PROTOKOŁY NOWSZEJ GENERACJI - TCP/IP

Autorka: Misiak Anna, IVFDS

STRESZCZENIE

Tematem pracy jest zapoznanie czytelnika z zestawem protokołów intersieci TCP/IP ze szczególnym naciskiem na dwa: IP i TCP. Zestaw protokołów TCP/IP jest pierwszym zestawem, jaki opracowano na potrzeby intersieci. Prace nad protokołami TCP/IP rozpoczęto w latach siedemdziesiątych mniej więcej w tym samym czasie, kiedy powstało pojecie sieci lokalnych. Badania te były w głównej mierze finansowane przez armię USA za pośrednictwem Advanced Research Project Agency (ARPA). To właśnie armia była jedną z pierwszych organizacji, które potrzebowały połączyć wiele sieci lokalnych, a co się z tym wiąże zapewnienie jednolitych usług.
Pierwsza część projektu prezentuje ogólny przegląd protokołów, model warstwowy TCP/IP oraz jego porównanie z modelem odniesienia ISO/OSI. Druga część reprezentuje opis protokołu IP jego sukces i rokowania na przyszłość. Ostatnia część to usługa niezawodnego przesyłania – TCP, jego cechy, sterowanie i retransmisja.

SPIS TREŚCI

1. OGÓLNY PRZEGLĄD PROTOKOŁÓW

TCP/IP jest zestawem protokołów dla pakietowej sieci rozległej WAN, o nazwie pochodzącej od dwóch protokołów składowych: TCP – *Transmission Control Protocol* oraz IP – *Internet Protocol*. Omawiana rodzina protokołów nazywa się TCP/IP, ale oprócz protokołów TCP oraz IP należą do niej również inne protokoły, które pobieżnie zostaną omówione poniżej:

IPv4 Wersja 4 protokołu IP, którą często nazywamy po prostu IP, od wczesnych lat osiemdziesiątych był najczęściej stosowanym protokołem z rodziny protokołów Internetu. Używa się w nim 32 – bitowych adresów. Protokół ten obsługuje dostarczanie pakietów danych dla protokołów: TCP, UDP, ICMP i IGMP.

IPv6 Wersję 6 protokołu IP utworzono w połowie lat dziewięćdziesiątych, aby zastąpić nią wersję 4 protokołu IP. Główna zmiana polegała na powiększeniu adresów do 128 bitów i była odpowiedzią na gwałtowny rozwój Internetu przypadający na lata dziewięćdziesiąte. Protokół ten obsługuje dostarczanie pakietów danych dla protokołów: TCP, UDP i ICMPv6.

Gdy rozróżnienie między obu wersjami protokołu IP nie ma znaczenia, wówczas używa się często po prostu nazwy IP.

TCP Protokół sterowania transmisją jest protokołem obsługi połączeniowej procesu użytkownika, umożliwiającym niezawodne i w pełni dwukierunkowe przesyłanie strumienia bajtów. Do zadań tego protokołu należy potwierdzenie, uwzględnienie czasu oczekiwania, dokonywanie retransmisji itp. Protokół TCP może korzystać z IPv4 albo IPv6.

UDP (ang. *User Datagram Protocol*) Jest protokołem obsługi bezpołączeniowej procesów użytkownika. W odróżnieniu od protokołu TCP, który jest niezawodny, protokół UDP nie daje gwarancji, ze datagramy UDP zawsze dotrą do wyznaczonego celu. Tak jak protokół TCP, również on moze korzystać z IPv4 albo IPv6.

ICMP (ang. *Internet Control Message Protocol*) Obsługuje on komunikaty o błędach i informacje sterujące przesyłane miedzy ruterami a stacjami. Te komunikaty są zazwyczaj generowane i przetwarzane przez oprogramowanie sieciowe protokołów TCP/IP, nie zaś przez procesy użytkownika. Istnieje też oprogramowanie użytkowe, które używa protokołu ICMP (np. Program `ping`).

IGMP (ang. *Internet Group Management Protocol*) Protokół ten obsługuje rozsyłanie grupowe (ang. *multicasting*), które jest dodatkową możliwością w razie korzystania z protokołu IPv4.

ARP (ang. *Address Resolution Protocol*) Protokół ten służy do odwzorowywania adresów IPv4 na adresy sprzętowe (tj. adresy Ethernetu). Ten protokół jest zazwyczaj używany w sieciach, w których stosuje się rozgłaszanie, tj. Ethernet, Token Ring i FDDI, lecz nie jest potrzebny w sieciach o połączeniach dwupunktowych.

RARP (ang. *Reverse Address Resolution Protocol*) Służy on do odwzorowywania adresów sprzętowych na adresy IPv4. Używa się go czasami w razie uruchamiania węzła sieci, w którym nie ma napędu dysków [6].

2. MODEL WARSTWOWY TCP/IP

Tabela 1. Warstwy modelu TCP/IP

WARSTWA APLIKACJI
WARSTWA TRANSPORTOWA TCP/UDP
WARSTWA MIĘDZYSIECIOWA IP/ICMP
WARSTWA DOSTĘPU DO SIECI
WARSTWA FIZYCZNA

Warstwa1: **Fizyczna -** odpowiada bazowemu sprzętowi sieciowemu podobnie jak warstwa 1 modelu ISO. Zestaw protokołów TCP/IP może być, zatem implementowany w środowisku sieci lokalnych Token Ring oraz Ethernet.

Warstwa2: **Dostępu do sieci** – protokoły tej warstwy określają podział danych na ramki i zasady przesyłania ramek przez sieć, podobnie jak warstwa 2 modelu OSI.

Warstwa3: **Międzysieciowa –** protokoły tej warstwy określają format pakietów przesyłanych w intersieci oraz metody przekazywania pakietów od nadawcy za pośrednictwem rutenów do odbiorcy. Zawiera protokół IP oraz protokoły skojarzone. Warstwa ta realizuje funkcje doboru trasy dla pakietów na podstawie czterobajtowego adresu IP identyfikującego źródło informacji oraz ich przeznaczenie.

Warstwa4: **Transportowa** – protokoły tej warstwy, podobnie jak protokoły warstwy 4 modelu ISO, określają sposób realizacji usług niezawodnego przesyłania danych. Obejmuje ona protokoły: TCP i UDP. Protokół TCP realizuje usługę połączenia wirtualnego. Protokół ten pobiera strumień danych z protokołu warstwy aplikacji oraz wykonuje operacje:

- segmentacja danych;
- transmisja pakietu z wykorzystaniem protokołu IP;
- odtwarzanie danych użytkowych;

Protokół TCP eliminuje też wspomniane wady protokołu IP – błędne lub zagubione pakiety są retransmitowane, a pakiety odebrane są ustawiane w prawidłowej kolejności logicznej. Aplikacje obsługiwane w trybie datagramowym korzystają a protokołu UDP nie dającego gwarancji przekazania datagramu, ale też stosowany narzut jest sprowadzony do minimum.

Warstwa5: **Aplikacji** – warstwa ta odpowiada warstwom 5, 6 i 7 modelu ISO. Każdy z protokołów warstwy 5 odpowiada jednemu z programów użytkowych intersieci. Warstwa ta realizuje wiele usług sieciowych, np. SNMP (ang. *Simple Network Management Protocol*), Telnet, FTP (ang. *File Transfer Protocol)*, SMTP (ang. *Simple Mail Transfer Protocol*) oraz Ping [1].

2.1 Usługi warstwy aplikacji TCP/IP

SMTP (ang. *Simple Mail Transfer Protocol*) – jest protokołem wspierającym pocztę elektroniczną, która stanowi ok. połowę liczby realizowanych połączeń TCP. Protokół SMTP realizuje połączenie pomiedzy dwoma punktami MTA (ang. *Message Transfer Agencies*) – jeden z nich jest klientem, a drugi serwerem. Punkty MTA zapewniają styk do różnych aplikacji poczty elektronicznej realizowanych w sieciach użytkowników.

Telnet – jest usługą pozwalającą użytkownikowi terminala zarejestrować się na odległym serwerze. Usługa Telnet umożliwia współpracę terminala oraz hosta pracujących pod różnymi systemami operacyjnymi, a to dzięki aplikacji wirtualnego terminala sieciowego NVT (ang. *Network Virtual Terminal)* wykonywanej na obydwóch końcach połączenia.

FTP (ang. *File Transfer Protocol)* - jest aplikacją pozwalajacą ściągać pliki z serwera do komputera klienta. Aplikacja FTP przekazuje więcej bajtów danych aniżeli połączenie SMTP. Anonimowe serwry FTP umożliwiają ściąganie plików każdemu użytkownikowi.

SNMP (ang. *Simple Network Management Protocol*) – jest protokołem wspomagającym zarządzanie siecią TCP/IP. Protokół SNMP cieszy się większym zainteresowaniem wśród użytkowników w porównaniu do protokołu CMIP (ang. *Common Mangement Information Protocol*) zalecanego w architekturze siedmiowarstwowej ISO/OSI, ponieważ jego konstrukcja jest mniej skomplikowana. Protokół SNMP wykorzystuje do przekazywania swoich danych protokół UDP, narzut jest bowiem w tym przypadku mniejszy od narzutu występującego w połączeniu wirtualnym.

