę **11111111.11111111.11111111.11000000** binarnie czyli **255.255.255.192** dziesiętnie.

Z pewnością u wielu z Was pojawi się pytanie dlaczego w drugim przypadku „pożyczenie" kolejnej jedynki spowodowało podział od razu na **4** a nie na **3 podsieci**? Wynika to z logiki binarnej, a dokładniej z potęgi liczby 2, gdzie nie można uzyskać liczby nieparzystej. Ponadto przy podziale na podsieci stosuje się wzór:

liczba podsieci = $\mathbf{2^n}$

gdzie **n** = liczba „pożyczonych" bitów z części hosta

Stąd też jeśli „pożyczylibyśmy" dwie **1** z części hosta nasz wzór miałby postać $\mathbf{2^2}$ co daje nam **4** podsieci

Poniższa tabela pozwoli Wam łatwiej zrozumieć, jak dodawanie jedynki binarnej do maski przekłada się na liczbę możliwych do utworzenia podsieci.

Maska	Liczba **jedynek (1)** w postaci binarnej maski	Liczba podsieci	Liczba hostów w każdej z podsieci
255.255.255.0	**/24**	**1**	**254**
255.255.255.128	**/25**	**2**	**126**
255.255.255.192	**/26**	**4**	**62**
255.255.255.224	**/27**	**8**	**30**
255.255.255.240	**/28**	**16**	**14**
255.255.255.248	**/29**	**32**	**6**
255.255.255.252	**/30**	**64**	**2**

Uprzedzając następne pytanie, które brzmi „czy zatem nie można dokonywać podziału adresu sieci na 3 lub 5 podsieci?” odpowiadam można. Należy wówczas jednak zastosować technikę zmiennej długości maski, tzw. **VLSM** (ang. Variable Length Subnet Mask), o której opowiem w dalszej części artykułu.

Wróćmy do naszego pierwszego przykładu, czyli adresu **192.168.1.0**, maski **255.255.255.0** oraz podziału na **2 podsieci**. Jeśli wykorzystaliśmy część bitów hosta, aby „zaspokoić” część sieciową, to jest to równoznaczne z tym, że liczba hostów w tych dwóch podsieciach się zmniejszy. Jak zatem obliczyć liczbę hostów w nowo utworzonych podsieciach? Należy skorzystać z tego samego wzoru, który stosowaliśmy w poprzednim artykule czyli: $\mathbf{2^{(\text{liczba bitów adresu IP – skrócony zapis maski})} – 2}$, co w naszym przypadku da nam:

$$\mathbf{2^{(32-25)} – 2 = 2^{(7)} – 2 = 128 – 2 = 126}$$

Tak więc w każdej z podsieci będziemy mieli **126 hostów.**

Na koniec pozostaje nam jeszcze określić adresy nowo utworzonych podsieci, oraz ich adres rozgłoszeniowe. Pamiętajmy, że „wyjściowo” skorzystaliśmy z adresu **192.168.1.0** oraz maski **255.255.255.0** i co za tym idzie **3 pierwszych oktetów** tego adresu zmienić nie możemy. Poruszać się będziemy zatem tylko w ramach **ostatniego oktetu**.

Jeśli nasz początkowy adres miał postać **192.168.1.0** to nie ma żadnego powodu, aby nie wykorzystać go jako adres naszej **pierwszej podsieci**, zatem będzie ona miała adres **192.168.1.0**, maska dla tej podsieci to: **255.255.255.128**. Adres rozgłoszeniowy (ang. broadcast) wyliczamy w taki sam sposób, w jaki robiliśmy to w poprzednim artykule, czyli wykonujemy **operacją logiczną NOT** na postaci binarnej maski, następnie zamieniamy otrzymaną liczbę na postać dziesiętną i dodajemy ją do adresu sieci.

Operacja NOT

	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
NOT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Konwersja otrzymanej liczby na postać dziesiętną

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
	0							0							0							127									

Dodanie otrzymanej liczby do adresu sieci

$$\begin{array}{r} 192 \,.\, 168 \,.\, 1 \,.\, 0 \\ +\quad 0 \,.\, 0 \,.\, 0 \,.\, 127 \\ \hline 192 \,.\, 168 \,.\, 1 \,.\, 127 \end{array}$$

I w taki oto sposób wyliczyliśmy adres rozgłoszeniowy naszej pierwszej podsieci.

