Wstęp do sieci komputerowych

Piotr Fulmański[1]

26 stycznia 2007

[1]E-mail: fulmanp@math.uni.lodz.pl

Spis treści

Spis treści **3**

1 Co to jest sieć mamo? **5**
1.1 Rola sieci komputerowej 5
1.2 Charakterystyka sieci 6
1.2.1 Zasięg sieci 7
1.2.2 Topologia sieci 8
1.2.3 Architektura sieci 9
1.2.4 Sposób transmisji 13
1.2.5 Technologie przesyłowe 13
1.3 Pewne pojęcia 14
1.3.1 Pakiety i ramki 14
1.3.2 Protokół 14
1.3.3 Interfejs 16
1.4 Komu potrzebny jest stos? 16

2 ISO/OSI vs. TCP/IP **19**
2.1 ISO/OSI 19
2.1.1 Warstwa fizyczna 20
2.1.2 Warstwa łącza danych 20
2.1.3 Warstwa sieci 20
2.1.4 Warstwa transportu 21
2.1.5 Warstwa sesji 21
2.1.6 Warstwa prezentacji 21
2.1.7 Warstwa aplikacji 22
2.2 TCP/IP 22
2.2.1 Warstwa host-sieć 22
2.2.2 Warstwa internetu 22
2.2.3 Warstwa transportu 23

2.2.4 Warstwa aplikacji 23
2.3 Porównanie 23

3 Warstwa łącza danych 25
3.1 Nagłówek Ethernet-u 25

4 Protokół IP 27
4.1 Nagłówek IP 28
4.2 Adresy IP 32
4.2.1 Adresowanie klasowe 32
4.2.2 Notacja dziesiętna z kropką 33
4.2.3 Adresy IP specjalnego przeznaczenia 34
4.2.4 Maska sieci i podsieci 35
4.2.5 Translacja adresów sieciowych (NAT) 37
4.2.6 Bezklasowy ruting międzydomenowy (CIDR) 41
4.3 IPv6 43
4.3.1 Nagłówek podstawowy 43
4.3.2 Nagłówki dodatkowe 46

5 TCP 49
5.1 Nagłówek TCP 50

6 Przykład 55
6.1 Analiza przykładowego pakietu 55
6.1.1 Nagłówek Ethernetu 55
6.1.2 Nagłówek IP 56
6.1.3 Nagłówek TCP 57
6.1.4 Dane 58

7 Ćwiczenia 61
7.1 Zestaw 1 61
7.2 Zestaw 2 63

Bibliografia 69

Rozdział 1

Co to jest sieć mamo?

1.1 Rola sieci komputerowej

Jedną z najcenniejszych rzeczy w dzisiejszym świecie jest informacja. Skonstruowanie pierwszych komputerów w połowie XX w. pozwoliło na niespotykane przyspieszenie przetwarzania danych. Komputery, najpierw powoli, głównie za sprawą swojej ceny i skomplikowanej obsługi, z czasem, coraz szybciej, znajdowały zastosowanie w najprzeróżniejszych dziedzinach naszej egzystencji. Wraz ze wzrostem ich liczby, problemy zaczęła sprawiać wymiana danych pomiędzy nimi. Coraz więcej urządzeń musiało dysponować bądź takimi samymi danymi, bądź pojawiały się komputery, które posiadały informację użyteczną dla innych. W tej sytuacji potrzebne dane trzeba było wciąż kopiować na taśmę lub dyskietkę, przenosić i wgrywać w nowym miejscu. Rozwiązanie takie nie należało do najbezpieczniejszych, głównie za sprawą podatności na utratę zapisanych danych i przypadkowe zniszczenia fizyczne, a także potencjalną możliwość uzyskania do nich dostępu przez osoby do tego nieuprawnione. Ponieważ zarówno szybkość przetwarzania, jak i możliwe pola zastosowań ulegały ciągłemu zwiększaniu, aktualizacje zgromadzonych i przetwarzanych informacji musiały być coraz częstsze. W tej sytuacji najefektywniejszą metodą, zarówno pod względem szybkości, jak i niezawodności przenoszenia danych, stało się połączenie komputerów za pomocą przewodu. Z czasem, mimo związanych z tym kosztów i wzrostu stopnia komplikacji, łączono ze sobą coraz więcej systemów, gdyż korzyści, jakie dzięki temu osiągano, przeważały poniesione nakłady.

Dzięki takiemu rozwiązaniu stało się możliwe współużytkowanie, a raczej współdzielenie danych; wystarczyło, że składowane były one na jednym komputerze, a pozostałe, jeśli ich potrzebowały, miały możliwość szybkiego

ich otrzymania, dzięki istnieniu bezpośredniego, pracującego bez ingerencji człowieka, systemu ich przekazywania. Ponieważ w takiej sytuacji informacje przechowywane były w jednym miejscu, zatem ekonomicznie uzasadnione stało się konstruowanie wyspecjalizowanych systemów przeznaczonych do ich zarządzania, kontrolowania i ochrony – powstają bazy danych, serwery FTP, serwery WWW, serwery poczty itd.

Kolejne ważne zagadnienie to zwiększenie niezawodności i funkcjonalnej dostępności. W sytuacji, gdy wiele komputerów połączonych jest ze sobą i nastąpi awaria jednego lub nawet kilku z nich, ich funkcje przejmują automatycznie jednostki sprawne. Jeśli z kolei użytkownik prosi o pewne dane, znajdujące się na kilku komputerach, to otrzyma je od tego, który w danym momencie może zrobić to najszybciej, skracając tym samym czas dostępu do danych.

Poniżej wymieniamy kilka najczęściej spotykanych zastosowań sieci komputerowych:

zastosowania w biznesie
- współużytkowanie zasobów (drukarki, dyski, informacje);
- środek łączności;
- prowadzenia interesów z innymi firmami, klientami;
- handel elektroniczny.

zastosowania w nauce
- dostęp do informacji (bazy artykułów, wyniki badań);
- szybki środek komunikacja z innymi osobami;

zastosowania domowe
- dostęp do informacji i komunikacja z innymi osobami (gazety on-line, grupy dyskusyjne, chaty);
- rozrywka interkatywna;
- handel elektroniczny.

1.2 Charakterystyka sieci

Charakteryzując fizyczną strukturę jakiekolwiek sieć komputerowej podajemy zwykle jej **zasięg** i **topologię**.

1.2.1 Zasięg sieci

Pod względem zasięgu zasadniczo wyróżniamy sieci typu LAN, WAN i MAN, które pokrótce można scharakteryzować w następujący sposób:

LAN – (ang. *local area network*) sieć lokalna, łącząca ze sobą urządzenia znajdujące się w „niewielkiej" odległości. Termin „niewielkiej" równie dobrze może oznaczać zarówno 1 metr (gdy łączymy się z komputerem kolegi stojącym na biurku obok), jak i 200 metrów, gdy łączymy się z komputerem znajdującym się w sąsiednim budynku. Zawsze pozostanie to jednak ograniczony obszar, niewielki (wręcz punktowy) w skali kraju (czy świata).

WAN – (ang. *wide area network*) sieć rozległa. Cechami charakterystycznymi są: dużo większy zasięg niż sieci LAN oraz skalowalność rozwiązań. Sieci tego typu muszą umożliwiać rozbudowę, w miarę potrzeb, przy założeniu łączenia wielu węzłów rozmieszczonych w znacznych odległościach od siebie. Węzeł to zarówno jeden komputer, jak i równie dobrze cała sieć LAN. Sieci WAN integrują wiele różnorodnych technologii sieciowych w jeden system umożliwiający wymianę informacji bez względu na występujące różnice sprzętowe, programowe i logiczne.

MAN – (ang. *metropolitan area network*) sieć miejska jest czymś pośrednim między siecią lokalną a rozległą. Jeszcze 10 lat temu wyraźnie można było powiedzieć, która sieć jest siecią miejską. Obecnie jest to trudne, gdyż sieci miejskie są włączane w ogólny szkielet Internetu (sieć rozległą), co powoduje, że stają się jego częścią.

W tym klasycznym podziale sieci uwzględniającym zasięg, można jeszcze uwzględnić dwie pozycje, których pojawienie się to oznaka ewolucji jaką obserwujemy w dziedzinie IT. Tak więc możemy rownież mówić o **sieci osobistej** oraz **internecie**.

Pojawienie się kategorii sieci osobistych to konsekwencja powstawania co raz to większej liczby urządzeń potrafiących wymieniąć informacje z otoczeniem na bardzo małe odległości. Palmtop komunikujący się z odbiornikiem GPS przy pomocy technologii Bluetooth czy notatnik menadżerski synchronizujący dane z komputerem przez złącze IrDA to dzisiaj nic niezwykłego. Zasięg tych urządzeń ogranicza się do ich najbliższego otoczenia (względnie odległość ta jest mierzona w metrach, ale raczej już nie w dziesiatkach metrów).

Odległość	Opis „słowny”	Rodzaj
1 m	Metr kwadratowy	Sieć osobista
10 m 100 m 1000 m	Pomieszczenie Budynek Grupa	Sieć lokalna
10,000 m	Miasto	Sieć miejska
100,000 m 1,000,000 m	Kraj Kontynent	Sieć rozległa
10,000,000 m	Planeta	Internet

Tablica 1.1. Klasyfikację sieci pod względem zasięgu

Mianem internetu[1] (lub sieci złożonej) nazwiemy natomiast nie tyle konkretną sieć co cały ich zbiór. Ważne jest, że w tym przypadku w skład rozważanej sieci wchodzą różne sieci (w sensie budowy, logiki działania, używanych protokołów). Niezbędną w tym przypadku translację sprzętową jak i programową (protokoły nie muszą być zgodne, mogą być używane różne napięcia czy ilość sygnałów służących do tansmisji) rezlizują tak zwane bramy (ang. *gateway*).

Ostatecznie klasyfikację sieci pod względem zasięgu można przedstawić tak jak w tabeli 1.1

1.2.2 Topologia sieci

Mówiąc o topologii sieci mamy na myśli jej niejako ułożenie fizyczne, określające sposób połączenia wchodzących w jej skład urządzeń. I tak zasadniczo wyróżnia się trzy topologie występujące w sieciach LAN oraz jedną dodatkową stosowaną w MAN i WAN.

Topologia magistrali – uzyskuje się ją łącząc wszystkie urządzenia za pomocą pojedynczego kabla. Obecnie realizowane jest to jedynie za pomocą kabla koncentrycznego. Z racji dużej awaryjności oraz kłopotliwości w serwisowaniu powoli odchodzi się od tej topologii i technologii z nią związanych.

Topologia pierścienia – charakteryzuje się tworzeniem zamkniętego połączenia. Każdy komputer łączy się ze swoimi dwoma najbliższymi

[1] Zgodnie z ogólnymi standardami, będziemy pisać „internet” mając na myśli dowolną sieć złożoną, natomiast Internet (przez duże „i”) majac na myśli sieć złożoną obejmującą swoim zasięgiem całą Ziemię, z której korzystamy na co dzień.

sąsiadami i tylko z nimi wymienia on bezpośrednio informacje.

Topologia gwiazdy – wszystkie węzły łączą się w jednym punkcie.

Topologia oczek pełnych – tutaj każdy węzeł połączony jest ze wszystkimi pozostałymi. Otrzymujemy w ten sposób najbardziej niezawodną i odporną na uszkodzenia fizyczne konfigurację. Oczywistą wadę takiego rozwiązania stanowi szybki wzrost liczby połączeń, pociągający za sobą wzrost stopnia komplikacji i kosztów wraz ze wzrostem liczby obsługiwanych węzłów.

Topologia mieszana – stanowi dowolne połączenie powyższych rozwiązań; jest ona dość często stosowana. Przykładowo w firmie istnieje topologia magistrali. Zarząd nie chce podjąć decyzji o przebudowie istniejącej infrastruktury, natomiast jednocześnie chce podłączyć kolejne komputery. Wówczas nowe maszyny, ze względu na zmiany w obowiązującej technologii, mogą tworzyć topologię gwiazdy.

Mówiąc o topologii pamiętajmy, aby nie mylić **organizacji fizycznej** i **logicznej sieci**. *Fizyczna organizacja* to właśnie powyższe modele, które można zobaczyć gołym okiem albo dotknąć, jeśli ktoś będzie miał takie życzenie. *Organizacja logiczna* wykorzystuje natomiast fizycznie istniejące połączenia do zorganizowania przesyłania danych. Organizacja ta jest całkiem niezależna od organizacji fizycznej. Dlatego nic nie stoi na przeszkodzie, aby istniała sieć o fizycznej strukturze gwiazdy działająca na (logicznej) zasadzie pierścienia.

1.2.3 Architektura sieci

Kolejną z cech, jaką należy zasygnalizować jest **architektura sieci**. Niesie ona ze sobą informację o konfiguracji urządzeń podłączanych do jednej z wcześniej omawianych topologii. Tu wymienia się głównie architekturę **klient–serwer** oraz **równorzędną**. W celu lepszego przybliżenia powyższych idei, wprowadzimy wpierw terminy **klient** i **serwer**.

Serwerem nazywany jest dowolny komputer przyłączony do sieci i udostępniający swoje zasoby innym urządzeniom. Poprzez zasób rozumiemy tutaj zarówno dysk, pamięć, jak i moc obliczeniową. Na ogół sposób pracy i konstrukcja fizyczna serwera zoptymalizowane są pod kątem wykonywania określonej funkcji. Nie zapominajmy jednak, iż „serwer” **nie musi** oznaczać osobnego komputera; kilka serwerów może funkcjonować w ramach jednej

fizycznej maszyny, wtedy też mówi się nie o serwerze, a **usłudze** lub **usługach**, jakie udostępnia. Ze względu na sposób pracy wyróżnia się zwykle:

Serwer plików służy do składowania plików i ich udostępniania użytkownikom sieci. Mechanizm centralnego przechowywania plików daje łatwiejszą kontrolę nad tworzonymi dokumentami. Teraz każdy, zamiast „osobistej" wersji pewnych plików przechowywanych na swoim komputerze, korzysta z tych, zapisanych na serwerze. Gdy nastąpi jego zmiana, każdy przy następnym jego pobraniu automatycznie otrzyma uaktualnioną wersję. Uwalnia nas to od konieczności przeglądania wszystkich potencjalnych miejsc przechowywania i ciągłego kontrolowania zachodzących zmian. Kolejną niezaprzeczalną zaletą jest uproszczenie tworzenia kopii zapasowych, które są niezbędne do zapewnienia bezpieczeństwa w razie awarii. Dość łatwo można wykonać kopię zasobów jednego komputera (tego, na którym wszyscy przechowują pliki), podczas gdy dla większej ich liczby wymagany jest większy nakład pracy, sprawna organizacja i co najważniejsze – pilnowanie użytkowników, by w ogóle wykonywali te kopie.

Serwer aplikacji jest miejscem wykonywania (uruchamiania) programów wykonywalnych. Jego istotą jest właśnie to, że programy uruchamiane są na tymże serwerze, a nie na komputerze użytkownika, co miałoby miejsce, gdyby były przechowywane na serwerze plików. Innymi słowy serwer aplikacji udostępnia swoje zasoby poprzez bezpośrednie wykonywanie zainstalowanych na nim programów na rzecz odległego klienta. Stosowanie serwerów aplikacji daje sposobność wygodnego kontrolowania używanego oprogramowania.

Serwer wydruków. Pomimo, iż żyjemy w epoce komputerów i cyfrowej informacji, nadal często zachodzi potrzeba sporządzenia dokumentów w postaci zadrukowanej kartki papieru. W tym celu należy zapewnić użytkownikom dostęp do odpowiednich urządzeń. Indywidualna drukarka dla każdego, to rozwiązanie nieuzasadnione ekonomicznie. Zwykle liczba drukowanych stron to kilka, kilkadziesiąt miesięcznie. W takiej sytuacji dużo lepiej zainwestować w serwer wydruków, który wszystkim użytkownikom sieci udostępnia podłączone do niego drukarki, których jest znacznie mniej niż potencjalnych użytkowników. Przyjmuje on zlecenia druku od wszystkich urządzeń w sieci, ustawia je w kolejkę i kieruje do odpowiedniej drukarki. Dzięki temu 50 komputerów może korzystać z 2 drukarek – praktycznie w taki sam sposób,

jakby stały one tuż obok każdego stanowiska. Jedyna niedogodność, to konieczność „przespacerowania się" po gotowe wydruki do miejsca, gdzie stoją drukarki.