Ping – jest aplikacją stosowaną powszechnie w celach diagnostycznych, do stwierdzenia dostępności odległego hosta. Aplikacja Ping wysyła do sprawdzanego serwera komunikat protokołu ICMP (ang. *Internet Control Message Protocol*) żądający zwrotnego potwierdzenia, a nastepnie oczekuje na to potwierdzenie.[2]

2.2 Porównanie siedmiowarstwowego modelu odniesienia ISO/OSI oraz TCP/IP

Zestaw protokołów TCP/IP nie jest w pełni zgodny z siedmiowarstwowym modelem odniesienia ISO/OSI, ale protokół IP można przypisać do 3 (warstwa sieciowa), a protokół TCP – do warstwy 4 (warstwa transportowa). Ewidentną różnicą pomiędzy TCP/IP a modelem ISO/OSI jest to, że protokół może do realizacji swoich funkcji wykorzystywać protokół należący do warstwy niższej bez konieczności „przechodzenia" przez warstwę pośrednią (np. aplikacja ping korzysta z protokołu ICMP bezpośrednio, bez uciekania się do mechanizmów warstwy transportowej, a przecież protokół ten jest przypisany do warstwy sieciowej. Innym przykładem nieprzestrzegania ścisłej hierarchii w zestawie protokołów TCP/IP jest sytuacja, gdy protokół niższy zamyka protokół wyższy bez konieczności komunikacji protokołów wyższych.

Siłą napędową rozwoju zestawu protokołów TCP/IP są użytkownicy końcowi. W przypadku modelu ISO/OSI siłą sprawczą ewolucji mogą być wyłącznie duzi dostawcy sprzętu komputerowego i komunikacyjnego oraz krajowe organizacje normalizacyjne poprzez dominujących, krajowych operatorów telekomunikacyjnych. Konsekwencją tego stanu rzeczy jest również znacznie wolniejszy przebieg procesu zatwierdzenia siedmiowarstwowego modelu odniesienia aniżeli zestawu TCP/IP. Z drugiej jednakże strony, dyskusja siedmiowarstwowego modelu odniesienia jest znacznie bardziej pogłębiona aniżeli w przypadku zestawu protokołów TCP/IP, a oznacza to, że model ISO/OSI po jego ogłoszeniu zmienia się w mniejszym stopniu. Można powiedzieć, że zestaw protokołów TCP/IP ma charakter do pewnego stopnia eksperymentalny – pewne protokoły składowe mogą być testowane, niektóre z wynikiem negatywnym, a to może powodować brak kompatybilności nowych wersji z wersjami starymi, których nie zaktualizowano [2].

3. PROTOKÓŁ IP

IP jest najważniejszym protokołem usług bezpołączeniowych, określającym w pierwszym rzędzie podstawową jednostką przesyłania danych w sieciach TCP/IP – datagram i schemat adresowania. Protokół ten określa również reguły przetwarzania i przenoszenia datagramów oraz metodykę generowania komunikatów o błędach.

W rozdziale tym wprowadzony zostanie schemat adresowania wykorzystywany przez protokół intersieci (IP) oraz wyjaśniony sposób podziału adresów IP na klasy.

3.1 Wprowadzenie

Zadaniem intersieci jest udostępnienie jednolitego systemu komunikacyjnego. Aby ten cel można było zrealizować oprogramowanie protokołów intersieci musi ukrywać szczegóły sieci fizycznych i oferować udogodnienia dużej sieci wirtualnej. Zasadniczą różnicą między intersiecią a siecią fizyczną jest fakt, że intersieć jest jedynie modelem opracowanym przez jego projektantów i działa tylko dzięki oprogramowaniu. Projektanci takich intersieci mają wolną rękę w wybieraniu adresów, metod dostarczania oraz formatów pakietów niezależnie od szczegółów sprzętowych.

Krytycznym elementem modelu intersieci jest adresacja, ponieważ aby nadać obraz pojedynczemu systemowi, wszystkie komputery muszą mieć jednolity schemat adresowania oraz każdy adres musi być jednoznaczny. Tak, więc największym problemem jest to, że intersieć może być budowana na wielu technikach sieciowych, co powoduje, że fizyczny adres sieci nie wystarcza.

Aby zagwarantować jednolite adresowanie we wszystkich węzłach intersieci, oprogramowanie protokołów określa schemat adresowania, który jest niezależny od bazowych adresów fizycznych.

Jednorodne adresowanie realizowane przez ukrywanie szczegółów bazowych adresów fizycznych, pozwala na stworzenie wielkiej, jednolitej sieci.

W stosie protokołów TCP/IP adresowanie jest zdefiniowane w protokole intersieci (ang. Internal Protocol - IP). Standard ten określa dla każdego węzła **adres węzła w protokole intersieci**, który jest 32-bitowym numerem przypisanym węzłowi, zwany również **adresem IP** lub **adresem internetowym**. Pakiet wysyłany przez intersieć zawiera zarówno adres IP nadawcy (źródła) oraz odbiorcy (celu).

3.2 Hierarchia adresów IP

Adres IP jest dzielony na dwie części, co zapewnia efektywne wyznaczanie tras:

- prefiks – jest to adres identyfikujący sieć fizyczną, do której jest podłączony dany komputer;
- sufiks – wskazuje konkretny komputer w danej sieci;

Oznacza to, że w każdej sieci fizycznej jest przypisana jednoznaczna wartość – **numer sieci**. Pojawia się on jako prefiks w adresie każdego komputera podłączonego do danej sieci. Każdy też komputer w danej sieci fizycznej ma przypisany jednoznaczny sufiks adresu.

Żadne dwie sieci nie mogą mieć przyznanego tego samego numeru ani też żadne dwa komputery w ustalonej sieci nie mogą mieć przyznanego tego samego sufiksu. Wartość sufiksu może być jednak wykorzystywana w więcej niż jednej sieci. Jeśli intersieć składa się z dwóch sieci, to można im przyznać numery 1 i 2. Trzy komputery podłączone do sieci 1 mogą mieć przypisane sufiksy 7, 8 i 9, natomiast komputery podłączone do sieci 2 mogą mieć przypisane sufiksy 3, 7 i 8.

Hierarchia adresów IP zapewnia dwie istotne własności:

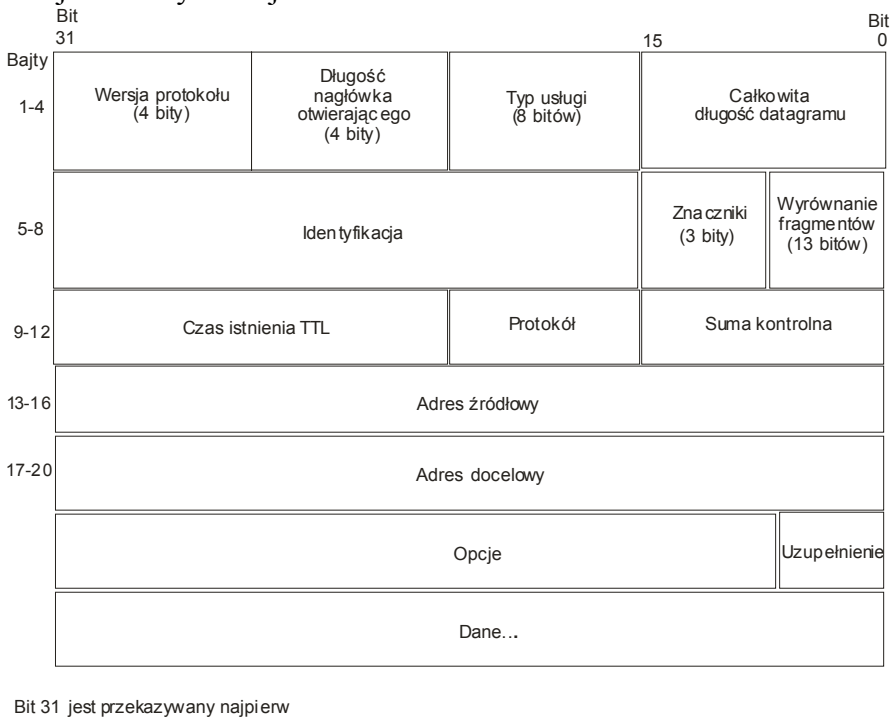
1. Każdy komputer ma przyznany jednoznaczny adres - dany adres nie jest nigdy przypisany do więcej niż jednego komputera. Własność ta jest zapewniona, gdyż pełny adres zawiera

zarówno prefiks, jak i sufiks, które są przyznane tak, aby zagwarantować jednoznaczność. Jeśli dwa komputery są przyłączone do różnych sieci fizycznych, to ich adresy mają różne prefiksy. Jeśli zaś dwa komputery są podłączone do tej samej sieci fizycznej, to ich adresy mają różne sufiksy.

2. Pomimo że przypisania numerów sieci muszą być koordynowane globalnie, sufiksy muszą być przyznane lokalnie bez globalnego uzgadniania [5].

3.3 Nagłówek protokołu IP

Budowę datagramu IP obrazuje poniższy rysunek (rys.1). Standardowy rozmiar nagłówka otwierającego IP jest równy 20 bajtów.