Jak w takim razie określić adres drugiej podsieci? Zasada jest bardzo prosta i obowiązywać będzie zawsze przy podziale na podsieci więc warto ją zapamiętać: adres każdej kolejnej podsieci to adres rozgłoszeniowy poprzedniej plus 1, czyli w naszym przypadku **192.168.1.128**. Maska jest taka sama, tak więc łatwo można obliczyć adres rozgłoszeniowy tej podsieci, wystarczy, że dodamy do ostatniego oktetu adresu podsieci wyliczoną wcześniej liczbę **127**, co da nam **192.168.1.255**. Adresy pierwszych i ostatnich hostów również obliczamy, tak jako było to omówione w poprzednim artykule, czyli adres pierwszego hosta każdej podsieci to adres tej podsieci plus 1, adres każdego ostatniego hosta podsieci to adres rozgłoszeniowy minus 1.

Podsumowując, podział adresu **192.168.1.0** z maską **255.255.255.0** przedstawia się następująco:

	Podsieć 1	Podsieć 2
Adres podsieci	*192.168.1.0*	*192.168.1.128*
Maska podsieci	*255.255.255.128*	*255.255.255.128*
Adres rozgłoszeniowy	*192.168.1.127*	*192.168.1.255*
Liczba hostów	*126*	*126*
Adres 1 hosta	*192.168.1.1*	*192.168.1.129*
Adres ostatniego hosta	*192.168.126*	*192.168.1.254*

W celu przećwiczenia zagadnienia, proponuję abyście samodzielnie spróbowali podzielić sieć z większą ilością możliwych do zaadresowania hostów o adresie **172.16.0.0** z maską **255.255.0.0** na **4 podsieci**. Zauważcie, że w tym przykładzie maska jest mniejsza niż we wcześniej omawianym, dlatego też, zmiana adresów będzie następować na dwóch ostatnich oktetach, a nie tylko na ostatnim. Dla ułatwienia udostępniam Wam tabelę z rozwiązaniem, abyście mogli sprawdzić poprawność wykonania zadania. Powodzenia!

	Podsieć 1	Podsieć 2	Podsieć 3	Podsieć 4
Adres podsieci	*172.16.0.0*	*172.16.64.0*	*172.16.128.0*	*172.16.192.0*
Maska podsieci	*255.255.192.0*	*255.255.192.0*	*255.255.192.0*	*255.255.192.0*
Adres rozgłoszeniowy	*172.16.63.255*	*172.16.127.255*	*172.16.191.255*	*172.16.255.255*
Liczba hostów	*16382*	*16382*	*16382*	*16382*
Adres 1 hosta	*172.16.0.1*	*172.16.64.1*	*172.16.128.1*	*172.16.192.1*
Adres ostatniego hosta	*172.16.63.254*	*172.16.127.254*	*172.16.191.254*	*172.16.255.254*

Wróćmy teraz do wspomnianej przeze mnie wcześniej techniki **VLSM**. Polega ona na zastosowaniu różnej długości maski w zależności od wielkości podsieci. Dwa przypadki, które przećwiczyliśmy wcześniej, dzieliły adres IP na równe podsieci, czyli na podsieci z taką samą ilością hostów (podsieci miały taką samą maskę). Stosowanie stałych masek nie zawsze ma jednak sens, ponieważ czasami zdarzają się sytuacje, gdzie w podsieciach pracować będzie różna liczba hostów, np. w jednej podsieci 100, a w drugiej tylko 10. Jeśli podzielimy adres IP na dwie równe podsieci, to w tej drugiej wiele adresów będzie niewykorzystanych, zostaną zmarnowane, co jest sytuacją niepożądaną. Aby uniknąć takiej sytuacji przy podziale na podsieci, uwzględnić należy „zapotrzebowanie" na liczbę hostów i tak ustawić maskę, aby odpowiadała temu „zapotrzebowaniu". Podsumowując VLSM stosuje się wówczas, kiedy chcemy podzielić adres IP na mniejsze podsieci, ale z różną ilość hostów możliwych do zaadresowania.