Serwer bazy danych jest miejscem przechowywania danych. Istotnym czynnikiem odróżniającym go od serwera plików, na którym również można składować dane, jest zdolność do zarządzania, kontroli i przetwarzania przechowywanych informacji. Korzystając z serwera plików wysyłamy do niego lub otrzymujemy plik, którego zawartości nadajemy znaczenie dopiero my sami. Natomiast między użytkownikiem a serwerem baz danych przesyłana jest tylko konkretna informacja. Załóżmy, że zapisujemy codziennie wielkość naszego majątku oraz ile i na co wydaliśmy nasze zasoby pieniężne. Po pewnym czasie chcąc dowiedzieć się, kiedy wydaliśmy więcej niż 100 złotych i korzystając z serwera plików pobieramy plik z wydatkami i przeglądamy go ręcznie w poszukiwaniu żądanych pozycji. Korzystając natomiast z usług serwera baz danych, wysyłamy do niego zapytanie, które nieformalnie może brzmieć: „Podaj mi te dane, w których $dziennywydatek > 100$". Serwer odpowiada na nasze zapytanie, przesyłając nam tylko poszukiwane informacje.

Serwer poczty umożliwia użytkownikom wymianę informacji na zasadach zbliżonych do funkcjonowania poczty tradycyjnej. Serwer udostępnia swoje usługi w postaci wysyłania lub odbierania wiadomości z określonego miejsca sieci oraz jej przechowywania. Wysyłając do kogoś list elektroniczny podajemy adres według wzoru:

```
nazwa_użytkownika@serwer.poczty
```

Część po znaku `@` określa miejsce w sieci, gdzie list ten zostanie przesłany (jest to nazwa fizycznego komputera – hosta), natomiast pozostała informuje, do której „skrzynki" należy go przekazać, czyli kto jest jego adresatem.

Serwer WWW , udostępniając swoje usługi, pozwala na tworzenie ogólnodostępnych, interaktywnych, niezależnych od platformy sprzętowej, hipertekstowych dokumentów. Utożsamiany często z samym Internetem, faktycznie wraz z językiem HTML (HyperText Markup Language), przyczynił się znacznie do jego rozwoju.

Serwer. . . „czegoś tam” . Powyżej wymieniliśmy tylko najczęściej spotykane serwery. W zasadzie każde urządzenie przyłączone do sieci i wykonujące jakąś usługę na rzecz innych maszyn nazywamy serwerem.

Klientem nazywamy urządzenie korzystające z usług serwera za pomocą sieci. Ważne jest to, że każdy komputer może być zarówno klientem jakiejś usługi, jak i serwerem innej, wszystko to zależy od konfiguracji oprogramowania.

Architektura równorzędna

W sieciach równorzędnych (P2P, ang. *peer-to-peer*), zwanych również sieciami każdy-z-każdym, nie występuje tzw. dedykowany serwer. Oznacza to, iż każde z urządzeń może pełnić funkcję zarówno klienta, jak i serwera. Sieci tego typu zbudować można w oparciu o dowolną topologię, co więcej nie ma konieczności stosowania specjalnych środowisk pracy czy systemów operacyjnych. Dzięki temu ich koszt i niezbędny nakład pracy związany z instalacją zredukowane są do minimum. Praktycznie w oparciu o każdy dzisiejszy komputer (a raczej ich zbiór) można zbudować sieci tego typu bez większego nakładu pracy i środków. Trzeba jednak przy tym pamiętać o pewnych zagrożeniach i ograniczeniach wynikających z takiej struktury. Bezpieczeństwo w tego typu systemach zależy od każdego z użytkowników. Wszyscy muszą przestrzegać pewnych zasad i procedur postępowania. Niewiele warte staną się nawet najwymyślniejsze hasła użytkowników, gdy jeden z nich ujawni (celowo lub przez przypadek) swoje. Ponadto współpraca kilkunastu osób wymagać może ciągłej wymiany pewnych dokumentów. Problemem staje się wówczas zapewnienie dostępu do ich najaktualniejszej wersji. Zwykle poszukiwany plik będzie znajdował się na kilku maszynach, co zmusi nas do ich przejrzenia i wybrania tego właściwego. Co więcej, może się okazać, że komputer z najaktualniejszą wersją właśnie został wyłączony, gdyż jego użytkownik postanowił właśnie zakończyć pracę i zapomniał poinformować o tym fakcie pozostałych.

Ze względu na wspomniane cechy, sieci typu każdy-z-każdym idealnie nadają się dla małych firm czy grup roboczych wchodzących w skład większych organizacji. Doskonale powinny zdać także egzamin w przypadku „sąsiedzkiego” połączenia w jednym bloku mieszkalnym czy odcinku ulicy.

Architektura klient–serwer

Ograniczenia i problemy związane z bezpieczeństwem i administracją w sieciach równorzędnych obejść można stosując architekturę klient–serwer.

Jedną z kluczowych cech w tej architekturze jest centralne zarządzanie, dzięki któremu ulega znacznej poprawie bezpieczeństwo danych. Jeśli zachodzi potrzeba stworzenia np. kopii zapasowej, dotyczy to tylko serwera i zadanie to można powierzyć jednej osobie lub wręcz zautomatyzować. Nie trzeba obarczać indywidualnych użytkowników, którzy najczęściej nie rozumieją potrzeby tworzenia kopii bezpieczeństwa dopóki nie nastąpi jakaś awaria i utrata danych. Konsekwencją centralnej weryfikacji tożsamości jest znacznie sprawniej działający mechanizm kontroli uprawnień nadawanych osobom korzystającym z sieci. W sieciach klient–serwer jedynie serwer odpowiada za przetwarzanie zapytań klientów. Odciąża to znacznie poszczególne komputery klienckie. W takiej sytuacji wydajność sieci zależy często od wydajności serwera. Zwykle jest to wyspecjalizowany system komputerowy zoptymalizowany pod kątem pełnienia pewnej określonej funkcji i jego wydajność zapewnia właściwe działanie sieci. Trzeba wyraźnie zaznaczyć, że serwer pozostaje jednak najbardziej newralgicznym ogniwem sieci. Jego awaria powoduje zatrzymanie pracy wszystkich użytkowników. Dlatego też przykłada się wielką wagę do zapewnienia ciągłej i nieprzerwanej pracy serwerów, czyniąc je tym samym urządzeniami skomplikowanymi i niestety drogimi. Dodatkowe nakłady finansowe związane są z koniecznością stosowania specjalnego oprogramowania nadzorującego pracę sieci, a także samo jej wdrażanie i codzienna obsługa. Właśnie ze względu na czynnik ekonomiczny stosowanie sieci klient–serwer wydaje się ekonomicznie uzasadnione w przypadku dużych instytucji, gdzie wymaga się zwiększonego bezpieczeństwa oraz jednolitego zarządzania i korzystania z zasobów sieci.

1.2.4 Sposób transmisji

Ze względu na sposób transmisji dzielić będziemy sieci na połączeniowe i bezpołączeniowe.

Transmisja połączeniowe gwarantuje jakość usług.

Używając transmisji bezpołączeniowej możemy osiągnąc za to większą wydajność niż w sieciach połączeniowych. Możemy osiągnąć, gdyż

1.2.5 Technologie przesyłowe

Sieci rozgłoszeniowe

Sieci typu dwupunktowego (ang. *point-to-point*)

Zwykle technologie rozgłoszeniowe wykorzystywane są w „małych” sieciach, natomiast wieksze wykorzystują połączenia dwupunktowe.

1.3 Pewne pojęcia

1.3.1 Pakiety i ramki

Zazwyczaj dane przesyłane przez sieć nie stanowią ciągłego strumienia bitów. Dane przeznaczone do wysłania dzielone są na pewne „porcje" zwane **pakietami** i dopiero wtedy transmitowane. Są dwa główne powody stosowania pakietów. Po pierwsze, wspierają obsługę błędów transmisji, gdyż dużo łatwiej jest powtórzyć transmisję niewielkiego bloku danych niż np. kilkumegowego pliku. Po drugie, umożliwiają prawie równoległą transmisję danych pochodzących od wielu nadawców za pomocą jednego ośrodka. Łatwo wyobrazić sobie sytuację, gdy jeden komputer przesyłający plik o rozmiarze 100 MB uniemożliwia pracę pozostałym użytkownikom sieci. Dzięki stosowaniu pakietów, każdy w pewnym fragmencie przydzielonego czasu może nadać małą porcję danych (właśnie pakiet), co chroni przed zablokowaniem dostępu dla pozostałych komputerów.

Ramką będziemy nazywali ciąg bitów o ustalonym znaczeniu i kolejności w pakiecie. Ogólny schemat ramki przedstawia rys. tutu. Jak widać oprócz danych użytkownika przesyłane są także pewne dodatkowe informacje, są nimi np. SOH (ang. *start of header* – początek ramki), oraz EOT (ang. *end of transmission* – koniec ramki). Wadą takiego rozwiązania jest „marnowanie" części pakietu na dodatkowe dane, które nie niosą ze sobą faktycznej informacji. Jednak patrząc na zagadnienie z drugiej strony, stosowanie ramek i większych nagłówków jest konieczne. Zauważmy, że musimy przecież gdzieś umieścić choćby informacje o adresacie przesyłki. Przyda się także numer kolejny przesyłki, pozwalający złożyć z danych transmitowanych w pakiecie oryginalną daną przesłaną przez użytkownika, jeśli zaszłaby konieczność jej podziału. Jeśli zamierzamy potwierdzać otrzymanie przesyłki, to ramka powinna zawierać także adres nadawcy. Należy ponadto wziąć pod uwagę fakt, iż rzadko kiedy transmisja w sieci jest bezproblemowa. To właśnie dzięki nadmiarowym informacjom znajdującym się w nagłówkach możliwe jest zidentyfikowanie i odtworzenie lub retransmisja błędnej informacji.

1.3.2 Protokół

Mianem **protokołu** określamy zbiór pewnych zasad i sposobów postępowania prowadzących do osiągnięcia pewnego celu. W kontekście sieci komputerowych (ale nie tylko), protokołem nazywać będziemy umowę precyzującą w jaki sposób i na jakich zasadach odbywa się komunikacja pomię-

dzy stronami. Zauważmy, że na co dzień spotykamy się z protokołami, nie zdając sobie nawet z tego sprawy. Telefonując do kogoś realizujemy „protokół nawiązania połączenia głosowego z odległym klientem". Gdy nasz rozmówca podniesie słuchawkę, nie mówimy od razu tego, co mamy do przekazania, lecz wpierw „inicjujemy wymianę danych". Zwykle następuje wymiana informacji niezbędnych do nawiązania połączenia w rodzaju:

Przykład 1.1. Prosty protokół nawiązywania połączenia zakończony sukcesem.

Odbiorca: Halo.
Nadawca: Dzień dobry, mówi X.
Odbiorca: Witam serdecznie, w czym mogę pomóc.
Nadawca: Dzwonię w sprawie...
połączenie zostało nawiązane

Przykład 1.2. Prosty protokół nawiązywania połączenia zakończony porażką.

Nadawca: Mówi X. Czy mogę prosić Y?
*Odbiorca: XyCVFR!*_?GHJAA;p[];,*
Nadawca: Bardzo przepraszam.
połączenie nie zostało nawiązane z powodu niezgodności protokołów

Savoir-vivre to nic innego jak zbiór zasad i ogólnie przyjętych norm postępowania w różnych sytuacjach życiowych. Nie obcy jest nam przecież termin *protokół dyplomatyczny*. Ustalenia takie chronią przed niezręczną lub kłopotliwą sytuacją, o co nie trudno w przypadku dyplomatów mających kontakty z ludźmi wychowanymi w różnych kulturach. Podobnie korzystanie z elektronicznych form przekazu wymaga ustalenia znaczenia pewnych sygnałów i kolejności ich wymiany w celu zainicjowania pewnego działania. Fizyczne połączenie jest warunkiem koniecznym, ale niewystarczającym dla pomyślnej realizacji komunikacji sieciowej. Jako przykład szeroko rozpowszechnionych i obecnie najczęściej wykorzystywanych protokołów sieciowych posłużyć może **stos protokołów TCP/IP**.

Rys. 1.1. Zależność pomiędzy warstwą, protokołem i interfejsem

Rys. 1.2. Przykładowy przepływ informacji w kanale wirtualnym i rzeczywistym dla warstwy 5

1.3.3 Interfejs

Interfejs określa zbiór działań jakie możemy wywołać na rzecz pewnego obieku celem osiągnięcia konkretnego efektu. Dobrze zdefiniowany interfejs umożliwia łatwe wykorzstanie możliwości obiektu. tutu przykład z telewizorem

1.4 Komu potrzebny jest stos?

Większość sieci zorganizowana jest w postaci stosu warstw lub też mówiąc inaczej poziomów.

Zadaniem każdej z warstw jest oferowanie określonych usług warstwom wyższym oraz izolowanie ich od szczegółów faktycznej implementacji świadczonych usług (ukrywanie informacji, abstakcyjne typy danych, kapsułkowanie danych, programowanie obiektowe).

Warstwa n na jednej maszynie prowadzi wymiane informacji z warstwą n na drugiej maszynie.

Reguły i konwencje używane podczas tej wymiany informacji nazywane są *protokołem warstwy n*.

Lista protokołów nosi nazwę *stosu protokołów*.

Podczas projektowania warstw należy zwrócić uwagę na:

adresowanie W każdej warstwie potrzebny jest mechanizm identyfikujący nadawcę i odbiorcę. Ponieważ jednocześnie może komunikować się wiele programów znajdujących się na tym samym bądź też na różnych komputerach, zatem niezbędny jest sposób wskazywania które programy ze sobą się komunikują – a więc niezbędny jest adres miejsca źródłowego i miejsca przeznaczenia informacji.

reguły transferu danych Transmisja w jednym czy w dwóch kierunkach? Ile kanałów? Czy i jakie są priorytety kanałów?

kontrolę błędów Jakie kody kontroli i korekcji używać? Sposób informowania o odebraniu / lub nie informacji.

sterowanie przepływem ... w celu zachowania kolejności przesyłanych informacji.

problemy związane z długością wiadomości mechanizmy podziału, transmisji i składania wiadomości... mechanizmy łączenia małych komunikatów w jeden duży

multipleksowanie i demultipleksowanie jedno połączenie dla wielu niezwiązanych komunikacji

trasowanie gdy pomiędzy nadawcą i odbiorcą istnieje kilka różnych ścieżek

O ilości warst i sposobie podziału na nie decydować będą następujące czynniki:

1. Tworzymy warstwy tam, gdzie potrzebna jest osobna abstrakcja.
2. Każda karstwa powinna pełnić dobrze określone i zefiniowane funkcje.
3. Granice warstw powinny zostać wobrane w taki sposób, aby zminimalizować przepływ informacji przez interfejsy.
4. Liczba warstw powinna być na tyle liczna, aby nie gromadzić różnych (różnych co do stosowalności tutu) funkcji w jednej warstwie, i na tyle mała, aby była wygodna w używaniu.

Rozdział 2

ISO/OSI vs. TCP/IP

Po wprowadzeniu podstawowych pojęć związanych z sieciami, możemy przejść do bliższego im przyjrzenia się. Zgodnie z tym co zostało powiedziane we wcześniejszym rozdziale, sieć rozważać będziemy na kilku niezależnych poziomach abstrakcji nazywanych warstwami, głównie dlatego aby móc skupić się na konkretnych zagadnieniach bez konieczności zajmowania się nieistotnymi z danego punktu widzenia zagadnieniami. Aby to jednak uczynić musimy przyjąć jakiś model sieci. W tym rozdziale przyjrzymy się właśnie dwum modelom: modelowi ISO/OSI, oraz modelowi TCP/IP.