Rys.1. Budowa nagłówka datagramu IP

Do najważniejszych pól w nagłówku IP należy zaliczyć:

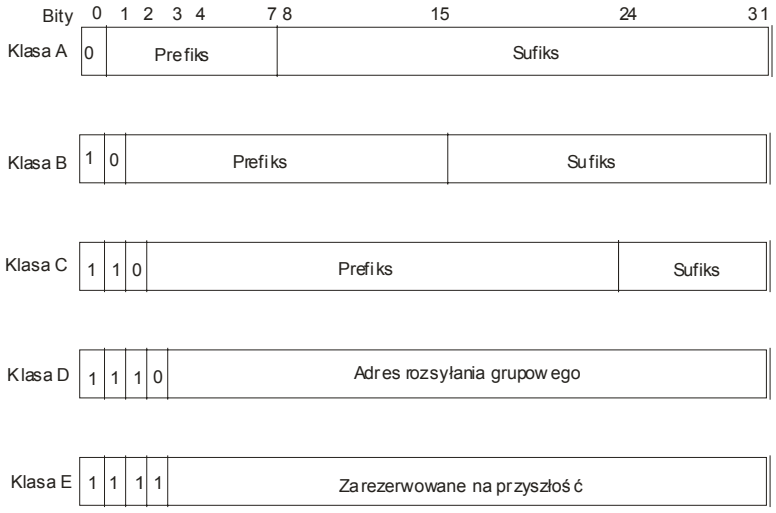
- ***Wersja protokołu*** — podaje numer używanej aktualnie wersji protokołu IP. To pole ma długość 4 bitów.
- ***Długość nagłówka otwierającego*** — wskazuje liczbę słów 32-bitowych (4-bajtowych) w nagłówku IP. To pole ma długość 4 bitów i zawiera wartość 0x5 (20 bajtów) lub większą. IP może dodatkowo rozszerzyć długość nagłówka o 4 bity naraz. Jeżeli dana opcja IP nie wykorzysta wszystkich 32 bitów słowa, to pozostałe bity zostają uzupełnione o zera tak, aby długość nagłówka IP była zawsze wielokrotnością 32 bitów.
- *Pierwszeństwo i typ usługi* — wskazuje ustawienia jakości usługi (QoS). To pole ma długość 8 bitów i zawiera informacje dotyczące pierwszeństwa, opóźnienia, przepustowości oraz parametry niezawodności.
- ***Całkowita długość datagramu*** — określa rozmiar datagramu w bajtach (nagłówek plus ładunek). To pole ma długość 16 bitów i zawiera liczbę słów 32-bitowych w datagramie.
- ***Identfikacja*** — identyfikuje określony datagram IP. To pole ma długość 16 bitów. Jeżeli datagram IP ulegnie fragmentacji podczas routingu, informacja zawarta w tym polu jest wykorzystywana do ponownego złożenia w miejscu przeznaczenia.
- ***Znaczniki*** — zawiera znaczniki fragmentacji. To pole ma długość 3 bitów, ale obecnie wykorzystuje się tylko dwa spośród nich. Bit najmniej znaczący wskazuje czy jest to

fragment końcowy w datagramie (czy też będzie ich więcej). Drugi bit najmniej znaczący wskazuje czy datagram może być fragmentowany, czy nie.

- ***Wyrównanie fragmentów*** — wskazuje pozycję fragmentu w stosunku do oryginalnego ładunku IP. To pole ma długość 13 bitów.
- ***Czas istnienia (TTL)*** — wskazuje w sekundach czas, przez jaki dany datagram pozostanie w sieci, zanim zostanie odrzucony. Ilekroć dany datagram przechodzi przez router, czas istnienia (TTL) zostaje zmniejszony, o co najmniej jedną sekundę. Ponieważ router normalnie przekazuje schemat IP w czasie krótszym niż jedna sekunda, ustawienie TTL staje się faktycznie liczbą przeskoków. To pole ma długość 13 bitów. To pole jest ustawiane przez komputer źródłowy i zmniejszane w każdym węźle sieci. Jeżeli wartość czasu życia spadnie do zera, datagram jest niszczony, a protokół ICMP wysyła do hosta komunikat o wystąpieniu błędu. Taki sposób postępowania zabezpiecza sieć przed przeciążeniem pakietami, które z różnych powodów nie mogą dotrzeć do hosta.
- ***Protokół*** — wskazuje protokół, który dał protokołowi IP ładunek do wysłania. To pole ma długość 8 bitów. Informacja zawarta w tym polu jest wykorzystywana przez warstwy wysokiego poziomu w hoście docelowym do przetwarzania ładunku. ICMP na przykład, sygnalizowany jest przez wartość równą jeden w tym polu.
- ***Suma kontrolna*** — wykorzystywana jest wyłącznie do sprawdzania integralności nagłówka i w związku z tym, bywa czasem określana jako *suma kontrolna nagłówka*. Ładunek może zawierać własną sumę kontrolną. To pole ma długość 16 bitów. Ponieważ TTL zmienia się przy każdym przeskoku, suma kontrolna jest ponownie obliczana ilekroć datagram przechodzi przez router. Jest ona wyznaczana z wykorzystaniem metody dopełnień do jedności wyłącznie dla danych zapisanych w nagłówku.
- ***Adres źródłowy*** — zawiera adres źródłowy IP. To pole ma długość 32 bitów w przypadku IPv4 (128 bitów w przypadku IPv6).
- ***Adres docelowy*** — zawiera adres docelowy IP. To pole ma długość 32 bitów w przypadku IPv4 (128 bitów w przypadku IPv6).
- ***Opcje*** *[długość pola jest zmienna]* – mogą zajmować przestrzeń na końcu nagłówka IP.
- ***Uzupełnienie*** *[długość pola jest zmienna]* – jeśli pole opcji nie zajmuje pełnego słowa to zostaje uzupełnione do 32 bitów [9].

3.4 Klasy adresów IP

Po obraniu rozmiarów adresów IP i podziale każdego z nich na dwie części potrzeba jeszcze określić ile bitów umieścić w każdej części. Prefiks musi mieć wystarczającą liczbę bitów, aby umożliwiał przypisanie każdej sieci fizycznej jednoznacznego numeru w intersieci. Sufiks musi mieć wystarczającą liczbę bitów, aby umożliwić przypisanie jednoznacznego sufiksu każdemu komputerowi podłączonemu do sieci. Opracowany został schemat adresowania, który może działać przy kombinacji dużych i małych sieci. Przestrzeń adresowa została podzielona na podstawowe **klasy**, z których każda ma inny rozmiar prefiksu i sufiksu.
Pierwsze cztery bity adresu określają klasę - do której należy adres, oraz sposób podziału pozostałej części adresu na prefiks i sufiks.



Rys.2 Pięć klas adresów IP.

Na rysunku (Rys.2) przedstawiony został podział klas adresów IP (pięć klas), pierwsze bity wykorzystywane do identyfikacji klasy oraz podział na prefiks i sufiks. Jest on zgodny z konwencją używaną w protokołach TCP/IP, gdzie bity numeruje się od lewa do prawa, oznaczając pierwszy z nich zerem. Adresy przypisywane komputerom należą do klasy A,B lub C. Prefiks identyfikuje sieć, a sufiks – komputer w ramach sieci.
Klasy A, B i C są nazywane **klasami pierwotnymi**, gdyż są przeznaczone na adresy komputerów. Klasa D jest wykorzystywana przy rozsyłaniu grupowym, które umożliwia dostarczanie do zbioru komputerów. Aby użyć rozsyłania grupowego IP, zbiór węzłów musi zgodzić się na wspólny adres rozsyłania grupowego. Po ustawieniu grupy rozsyłania kopia każdego pakietu wysłanego pod danym adresem rozsyłania będzie dostarczona do każdego komputera będącego członkiem tej grupy [5].

3.4.1 Sieci klasy A

W sieci klasy A tożsamość sieci określana jest przez wartość pierwszego oktetu (ośmiu bitów). Dlatego są one często określane jako ***sieci /8***. Ponieważ zakres wartości dla pierwszego oktetu adresu klasy A to 1 do 126, jest 126 niepowtarzalnych sieci klasy A. Pozostałe 24 bity adresu identyfikują hosta. Tożsamości hostów nie mogą być wyłącznie jedynkami, ani wyłącznie zerami, więc maksymalna liczba hostów w każdej sieci klasy A to 2^{24}–2.
Blok adresowy *klasy A* zawiera 2^{31} indywidualnych adresów (łącznie z zarezerwowanymi wartościami pierwszego oktetu, wynoszącymi 0 oraz 27), a przestrzeń adresowa IPv4 zawiera 2^{32} adresów. Stąd przestrzeń adresowa klasy A to 50% całkowitej przestrzeni adresowej IPv4. Wszystkie adresy protokołu IP muszą być niepowtarzalne w swojej własnej sieci. Jeśli jednak dwie sieci złożone nie wiedzą o sobie nawzajem i nie mogłyby nigdy pojawić się na tej samej trasie, to ten sam adres IP mógłby pojawić się w obu z nich.
Zazwyczaj do wewnętrznego adresowania w intersieci wykorzystywana jest sieć klasy A 10.0.0.0. Jeżeli hosty w danej sieci 10.0.0.0 mają mieć dostęp do Internetu, to musi zostać zaimplementowana usługa translacji adresów sieciowych (NAT). Przykładowy adres tej klasy ma postać: 10.0.0.0, gdzie Id. Sieci: 10.

3.4.2 Sieci klasy B

W sieci klasy B tożsamość sieciowa określana jest przez wartość pierwszych dwóch oktetów, (16 bitów). Sieci klasy B są, zatem czasami określane jako ***sieci /16***. Dwa pierwsze bity identyfikują daną sieć jako sieć klasy B, co pozostawia 14 bitów na określenie niepowtarzalnych tożsamości sieciowych. Stąd też można zdefiniować 2^{14}, albo 16 384, sieci klasy B, przy czym każda z nich może mieć 2^{16}–2 hostów. Blok adresowy klasy B zawiera 2^{30}

adresów i stanowi 25% całkowitej przestrzeni adresowej IPv4. Przykładowy adres tej klasy ma postać: 128.3.2.3, gdzie Id. sieci: 128.3; Id. węzła: 2.3.