Aby przybliżyć Wam stosowanie tej techniki ponownie posłużę się przykładem:

Mamy do dyspozycji adres **192.168.1.0** z maską **255.255.255.0** i musimy dokonać podziału na **3 podsieci** o określonej liczbie hostów:

1 podsieć: minimum **100 hostów**,

2 i 3 podsieć: minimum po **50 hostów**.

Aby wykonać to zadanie, na początek trzeba wskazać jaką zastosować maskę, aby możliwe było zaadresowanie liczby hostów podanej w przykładzie. Tak więc jeśli dodamy **jedynkę** do maski i będzie ona miała postać **255.255.255.128 (/25)** będziemy mogli zaadresować do **126 hostów** (wynika to z podanego wcześniej wzoru $\mathbf{2^{(\text{liczba bitów adresu IP – skrócony zapis maski})} - 2}$).

Jeśli dodamy **2 jedynki** to maska będzie miała postać **255.255.255192 (/26)**, czyli będziemy mogli zaadresować do **62 hostów**. Jak widać, zarówno w pierwszym, jak i w drugim przypadku mieścimy się w podanych wcześniej założeniach. Podsumowując zatem, maski dla poszczególnych podsieci przedstawiają się następująco:

1 podsieć: **255.255.255.128**

2 i 3 podsieć: **255.255.192**

Jeśli mamy już określone maski, możemy przystąpić do wyliczania pozostałych danych, czyli adresów sieci, adresów rozgłoszeniowych oraz adresów pierwszych i ostatnich hostów w podsieciach. Zasady są dokładnie takie same jak w poprzednich przypadkach.

Przypominam, że jeśli mamy do podzielenia adres IP **192.168.1.0** to adres ten użyty będzie jako adres **1 podsieci**. Jeśli maska ma postać **255.255.255.128** to po zamianie na postać binarną (11111111.11111111.11111111.10000000), wykonaniu negacji (operacji NOT) i po konwersji na liczbę dziesiętną uzyskamy postać **0.0.0.127**. Dodając tę wartość do adresu podsieci otrzymamy adres rozgłoszeniowy **192.168.1.127**. Adres pierwszego hosta to **192.168.1.1** (adres sieci + 1), natomiast ostatniego to **192.168.1.126** (adres rozgłoszeniowy – 1)

Adres drugiej podsieci będzie miał postać **192.168.1.128** (+ 1 do adresu rozgłoszeniowego poprzedniej podsieci). Po zamianie na postać binarną maski podsieci (11111111.11111111.11111111.11000000), wykonaniu operacji NOT i ponownej konwersji na liczbę dziesiętną otrzymamy postać **0.0.0.63**. Po dodaniu tej wartości do adresu sieci, otrzymamy adres rozgłoszeniowy w tej podsieci w postaci **192.168.1.191**. Adres hosta pierwszego to **192.168.129**, a ostatniego **192.168.1.190**

Adres trzeciej podsieci będzie miał postać **192.168.1.192**, maska jest taka sama, jak w poprzedniej podsieci tak więc ponownie wartość **0.0.0.63** dodajemy do adresu podsieci aby otrzymać adres rozgłoszeniowy, który ostatecznie będzie miał postać **192.168.1.255** Adres pierwszego hosta to **192.168.1.193**, natomiast ostatniego **192.168.1.254.**

Podsumowanie przedstawiam Wam w poniższej tabeli.

	Podsieć 1	Podsieć 2	Podsieć 3
Adres podsieci	*192.168.1.0*	*192.168.1.128*	*192.168.1.192*
Maska podsieci	*255.255.255.128*	*255.255.255.192*	*255.255.255.192*
Adres rozgłoszeniowy	*192.168.1.127*	*192.168.1.191*	*192.168.1.255*
Liczba hostów	*126*	*62*	*62*
Adres 1 hosta	*192.168.1.1*	*192.168.1.129*	*192.168.1.193*
Adres ostatniego hosta	*192.168.1.126*	*192.168.1.190*	*192.168.1.254*

FILM ADRESOWANIE IP: **https://youtu.be/t3IceGlTjig**