2.1 ISO/OSI

Najpierw spróbujmy rozszyfrować nazwę modelu. Skrót **ISO** pochodzi od angielskich słów **International Organization for Standardization** czyli Międzynarodową Organizację Normalizacji[1] OSI to natomiast **Open System Interconnection**, czyli Łączenie Systemów Otwartych. Pojęcie „otwarty" zapewne jest odbiciem idei działania, tj. otwartości i gotowości systemów tworzonych w oparciu o taki model, na komunikację z innymi systemami.

Przejdziemy teraz do omówienia modelu, a więc przyjrzymy się wszystkim warstwom. Zanim do tego przejdziemy pragniemy zwrócić uwagę, iż model ten nie definiuje architektury sieci, nie definiuje konkretnych usług ani protokołów. Model ten raczej mówi co warstwa ma robić a nie jak. Model warstwowy ISO/OSI przedstawiono na rysunku 2.1.

[1] Skrót ten rozwijany jest także jako **International Standards Organization** czyli Organizacji Standardów Międzynarodowych.

Rys. 2.1. Model ISO/OSI

2.1.1 Warstwa fizyczna

Warstwa fizyczna odpowiedzialna jest za przesyłanie bitów; odpowiada jej karta sieciowa lub modem. Na tym poziomie realizowana jest fizyczna transmisja danych bez „kontroli ruchu" i bez uwzgledniania rodzaju informacji. Widzi ona tylko ciag jedynek i zer, nie potrafi nadać im znaczenia ani określić ich wagi. Ciagłość transmisji nie jest zabezpieczona – jeśli medium zostanie zablokowane lub uszkodzone, komunikacja zostanie przerwana. Warstwa ta nie jest tym samym co fizyczny nośnik danych (sygnałów) - przewody nie są częścią warstwy fizycznej i w modelu tym umiejscowione są poniżej warstwy fizycznej[2].

Zagadniena projektowe w tej warstwie wiążą się głównie z **interfejsami mechanicznymi** (np. ilość styków), **elektrycznymi** (np. napięcia reprezentujące 0 i 1), **zależnościami czasowymi** (np. czas trwania sygnału reprezentującego bit) a także **logiką działania** (np. czy transmisja może mieć miejsce jednocześnie w obu kierunkach, sposób nawiązywania i kończenia połączenia).

2.1.2 Warstwa łącza danych

Warstwa łącza danych steruje fizyczną wymianą bitów; **układa bity w ramki**, **sprawdza poprawność danych**, **zarządza warstwą 1**. Odpowiada za końcową zgodność przesyłanych bitów z oryginalnie nadanymi. W większości przypadków obie wymienione warstwy połączone są w jedną całość, tworząc w ten sposób kartę sieciową.

Zasadniczym zadaniem tej warstwy jest **dostarczenie warstwie sieciowej łącza transmisyjnego wolnego od błędów**.

2.1.3 Warstwa sieci

Warstwa sieci zamienia ciag bitów w kanał komunikacyjny: wyznaczając odpowiednią trasę dba o to aby informacje przepływały między odpowiednimi komputerami. Dane wymieniane są w postaci pakietów wysyłanych od nadawcy do odbiorcy, nie jest jednak sprawdzana ich zawartość.

Podstawowe problemy projektowe to:

[2]Mówiąc precyzyjniej – model nie obejmuje fizycznych nośników.

1. Sposób trasowania czyli wybierania drogi jaką mają pokonać pakiety aby przejść od nadawcy do odbiorcy. I tak mamy do czynienia z
   (a) trasowanie statyczne – trasa zapisana jest „na sztywno" i rzadko się zmienia;
   (b) trasowanie dynamiczne – trasa ustalana jest przy każdorazowym nawiązaniu połączenia;
   (c) trasowanie wysoce dynamiczne – trasa wybierana jest osobno dla każdego pakietu aby reagować na bieżące obciążenia w sieci.
2. Jakość świadczonych usług, czyli niedopuszczanie do sytuacji gdy zbyt duża liczba pakietów przesyłana w tym samym czasie powoduje tworzenie wąskiego gardła w sieci.
3. Jak łączyć ze sobą różne sieci? Jest to ważne pytanie, ponieważ różne sieci mogą używać innych sposobów adresowania, może mieć pakiety innej postaci czy też mogą wystąpić różnice w protokole. Warstwa sieciowa powinna izolować nas od tych problemów zapewniając, z punktu widzenia całości, jednolite adresowanie, jednolity format pakietu, jednolite protokoły itd.

2.1.4 Warstwa transportu

Warstwa transportu przesyła wiadomości kanałem stworzonym przez warstwę sici. Dopiero ta warstwa troszczy się o **bezpieczeństwo** i **pewność wymiany danych**. Wszystkie trzy wcześniejsze warstwy nie przykładają żadnej wagi do bezpieczeństwa, skupiając się na zapewnieniu maksymalnej szybkości.

2.1.5 Warstwa sesji

Warstwa sesji zarządza przebiegiem komunikacji podczas połączenia między dwoma komputerami. Przez sesję rozumiemy relację zachodzące podczas współpracy dwóch odległych komputerów. Sesje sterują m.in. wymianą informacji (śledzenie, kto powinien nadawać) czy np. synchronizują transmisje pozwalając na ich wznowienie od pewnego punktu.

2.1.6 Warstwa prezentacji

Warstwa prezentacji zajmuje się przetworzeniem dane dostarczonych z niższych warstw w taki sposób, aby mogły być odebrane przez aplikacje

Rys. 2.2. Modele odniesienia ISO/OSI i TCP/IP

użytkownika. Tutaj dokonywana jest na przykład konwersja, jeśli komputer klient używa innego formatu liczb niż komputer serwer. Można powiedzieć, że warstwa ta zajmuje się przemieszczaniem bitów, dbając o zachowanie odpowiedniej składni i semantyki specyficznej dla danego rodzaju urządzenia.

2.1.7 Warstwa aplikacji

Warstwa aplikacji stanowi interfejs realizujący zapytania aplikacji lub po prostu program komunikacyjny, a więc na przykła przeglądarkę WWW.

2.2 TCP/IP

Podobnie jak poprzednio, zaczniemy od wyjaśnienia znaczenia użytych skrótów. TCP to inaczej **protokół transmisji danych**) (ang. *Transmission Control Protocol*), IP zaś oznacza **protokół intersieci** (*Internet Protocol*). W modelu tym mamy trochę inną strukturę warstw, których są w tym przypadku 4. Zależności pomiędzy warstwami w modelu ISO/OSI i TCP/IP przedstawia rysunke 2.2

2.2.1 Warstwa host-sieć

Cytując [3], s.55: „*Pod warstwą internetową [czyli warstwą nadrzędną w stosunku do warstwy host-sieć] leży wielkie Nic*". Model TCP/IP nie precyzuje zadań tej warstwy, zwracając jedynie uwagę, że powinna ona zapewnić możliwość przyłączenia hosta do sieci tak aby możliwe stało się przesyłanie pakietów IP.

2.2.2 Warstwa internetu

Warstwa ta stanowi niejako fundament działania intersieci (a więc sieci powstałej z połączenia wielu różnych rodzajów sieci). Do zadań tej warstwy należy umożliwienie hostom niezależnego kierowania pakietów do dowlnego celu w dowolnej sieci. Nie dbamy w tym miejscu o kolejność pakietów. Mówiąc inaczje: pakiet ma być dostarczony (nienaruszony); reszta nie jest istotna. Jako model komunikacji wybrano komunikację bezpołączeniową.

Warstwa interntu definiuje konkretny rodzaj format pakietu i protokół o wspomnianej już nazwie IP.

2.2.3 Warstwa transportu

W ramach warstwy transportu zdefiniowan dwaprotokoły: wspomniany TCP oraz UDP (ang. *user datagram protocol*).

TCP jest dwupunktowym protokołem połączeniowym. Zapewnia on niezawodne przesłanie strumienia bajtów od nadawcy do odbiorcy. (tutu dzielenie, sterownie przepływem itp.)

UDP jest dwupunktowym protokołem bezpołączeniowym. Protokół ten z definicji jest protokołem zawodnym. Może jednak być wykorzystany przez inne protokoły, które niezawodność dostarczą. UDP stosowany jest wszędzie tam, gdzie szybkość ważniejsza jest od dokładności (np. strumień muzyczny) czy też w jednokrotnych i krótkich zapytaniach typu klient-serwer.

2.2.4 Warstwa aplikacji

Warstwa ta zawiera protokoły wyższego poziomu jak FTP czy SMTP i funkcjonalnie odpowiada analogicznej warstwie z modelu ISO/OSI.

2.3 Porównanie

Model ISO/OSI wprowadza pojęcia usługi, interfejsu i protokołu. Co wiecej, dba aby były one dobrze zdefiniowane i wyraźnie je rozgrancza. Dzieki temu są one lepiej ukryte i mogą być z łatwością zastępowane przez nowe wraz ze zmianą technologii (nowe protokoły, nowe interfejsy itd.). Niestety w przypadku TCP/IP taka precyzja nie występuje.

Oba modele różnią się sposobem zestawiania komunikacji. Model ISO/OSI obsługuje komunikację połączeniową i bezpołączeniową w warstwie sieciowej, natomiast w transportowej dostępne jest tylko komunikacja połączeniowa. W modelu TCP/IP w warstwie sieciowej (do której użytkownik i tak nie ma dostępuj, więc prawdę mówiąc jest jemu wszystko jedno) mamy komunikację bezpołączeniową, natomiast w warstwie transportowej mamy do wyboru oba tryby.

Porównując oba modele i oceniając je należy mieć na uwadze, czas ich powstania. Model ISO/OSI powstał przed pojawieniem się większości protokołów, przez co nie był (i nadal nie jest) ukierunkowany na żaden konkretny ich zbiór. Teoretycznie uniwersalny, przez co bardzo ogólny. Konsekwencją tego jest to, że projektanci nie mieli doświadczenia i nie wiedzieli zbyt dobrze jaką funkcjonalność umieścić w jakiej warstwie. Dla kontrastu, w modelu TCP/IP najpierw powstały protokoły a model jest po prostu ich opisem.

Niestety, jednak taki model nie pasował do żadnego z już istniejących. Model TCP/IP w żadnym tego słowa znaczeniu nie jest ogólny, przez co nie nadaje się jako wytyczne do tworzenia przyszłych sieci z uwzględnieniem nowych technologii. Ponadto jako jego ewidentną wadę należy wynienić czarną dziure nazywana warstwa host-sieć (w przypadku modelu ISO/OSI taka czarną dziura, choć trochę mniejszą, jest warstwa sesji).

Oderwanie od rzeczywistości to jeden z powodów dla których ISO/OSI nie przyjął się w praktyce. Drugi powód to właśnie praktyka. ISO/OSI pojawia się wtedy gdy funkcjonowało już TCP/IP. Co ważniejsze, funkcjonowało sprawnie i bez żadnych problemów. Po cóż więc zmieniać coś co działa na coś nowego i niepewnego. Poza tym musimy w tym miejscu uświadomić sobie mentalność ludzi odpowiedzialnych w tamtych latach za Informatyke (w szerokim tego słowa znaczeniu). W ogólności byli to „buntownicy", dumnie ze tego co udało im się stworzyć i dbający o to. Wierni pewnym ideałom i niechętni narzucaniu czegoś przez osoby i instytucje nie związane z tematem. Większość oprogramowania i ogólnych rozwiazań tamtych czasów powstawała w ośrodkach uniwersyteckich, gdzie programistom bliższe były zasady „wolności, równości i braterstwa" niż chęci do przyjmowania standardów wymyślonych i narzuconych przez pewne instytucje.

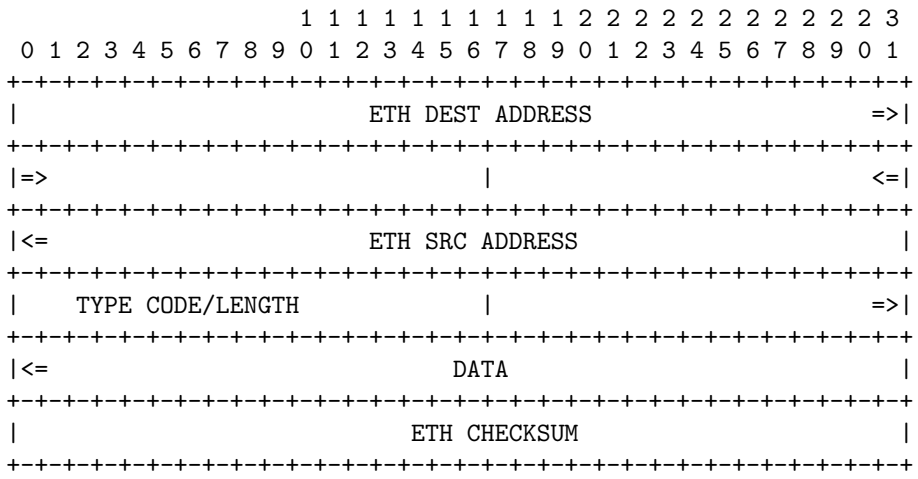
Ponad to nie ukrywajmy, ale model ISO/OSI jest, prze swoją ogólność, modelem bardzo złożonym, przez co wiele aspektów, mimo ich precyzyjnego opisania, jest trudna do zrozumienia i wprowadzenia w życie a niektóre funkcjonalność powtarzając się w różnych warstwach (np. kontrola błędów występuje w warstwie łącza danych jak i transportu). Wynikiem złożoności były złe implementacje co tylko dodatkowo przysparzało modelowi przeciwników.

Podsumowując wszystko powyższe, stwierdzić można, że model ISO/OSI (może z pominieciem warstwy sesji a czasem i prezentacji jako, że role tych warstw przejęła warstwa aplikacji) jest wyjątkowo przdatny przy omawianiu dowolnych sieci (bo przecież po to został stworzony). Co do modelu TCP/IP często zaś mówi się, że model praktycznie nie istnieje, ale działają (i to dosyć dobrze) protokoły na potrzeby których model ten został stworzony. Tak więc ISO/OSI do nauki, TCP/IP do pracy.

Rozdział 3

Warstwa łącza danych

3.1 Nagłówek Ethernet-u



Znaczenie poszczególnych pól

ETH DEST ADDRESS (*ang. Ethernet Destination Address*, 48 bitów) Docelowy adres urządzenia pracującego w sieci Ethernet.

ETH SRC ADDRESS (*ang. Ethernet Source Address*, 48 bitów) Źródłowy adres urządzenia pracującego w sieci Ethernet.

TYPE CODE (16 bitów) Kod określający rodzaj protokołu użytego do przesyłania danych lub ilość danych wyrażona w bajtach.

ETH CHECKSUM (*ang. Ethernet Checksum*, 32 bity)

Każdy pakiet poprzedzony jest jeszcze specyficzną sekwencją 8 bitów (ang. *preamble*) 01111110; analogiczna sekwencja kończy pakiet.

Rozdział 4

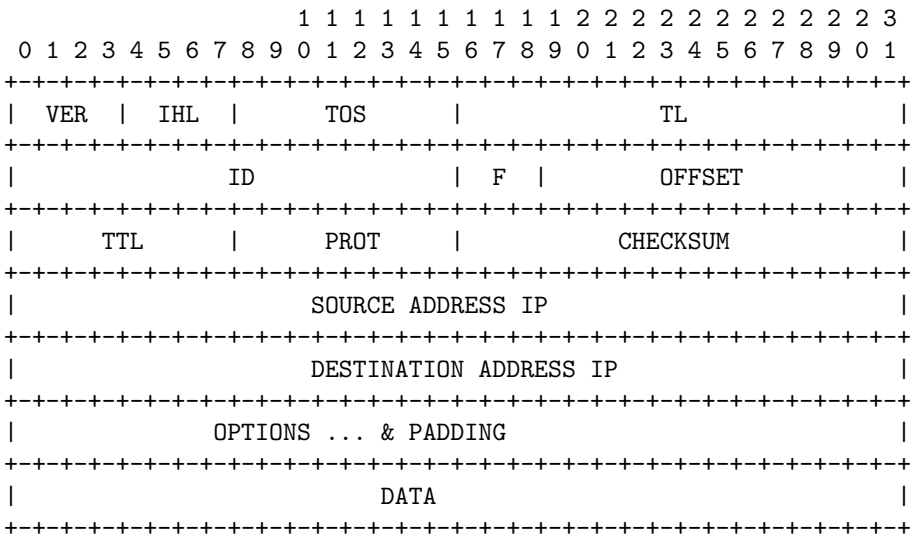
Protokół IP

Protokół IP ([4]) (ang. *internet protocol*) jest protokołem warstwy 3 modelu ISO/OSI. Mówiąc najogólniej, dzięki przenoszonym przez niego informacjom adresowym dane mogą zostać dostarczone od nadawcy do odbiorcy.