3.4.3 Sieci klasy C

W sieci klasy C tożsamość sieciowa określana jest przez wartość pierwszych trzech oktetów, (24 bitów). Sieci klasy C są, zatem czasami określane jako ***sieci /24***. Trzy pierwsze bity identyfikują daną sieć jako sieć klasy C, co pozostawia 21 bitów na określenie niepowtarzalnych tożsamości sieciowych. Stąd też można zdefiniować 2^{21} sieci klasy C, przy czym każda z nich może mieć do 2^8–2, lub 254 hostów. Blok adresowy klasy C zawiera 2^{29} adresów i stanowi 12,5% całkowitej przestrzeni adresowej IPv4. Przykładowy adres tej klasy ma postać: 192.0.1.255, gdzie Id. sieci: 192.0.1; Id. węzła: 255 [9].

3.4.4 Sieci klasy D i E

Sieci klasy D wykorzystywane są do multiemisji, gdzie pojedynczy adres sieciowy identyfikuje grupę hostów. Sieci klasy E zarezerwowane są do celów doświadczalnych. Blok klasy D stanowi 6,25% całkowitej przestrzeni adresowej IPv4, a blok klasy E nieznacznie mniejszą jej część, ponieważ 255 nie jest wykorzystywane jako wartość pierwszego oktetu [9].

3.5 Obliczanie klasy adresu IP

Adresy IP są nazywane **samoidentyfikującymi się** gdyż klasa adresu może być obliczona na podstawie jego samego.
Na rysunku (Rys.3) pokazana została tablica, która może być wykorzystywana przy obliczaniu klasy adresu. Osiem kombinacji, które zaczynają się od 0 odpowiada klasie A. Cztery kombinacje, które zaczynają się od 01 – odpowiadają klasie B, a dwie kombinacje zaczynające się od 110 – klasie C. Adres zaczynający się od 111 należy do klasy D. Adres, który zaczyna się od 1111 należy do klasy E, która nie jest obecnie używana.

Pierwsze 4 bity adresu	Indeks w tablicy (dziesiętnie)	Klasa adresu
0000	0	A
0001	1	A
0010	2	A
0011	3	A
0100	4	A
0101	5	A
0110	6	A
0111	7	A
1000	8	B
1001	9	B
1010	10	B
1011	11	B
1100	12	C

Rys.3 Tablica, która może być wykorzystywana przy obliczaniu klasy adresu.
Pierwsze 4 bity adresu są wydobywane i wykorzystywane jako indeks w tablicy

Adresy IP mogą być zapisywane w systemie binarnym (na przykład: 11000011101000101110011000000001), ale jest to nieporęczne. Mogą też być zapisywane w systemie szesnastkowym (np.C3A2CB01). Jest to krótsze, ale i tak trudne do zapamiętania. Mogą być również przekształcane bezpośrednio na system dziesiętny (3 282 225 921 powyższy przykład), ale ten format jest prawie tak trudny do zapamiętania, jak szesnastkowy. Jest on również znacznie mniej użyteczny, ponieważ wartość każdego z 4 bajtów w liczbie 32-bitowej jest ważna i nie jest łatwo obliczalna z wartości dziesiętnej [5].

3.5.1 Klasy a notacja dziesiętna z kropkami

Normalną praktyką jest dzielenie danego adresu IP na 4 bajty (oktety) a następnie obliczanie wartości dziesiętnej dla każdego z oktetów. Oktety oddzielone są kropkami i stąd wywodzi się termin ***kropkowa notacja dziesiętna***. W tym sposobie zapisu nie było nic szczególnego, kiedy go wybierano. Był to po prostu kompromis pomiędzy czytelnością a użytecznością.
Format dziesiętny kropkowy wykorzystuje się do wpisywania i wyświetlania adresów IP w szerokiej gamie graficznych interfejsów użytkownika (GUI), ale należy zawsze pamiętać, że adres IP to po prostu 32-bitowa wartość binarna.
W wypadku adresów klasy A ostatnie trzy oktety odpowiadają sufiksowi komputera. W adresach klasy B są dwa oktety sufiksu komputera, a w adresach klasy C – jeden oktet.

Tabela 2. Zakresy wartości dziesiętnych odpowiadające poszczególnym klasom adresów

Klasa	Zakres wartości
A	Od 0 do 127
B	Od 128 do 191
C	Od 192 do 223
D	Od 224 do 239
E	Od 240 do 255

Niestety, ponieważ w notacji dziesiętnej z kropkami nie widać poszczególnych bitów adresu, klasę musimy rozpoznawać na podstawie wartości dziesiętnej pierwszego oktetu. W tabeli (Tabela2) pokazane są odpowiadające poszczególnym klasom zakresy wartości dziesiętnych.

3.6 Maska podsieci

W roku 1985 dokument RFC 950 określił standardową procedurę obsługującą podział na podsieci, która została wprowadzona, ponieważ dany administrator lokalny, który potrzebował drugiej sieci, zmuszony był żądać innego numeru sieci, pomimo że wciąż były dostępne adresy hostów w sieci pierwotnie przydzielonej.
Podział na podsieci dzieli standardowe klasowe pole numeru hosta na dwie części — numer podsieci oraz numer hosta w tej podsieci. Praktycznie rzecz ujmując podział na podsieci bierze bity z adresu hosta i zamiast tego przydziela te bity adresowi sieci, w ten sposób dokonując dalszego podziału sieci
Maska podsieci, podobnie jak adres IP, jest 32-bitową liczbą binarną, ale posiada bardzo specyficzny format. Musi ona składać się z grupy jedynek poprzedzającej grupę zer — na przykład 11111111111111110000000000000000. Maski podsieci są zazwyczaj zapisywane albo przy użyciu kropkowej notacji dziesiętnej (255.255.0.0), albo w formacie ukośnikowym, gdzie wartość po ukośniku reprezentuje liczbę jedynek (/16).
Funkcją maski podsieci jest identyfikowanie, która część adresu IP określa sieć, a która część określa hosta. Jedynki określają, że odpowiadające im bity w adresie IP to bity sieci, a zera określają bity hosta. W przypadku tradycyjnego adresowania klasowego, początkowe bity adresu określają klasę adresu, która z kolei określa zakres hosta i sieci. Stąd, kiedy wprowadzano adresy IP oraz adresowanie klasowe, nie zostały zaimplementowane maski sieci.
Jednak analiza początkowych bitów adresu jest nużąca, a maski podsieci upraszczają ten proces. Binarna operacja AND sprawia, że zera w masce podsieci maskują część hosta w adresie IP, pozostawiając tylko te bity, które identyfikują sieć, albo prefiks sieci. Adresy klasy A (adresy */8*) mają domyślną maskę podsieci */8* (255.0.0.0). Klasy B *i* C mają domyślne maski podsieci, odpowiednio */16* (255.255.0.0) i */24* (255.255.255.0).
Czasem bywają mylone pojęcia podziału na podsieci i **masek podsieci o zmiennej długości (VLSM).** Wydaje się to zrozumiałe, gdyż technika podziału na podsieci polega na zmianie długości maski podsieci. Jednak, kiedy dzielimy sieć na podsieci, rozbijamy ją na segmenty, z których wszystkie są tej samej wielkości. Pojedynczą maskę podsieci (nie domyślną maskę

podsieci) stosuje się wobec całej sieci. Korzyści płynące z przydzielania wielu masek podsieci danemu numerowi IP sieci:

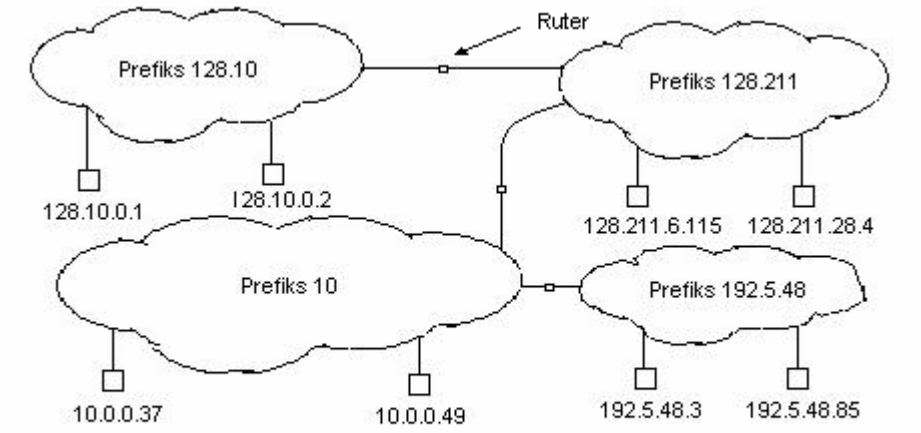
- umożliwiają one bardziej wydajne wykorzystanie przydzielonej danemu przedsiębiorstwu przestrzeni adresów IP;
- umożliwiają one zespalanie tras, co może znacząco ograniczyć ilość informacji dotyczących routingu w obrębie domeny routingu danej organizacji.

3.7 Przykład przydziału adresu

Rozpatrujemy organizację, która zdecydowała się utworzyć prywatną intersieć TCP/IP, składającą się z czterech sieci fizycznych. Organizacja musi nabyć rutery, aby połączyć te cztery sieci, a następnie przyznać adresy IP. Na początek obieramy jednoznaczny prefiks dla każdej sieci.

Przy przyznawaniu prefiksu sieciowego musi zostać obrana liczba z klasy A, B lub C – wybór zależy od rozmiaru sieci fizycznej. Zwykle sieciom przyznawane są adresy klasy C. W przypadku sieci podłączonych do globalnego Internetu wyboru, jaką klasę dokonuje usługodawca sieciowy. Dla intersieci prywatnej klasę obiera lokalny administrator.

Administrator sieciowy szacuje docelowy rozmiar każdej sieci fizycznej i wykorzystuje go przy wyborze prefiksu. Jeśli organizacja potrzebuje jednej małej sieci, dwu sieci średniego rozmiaru oraz jednej bardzo dużej, administrator może zdecydować się na przyznanie prefiksu klasy C (np. 192.5.48), dwu prefiksów klasy B (np. 128.10 oraz 128.211) oraz prefiksu klasy A (np. 10). Rysunek (Rys.4) przedstawiona została intersieć złożona z czterech sieci fizycznych, którym przyznano takie prefiksy, oraz przykłady adresów IP przypisanych węzłom.



Rys.4 Przykład prywatnej intersieci z adresami IP przyznanymi komputerom.

Rozmiar chmurki użytej do oznaczenia sieci fizycznej odpowiada liczbie spodziewanych w niej węzłów. Rozmiar sieci określa klasę przyznawanego adresu. Adres IP maszyny zawsze zaczyna się od prefiksu przydzielonego sieci fizycznej, do której dana maszyna jest podłączona. Sufiksy mogą być zupełnie dowolne. Przykładowe adresy przydzielone maszynom na rysunku pokazują, że sufiksy mogą mieć dowolne wartości, np. 49 lub 3 [5].

3.8 Adresy IP specjalnego przeznaczenia

Dla wygody, zamiast przyznawania adresu każdemu komputerowi, dobrze jest określić adresy, które mogą być przypisywane do pewnej sieci lub zbiorów komputerów. IP określa zestawy adresów o szczególnej postaci, które są zarezerwowane – nie są one nigdy przyznawane komputerom. Są nimi:

– ***Adresy sieciowe*** odnoszą się do samej sieci, a nie do komputerów podłączonych do niej. Nie powinien się nigdy pojawiać jako adres docelowy w pakiecie. Adres sieci tworzymy przepisując niezmienione wszystkie bity adresu IP, dla których odpowiednie bity maski mają

wartość jeden. Resztę uzupełniamy zerami.

- ***Adres rozgłaszania ukierunkowanego*** – jeśli wygodniej jest wysłać kopię pakietu do wszystkich węzłów sieci fizycznej to wysyłamy pakiet pod tym właśnie adresem danej sieci. Przez intersieć podróżuje tylko jedna jego kopia, aż dotrze do danej sieci. Następnie pakiet ten jest dostarczany do wszystkich węzłów tej sieci. Adres ten jest tworzony przez dodanie do jej prefiksu sufiksu, który cały składa się z jedynek.
- ***Adresy rozgłaszania ograniczonego*** odnoszą się do rozgłaszania w lokalnej sieci fizycznej. Jest ono używane przy starcie systemu przez komputery, które nie znają w tym momencie numeru sieci. IP na rozgłaszanie ograniczone rezerwuje adres składający się z samych jedynek. W ten sposób pakiet wysłany pod tym adresem oprogramowanie IP rozgłosi w sieci lokalnej.
- ***Adresy pętli zwrotnej*** służą do komunikacji z wykorzystaniem protokołu IP z lokalnym komputerem (*localhost*). Jest to adres zawsze przypisany komputerowi, na którym właśnie pracujemy, ponieważ pakiety z takimi adresami nie powinny wydostawać się na zewnątrz komputera, nie powoduje to żadnych konfliktów. Protokół IP rezerwuje prefiks sieciowy klasy A równy 127 na adresy pętli zwrotnej. Adres węzła (sufiks) używany przy tym jest bez znaczenia. Najpopularniejszym adresem pętli zwrotnej jest 127.0.0.1.
- ***Adres bieżącego komputera*** – zestaw protokołów TCP/IP obejmuje protokoły, których komputer może używać przy automatycznym uzyskiwaniu adresu IP przy starcie. Komputer korzystając z takich protokołów uruchomieniowych nie może podać prawidłowego adresu IP nadawcy. Radzi sobie w takich sytuacjach tak, że adres IP składa się z samych zer i oznacza adres bieżącego komputera [5].

Tabela 3. Postacie adresów IP specjalnego przeznaczenia

Prefiks	Sufiks	Typ adresu	Przeznaczenie
Same zera	Same zera	Bieżący komputer	Używany przy rozruchu
Sieciowy	Same zera	Sieć	Identyfikuje sieć
Sieciowy	Same jedynki	Rozgłaszanie ukierunkowane	Rozgłaszanie w określonej sieci
Same jedynki	Same jedynki	Rozgłaszanie ograniczone	Rozgłaszanie w sieci lokalnej
127	Cokolwiek	Pętla zwrotna	Testowanie

4. PRZYSZŁOŚĆ PROTOKOŁU IP

Protokół TCP/IP jest też nazywany *standardem otwartym* (ang. *open standard*). Oznacza to, że żadna firma ani osoba nie kontroluje specyfikacji tego protokołu lub sposobu, w jaki działa. Jego ewolucję nadzoruje zespół Internet Engineering Task Force (IETF), który skupia ekspertów z dziedziny przemysłu sieciowego oraz reprezentantów przedsiębiorstw. Grupy robocze, które działają w obrębie IETF rewidują, omawiają, zalecają i zatwierdzają proponowane zmiany w standardzie (korzystają przy tym z raportów technicznych zwanych *Request For Comments* (RFC)) [4].

W tym rozdziale określone zostaną mocne strony i ograniczenia IPv4. Pokazany zostanie zmiennik tej wersji zaproponowanej przez IETF, czyli IPv6 (cechy i pokonanie niektórych ograniczeń w odniesieniu do wersji IPv4).

4.1 Sukces protokołu IP

Wersja protokołu IPv4 (nazwa użytkowa IP) osiągnęła ogromny sukces. Protokół ten umożliwił Internetowi:

- poradzenie sobie z różnorodnymi sieciami,
- istotnymi zmianami w technologii sprzętowej,
- z ogromnymi zmianami skali.

Aby obsłużyć różnorodność sieci IP definiuje jednolity format pakietów (datagram IP – jest podstawową jednostką komunikacyjną w Internecie; gdy program użytkowy przesyła dane przez Internet z jednego komputera do drugiego, podróżują one w datagramie IP) i mechanizm ich przesyłania.

Protokół IP definiuje także zbiór adresów, które umożliwiają programom użytkowym oraz wyższym warstwom protokołów komunikację poprzez różnego rodzaju sieci bez względu na różnice w adresach sprzętowych wykorzystywanych w systemach sieciowych.

Może się on także dostosować do zmian w technologii sprzętu - jest on obecnie stosowany w sieciach działających o kilka rzędów wielkości szybciej, od tych, które były wykorzystywane w czasie jego projektowania. Co więcej niektóre nowoczesne sieci udostępniają rozmiary ramek, które mają większe rozmiary niż te za czasów jego powstawania. Protokół IP działa efektywnie z tymi sieciami, gdyż ma możliwość wykorzystywania zwiększonego rozmiaru ramek.

4.2 Powody zmian

Jeśli protokół IP działa tak dobrze to, dlaczego cokolwiek zmieniać? Spowodowane jest to ograniczonością przestrzeni adresowej. Gdy IP był definiowany użyte zostało 32 bity na jego adres. Jednak projektanci nie przewidzieli do końca tak szybkiego i gwałtownego rozrostu Internetu, który rozrasta się wykładniczo podwajając rozmiary w czasie krótszym niż rok. Dlatego aby pozwolić na ciągły wzrost Internetu konieczne są dłuższe adresy.

Do mniej ważnych powodów zmian w protokole IP można zaliczyć stosowanie nowych programów użytkowych w Internecie. Dobrym przykładem są programy zajmujące się przekazem dźwięku i obrazu, które muszą dostarczać dane w regularnych odstępach. Aby utrzymać przepływ takich danych bez zakłóceń należy unikać częstego zmieniania tras. Protokół nie określa niestety typu, który może być wykorzystywany do dostarczana dźwięku i obrazu w czasie rzeczywistym. Stąd, aby wyjść naprzeciw trudnościom i ograniczeniom powstała nowsza, „ulepszona” wersja protokołu IP pod nazwą **IPv6** [5].

4.3 Charakterystyka IPv6

Podobieństwem protokołów IPv4 i IPv6 jest to, że oba są protokołami bezpołączeniowymi (każdy datagram zawiera adres odbiorcy i ma wyznaczoną trasę niezależnie od innych datagramów).