Aby umożliwić intersieci udostępnianie jednolitego systemu komunikacyjnego, oprogramowanie ukrywa szczegóły sieci fizycznych, oferując udogodnienia dużej sieci wirtualnej. Wirtualna intersieć działa podobnie do każdej innej sieci, umożliwiając komputerom wysyłanie i odbieranie pakietów z informacjami. Główna różnica polega na tym, że intersieć jest jedynie modelem działającym dzięki odpowiedniemu oprogramowaniu.

Krytycznym elementem modelu intersieci jest adresowanie. Aby dawać obraz pojedyńczego, jednolitego systemu, wszystkie komputery muszš wykorzystywać jednolity schemat adresowania, a każdy adres musi być jednoznaczny. Fizyczne adresy sieciowe są w tym przypadku nieodpowiednie, gdyż intersieć może obejmować wiele technik sieciowych, z których każda może definiować własny format adresu.

Aby zagwarantować jednolite adresowanie we wszystkich węzłach, oprogramowanie protokołów określa schemat adresowania, który jest **nie zależny** od bazowych adresów fizycznych. Chociaż schemat adresowania w intersieci jest realizowany przez oprogramowanie, adresy protokołowe są wykorzystywane jako punkty docelowe w intersieci, tak jak adresy sprzętowe są wykorzystywane jako punkty docelowe w sieciach fizycznych. Aby wysłać pakiet przez intersieć, nadawca umieszcza protokołowy adres odbiorcy w pakiecie i przekazuje pakiet do oprogramowania protokołu w celu wysłania. Oprogramowanie wykorzystuje protokołowy adres docelowy przy przekazywaniu pakietu poprzez intersieć do komputera odbiorcy. Dwa programy komunikują się, nie znając swoich adresów fizycznych. W stosie protoko-



Rys. 4.1. Nagłówek protokołu IP

łów TCP/IP adresowanie zdefiniowane jest w protokole intersieci – Internet Protocol – IP.

Opis protokołu rozpoczniemy od przedstawienia datagramu (pakietu) IP. Zaznaczmy przy tym, że mówiac protokół IP mamy zawsze na myśli wersję 4 tego protokołu (czyli IPv4). Datagram taki składa się z nagłówka IP oraz następujących po nim danych.

4.1 Nagłówek IP

Nagłówek IP składa się z części stałej (20 bajtowej) oraz części opcjonalnej o zmiennej długości. Format nagłówka przedstawiono na rysunku 4.1

VER (*ang. Version*, 4 bity) Wersja nagłówka IP; określa format nagłówka IP. Wartość równa 4 oznacza standardowy nagłowek wersji 4 protokołu IP.

IHL (*ang.Internet Header Length*, 4 bity) Określa długość pakietu IP w 32 bitowych słowach[1]. Minimalna wartość dla poprawnego nagłówka wynosi 5.

[1]To znaczy wielokrotnościach 32 bitów; 32 bity to 4 bajty.

TOS (*ang. Type Of Service*, 8 bitów) Rodzaj usługi. Parametry te mogą wpływać na sposób traktowania pakietu przez urządzenia sieci podczas jego przesyłania. Na przykład bardzo ważne pakiety można oznaczyć etykietą „wysokiego priorytetu". Bardziej szczegółowo pole to dzieli się na następujące fragmenty

```
 012.3.4.5.6.7
|P  |D|T|R|M|0|
```

Znaczenie ich jest następujące

P (*ang. Precedence*, 3 bity) Określa „ważność" pakietu – od 0 (zwykły) do 7 (sterujący siecią a więc najważniejszy).

```
0 - Routine
1 - Priority
2 - Immediate
3 - Flash
4 - Flash override
5 - CRITIC/ECP
6 - Internetwork control
7 - Network control
```

D (*ang. Delay*, 1 bit) Określa opóźnienie.

```
0 - Normal delay
1 - Low delay
```

T (*ang. Throughput*, 1 bit) Określa przepustowość.

```
0 - Normal throughput
1 - High throughput
```

R (*ang. Reliability*, 1 bit) Określa niezawodność.

```
0 - Normal reliability
1 - High reliability
```

M (*ang. Monetary*, 1 bit) Określa koszt przesłania.

```
0 - Normal monetary cost
1 - Minimize monetary cost
```

0 (1 bit) Puste

TL (*ang. Total Length*, 16 bitów) Długość datagramu a więc długość nagłówka razem z danymi. Największą liczbą jaką na 16 bitach można zapisać jest 65536 (2^{16}).

ID (*ang. Identification*, 16 bitów) Pozwala jednoznacznie odróżnić wysyłane datagramy. Mówiąc precyzyjniej: w przypadku podziału datagramu na mniejsze części, pole to pozwala na jego ponowne „złożenie". Pole to musi mieć unikalną wartość dla pary nadawca-odbiorca.

F (*ang. Flags*, 3 bity) Flagi

```
 0. 1. 2
|R|DF|MF|
```

Znaczenie ich jest następujące

R (*ang. Reserved*, 1 bit) Zarezerwowany; jego wartość powinna być równa 0.

DF (*ang. Don't Fragment*, 1 bit) Oznacza sposób przesyłania datagramu.

```
0 - Fragment if necessary
1 - Do not fragment
```

MF (*ang. More Fragments*, 1 bit) Określa czy dany fragment jest ostatnim fragmentem datagramu.

```
0 - This is the last fragment
1 - More fragments follow this fragment
```

OFFSET (*ang. Fragment Offset*, 13 bitów) Używany do określenia kolejności fragmentów wiadomości wchodzących w skład większego datagramu. Wszystkie fragmenty jednego datagramu mają to samo ID. Wszystkie fragmenty datagramu z wyjątkiem ostatniego muszą mieć długość równą wielokrotności 8 bajtów. Ponieważ pole to ma 13 bitów, więc datagram może mieć maksymalnie 8192 fragmenty (2^{13}) czyli 65536 bajtów ($8192 \cdot 8$) a więc o jeden więcej niż pozwala na to pole TL.

(TTL) (*ang. Time To Live*, 8 bitów) Określa czas życia datagramu. Gdy liczba zapisana w tym polu osiągnie wartość 0, datagram jest usuwany.

PROT (*ang. Protocol*, 8 bitów) Określa jaki protokół zawarty jest w datagramie IP. Możliwe wartości to między innymi[2]

```
 1 - ICMP - Internet Control Message Protocol
 2 - IGAP - IGMP for user Authentication Protocol
     IGMP - Internet Group Management Protocol
     RGMP - Router-port Group Management Protocol
 6 - TCP  - Transmission Control Protocol
17 - UDP  - User Datagram Protocol
```

CHECKSUM (*ang. Header Checsum*, 16 bitów) Suma kontrolna nagłówka IP. Według algorytmu, dodawane są, przy użyciu arytmetyki uzupełnień jedynkowych, 16-bitowe liczby; jako sumę kontrolną bierzemy uzupełnienie jedynkowe liczby otrzymanej w wyniku tego dodawania.

SOURCE IP ADDRESS (16 bitów)Adres IP nadawcy.

DESTINATION IP ADDRESS (16 bitów) Adres IP odbiorcy.

OPTIONS (różna długość) Opcje. Pole to dzieli się na następujące fragmenty

```
 0.12.34567
|C|CL| OPT |
```

Znaczenie ich jest następujące

C (*ang. Copy*, 1 bit)

```
0 - Do not copy
1 - Copy
```

CL (*ang. Class*, 2 bity)

```
0 - Control
1 - Reserved
2 - Debugging and measurement
3 - Reserved
```

OPT (*ang. Option*, 5 bitów) Jedną z dostępnych opcji jest na przykład możliwość śledzenia trasy jaką podąża datagram.

[2]Kiedyś wartości te były określone przez RFC 1700; aktualny obecnie spis dostępny jest pod adresem `www.iana.org`

PADDING Używane do wypełnienia pustego miejsca aby zapewnić, że dane zaczną się na granicy 32 bitowego słowa.

DATA Dane

4.2 Adresy IP

4.2.1 Adresowanie klasowe

Standard IP określa, że każdy węzeł ma przypisany 32-bitowy numer, zwany adresem węzła w protokole intersieci, lub po prostu adresem IP. Każdy pakiet wysyłany przez intersieć zawiera zarówno adres IP odbiorcy jak i nadawcy. Każdy adres IP podzielony jest na dwie części: **prefiks** i **sufiks**. Prefiks identyfikuje sieć fizyczną, do której jest podłączony fizycznie komputer. Sufiks wskazuje konkretny komputer w danej sieci. Żadne dwie sieci nie mogą mieć przyznanego tego samego numeru jak i żadne dwa komputery w ustalonej sieci nie mogą posiadać identycznego numeru. Inaczej: jeśli dwa komputery są przyłączone do różnych sieci fizycznych, to mają różne prefiksy; jeśli dwa komputery są podłączone do tej samej sieci fizycznej, to ich adresy mają różne sufiksy. Hierarchia adresów IP gwarantuje dwie ważne własności:

* każdy komputer ma przyznany jednoznaczny adres,

* chociaż przypisanie numerów sieci **muszą** być koordynowane globalnie, sufiksy mogą być przyznawane lokalnie bez globalnego uzgadniania.

Zauważmy, że takie rozwiązanie pozwala na rozsądne powiązanie urządzeń ze sobą - na podstawie analizy adresu można odgadnąć np. gdzie znajduje się dana maszyna. Dużo łatwiejsze jest także zarządzanie adresami bo adres globalnie przydziela się sieci a nie każdemu hostowi z osobna a taka sytuacja miałaby miejsce gdyby nie opisany podział.

Po określeniu rozmiaru pojedyńczego adresu należało zdecydować ile bitów przeznaczyć na każdą z dwóch części. Żaden prosty wybór nie był tu możliwy, ponieważ dodanie bitów do jednej z nich powodowało zabranie ich z drugiej. Obranie długiego prefiksu jest odpowiednie w przypadku istnienia wielu sieci, ale powoduje ograniczenia rozmiaru każdej z nich. Obranie długiego sufiksu oznacza, że każda sieć fizyczna może zawierać wiele komputerów, ale całkowita liczba sieci jest wówczas ograniczona.

Klasa A: max. liczba sieci – 128, max. liczba komputerów w sieci 16777216

```
0|1 2 3 4 5 6 7 | 8 9 10 11 12 13 14 ... 25 26 27 28 29 30 31
0| Prefiks - 7 b|    Sufiks - 24 bity
```

Klasa B: max. liczba sieci – 16384, max. liczba komputerów w sieci 65536

```
0 1|2 3 4 5 6 ... 12 13 14 15|16 17 18 19  ...  28 29 30 31
1 0| Prefiks - 14 bitów      |   Sufiks - 16 bitów
```

Klasa C: max. liczba sieci – 2097152, max. liczba komputerów w sieci 256

```
0 1 2|3 4 5 6 ... 18 19 20 21 22 23|24 25 26 27 28 29 30 31
1 1 0| Prefiks - 21 bitów          | Sufiks - 8 bitów
```

Klasa D – adres rozgłaszania grupowego (RFC 1112)

```
0 1 2 3|4 5 6 ... 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1 1 1 0|  28 bitów
```

Klasa E – zarezerwowane na przyszłość

```
0 1 2 3|4 5 6 ... 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1 1 1 1|  28 bitów
```

Rys. 4.2. Klasy adresowe

Ze względu na to, że intersieć może obejmować dowolne techniki sieciowe zbudowane z „mieszaniny" dużych i małych sieci, podzielono przestrzeń adresową na trzy podstawowe klasy[3] (A, B, C) o różnych rozmiarach prefiksu i sufiksu (patrz rysunek 4.2).

4.2.2 Notacja dziesiętna z kropką

Chociaż adresy IP są 32-bitowymi liczbami binarnymi, użytkownicy rzadko wpisują lub czytają ich wartość w tej postaci, stosując zamiast niej notację dziesiętną z kropką. W tym sposobie zapisu każda 8-bitowa część 32-bitowej liczby jest wyrażona jako wartość dziesiętna, zaś kropki są wykorzystywane jako separatory części.

[3]Obecnie adresy IPv4 uważa się za adresy bezklasowe. Pozostawiam jednak ten fragment mając nadzieję, że pomoże on zrozumieć inne zagadnienia czy idee.

Przykład 4.1. Notacja "zwykła" i dziesiętna z kropką.

Notacja "zwykła"

```
10000001 00110100 00000110 00000000
```

Ten sam adres zapisany z wykorzystaniem notacji dziesiętnej z kropką

```
129.52.6.0
```

W ten sposób adresy dziesiętne z kropką sięgają od 0.0.0.0 do 255.255.255.255. Notacja z kropkami jest odpowiednia dla adresów IP, gdyż w adresach tych podział na prefiks i sufiks jest na granicy oktetów. W przypadku adresów klasy A ostatnie trzy oktety odpowiadają sufiksowi komputera, klasy B – ostatnie dwa, a w adresach klasy C – jeden oktet. Niestety w tej postaci nie widać poszczególnych bitów adresu, przez co klasę musimy rozpoznwać na podstawie wartości dziesiętnej pierwszego oktetu.

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

4.2.3 Adresy IP specjalnego przeznaczenia

IP określa zestaw adresów o szczególnej postaci, które są zarezerwowane

Adresy sieciowe. (NET.0) IP rezerwuje adres zerowy węzła w danej sieci i wykorzystuje go przy odwoływaniu się do sieci. Odnosi się on do samej, a nie do komputerów podłączonych do niej. Adres ten oznacza prefiks przyznany sieci. Na przykład adres 128.211.0.0 oznacza sieć, której przyznano prefiks klasy B równy 128.211.

Adres rozgłaszania ukierunkowanego. (NET.255) Używany jest w celu przesłania pakietu do wszystkich węzłów w danej sieci fizycznej. Gdy

jest wysyłany pakiet pod adres rozgłaszania ukierunkowanego danej sieci, przez intersieć podróżuje tylko jedna jego kopia, aż dotrze do sieci. Następnie pakiet ten jest dostarczany do wszystkich węzłów tej sieci.

Adres rozgłaszania ukierunkowanego danej sieci jest tworzony przez dodanie do jej prefiksu sufiksu, któty składa się z samych jedynek. IP rezerwuje adres węzła, którego wszystkie bity to jedynki. Zatem sufiks składa się z liczb 255.

Adres rozgłaszania ograniczonego. (255) Termin rozgłaszanie ograniczone odnosi się do rozgłaszania w lokalnej sieci fizycznej. Jest ono używane na przykład przy starcie systemu przez komputey, które nie znają w tym momencie numeru sieci. IP na rozgłaszanie ograniczone rezerwuje adres składający się z samych jedynek.

Adres bieżącego komputera. (0) Każdy pakiet musi zawierać adres odbiorcy i nadawcy; komputer by wysłać i odebrać pakiety, musi znać swój adres IP. Zestaw protokołów TCP/IP obejmuje protokoły, których można użyć przy automatycznym uzyskiwaniu adresu IP przy starcie. Co ciekawe, protokoły startowe do komunikacji używają IP. Komputer korzystając z takich protokołów uruchomieniowych, nie może podać prawidłowego adresu IP nadawcy. Aby radzić sobie w takich sytuacjach, w IP zarezerwowano adres, który składa się z samych zer, na oznaczenie bieżącego komputera.