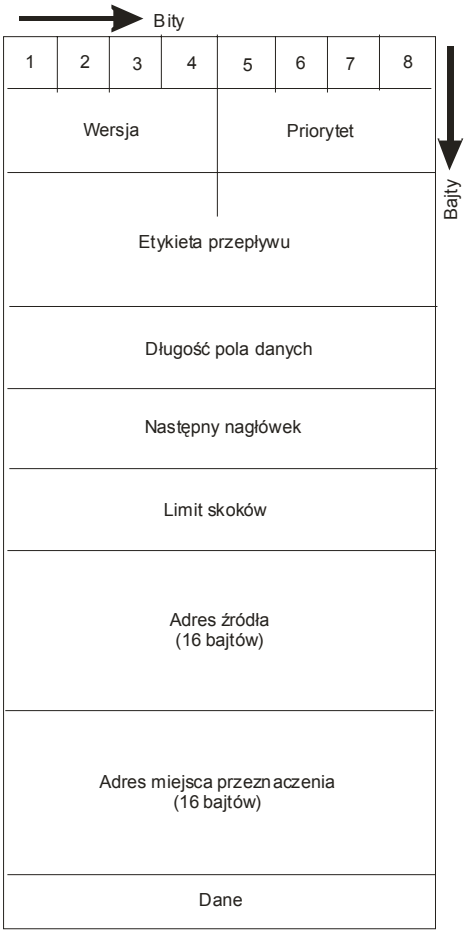
Nowe cechy IPv6:

- *rozmiar adresu* – każdy adres IPv6 ma 128 bitów zamiast 32. Powstała przestrzeń adresowa jest niewyobrażalna do wyczerpania w dającej się przewidzieć przyszłości.
- *format nagłówka* – jest zupełnie inny niż nagłówek IPv4. Prawie każde pole zostało zmienione, niektóre usunięto.
- *nagłówki dodatkowe* – inaczej niż w IPv4, który używa pojedynczego formatu nagłówka dla wszystkich datagramów, IPv6 koduje informacje w oddzielnych nagłówkach. Datagram składa się tu z podstawowego nagłówka IPv6, po którym mogą się znajdować nagłówki dodatkowe z następującymi po nich danymi.
- *wsparcie dla dźwięku i obrazu*- IPv6 obejmuje mechanizm, który umożliwia nadawcy i odbiorcy ustawienie ścieżki wysokiej jakości przez sieci bazowe i powiązanie datagramów z tą ścieżką. Chociaż mechanizm ten jest przeznaczony do wykorzystania przez programy do przesyłania dźwięku i obrazu. Wymagają one zagwarantowania wysokiej jakości połączenia Protokół ten może też być wykorzystany do wiązania datagramów ze ścieżkami o niskim koszcie.
- *Rozszerzalny protokół* – IPv6 nie określa wszystkich możliwych funkcji – przeciwnie do IPv4. W zamian za to możliwy jest schemat dający prawo nadawcy dodania do datagramu dodatkowej informacji. Ten schemat rozszerzeń w IPv6 stanowi przewagę jego nad IPv4 i czyni go bardziej elastycznym – nowe elementy mogą być dodawane do projektu w miarę potrzeb.

Choć długie adresy rozwiązują problem niewystarczającej przestrzeni, to pojawia się inny, równie interesujący. Ludzie zajmujący się administracją sieciami muszą tymi adresami operować. Notacja kropkowo-dziesiętna używana w IPv4 nie nadaje się, gdyż adresy są za długie. Jako rozwiązanie zaproponowano używanie notacji szesnastkowej z dwukropkami, co umożliwia dodatkowo także kompresję zer. Przykładowy adres kropkowo-dziesiętny dla IPv6 wyglądałby tak: 104.230.140.100.255.255.255.255.0.0.17.128.150.10.255.255 stosując krótszą formę - zapis szesnastkowy: 68E6:8C64:FFFF:FFFF:0:1180:96A:FFFF. Możliwa jest także tzw. *kompresja zer* – ciąg powtarzających się zer jest zastępowany przez parę dwukropków. Adres FF05:0:0:0:0:0:0:B3 może zostać zapisany jako FF05::B3. Aby zapewnić, że kompresja zer nie powoduje niejednoznaczności w zapisie, może być ona zastosowana tylko raz [8].

4.3.1 Format nagłówka protokołu IPv6

Datagram IPv6 składa się z dwu elementów: Nagłówka IPv6 i danych. Długość nagłówka protokołu IPv6 wynosi 320 bitów (40 oktetów). Datagram protokołu IPv6 został przedstawiony na rysunku (Rys.5).



Rys.5 Format datagramu protokołu IP v 6

Podstawowe pola datagramu to:
Wersja *(Version)* – określa numer wersji protokołu, ma 4 bity (6 na tym polu oznacza, że jest nagłówek protokołu IPv6);
Priorytet *Priority)* – określa priorytet pakietu w stosunku do innych pakietów pochodzących z tego samego źródła, ma 4 bity;
Etykieta przepływu *(Flow label)* – identyfikuje wymagający specjalnej obsługi przepływ pakietu, zajmuje 24 bity;
Długość pola danych*(Payload)* – określa wyrażoną w oktetach długość pozostałej, następującej po nagłówku części pakietu, jest liczbą 16-bitową;

Następny nagłówek *(Next Header)* – identyfikuje nagłówek następujący po nagłówku IPv6, zajmuje 8 bitów;
Limit skoków *(Hop limit)* – liczba, która zmniejszona o jeden, gdy pakiet przechodzi przez węzeł. Jeśli limit skoków osiągnie zero, to pakiet zostanie odrzucony, ma długość 8 bitów;
Adres źródła *(Source address)* – zawiera adres nadawcy (adres źródłowy hosta) pakietu;
Adres miejsca przeznaczenia *(Destination address)* – zawiera adres odbiorcy pakietu;
Dane *(Data)* – zawiera dane przesyłane pakietem (nagłówek TCP, dane TCP itp.) [10].

4.3.2 Nagłówki opcjonalne – obsługa i powody użycia

Podstawowy nagłówek IPv6 jest tylko dwa razy dłuższy od nagłówka IPv4 (40 oktetów w stosunku do 20). Zwiększenie wydajności uzyskuje się przez optymalizację operacji związanych z nagłówkiem pakietu i przesuwając niektóre opcjonalne funkcje do nagłówków rozszerzenia. Aby przejść do elementu następnego po nagłówku, IPv6 po prostu dodaje 40 do adresu nagłówka podstawowego.

Pakiet IPv6 może mieć zero, jeden lub większą liczbę nagłówków rozszerzających. Pole *Next Header* identyfikuje nagłówek, który przychodzi w następnej kolejności.

Wyróżniamy sześć opcjonalnych, rozszerzających nagłówków:

- ***Hop-by-Hop*** *(skok po skoku)* – przenosi informację, która musi być sprawdzana i przetwarzana w każdym węźle wzdłuż drogi przesyłania pakietu, również w węźle docelowym;
- ***Destination*** *(miejsca przeznaczenia)* – przenosi informację, która wymaga sprawdzenia pakietu tylko w miejscu przeznaczenia. Obecnie brak jest przykładu na zastosowanie tej opcji;
- ***Routing Header*** *(nagłówek routingu)* – specyfikuje pośrednie węzły tworzące ścieżkę od źródłą do miejsca przeznaczenia;
- ***Fragment Header*** *(nagłówek fragmentacji)* – używany jest przez węzeł źródłowy w celu działania komunikatu na fragmenty, które mogą być przetwarzane przez znajdujące się po drodze routery;
- ***Authentication*** *(uwierzytelnianie)* – zapewnia integralność i uwierzytelnianie danych;
- ***Encapsulating Security Payload*** *(ESP)* – zapewnia poufność danych;[10]

Dlaczego w IPv6 są stosowane oddzielne nagłówki dodatkowe? Rozdzielenie funkcji datagramu na wiele nagłówków jest uzasadnione ekonomicznie – pozwala oszczędzać miejsce. Istnienie oddzielnych nagłówków w IPv6 umożliwia zdefiniowanie dużego zestawu opcji bez wymagania, aby każdy nagłówek datagramu miał choćby po jednym polu dla każdej z nich. Przykład: nagłówek IPv4 zawiera pola wykorzystywane do przechowywania informacji o fragmentowaniu, w IPv6 miejsce na pola fragmentacji nie jest zajmowane, jeśli datagram nie jest fragmentowany. Większość datagramów ma jedynie kilka nagłówków, dlatego unikanie niepotrzebnych pól w nagłówkach zaoszczędza w znaczący sposób miejsce. Mniejsze datagramy są również krócej transmitowane. Stąd wniosek, że zmniejszenie rozmiaru datagramu oznacza zmniejszenie zajmowanej części przepustowości [5].

4.3.3 Adresowanie w IPv6

W protokole IPv6 (analogicznie jak w IPv4) każdemu połączeniu sieci z komputerem przypisywany jest jednoznaczny adres. Również podobnie jak w IPv4 protokół IPv6 rozdziela każdy taki adres na prefiks identyfikujący sieć oraz sufiks określający dany komputer w sieci. Jednak mimo podobieństw adresowanie w protokole IPv6 różni się w znaczny sposób od adresowania w IPv4. **Różnice:**
- adresy nie mają określonej klasy;
- granica między prefiksem i sufiksem może znajdować się w dowolnym miejscu i nie może być wyznaczona na podstawie samego adresu;

- każdemu adresowi może być przypisana długość prefiksu, dzięki temu możliwe jest określenie gdzie kończy się prefiks;
- zestaw adresów specjalnego przeznaczenia jest całkowicie odmienny;
- w IPv6 nie udostępniono specjalnego adresu do rozgłaszania w danej sieci;
- każdy z adresów IPv6 należy do jednego z *trzech podstawowych typów:*
  - **Adres jednostkowy (Unicast)** – odpowiada on pojedynczemu komputerowi; datagram wysyłany jest najkrótszą ścieżką do danego komputera;
  - **Adres rozsyłania grupowego (Multicast)** – odpowiada on zbiorowi komputerów znajdujących się być może w różnych miejscach; gdy pod takim adresem jest wysyłany datagram, jest on dostarczany za pomocą IPv6 do każdego członka grupy;
  - **Adres grona (Anycast)** – odpowiada on zbiorowi komputerów który mają wspólny prefiks adresowy; datagram jest przesyłany najkrótszą drogą i dostarczany dokładnie do jednego komputera [5].