Adres pętli zwrotnej. (127.X) Protokół IP rezerwuje prefiks sieciowy klasy A równy 127 na adres pętli zwrotnej. Adres węzła (sufiks) używany przytym jest bez znaczenia. Najpopularniejszym adresem pętli zwrotnej jest 127.0.0.1. Adres ten przydaje się na przykład przy testowaniu programów sieciowych. Używajšc pętli zwrotnej, żadne pakiety nigdy nie opuszczš komputera – oprogramowanie IP przekazuje pakiety z jednego programu użytkownika do drugiego.

4.2.4 Maska sieci i podsieci

Jak powiedzieliśmy wcześniej, wszystkie hosty w sieci lokalnej **muszą** mieć ten sam numer sieci. Przy większej liczbie hostów takie wymaganie jest uciążliwe. Rozwiązaniem jest wewnętrzny (a więc nie widoczny z zewnątrz) podział sieci na osobne niezależne logicznie fragmenty – podsieci.

Zasada podziału jest bardzo prosta. Oto **nie ruszając** numeru sieci, na potrzeby wskazania numeru podsieci „zabieramy" część bitów przeznaczonych na numer hosta. O tym które bity są adresem sieci i podsieci informuje nas **maska podsieci**. Spójrzmy na następujący przykład. Załóżmy, że przyznano nam następujący adres sieci klasy B

```
130.12.0.0
```

czyli dwójkowo

```
10000010.00001100.00000000.00000000
```

Zapiszmy teraz pod dwójkowym ciągiem będącym adresem sieci, pewien ciąg złożony z zer i jedynek według następującej zasady: 1 piszemy pod bitami wchodzącymi w skład adresu sieci, 0 pod bitmi wchodzącymi w ewentualne numery hostów w ramach tejże sieci.

```
10000010.00001100.00000000.00000000
11111111.11111111.00000000.00000000
```

Otrzymaliśmy w ten sposób maskę sieci a więc „coś" informującego nas o tym jaka część adresu jest adresem sieci a jaka jest adresem hosta. Zwykle zarówno adres jak i związaną z nim maskę będziemy zapisywać z użyciem notacji kropkowej; a więc w tym przypadku otrzymujemy

```
130.12.0.0 - adres sieci,
255.255.0.0 - maska sieci

lub równoważnie

130.12.0.0/16.
```

Spróbujmy teraz podzielić naszą sieć na kawałki. Załóżmy, że chcemy mieć 8 podsieci. Tak więc, nie ruszając adresu sieci już przyznanego, możemy pożyczyć jeszcze 3 bity z puli bitów przeznaczonych na adres hosta i dołączyć je do adresów sieci.

```
10000010.00001100.xxx00000.00000000

x - bity, które staną się adresem podsieci w ramach sieci 130.12.0.0

maska:
```

```
10000010.00001100.xxx00000.00000000
11111111.11111111.11100000.00000000 - maska

czyli

255.255.224.0

lub też

130.12.0.0/19
```

Tak więc pierwsza podsieć będzie używać adresów zaczynających się od 130.12.0.1, druga 130.12.32.1, trzecia 130.12.64.1 itd. Wyjaśnienie tak przyznanych adresów jest następujące.

Pierwszych 16 bitów, jako stanowiących odgórnie przyznany nam adres, nie możemy zmienić, stąd też dwie pierwsze liczby pozostaną niezmienione, a więc mamy 130.12. Następnie w ramach dostępnej przestrzeni adresowej 3 pierwsze bity postanawiamy przeznaczyć na adresy podsieci. Tak więc niech pierwsza podsiec ma numer `000`, druga `001`, trzecia `010` itd. Pozostaje nam do dyspozycji jeszcze 13 bitów. Najmniejszą możliwą do zapisania liczbą 13-bitową będącą poprawnym adresem hosta jest `00000 00000001`. Składając to wszystko razem, otrzymujemy, że ostatnie 16 bitów będące adresem pierwszego hosta w każdej z podsieci będzi postaci

```
dla podsieci ,,pierwszej'', tj. 000:
00000000 00000001 czyli 0.1

dla podsieci ,,drugiej'', tj. 001:
00100000 00000001 czyli 32.1

dla podsieci ,,trzeciej'', tj. 010:
01000000 00000001 czyli 64.1
```

4.2.5 Translacja adresów sieciowych (NAT)

Potencjalnie bardzo duża przestrzeń adresowa ($2^{32} = 4.294.967.296$) ze względu na przyjęty klasowy sposób podziału adresów powoduje „marnowanie" pewnej ich ilości. I tak jeśli organizacja potrzebuje 60.000 adresów wówczas w naturalny sposób otrzyma klasę adresową B. Niestety w takiej sytuacji pozostanie niewykorzystane ponad 5.000 adresów – adresów tych nie można przydzielić już nikomu innemu.

Zatem należało znaleźć sposób na obejście pojawiających się ograniczeń związanych z dostępną przestrzenią adresową. Jedną z możliości jest tak zwana **translacja adresów sieciowych** (ang. *network address translation*).

Przy translacji adresów wykorzystano pewną obserwację pocznioną na temata protokołów wykorzystywanych w sieci. Otóż okazało się, że znaczna część pakietów zawiera w sobie nagłówki protokołów IP oraz TCP.

Postąpujemy teraz według następującego schematu. Każdy pakiet jaki wychodzi z naszej lokalnej sieci ma nadany unikalny adres nadawcy (nazwijmy go HOSTADDR). Pakiet ten wychodząc z sieci podmieniany ma adres nadawcy na adres komputera łączącego naszą sieć lokalną ze światem (nazwijmy go GTWADDR). Następnie pakiet wędruje do odbiorcy (nazwijmy go DESTADDR). Odbiorca przetwarza go i odsyła będąc przekonanym, że pakiet ten powinien trafić do GTWADDR (po przecież to właśnie adres GTWADDR był w pakiecie jaki naszedł do DESTADDR). Teraz, gdy GTWADDR otrzyma pakiet, podmienia w nim ponownie adres na HOSTADDR i wysyła do sieci lokalnej. Opisane tutaj operacje zestawiono w tabeli 4.1 Wszystko jak do tej pory wygląda bardzo obiecująco z jednym małym wyjatkiem. Skąd mianowicie GTWADDR wie do jakiekiego hosta w sieci lokalnej wysłać pakiet?! Przecież w sieci lokalnej może być wiele komputerów i wiel z nich może prowadzić taką komunikację. Stąd też GTWADDR otrzyma wiele pakietów gdzie w polu adresat będzie widniał jego adres. Teraz należy jakoś pakiety te porozdzielać a część być może zatrzymać bo część faktycznie może być adresowana do samego GTWADDR. I właśnie w tym miejscu przydaje się wspomniana na początku obserwacja. Otóż w nagłówku IP nie mam już miejsca na dołączenie dodatkowej informacji związanej z tym do kogo przesłać pakiet. Miejsce takie jest na przykład w nagłówku pakietu TCP. Występuje tam pole przechowujące numer portu pod który należy przekazać pakiet. Czym jest numer portu powiemy sobie dokładniej przy okazji omawiana protokołu TCP. Teraz przyjmijmy, że jest to numer identyfikujący rodzaj aplikacji dla której przeznaczony jest pakiet. Numer tej jest wystarczająco pojemny aby móc odróżnić wszystkie komputery w podsieci. Tak więc faktyczne operacje realizowane podczas translacji należy uzupełnić o zmianę numeru portu (pamiętajmy jednk, że zmiana ta ziwże się z „grzebaniem" w pakiecie TCP a więc warstwa sieci – zagląda i modyfikuje zawartość otrzymną od wartwy transportowej; zatem robi coś co jest zaprzeczeniem idei warstwowości i niezaleśnoci kolejnych poziomów stosu!). Uzupełnione operacje zestawiono w tabeli 4.2. Jak więc widać zasadniczą wadą tego sposobu postępowania jest bazowanie na struk-

Wpisy w nagłówku IP (adres nadawcy i odbiorcy)	Operacja
`SRC: HOSTADDR` `DEST: DESTADDR`	Nadawca przygotowywuje pakiet ustawiając **swój** adres IP i adres docelowego odbiorcy.
`SRC: HOSTADDR` `DEST: DESTADDR`	Pakiet jaki dochodzi do GTWADDR od strony sieci lokalnej.
`SRC: GTWADDR` `DEST: DESTADDR`	Pakiet jaki wychodzi z GTWADDR i jest dalej przesyłany do DESTADDR.
`SRC: GTWADDR` `DEST: DESTADDR`	Pakiet jaki dochodzi do docelowego odbiorcy.
`SRC: DESTADDR` `DEST: GTWADDR`	Pakiet jaki wychodzi od docelowego odbiorcy i jest wysyłany do nadawcy.
`SRC: DESTADDR` `DEST: GTWADDR`	Pakiet jaki dochodzi do GTWADDR od strony Sieci.
`SRC: DESTADDR` `DEST: HOSTADDR`	Pakiet jaki wychodzi z GTWADDR i jest dalej przesyłany do HOSTADDR.

Tablica 4.1. Zmiany adresów nadawcy i odbiorcy przy stosowaniu techniki NAT

Wpisy w nagłówku IP (adres nadawcy i odbiorcy)	Operacja
IP SRC: HOSTADDR IP DEST: DESTADDR PORT SRC: HOSTPORT PORT DEST: DESTPORT	Nadawca przygotowywuje pakiet ustawiając **swój** adres IP i adres docelowego odbiorcy. Ustawia także numer portu aplikacji nadającej i aplikacji odbierającej.
IP SRC: HOSTADDR IP DEST: DESTADDR PORT SRC: HOSTPORT PORT DEST: DESTPORT	Pakiet jaki dochodzi do GTWADDR od strony sieci lokalnej.
IP SRC: GTWADDR IP DEST: DESTADDR PORT SRC: UNIQUE PORT DEST: DESTPORT	Pakiet jaki wychodzi z GTWADDR i jest dalej przesyłany do DESTADDR. Przed wysłaniem, GTWADDR zmienia numer portu na pewną uniklaną wartość i **zapamiętuje**, związek UNIQUE <-> (HOSTPORT,HOSTADDR)
IP SRC: GTWADDR IP DEST: DESTADDR PORT SRC: UNIQUE PORT DEST: DESTPORT	Pakiet jaki dochodzi do docelowego odbiorcy.
IP SRC: DESTADDR IP DEST: GTWADDR PORT SRC: DESTPORT PORT DEST: UNIQUE	Pakiet jaki wychodzi od docelowego odbiorcy i jest wysyłany do nadawcy.
IP SRC: DESTADDR IP DEST: GTWADDR PORT SRC: DESTPORT PORT DEST: UNIQUE	Pakiet jaki dochodzi do GTWADDR . od strony Sieci. Teraz na podstawie **zapamiętanej** zależności UNIQUE <-> (HOSTPORT,HOSTADDR) przywracane są odpowiednie wartości w polach IP DEST oraz PORT DEST
IP SRC: DESTADDR IP DEST: HOSTADDR PORT SRC: DESTPORT PORT DEST: HOSTPORT	Pakiet jaki wychodzi z GTWADDR i jest dalej przesyłany do HOSTADDR.

Tablica 4.2. Zmiany adresów nadawcy i odbiorcy przy stosowaniu techniki NAT

turze nagłówka warstwy wyższej. Co gorsza, w przypadku zmiany formatu nagłowka w warstwie transportowej **muszą** nastąpić pewne zmiany w warstwie sieci. Dodatkowo sytuacje komplikuje sytuacja kiedy to w nagłówku warstwy nadrzędnej dla warstwy sieci **nie ma** numeru portu (bo przecież nie musi!).

Przy wykorzystaniu NAT w sieci wewnętrznej powinniśmy stosować adresy z jednego z trzech poniższych zakresów (mamy bowiem gwarancję, że w Internecie nie może pojawić się pakiet zawierający adres z tych zakresów):

```
10.0.0.0    - 10.255.255.255  / 8  (max 16.777.216 hostów)
172.16.0.0  - 172.31.255.255  /12  (max 1.048.576 hostów)
192.168.0.0 - 192.168.255.255 /16  (max 65.536 hostów)
```

4.2.6 Bezklasowy ruting międzydomenowy (CIDR)

Opisana w rozdziale 4.2.5 technika translacji adresów sieciowych działa sprawnie, ale posiada jeszcze jedną wadę, o ktorej do tej pory nie wspomnieliśmy – osoby będące w sieci lokalnej nie mogą posiadać adresu IP widocznego na zewnątrz. Innymi słowy nie ma możliwości zainicjowania komunikacji z zewnątrz, czyli np. taki adres będzie całkowicie nieprzydatny dla komputera mającego być serwerem stron www. Tak więc ponownie natrafiamy na, pozornie rozwiązany już, problem braku wystarczającej ilości adresów. W sytuacji gdy po pierwsze należało wygospodarować pewną liczbę adresów a po drugie adresy te musiały być „prawdziwymi" adresami IP zaproponowano (i faktycznie wprowadzono) rozwiązanie znane pod nazwą CIDR ([9]) (ang. *Classless InterDomain Routing* – bezklasowy ruting międzydomenowy, dla odróżnienia od routingu używającego adresowania klasowego (ang. *classful addressing*), opisanego w 4.2.1). Podstawowym założeniem CIDR jest przydzielanie pozostałych adresów IP w blokach o różnych wielkościach bez zwracania uwagi na klasy. Wymaga to rozszerzenia każdego adresu IP o maskę sieci, gdyż teraz nie wiemy jaka część adresu jest adresem sieci.

Przykład

Weźmy pod uwagę blok 2048 klasy adresowych C (w ramach każdej klasy mamy oczywiście 256 możliwych adresów, więc łącznie mamy do dyspozycji $2048 \cdot 256 = 524288$ adresów) od 192.24.0.0 do 192.31.255.0 przyznany jednej instytucji (single network provider) RA. Dostęp do tej sieci (a raczej bloku tych adresów) jest przez adres 192.24.0.0 z maską 255.248.0.0.

Załóżmy teraz, że RA przyłącza 6 klientów w następującej kolejności

1. C1 requiring fewer than 2048 addresses (8 class C networks),
2. C2 requiring fewer than 4096 addresses (16 class C networks),
3. C3 requiring fewer than 1024 addresses (4 class C networks),
4. C4 requiring fewer than 1024 addresses (4 class C networks),
5. C5 requiring fewer than 512 addresses (2 class C networks),
6. C6 requiring fewer than 512 addresses (2 class C networks).

We wszystkich przypadkach, każdy z klientów dopuszcza wzrost zapotrzebowania na dostępne adresy. RA przyznaje adresy w następujący sposób

1. C1: allocate 192.24.0 through 192.24.7. This block of networks is described by the supernet route 192.24.0.0 and mask 255.255.248.0
2. C2: allocate 192.24.16 through 192.24.31. This block is described by the route 192.24.16.0, mask 255.255.240.0
3. C3: allocate 192.24.8 through 192.24.11. This block is described by the route 192.24.8.0, mask 255.255.252.0
4. C4: allocate 192.24.12 through 192.24.15. This block is described by the route 192.24.12.0, mask 255.255.252.0
5. C5: allocate 192.24.32 and 192.24.33. This block is described by the route 192.24.32.0, mask 255.255.254.0
6. C6: allocate 192.24.34 and 192.24.35. This block is described by the route 192.24.34.0, mask 255.255.254.0

```
C1:
F: 11000000 00011000 00000000 xxxxxxxx
T: 11000000 00011000 00000111 xxxxxxxx
M: 11111111 11111111 11111000 00000000
R: 11000000 00011000 00000000 00000000

C2:
F: 11000000 00011000 00010000 xxxxxxxx
T: 11000000 00011000 00011111 xxxxxxxx
M: 11111111 11111111 11110000 00000000
R: 11000000 00011000 00010000 00000000
```

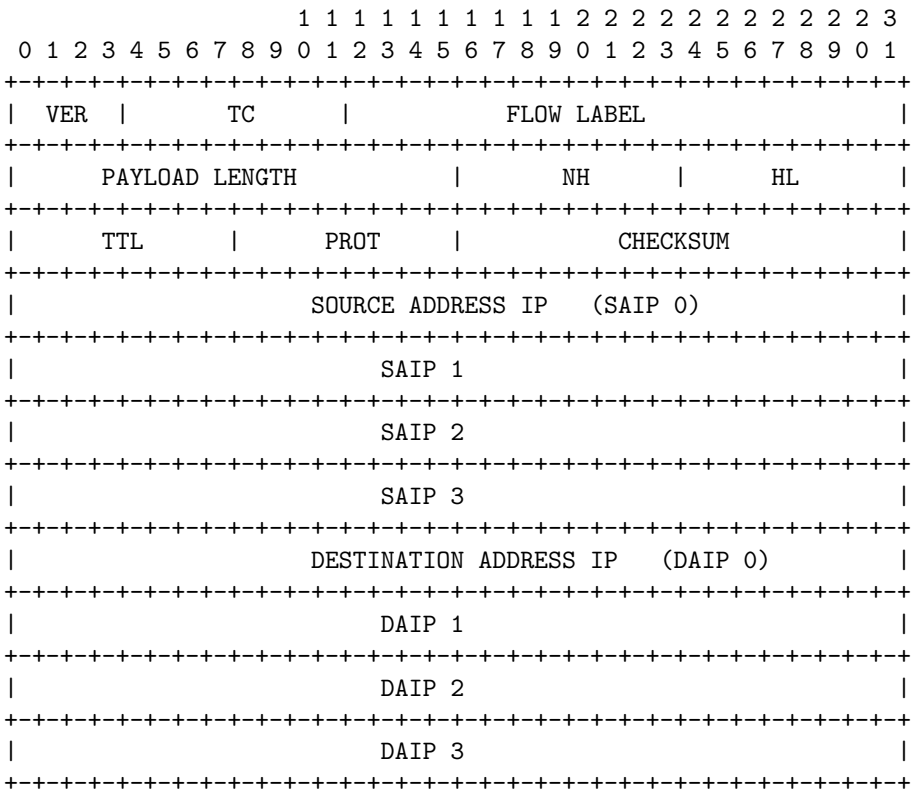
```
C3:
F: 11000000 00011000 00001000 xxxxxxxx
T: 11000000 00011000 00001011 xxxxxxxx
M: 11111111 11111111 11111100 00000000
R: 11000000 00011000 00001000 00000000

C4:
F: 11000000 00011000 00001100 xxxxxxxx
T: 11000000 00011000 00001111 xxxxxxxx
M: 11111111 11111111 11111100 00000000
R: 11000000 00011000 00001100 00000000

C5:
F: 11000000 00011000 00100000 xxxxxxxx
T: 11000000 00011000 00100001 xxxxxxxx
M: 11111111 11111111 11111110 00000000
R: 11000000 00011000 00100000 00000000

C6:
F: 11000000 00011000 00100010 xxxxxxxx
T: 11000000 00011000 00100011 xxxxxxxx
M: 11111111 11111111 11111110 00000000
R: 11000000 00011000 00100010 00000000

R - adres sieciowy podsieci otrzymanej w ramach podziału
póli adresowej
```

4.3 IPv6

4.3.1 Nagłówek podstawowy

Kłopoty z adresowaniem, które zostały odsunięte w czasie przez wykorzystanie NAT i CIDR, sugerowały przyjrzenie się protokołowi IP i wprowadzenie w nim pewnych zmian. Główne założenia brane pod uwagę przy projektowaniu kolejnej wersji protokołu IP to:

1. Zwiększenie dostępnej puli adresowej przez wprowadzenie adresów zapisywanych na większej ilości bitów.
2. Uproszczenie samego protokołu (w szczególności jego nagłowka).
3. Zapewnienie większego bezpieczeństwa (a właściwie to bezpieczeństwa bo IPv4 nie zapewnia jakiegokolwiek bezpieczeństwa tj. ani prywatności aniuwierzytelniania).
4. Zwrócenie większej uwagi na prowadzenie transmisji w czasie rzeczywistym.
5. Umożliwienie współistnienia obu protokołow tj. istniejącego IPv4 i właśnie planowanego w okresie przejściowym (co może potrwać od kilku do kilkunastu lat).
6. Umożliwienie łatwego wprowadzania zmian w protokole w poźniejszych jego wersjach.



Rys. 4.3. Nagłówek protokołu IPv6

Nagłówek IPv6 składa się, w odróżnieniu od IPv4 tylko z części stałej (40 bajtowej). Format nagłówka przedstawiono na rysunku 4.3

VER (*ang. Version*, 4 bity) Wersja nagłówka IP; określa format nagłówka IP. Wartość równa 6 oznacza standardowy nagłowek wersji 6 protokołu.

TC (*ang.Traffic Class*, 8 bitów) Klasa ruchu (ang. *Internet traffic priority delivery value*). Pole wykorzystywane jest do rozróżniania pakietów o różnych wymaganiach związanych z czasem dostarczenia (głównie chodzi tutaj o transmisje w czasie rzeczywistym).

FLOW LABEL (20 bitów) Etykieta przepływu. Pole wykorzystywane do tworzenia pseudopołączeń[4] spełniających określone wymagania.

PAYLOAD LENGTH (16 bitów) Długość ładunku. W odróżnieniu do IPv4, nagłówek nie jest wliczany do długości – stąd zmiana nazwy pola.

[4]Pseudopołączeń, gdyż jak pamiętamy IP jest protokołem, ze swej natury, bezpołączeniowym.

NH (*NEXT HEADER*, 8 bitów) Następny nagłówek. Pole to określa który z sześciu opcjonalnych nagłówków następuje po bieżącym (jeśli oczywiście następuje, bo tak być nie musi).

HL (*HOP LIMIT*, (8 bitów) Liczba przeskoków. Odpowiada polu TTL z nagłówka IPv4. Zmiana nazwy jest tylko kosmetyką. Teoretycznie wartość w polu TTL oznaczać miała sekundy „życia" pakietu; w praktyce pole to było wykorzystywane raczej jako własnie licznik przeskoków.

SOURCE ADDRESS i DESTINATION ADDRESS, (128 bitów) Adres źródłowy i docelowy. Ze względu na znaczną długość adresu wprowadzono nowy sposób ich zapisu. Otóż każdy adres zapisujemy w **ośmiu** grupach po **cztery** cyfry szesnastkowe; grupy rozdzielamy znakiem dwukropka, np.:

`1000:0000:ADFF:0000:0000:0000:12A3:F2B1`

Zapewne jeszcze przez długi czas adresy będą zawierały, tak jak ten powyższy, długie ciągi zer. Dlatego dopuszczono kilka sposobó skracania zapisu.

- Można pominąć zera zaczynające każdą z grup. Tak więc zamiast

  `0020:00B2:ADFF:0000:0000:0000:12A3:F2B1`

  możemy napisać

  `20:B2:ADFF:0:0:0:12A3:F2B1`

- Jedną lub więcej grup złożoną z samych zer można zastąpić parą dwukropków[5]. Tak więc zamiast

  `0020:00B2:ADFF:0000:0000:0000:12A3:F2B1`

  możemy napisać

  `0020:00B2:ADFF::12A3:F2B1` lub
  `20:B2:ADFF::12A3:F2B1`

- Końcówkę adresu można zapisać w notacji kropkowo-dziesiętnej. Tak więc poprawny jest następujący adres

  `0000:0000:0000:0000:0000:0000:192.65.33.1`

  który może oczywiście zostać zapisany jako

  `::192.65.33.1`

Przyjrzyjmy się teraz zmianom jakie pociąga za sobą nowy format nagłówka. Po pierwsze 16-bajtowy adres zapewnia niewyczerpany zakres adresów. Oczywiście, ktoś może zaprotestować, że za jakiś czas gdy ludzkość przeniesie się w inne galaktyki, przybędzie nam mieszkańców i wówczas wyczerpie się ta, teoretycznie niewyczerpywalna, pula adresów. Pragnę więc uspokoić takie osoby, że nie grozi nam takie zdarzenie, bo ludzkość tego momentu nie dożyje :D

Drugim istotnym ulepszeniem jest uproszczenie nagłówka, pozwalające na jego znacznie szybsze przetwarzanie.

[5]Celem uniknięcia niejednoznaczności operację taką możemy zastosować tylko raz

Znacznie lepiej obsługiwane są opcje. Dzieje się to za sprawą dodatkowych nagłówków, dołączanych tylko wówczas gdy zachodzi potrzeba skorzystania z opcji.

Większą uwagę poświęcono jakości usług, głównie za sprawą rozwijających się dynamicznie ogólnie pojętych multimediów.

Piątą cechą, której nie można zauważyć bezpośrenio na podstawie podanego nagłówka, jest zwiększenie bezpieczeństwa czyli wprowadzenie mechanizmów uwierzytelniania i ochrony prywatności.

4.3.2 Nagłówki dodatkowe

W [13] zdefiniowano 6 możliwych nagłówków opcjonalnych:

1. Hop-by-Hop Options The Hop-by-Hop Options header is used to carry optional information that must be examined by every node along a packet's delivery path.
2. Routing (Type 0) The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visitedón the way to a packet's destination. This function is very similar to IPv4's Source Route options.
3. Fragment The Fragment header is used by an IPv6 source to send packets larger than would fit in the path MTU to their destinations. (Note: unlike IPv4, fragmentation in IPv6 is performed only by source nodes, not by routers along a packet's delivery path – see section 5.) The Fragment header is identified by a Next Header value of 44 in the immediately preceding header
4. Destination Options The Destination Options header is used to carry optional information that need be examined only by a packet's destination node(s).
5. Authentication The Authentication Header is a mechanism for providing strong integrity and authentication for IP datagrams. It might also provide non-repudiation, depending on which cryptographic algorithm is used and how keying is performed. For example, use of an asymmetric digital signature algorithm, such as RSA, could provide non- repudiation.

   Confidentiality, and protection from traffic analysis are not provided by the Authentication Header. Users desiring confidentiality should consider using the IP Encapsulating Security Protocol (ESP) either in lieu of or in conjunction with the Authentication Header [Atk95b]. This document assumes the reader has previously read the related IP Security Architecture document which defines the overall security architecture for IP and provides important background information for this specification [Atk95a].
6. Encapsulating Security Payload This document describes the IP Encapsulating Security Payload (ESP). ESP is a mechanism for providing integrity and confidentiality to IP datagrams. In some circumstances it can also provide authentication to IP datagrams. The mechanism works with both IPv4 and IPv6.

   1. INTRODUCTION

   ESP is a mechanism for providing integrity and confidentiality to IP datagrams. It may also provide authentication, depending on which algorithm and algorithm mode are used. Non-repudiation and protection from traffic analysis are not provided by ESP. The IP Authentication Header (AH) might provide non-repudiation if used with certain authentication algorithms [Atk95b]. The IP Authentication Header may be used in conjunction with ESP to provide authentication. Users

> desiring integrity and authentication without confidentiality should use the IP Authentication Header (AH) instead of ESP. This document assumes that the reader is familiar with the related document "IP Security Architecture", which defines the overall Internet-layer security architecture for IPv4 and IPv6 and provides important background for this specification [Atk95a].

Sugerowana jest następująca kolejność nagłówków:

1. IPv6 header
2. Hop-by-Hop Options header
3. Destination Options header (note 1)
4. Routing header
5. Fragment header
6. Authentication header (note 2)
7. Encapsulating Security Payload header (note 2)
8. Destination Options header (note 3)
9. upper-layer header

Rozdział 5

TCP

Protokół TCP ([5],[6],[7]) (ang. *Transmission Control Protocol*) jest protokołem warstwy 4 modelu ISO/OSI. Mówiąc najogólniej jego zadaniem jest zapewnienie niezawodnej transmisji strumienia bajtów (jakim są dane otrzymywane z warstw wyższych) na bazie (zawodnych z natury) usług warstw niższych.

TCP daje nam połączenie:

- zawsze dwupunktowe (brak np. obsługi multicastingu);
- pełnodupleksowe – jednocześnie przesyłane są dane w obu kierunkach;
- traktujace dane przesyłane w ramach połączenia TCP jak strumień bajtów.

Usługa TCP realizowana jest w oparciu o tak zwane gniazda (ang. *sockets*). Gniazdo tworzone jest zarówno przez nadawcę jak i odbiorcę. Jednoznacznie identyfikowane jest ono prze numer IP urządzenia i loklanego (przyznawanego w ramach urządzenia) numeru nazywanego portem (ang. *port*). Ważne aby pamiętać, że określone gniazdo może być wykorzystywane w tym samym czasie w kilku różnych połączeniach; identyfikację połączenia stanowi para złożona z dwóch gniazd reprezentujących dwie strony kanału komunikacyjnego[1].

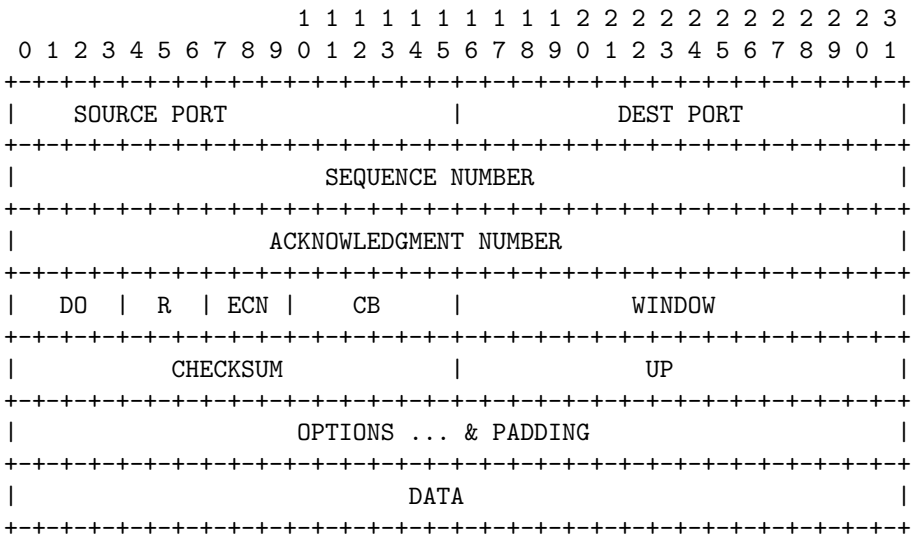
Porty o numerach mniejszych od 1024 określane są mianem dobrze znanych portów (ang. *well-known ports*) i ich wykorzystanie zarezerwowane jest na potrzeby usług standardowych[2]. Niektóre z częściej używanych portów przedstawia tabela 5.1.

[1] Celowo unikam w tym miejscu użycia terminów nadawca i odbiorac, gdyż utworzony kanał komunikacyjny nie wskazuje jednoznacznie żadngo z nich.

[2] Oczywiście nikt nie zabroni nam napisania programu wykorzystującego jeden z portów poniżej 1024 tylko, że wówczas może to rodzić spore komplikacje.

Wpisy w nagłówku IP (adres nadawcy i odbiorcy)	Operacja

Tablica 5.1. Niktóre z częściej używanych portów



Rys. 5.1. Nagłówek protokołu TCP

5.1 Nagłówek TCP

Nagłówek TCP składa się z części stałej (20 bajtowej) oraz części opcjonalnej o zmiennej długości. Format nagłówka przedstawiono na rysunku 5.1

SOURCE PORT (16 bitów) Port źródłowy.

DEST PORT (16 bitów) Port docelowy.

SEQUENCE NUMBER (32 bity) Numer sekwencyjny pierwszego oktetu danych. Jeśli obecny jest znacznik SYN to numer sekwencyjny jest początkowym numerem sekwenycjnym a pierwszy oktet ma numer o jeden większy od tego numeru.

ACKNOWLEDGMENT NUMBER (32 bity) Jeśli bit ACK jest ustawiony (ma wartość 1) pole to zawiera liczbę określającą kolejny numer danych oczekiwanych przez odbiorcę.

DO (*ang. Data Offset*, 4 bity) Określa długość nagłówka TCP w 32 bitowych słowach.

R (*ang. Reserved*, 3 bity) Zarezerwowane. Wartość tego pola powinna wynosić 0.