4.4 Podsumowanie

Mimo iż już dzisiaj działają sieci bazujące na protokole IPv6 to jednak przewiduje się, że protokół IPv4 będzie z powodzeniem panował jeszcze przez ok 5-10lat. Niemniej jednak niedawno dokonano oficjalnego przydziału adresów IPv6 dla amerykańskiego ISP z puli adresów nie testowych. Potwierdza to fakt, iż IPv6 zdobywa coraz większą popularność nie tylko w środowisku administratorów - eksperymentatorów. Planuje się, że przejście na protokół IPv6 będzie odbywać się stopniowo, a sieci IPv4 i IPv6 będą przez jakiś czas współistnieć. Komunikację pomiędzy obiema sieciami mają zapewnić translatory nagłówków oraz proxy.

5. TCP – USŁUGA NIEZAWODNEGO PRZESYŁANIA

TCP to główny protokół transportowy w zestawie TCP/IP zapewniający niezawodne dostarczanie danych. Oprogramowanie tego protokołu wykonuje pozornie niemożliwe zadanie – używa oferowanej przez protokół IP zawodnej usługi przenoszenia datagramów do wysyłania danych do innego komputera, udostępniając przy tym programom użytkowym usługę niezawodnego dostarczania danych. Oprogramowanie TCP, aby zapewnić efektywne przesyłanie danych, musi kompensować powstające straty czy opóźnienia bez przeciążenia bazowych sieci oraz ruterów.

W rozdziale tym zostanie opisane, w jaki sposób protokół ten zapewnia niezawodne dostarczanie danych, jakie są metody używane w TCP do zapewnienia niezawodności przesyłania, co to jest retransmisja i jaki jest format segmentu TCP.

5.1 Główne cechy usługi TCP

Z punktu widzenia programu użytkowego usługa oferowana przez TCP ma następujące główne cechy:

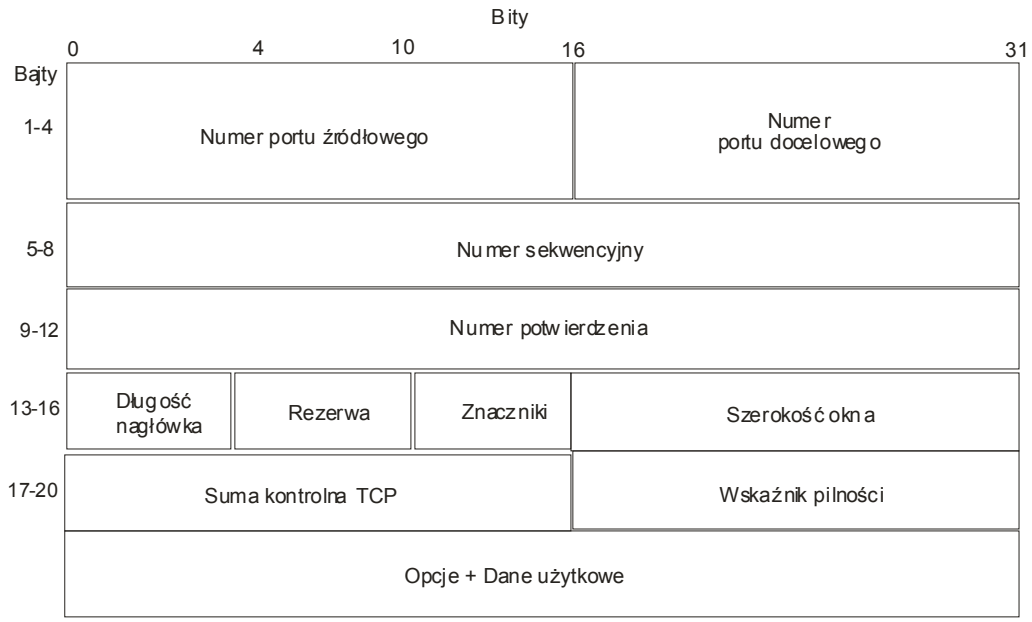
1. **Zorientowanie na połączenie** – TCP zapewnia usługę zorientowaną połączeniowo, w której program użytkowy musi najpierw poprosić o połączenie do odbiorcy, a następnie używać go do przesyłania danych.
2. **Komunikacja punkt-do-punktu** – każde połączenie TCP ma dokładnie dwa końce.
3. **Pełna niezawodność** – TCP gwarantuje, że dane wysyłane połączeniem będą dostarczone dokładnie tak, jak były wysyłane, beż żadnych braków czy dostarczaniu dokładnie tak, jak były wysyłane, bez żadnych braków czy dostarczania nie w kolejności.
4. **Komunikacja w pełni dwukierunkowa** – połączenie TCP pozwala, aby dane przepływały w każdym kierunku, a każdy program może wysyłać dane w dowolnym momencie. TCP umożliwia buforowanie wychodzących i przychodzących danych z obu kierunków – pozwala to programom na wysyłanie danych i dalsze wykonywanie obliczeń,

podczas gdy te informacje są przesyłane.

5. **Interfejs strumieniowy** – oprogramowanie TCP zapewnia interfejs sieciowy, w którym program wysyła połączeniem ciągłą sekwencję oktetów. Oznacza to, że TCP nie udostępnia pojęcia rekordu ani nie gwarantuje, że dane zostaną dostarczone do programu odbierającego w kawałkach tego samego rozmiaru, co wysyłane przez program nadający.
6. **Brak strukturalizacji strumienia** – usługa przesyłania za pomocą strumieni TCP/IP nie uwzględnia strukturalizacji strumienia danych. Programy użytkowe wykorzystujące usługi przesyłania za pomocą strumieni muszą interpretować zawartość strumienia i jeszcze przed rozpoczęciem połączenia zgadzać się na format strumienia.
7. **Niezawodne kończenie połączenia** – protokół TCP wymaga, aby obie strony połączenia się na nie zgodziły; duplikaty pakietów używanych w poprzednich połączeniach nie będą wyglądały jak prawidłowe odpowiedzi ani też w inny sposób nie będą zakłócać nowego połączenia.
8. **Łagodne kończenie połączenia** – program użytkowy może otworzyć połączenie, wysłać dowolną ilość danych, a następnie poprosić, aby połączenie zostało zamknięte. Protokół ten gwarantuje niezawodne dostarczenie wszystkich danych przed zamknięciem połączenia [7].

5.2 Sterowanie przepływem TCP

Nagłówek TCP, którego długość musi być wielokrotnością 32 bitów. Jego postać została przedstawiona na rysunku (Rys.6).



Rys.6. Budowa nagłówka segmentu TCP

Zawiera on następujące pola:

Numer portu źródłowego i numer portu docelowego – podają numery portów procesów aplikacyjnych wykonywanych w segmentach końcowych host.

Numer sekwencyjny SN *(ang. Sequensc Number)* ***i numer potwierdzenia ACK*** *(ang. Acknowledgement Number)* – są wykorzystywane do sterowania przepływem oraz do usuwania błędów transmisji.

Szerokość okna WND *(ang. Window)* – umożliwia przystosowanie transmisji do warunków sieci: większy natłok – mniejsza szybkość transmisji. W to pole urządzenie odbiorcze wpisuje liczbę bajtów danych, które są w stanie przyjąć jego bufory. Jeśli wpisze zero – nadawca musi

przerwać nadawanie. Wznowienie transmisji nastąpi wtedy, gdy odbiorca wpisze liczbę większą od zera.
Długość nagłówka – zawiera liczbę całkowitą, która określa długość nagłówka segmentu mierzoną w wielokrotnościach 32 bitów.
Rezerwa – pole to jest pozostawione do wykorzystania w przyszłości.
Suma kontrolna TCP – zawiera 16-bitową liczbę całkowitą służącą do sprawdzenia, czy dane i nagłówek TCP nie zostały naruszone.
Wskaźnik pilności – zaznacza czy przy transmisji danych w segmencie czy są one pilne.
Prawidłowe ustawienie sterowania przepływem TCP zależy nie tylko od zarządzania oknem, ale również od implementacji następujących zasad:

- **Zasada nadawania** – określa sposób tworzenia nadawanego segmentu TCP. Może on być tworzony dla każdego bloku danych dostarczonego z poziomu aplikacji, ale moduł TCP może również zbierać dane użytkownika w większą całość.
- **Zasada przyjmowania** – dotyczy wyłącznie tych przypadków, gdy segmenty TCP pojawiają się poza kolejnością. Możemy usuwać wszystkie segmenty poza kolejnością, co wiąże się ze zwiększonym obciążeniem sieci. Może przyjmować także poprawne segmenty poza kolejnością, ale pod warunkiem znacznego skomplikowania procedur obsługujących bufor odbiorczy.
- **Zasada dostarczania** – jest zwierciadlanym odbiciem zasady nadawania, a dotyczy sposobu oddawania danych użytkowych z poziomu TCP do aplikacji.
- **Zasada retransmisji** – definiuje sposób powtórnego nadawania segmentów, dla których upłynęła już kontrolowana zwłoka czasowa – czyli czas na potwierdzenie odbioru. Wyróżniamy:
  - zasadę indywidualnych retransmisji – wiąże z każdym segmentem zegar odmierzający zwłokę czasową.
  - zasadę grupowej retransmisji – wiąże jeden zegar z całą kolejką segmentów oczekujących na potwierdzenia. Potwierdzone segmenty są usuwane, zegar kasowany i dla pozostałej kolejki uruchamiany ponownie.
- **Zasada potwierdzania** – określa czas wysyłania potwierdzenia odebranych segmentów. Możliwe są dwa przeciwne warianty: potwierdzenie bezzwłoczne i potwierdzenie zwłoczne [3].