ECN (*ang. Explicit Congestion Notifiction*, 3 bity)

```
 0.1.2
|N|C|E|
```

N (*ang. Nonce Sum*, 1 bit)

C (*ang. CWR*, 1 bit)

E (*ang. ECE, ECN-Echo*, 1 bit)

CB (*ang. Control Bits*, 6 bitów)

```
 0.1.2.3.4.5
|U|A|P|R|S|F|
```

U, URG (*ang. Urgent*, 1 bit) – pole wskaźnika do pilnych danych ma znaczenie;

A, ACK (*ang. Acknowledgment*, 1 bit) – pole potwierdzenia ma znaczenie;

P, PSH (*ang. Push*, 1 bit) – wymuszenie transmisji danych;

R, RST (*ang. Reset*, 1 bit) – zerowanie połączenia;

S, SYN (*ang. Synchronize*, 1 bit) – synchronizacja numerów sekwencyjnych;

F, FIN (1 bit) – koniec danych od nadawcy;

WINDOW (16 bitów) Ilość danych jaką może przyjąć odbiorca. W RFC 1323 określono tzw. okno skalowalne, czyli możliwość interpretowania zawartości tego pola w jednostkach większych niż bajt. Jako maksymalną jednostkę przyjęto 2^{14} bajtów co w połączeniu z samą wielkością pola, pozwala definiować pola o rozmiarze 1GB ($2^30 = 1.073.741.824$).

CHECKSUM (16 bitów) Suma kontrola nagłówka TCP (wraz z danymi) uzupełnionego o pseudonagłowek zawierający informacje z nagłówka IP oraz TCP. Format tego pseudonagłówka jest następujący

```
                     1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          SOURCE IP ADDRESS                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       DESTINATION IP ADDRESS                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       PADDING     |  TCP PROT |               TL              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

gdzie `PADDING` powinno być wypełnione zerami, `TCP PROT` jest numerem wersji protokolu TCP, natomiast `TL` jest rozmiarem pakietu TCP.

Najpierw wpisujemy w pole sumy kontrolnej wartość zero. Następne dodajemy do siebie wszystkie 16-bitowe słowa tworzące pseudonagłówek i nagłówek właściwy TCP (wraz z danymi). Przy dodawaniau wykorzystujemy tylko 16 młodszych bitów wyniku (zatem ignorujemy przeniesienia na pozycje dalsze niż na 16 bit). Sumę kontrolną stanowi zanegowany wynik tych dodawań.

UP (*ang. Urgent Pointer*, 16 bitów) Jeśli flaga URG jest ustawiona, pole to wskazuje na kolejność ważnych danych.

OPTIONS (różna długość; od 0 do 44 bajtów)

PADDING Używane do wypełnienia pustego miejsca aby zapewnić, że dane zaczną się na granicy 32 bitowego słowa.

DATA Dane

Oto fragment listy dostępnej pod adresem
`http://www.iana.org/assignments/port-numbers`
opisującej porty przypisane do różnych usług:

PORT NUMBERS

(last updated 17 November 2005)

The port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports.

The Well Known Ports are those from 0 through 1023.

The Registered Ports are those from 1024 through 49151

The Dynamic and/or Private Ports are those from 49152 through 65535

```
*************************************************************************
* PLEASE NOTE THE FOLLOWING:                                            *
*                                                                       *
* 1. UNASSIGNED PORT NUMBERS SHOULD NOT BE USED.  THE IANA WILL ASSIGN *
* THE NUMBER FOR THE PORT AFTER YOUR APPLICATION HAS BEEN APPROVED.    *
*                                                                       *
* 2. ASSIGNMENT OF A PORT NUMBER DOES NOT IN ANY WAY IMPLY AN          *
* ENDORSEMENT OF AN APPLICATION OR PRODUCT, AND THE FACT THAT NETWORK  *
* TRAFFIC IS FLOWING TO OR FROM A REGISTERED PORT DOES NOT MEAN THAT   *
* IT IS "GOOD" TRAFFIC. FIREWALL AND SYSTEM ADMINISTRATORS SHOULD      *
* CHOOSE HOW TO CONFIGURE THEIR SYSTEMS BASED ON THEIR KNOWLEDGE OF    *
* THE TRAFFIC IN QUESTION, NOT WHETHER THERE IS A PORT NUMBER          *
* REGISTERED OR NOT.                                                    *
*************************************************************************
```

WELL KNOWN PORT NUMBERS

The Well Known Ports are assigned by the IANA and on most systems can only be used by system (or root) processes or by programs executed by privileged users.

Ports are used in the TCP [RFC793] to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port".

To the extent possible, these same port assignments are used with the UDP [RFC768].

The range for assigned ports managed by the IANA is 0-1023.

Port Assignments:

Keyword	Decimal	Description	References
echo	7/tcp	Echo	
echo	7/udp	Echo	
daytime	13/tcp	Daytime (RFC 867)	
daytime	13/udp	Daytime (RFC 867)	
qotd	17/tcp	Quote of the Day (RFC 865)	
qotd	17/udp	Quote of the Day (RFC 865)	
ftp-data	20/tcp	File Transfer [Default Data]	
ftp-data	20/udp	File Transfer [Default Data]	
ftp	21/tcp	File Transfer [Control]	
ftp	21/udp	File Transfer [Control]	
ssh	22/tcp	SSH Remote Login Protocol	
ssh	22/udp	SSH Remote Login Protocol	
telnet	23/tcp	Telnet	
telnet	23/udp	Telnet	
smtp	25/tcp	Simple Mail Transfer	
smtp	25/udp	Simple Mail Transfer	
time	37/tcp	Time (RFC 1305)	
time	37/udp	Time (RFC 1305)	
domain	53/tcp	Domain Name Server	
domain	53/udp	Domain Name Server	
http	80/tcp	World Wide Web HTTP	
http	80/udp	World Wide Web HTTP	
www	80/tcp	World Wide Web HTTP	
www	80/udp	World Wide Web HTTP	
www-http	80/tcp	World Wide Web HTTP	
www-http	80/udp	World Wide Web HTTP	

REGISTERED PORT NUMBERS

The Registered Ports are listed by the IANA and on most systems can be used by ordinary user processes or programs executed by ordinary users.

Ports are used in the TCP [RFC793] to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port.

The IANA registers uses of these ports as a convenience to the community.

To the extent possible, these same port assignments are used with the

UDP [RFC768].

The Registered Ports are in the range 1024-49151.

Port Assignments:

Keyword	Decimal	Description	References
interwise	7778/tcp	Interwise	
interwise	7778/udp	Interwise	
quake	26000/tcp	quake	
quake	26000/udp	quake	

DYNAMIC AND/OR PRIVATE PORTS

The Dynamic and/or Private Ports are those from 49152 through 65535

Rozdział 6

Przykład

6.1 Analiza przykładowego pakietu

Załóżmy, że udało nam się przechwycić następujący pakiet

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	208	183	116	229	103	0	2	179	101	64	208	8	0	69	0
1	0	56	200	213	64	0	64	6	70	72	212	191	65	33	212	191
2	65	2	5	20	10	109	166	211	34	168	138	91	184	29	80	24
3	22	208	85	26	0	0	84	11	32	106	101	115	116	32	116	101
4	107	115	99	105	107	10	0	0	0	0	0	0	0	0	0	0

Aby móc sprawdzać poprawność analizy dodam jedynie, że wiadomość wysłana była z komputera o adresie IP `212.191.65.33` a odbiorca miał adres `212.191.65.2`. Wiadomością był tekst w jezyku polskim zakodowany przy użyciu standardowych znaków wchodzących w skład kodu ASCII. Adres karty sieciowej nadawcy to `00-02-B3-65-40-D0` odbiorcy zaś
`00-D0-B7-74-E5-67`. Numer portu nadwcy wynosił 1300.

Analizując ten pakiet będe podawał współrzędne liczb w postaci (x,y) co należy czytać: wiersz x, kolumna y. Na przykład zapis (2,6) oznacza 2 wiersz, 6 kolumna i jest tam wpisana liczba 166.

6.1.1 Nagłówek Ethernetu

1. (0,0)-(0,5) – ETH DEST ADDRESS. Są tam zapisane liczby: 0, 208, 183, 116, 229, 103, które po zamianie na liczby zapisane w systemie szesnastkowym dają `00-D0-B7-74-E5-67`. Jest to sprzętowy adres karty sieciowej komputera, do którego adresowany jest pakiet.
2. (0,6)-(0,11) – ETH SRC ADDRESS. Są tam zapisane liczby: 0, 2, 179, 101, 64, 208, które po zamianie na liczby zapisane w systemie szesnastkowym dają `00-02-B3-65-40-D0`. Jest to sprzętowy adres karty sieciowej komputera, którego wysłał pakiet.

3. (0,12)-(0,13) – TYPE CODE. Są tam zapisane liczby 8 i 0 co oznacza, że jako dane przesyłany jest pakiet IP, dlatego następne bajty należy interpretować zgodnie ze specyfikacją pakietu IP.

6.1.2 Nagłówek IP

1. Nagłówek IP rozpoczyna się w (0,14). Jest tam wpisana liczba 69. Liczba ta jest 8-bitowa, zatem opisuje pola VER (4 bity) i IHL (4 bity). Przeliczymy teraz liczbę dziesiętną 69 na jej dwójkową reprezentację.

```
69 (10) = 01000101 (2)

Zatem
0100 (2) = 4 (10) to wartość z pola VER
0101 (2) = 5 (10) to wartość z pola IHL
```

   Wnosimy stąd, że mamy do czynienia ze standardowym nagłówkiem protokołu IP oraz, że całkowita długość nagłówka IP wynosi 20 bajtów (pamiętamy, że długość podawana jest w wielokrotnościach 32 bitów, a więc 4 bajtów). Nagłówek IP powinien wiec skończyć się w (2,1).

2. (0,15) – TOS. Jest tam liczba 0 a więc znaczenie poszczególnych fragmentów tego pola jest następujące

```
P (3 bity)
0 - Routine

D (1 bit)
0 - Normal delay

T (1 bit)
0 - Normal throughput

R (1 bit)
0 - Normal reliability

M (1 bit)
0 - Normal monetary cost
```

3. (1,0) – (1,1) – TL. Są tam wpisane liczby 0 i 56.

```
 0 (10) = 00000000 (2)
56 (10) = 00111000 (2)

0000000000111000 (2) = 56 (10)
```

   Zatem całkowita długość pakietu IP wraz z nagłówkiem wynosi 56 bajtów a więc pakiet powienien kończyć się w (4,5).

4. (1,2) – (1,3) – ID. Są tam wpisane liczby 200 213.

```
200 (10) = 11001000 (2)
213 (10) = 11010101 (2)

1100100011010101 (2) = 51413 (10)
```

5. (1,4) – (1,5) – F (3 bity) , OFFSET (13 bitów). Są tam wpisane liczby 64 i 0.

```
64 (10) = 01000000 (2)
 0 (10) = 00000000 (2)

Zatem
010 (2) to wartość z pola F
  R (1 bit) Zarezerwowany; jego wartość powinna być równa 0 i jest równa 0 :))

  DF (1 bit)
  1 - Do not fragment

  MF (1 bit)
  0 - This is the last fragment
  (faktycznie jest to ostatni fragment, a nawet jedyny :)), gdyż wiadomość
   do najdłuższych nie należy :)) )

0000000000000 (2) = 0 (10) to wartość z pola OFFSET
```

6. (1,6) – TTL. Jest tam wpisana wartość 64. 64 to standardowa wartość nadawana temu polu.
7. (1,7) – PROT. Jest tam wpisana wartość 6 co oznacza, że datagram IP przenosi w sobie datagram TCP.
8. (1,8) – (1,9) – CHECKSUM. Są tam wpisane wartości 70 i 72.

```
70 (10) =  01000110 (2)
72 (10) =  01001000 (2)
```

9. (1,10) – (1,13) – SOURCE IP ADDRESS. Są tam wpisane liczby: 212, 191, 65, 33 co zgodne jest z informacjami podanymi na początku.
10. (1,14) – (2,1) – DESTINATION IP ADDRESS. Są tam wpisane liczby: 212, 191, 65, 2 co zgodne jest z informacjami podanymi na początku. (2,1), godnie z wartością zapisaną w polu IHL, jest ostatnim bajtem nagłówka IP.

6.1.3 Nagłówek TCP

1. (2,2) – (2,3) – SOURCE PORT. Zapisane są tam liczby 5 i 20.

```
 5 (10) = 00000101 (2)
20 (10) = 00010100 (2)

Zatem
0000010100010100 (2) = 1300 (10)
```

2. (2,4) – (2,5) – DEST PORT. Zapisane są tam liczby 10 i 109.

```
 10 (10) = 00001010 (2)
109 (10) = 01101101 (2)

Zatem
0000101001101101 (2) = 2669 (10)
```

3. (2,6) – (2,9) – SEQUENCE NUMBER. Zapisane są tam liczby 166, 211, 34, 168.
4. (2,10) – (2,13) – ACKNOWLEDGMENT NUMBER. Zapisane są tam liczby 138, 91, 184, 29.
5. (2,14) – (2,15) – DO (4 bity) , R (3 bitów), ECN (3 bity), CB (6 bitów). Są tam wpisane liczby 80 i 24.

```
80 (10) = 01010000 (2)
24 (10) = 00011000 (2)

Zatem
0101 (2) = 5 (10) to wartość z pola DO

Wnosimy stąd, że całkowita długość nagłówka TCP wynosi
20 bajtów (pamiętamy, że długość podawana jest
w wielokrotnościach 32 bitów, a więc 4 bajtów).
Nagłówek TCP powinien wiec skończyć się w (3,5).

000 to wartość z pola R i powinna ona wynosić zero.
  DF (1 bit)
  1 - Do not fragment

000 to wartość z pola ECN
  N (1 bit) - 0
  C (1 bit) - 0
  E (1 bit) - 0

011000 to wartość z pola CB.
  U (1 bit) - 0
  A (1 bit) - 1
  P (1 bit) - 1
  R (1 bit) - 0
  S (1 bit) - 0
  F (1 bit) - 0
```

6. (3,0) – (3,1) – WINDOW. Zapisane są tam liczby 22, 208.

```
 22 (10) = 00010110 (2)
208 (10) = 11010000 (2)

Zatem
0001011011010000 (2) = 5840 (10)
```

7. (3,2) – (3,3) – CHECKSUM. Zapisane są tam liczby 85, 26.
8. (3,4) – (3,5) – UP. Zapisane są tam liczby 0, 0. (3,5), godnie z wartością zapisaną w polu DO, jest ostatnim bajtem nagłówka TCP.

6.1.4 Dane

1. (3,6) – (4,5) – DATA. Rozmiaru danych nie trzeba przesyłać, gdyż warstwa TCP otrzymuje tylko „swój” datagram o określonej długości wynikającej z rozmiaru

datagramu IP. Na dane składają się następujące liczby: 84, 111, 32, 106, 101, 115, 116, 32, 116, 101, 107, 115, 99, 105, 107, 10. Ponieważ wiemy, że danymi był tekst, zatem możemy odkodować go. Oto tablica kodów ASCII:

Dec	Hex	Symbol	Dec	Hex	Symbol	Dec	Hex	Symbol	Dec	Hex	Symbol
32	20	spacja	56	38	8	80	50	P	104	68	h
33	21	!	57	39	9	81	51	Q	105	69	i
34	22	"	58	3A	:	82	52	R	106	6A	j
35	23	#	59	3B	;	83	53	S	107	6B	k
36	24	$	60	3C	<	84	54	T	108	6C	l
37	25	%	61	3D	=	85	55	U	109	6D	m
38	26	&	62	3E	>	86	56	V	110	6E	n
39	27	’	63	3F	?	87	57	W	111	6F	o
40	28	(	64	40	@	88	58	X	112	70	p
41	29	)	65	41	A	89	59	Y	113	71	q
42	2A	*	66	42	B	90	5A	Z	114	72	r
43	2B	+	67	43	C	91	5B	[	115	73	s
44	2C	,	68	44	D	92	5C	\	116	74	t
45	2D	-	69	45	E	93	5D	]	117	75	u
46	2E	.	70	46	F	94	5E	^	118	76	v
47	2F	/	71	47	G	95	5F	_	119	77	w
48	30	0	72	48	H	96	60	‘	120	78	x
49	31	1	73	49	I	97	61	a	121	79	y
50	32	2	74	4A	J	98	62	b	122	7A	z
51	33	3	75	4B	K	99	63	c	123	7B	{
52	34	4	76	4C	L	100	64	d	124	7C	|
53	35	5	77	4D	M	101	65	e	125	7D	}
54	36	6	78	4E	N	102	66	f	126	7E	~
55	37	7	79	4F	O	103	67	g	127	7F	Del

Stąd otrzymujemy:

```
 84 - T
111 - o
 32 - spacja
106 - j
101 - e
115 - s
116 - t
 32 - spacja
116 - t
101 - e
107 - k
115 - s
 99 - c
105 - i
107 - k
 10 - '\n'
```

Unix:

```
'\n' = Ctrl + J = LF ang. line feed

DOS/WINDOWS:
'\n' = Ctrl + J = LF ang. line feed +
       Ctrl + M = CR ang. carriage return
```

Rozdział 7

Ćwiczenia

7.1 Zestaw 1

Ćwiczenie 7.1. *Ustalić czy adres 112.224.0.0 z maską 255.224.0.0 jest:*

1. *adresem sieci,*
2. *adrsem hosta,*
3. *broadcastem.*

```
112.224.0.0 = 01110000 11100000 00000000 00000000
255.224.0.0 = 11111111 11100000 00000000 00000000

adres jest adresem sieci
```

Ćwiczenie 7.2. *Jaki jest adres sieci dla hosta 201.100.5.33/27?*

```
201.100.5.33 = 11001001 01100100 00000101 00100001
               11111111 11111111 11111111 11100000 = 255.255.255.224
               11001001 01100100 00000101 00100000 = 201.100.5.32
adres sieci to: 201.100.5.32
```

Ćwiczenie 7.3. *Sieć o adresie 192.112.0.0/16 ma być podzielona na podsieci, z których każda musi pozwolić na zaadresowanie 238 urządzeń. Jaka będzie maska podsieci, przy założeniu, że chcemy uzyskać maksymalną ilość takich podsieci?*

```
192.112.0.0 = 11000000 01110010 00000000 00000000
              11111111 11111111 00000000 00000000 = 255.255.0.0

Na zapisanie 238 różnych adresów potrzeba co najmniej 8 bitów.
Tak więc ostatnie 8 bitów rezerwujemy na adresy hostów i w konsekwencji
maska przyjmie postać: 255.255.255.0
```

Ćwiczenie 7.4. *Określić, które z adresów mogą być użyte do zaadresowania hostów przy masce podsieci 255.255.255.240*

1. *17.61.12.31*
2. *17.61.12.93*
3. *17.61.12.144*
4. *17.61.12.33*
5. *17.61.12.56*
6. *17.61.12.192*

```
255.255.255.240 = 11111111 11111111 11111111 11110000

17.61.12.31     = 00010001 00111101 00001100 00011111 - N (host. addr. = same 1)
17.61.12.93     = 00010001 00111101 00001100 01011101 - T
17.61.12.144    = 00010001 00111101 00001100 10010000 - N (host. addr. = same 0)
17.61.12.33     = 00010001 00111101 00001100 00100001 - T
17.61.12.56     = 00010001 00111101 00001100 00111000 - T
17.61.12.192    = 00010001 00111101 00001100 11000000 - N (host. addr. = same 0)
```

Ćwiczenie 7.5. *Mając do dyspozycji sieć o adresie 200.5.4.128 z maską 255.255.255.128 zaadresować elementy w sieci według schematu. Sieć S1 musi umożliwić zaadresowanie 60 urządzeń a sieć S2: 30. Proszę podać 3: adresy sieci, maski, broadcasty i zakresy dostępnych adresów dla hostów.*

```
| PC11------|         |
|           |         |
| PC 12-----+-----Hub-+-Router1 ====== Router2 -+-S2
|           |         |
| PC 13-----+         |
|                     |
| ...                 |
+---------------------+
|         S1          |

200.5.4.128     = 11001000 00000101 00000100 10000000
255.255.255.128 = 11111111 11111111 11111111 10000000
zatem do dyspozycji mamy 7 bitów
Na 60 urządzeń potrzeba 6 bitów, co daje maskę:
                  11111111 11111111 11111111 11000000 = 255.255.255.192
Na 30 urządzeń potrzeba 5 bitów, co daje maskę:
                  11111111 11111111 11111111 11100000 = 255.255.255.224

Stąd dla sieci S1 mamy:
Adres sieci:      11001000 00000101 00000100 10000000 = 200.5.4.128
Maska sieci:      11111111 11111111 11111111 11000000 = 255.255.255.192
Breadcast:        11001000 00000101 00000100 10111111 = 200.5.4.191
Zakres adresów:   11001000 00000101 00000100 10000001 = 200.5.4.129 do
                  11001000 00000101 00000100 10111110 = 200.5.4.190

Stąd dla sieci S2 mamy:
```

```
Adres sieci:      11001000 00000101 00000100 11000000 = 200.5.4.192 (?)
Maska sieci:      11111111 11111111 11111111 11100000 = 255.255.255.224
Breadcast:        11001000 00000101 00000100 11011111 = 200.5.4.223 (?)
Zakres adresów:   11001000 00000101 00000100 11000001 = 200.5.4.193 do
                  11001000 00000101 00000100 11011110 = 200.5.4.222

Stąd dla sieci S3 (reszta czyli pozostałe 32 adresy: mieliśmy 128,
użyliśmy 64 + 32) mamy:
Adres sieci:      11001000 00000101 00000100 11100000 = 200.5.4.224
Maska sieci:      11111111 11111111 11111111 11100000 = 255.255.255.224
Breadcast:        11001000 00000101 00000100 11111111 = 200.5.4.255
Zakres adresów:   11001000 00000101 00000100 11100001 = 200.5.4.225 do
                  11001000 00000101 00000100 11111110 = 200.5.4.254
```

Ćwiczenie 7.6. *Komputery z sieci, w której znajduje się PC1 nie są w stanie komunikować się z serwerem S w sieci odległej. Zakładając, że hosty są zaadresowane następująco*

```
PC1---Switch---E0+Router1+S0======S1+Router1+E1---Switch---S

S:
IP: 100.1.1.96/28
Brama: 100.1.1.97

PC1:
IP: 100.1.1.18/28
Brama: 100.1.1.17

Router1:
E0: 100.1.1.17/28
S0: 100.1.1.49/28

Router2:
E1: 100.1.1.97/28
S1: 100.1.1.50/28
```

określić czy przyczyną braku komunikacji jest:

1. *niepoprawna brama PC1?*
2. *niepoprawny adres serwera? (tak)*
3. *niepoprawna brama serwera?*
4. *Serial 0 dla Routera 1 i S1 dla Routera 2 nie są z tej samej sieci?*

7.2 Zestaw 2

Ćwiczenie 7.7. *Ustalić czy adres 112.0.0.0 z maską 240.0.0.0 jest:*

1. *adresem sieci,*
2. *adrsem hosta,*

3. *broadcastem.*

```
112.0.0.0 = 01110000 00000000 00000000 00000000
240.0.0.0 = 11110000 00000000 00000000 00000000

adres jest adresem sieci
```

Ćwiczenie 7.8. *Jaki jest adres sieci dla hosta 201.100.5.99/27?*

```
201.100.5.99 = 11001001 01100100 00000101 01100011
               11111111 11111111 11111111 11100000 = 255.255.255.224
               11001001 01100100 00000101 01100000 = 201.100.5.96
adres sieci to: 201.100.5.96
```

Ćwiczenie 7.9. *Sieć o adresie 172.12.0.0/16 ma być podzielona na podsieci, z których każda musi pozwolić na zaadresowanie 458 urządzeń. Jaka będzie maska podsieci, przy założeniu, że chcemy uzyskać maksymalną ilość takich podsieci?*

```
172.12.0.0 = 10101100 00001100 00000000 00000000
             11111111 11111100 00000000 00000000 = 255.252.0.0

Na zapisanie 458 różnych adresów potrzeba co najmniej 9 bitów.
Tak więc ostatnie 9 bitów rezerwujemy na adresy hostów i w konsekwencji
maska przyjmie postać: 255.255.254.0
```

Ćwiczenie 7.10. *Określić, które z adresów mogą być użyte do zaadresowania hostów przy masce podsieci 255.255.255.224*

1. *217.63.12.192*
2. *217.63.12.93*
3. *217.63.12.159*
4. *217.63.12.56*
5. *217.63.12.63*
6. *217.63.12.87*

```
255.255.255.224 = 11111111 11111111 11111111 11100000

217.63.12.192   = 11011001 00111111 00001100 11000000 - N (host. addr. = same 0)
217.63.12.93    = 11011001 00111111 00001100 01011101 - T
217.63.12.159   = 11011001 00111111 00001100 10011111 - N (host. addr. = same 1)
217.63.12.56    = 11011001 00111111 00001100 00111000 - T
217.63.12.63    = 11011001 00111111 00001100 00111111 - N (host. addr. = same 1)
217.63.12.87    = 11011001 00111111 00001100 01010111 - T
```

Ćwiczenie 7.11. *Mając do dyspozycji sieć o adresie 200.5.4.0 z maską 255.255.252.0 zaadresować elementy w sieci według schematu. Sieć S1 musi umożliwić zaadresowanie 500 urządzeń a sieć S2: 200. Proszę podać 3: adresy sieci, maski, broadcasty i zakresy dostępnych adresów dla hostów.*

```
| PC11------|         |
|           |         |
| PC 12-----+-----Hub-+-Router1 ====== Router2 -+-S2
|           |         |
| PC 13-----+         |
|                     |
| ...                 |
+---------------------+
|         S1          |
```

```
200.5.4.0     = 11001000 00000101 00000100 00000000
255.255.252.0 = 11111111 11111111 11111100 00000000
zatem do dyspozycji mamy 10 bitów
Na 500 urządzeń potrzeba 9 bitów, co daje maskę:
                  11111111 11111111 11111110 00000000 = 255.255.254.0
Na 200 urządzeń potrzeba 8 bitów, co daje maskę:
                  11111111 11111111 11111111 00000000 = 255.255.255.0

Stąd dla sieci S1 mamy:
Adres sieci:      11001000 00000101 00000100 00000000 = 200.5.4.0
Maska sieci:      11111111 11111111 11111110 00000000 = 255.255.254.0
Breadcast:        11001000 00000101 00000101 11111111 = 200.5.5.255
Zakres adresów:   11001000 00000101 00000100 00000001 = 200.5.4.1 do
                  11001000 00000101 00000101 11111110 = 200.5.5.254

Stąd dla sieci S2 mamy:
Adres sieci:      11001000 00000101 00000110 00000000 = 200.5.6.0
Maska sieci:      11111111 11111111 11111111 00000000 = 255.255.255.0
Breadcast:        11001000 00000101 00000110 11111111 = 200.5.6.255
Zakres adresów:   11001000 00000101 00000110 00000001 = 200.5.6.1 do
                  11001000 00000101 00000110 11111110 = 200.5.6.254

Stąd dla sieci S3 (reszta czyli pozostałe 256 adresy: mieliśmy 1024,
użyliśmy 512 + 256) mamy:
Adres sieci:      11001000 00000101 00000111 00000000 = 200.5.7.0
Maska sieci:      11111111 11111111 11111111 00000000 = 255.255.255.0
Breadcast:        11001000 00000101 00000111 11111111 = 200.5.7.255
Zakres adresów:   11001000 00000101 00000111 00000001 = 200.5.7.1 do
                  11001000 00000101 00000111 11111110 = 200.5.7.254
```

Ćwiczenie 7.12. *Komputery z sieci, w której znajduje się PC1 nie są w stanie komunikować się z serwerem S w sieci odległej. Zakładając, że hosty są zaadresowane następująco*

```
PC1---Switch---E0+Router1+S0======S1+Router1+E1---Switch---S

S:
IP: 200.1.1.96/28
Brama: 200.1.1.97
```

```
PC1:
IP: 200.1.1.18/28
Brama: 200.1.1.17

Router1:
E0: 200.1.1.17/28
S0: 200.1.1.49/28

Router2:
E1: 200.1.1.97/28
S1: 200.1.1.50/28
```

określić czy przyczyną braku komunikacji jest:

1. *niepoprawna brama PC1?*
2. *niepoprawny adres serwera? (tak)*
3. *niepoprawna brama serwera?*
4. *Serial 0 dla Routera 1 i S1 dla Routera 2 nie są z tej samej sieci?*

Ćwiczenie 7.13. *Ustalić czy adres 112.0.0.0 z maską 240.0.0.0 jest:*

1. *adresem sieci,*
2. *adrsem hosta,*
3. *broadcastem.*

Ćwiczenie 7.14. *Sieć o adresie 172.12.0.0/16 ma być podzielona na podsieci, z których każda musi pozwolić na zaadresowanie 458 urządzeń. Jaka będzie maska podsieci, przy założeniu, że chcemy uzyskać maksymalną ilość takich podsieci?*

Ćwiczenie 7.15. *Określić, które z adresów mogą być użyte do zaadresowania hostów przy masce podsieci 255.255.255.224*

1. *217.63.12.192*
2. *217.63.12.93*
3. *217.63.12.159*
4. *217.63.12.56*
5. *217.63.12.63*
6. *217.63.12.87*

Ćwiczenie 7.16. *Mając do dyspozycji sieć o adresie 200.5.4.0 z maską 255.255.252.0 zaadresować elementy w sieci według schematu. Sieć S1 musi umożliwić zaadresowanie 500 urządzeń a sieć S2: 200. Proszę podać 3: adresy sieci, maski, broadcasty i zakresy dostępnych adresów dla hostów.*

```
S1-+-Router1 ====== Router2 -+-S2
```

Ćwiczenie 7.17. *Komputery z sieci, w której znajduje się PC1 nie są w stanie komunikować się z serwerem S w sieci odległej. Zakładając, że hosty są zaadresowane następująco*

```
PC1---Switch---E0+Router1+S0======S1+Router1+E1---Switch---S

S:                          Router1:
IP: 200.1.1.96/28           E0: 200.1.1.17/28
Brama: 200.1.1.97           S0: 200.1.1.49/28

PC1:                        Router2:
IP: 200.1.1.18/28           E1: 200.1.1.97/28
Brama: 200.1.1.17           S1: 200.1.1.50/28
```

określić czy przyczyną braku komunikacji jest:

1. *niepoprawna brama PC1?*
2. *niepoprawny adres serwera?*
3. *niepoprawna brama serwera?*
4. *Serial 0 dla Routera 1 i S1 dla Routera 2 nie są z tej samej sieci?*

Bibliografia

[1] Craig Hunt, *TCP/IP*, wydanie III, Wydawnictwo RM, Warszawa, 2003.

[2] Heather Osterloch, *TCP IP Szkoła programowania*, Helion, Gliwice, 2006.

[3] Andrew S. Tanenbaum, *Sieci komputerowe*, Wydawnictwo HELION, 2004.

[4] RFC 791, *Internet Protocol*.

[5] RFC 793, *Transmission Control Protocol*.

[6] RFC 1122, *Requirements for Internet Hosts - Communication Layers*.

[7] RFC 1323, *TCP Extensions for High Performance*.

[8] RFC 1518, *An Architecture for IP Address Allocation with CIDR*.

[9] RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*.

[10] RFC 1520, *Exchanging Routing Information Across Provider Boundaries in the CIDR Environment*.

[11] RFC 1826, *IP Authentication Header*.

[12] RFC 1827, *IP Encapsulating Security Payload (ESP)*.

[13] RFC 1883, *Internet Protocol, Version 6 (IPv6) Specification*.

[14] RFC 1884, *IP Version 6 Addressing Architecture*.

[15] RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*.

[16] RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture*.

[17] RFC 3587, *IPv6 Global Unicast Address Format*.