5.3 Uzyskiwanie niezawodności

Protokół transportowy, aby zapewnić niezawodność musi być opracowany z dużą starannością. Główne problemy to: niepewność dostarczania za pomocą bazowego systemu komunikacyjnego oraz restarty komputerów.
Aby to pojąć weźmy pod uwagę dwa programy użytkowe, które tworzą połączenie TCP, komunikują się, zamykają je, a następnie tworzą nowe połączenie. Istnieje możliwość, że każdy komunikat może zostać zgubiony, zduplikowany, dostarczony z opóźnieniem lub nie w kolejności itp. Dlatego komunikaty muszą być niedwuznaczne – inaczej protokół będzie akceptował zduplikowane komunikaty ze starego połączenia i pozwalał, aby zakłócały nowe.
Druga sytuacja problemowa to: dwa programy użytkownika ustanawiają połączenie, a następnie jeden z komputerów zostaje restartowany. Pomimo, że oprogramowanie protokołu na komputerze, który został restartowany nie ma informacji o wcześniejszym połączeniu, oprogramowanie na komputerze działającym non stop uważa połączenie za działające. Dlatego bardzo ważne jest, aby protokół umiał odrzucać pakiety sprzed restartu.

5.4 Gubienie pakietów i retransmisja

Odpowiedź na pytanie, w jaki sposób za pomocą protokołu TCP osiąga się niezawodność jest złożona. W tym celu wykorzystuje się różne metody. Jedną z najważniejszych jest retransmisja, której zasada po krótce została opisana wyżej. Schemat retransmisji w protokole TCP jest

kluczem do sukcesu tego protokołu, gdyż protokół obsługuje komunikację przez dowolną intersieć oraz pozwala na jednoczesne komunikowanie się wielu programów użytkowych. Oprogramowanie TCP musi być gotowe do retransmisji dowolnego komunikatu zgubionego na dowolnym z połączeń. Jak długo jednak TCP powinno czekać przed wykonaniem retransmisji? Czas potwierdzenia z komputera w sieci lokalnej będą przychodzić w czasie kilku milisekund. Zbyt długi czekanie na potwierdzenie powoduje zatrzymanie sieci i nie pozwala na maksymalizowanie przepływu. Dlatego w sieci lokalnej nie powinno się zbyt długo czekać z retransmisją. Natomiast retransmisja kilku milisekundowa w połączeniach satelitarnych nie jest odpowiednia, ponieważ niepotrzebny ruch powoduje marnotrawienie przepustowości i zmniejszenie przepływu.
Opóźnienie potrzebne, aby dane osiągnęły odbiorcę i powróciło potwierdzenie, zależy zarówno od ruchu w intersieci, jak i od odległości od odbiorcy. Ponieważ protokół TCP pozwala na komunikowanie się wielu programów użytkowych z wieloma odbiorcami naraz i ponieważ warunki ruchu wpływają na opóźnienia, oprogramowanie TCP musi obsługiwać różne, mogące się gwałtownie zmieniać opóźnienia.

5.4.1 Retransmisja z adaptacją

Przyjęcie stałej wartości czasu retransmisji w wypadku intersieci nie jest dobre. Dlatego przyjęto, że retransmisja w protokole TCP będzie *retransmisją z adaptacją*. Oznacza to, że TCP dla każdego połączenia śledzi aktualne opóźnienie i dostosowuje (zmienia) czas retransmisji, aby zaadoptować się do zmieniającej się sytuacji.
W rzeczywistości TCP nie może w każdym momencie znać dokładnych opóźnień dla wszystkich części intersieci. Zamiast tego szacuje się dla każdego aktywnego połączenia *opóźnienie przy podróży w obie strony*, mierząc czas potrzebny do uzyskania odpowiedzi. Oprogramowanie TCP, wysyłając komunikat, na który spodziewa się odpowiedzi, zapisuje czas jego wysłania. Gdy przybywa odpowiedź, odejmuje od czasu aktualnego czas wysłania komunikatu i otrzymuje nowe przybliżenie czasu podróży w obie strony dla danego połączenia.
Retransmisja z adaptacją w protokole TCP działa odpowiednio. Wykorzystując wariancję, oprogramowanie TCP może szybko reagować, gdy zwiększa się opóźnienie w związku z zalewem pakietów. Użycie średniej ważonej pomaga TCP ustawić zegar retransmisji, gdy opóźnienie wraca do mniejszej wartości po chwilowym zalewie. Gdy opóźnienie pozostaje stałe, TCP zmienia czas retransmisji na wartość odrobinę większą niż średnia czasu podróży w obie strony. Gdy opóźnienia zaczynają się zmieniać, TCP poprawia czas retransmisji na wartość większą niż średnia, dostosowując ją do nagłych skoków.

5.5 Kontrola przeciążenia

Jednym z najbardziej interesujących aspektów TCP jest mechanizm *kontroli przeciążenia*. W większości nowoczesnych intersieci utrata pakietu jest częściej spowodowana przeciążeniem niż usterką sprzętu. *Przeciążenie* to sytuacja, gdy powstały znaczące opóźnienia spowodowane natłokiem datagramów co najmniej jednym punkcie wymiany pakietów. Protokoły z retransmisją mogą dodatkowo pogorszyć jeszcze sytuację, wpuszczając do sieci dodatkowe kopie komunikatu. Jeśli przeciążenie powoduje nadmiarową retransmisję, cały system może osiągnąć analogiczny do korka na drodze stan ***zapaści z powodu przeciążenia***. Aby uniknąć tego problemu, protokół TCP jako miary przeciążenia używa ilości straconych pakietów i odpowiada na nie, zmniejszając szybkość retransmisji danych.
Gdy zgubiony zostaje komunikat, oprogramowanie TCP zaczyna procedurę kontroli przeciążenia. Zamiast retransmitować tyle danych, aby zapełnić bufor odbiorcy, wysyła pojedynczy komunikat z danymi. Gdy potwierdzenie przybędzie bez dodatkowej utraty danych, TCP podwaja ilość wysłanych danych i przesyła dwa kolejne komunikaty. Jeśli przybędą oba potwierdzenia, wysyła następne cztery itd. Taki wykładniczy wzrost trwa, aż do chwili, gdy TCP zacznie wysyłać połowę oferowanego przez odbiorcę okna. Wtedy następuje zwolnienie

tempa wzrostu.

Schemat ten zachowuje się dobrze przy zwiększonym ruchu w intersieci. Oprogramowanie TCP dzięki szybkiemu zmniejszaniu tempa wysyłania potrafi zmniejszać przeciążenie. Co ważniejsze, ponieważ unika ono retransmisji w przeciążonej intersieci, jego schemat kontroli przeciążenia pomaga w zabezpieczaniu przed powodowaną tym zjawiskiem zapaścią [5].

5.6 Podsumowanie

Protokół kontroli transmisji TCP zapewnia programom użytkowym niezawodną, obejmującą kontrolę przepływu, w pełni dwukierunkową, strumieniową usługę transportową. TCP gwarantuje dostarczenie danych po kolei i bez duplikowania. Oprogramowanie TCP na jednym komputerze komunikuje się z oprogramowaniem TCP na drugim, wymieniając komunikaty. Wszystkie takie komunikaty mają format segmentu TCP niezależnie od tego, jakie są to komunikaty. Każdy segment TCP wędruje w datagramie IP.

Oprogramowanie TCP dla zapewnienia niezawodnej obsługi używa całej gamy mechanizmów. Każdy segment jest opatrzony sumą kontrolną, a wszystkie zgubione komunikaty są retransmitowane. Aby protokół TCP mógł być używany w intersieci, w której opóźnienia zmieniają się wraz z czasem, czasy oczekiwania muszą być adaptowalne. W stosowanej przez TCP metodzie retransmisji z adaptacją mierzy się, oddzielnie dla każdego połączenia, aktualny czas trwania podróży w obie strony i wykorzystuje się go do obierania czasu oczekiwania na retransmisję.

LITERATURA

[1] Boczyński Tomasz – praca zbiorowa „Vademecum teleinformatyka II" IDG Poland S.A. Warszawa 2002

[2] Simmonds Andrew „Wprowadzenie do transmisji danych" WKŁ Warszawa 1999

[3] Papier Zdzisław „Ruch telekomunikacyjny i przeciążenia sieci pakietowych" WKŁ Warszawa 2001

[4] Leinwand Allan, Pinsky Bruce „Konfiguracja Routerów Cisco – podstawy, wydanie drugie" Wydawnictwo MIKOM Warszawa 2002

[5] Duglas E. Comer „Sieci komputerowe i intersieci" WNT Warszawa 2000

[6] Stevens W. Richard „UNIX: programowanie usług sieciowych, tom 1- API: gniazda i XTI" WNT Warszawa 2002

[7] Duglas E. Comer „Sieci komputerowe TCP/IP. Zasady, protokoły i architektura" WNT Warszawa 1997

[8] Hallberg Bruce „Sieci komputerowe, kurs podstawowy" Wydawnictwo „Edition 2000" Kraków 2001

[9] Zieliński Krzysztof „Ćwiczenia do laboratorium sieci komputerowych" AGH Uczelniane Wydawnictwo Naukowo-Dydaktyczne Kraków 1999

[10] Chustecki Janusz – praca zbiorowa „Vademecum teleinformatyka" IDG Poland S.A. Warszawa 1999