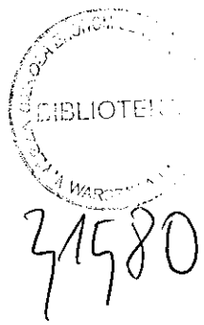
Projekt okładki: Zbigniew Szmigielski

Recenzent: prof. dr hab. inż. Leszek Jung

Redaktor: Maria Stefańska-Matuszyn

Fotografia na okładce: Wojciech Dittwald

ISBN 83-87444-57-X



Skład i łamanie: „Stilus"

Druk: „Soft Script" Warszawa

Iwona Dolińska

Sieci komputerowe

Wyższa Szkoła Ekonomiczno-Informatyczna
Warszawa 2005

Przedmowa

Niniejszy skrypt został opracowany jako pomoc dydaktyczna do trzydziestogodzinnego wykładu i ćwiczeń. Jego celem jest zaznajomienie zainteresowanych problematyką sieci komputerowych z podstawowymi zasadami budowy tych sieci oraz z ich oprogramowaniem, umożliwiającym przepływ informacji. Podane tu są także podstawowe zasady bezpieczeństwa, związane z pracą w sieci.

Zebrany materiał został podzielony na dwie zasadnicze części. Pierwsza – zawarta w rozdziałach 1–6 – omawia model warstwowy sieci, najważniejsze technologie warstwy fizycznej (sieci LAN i WAN) oraz rodzinę protokołów TCP/IP, która jest podstawą działania Internetu. Rodzina protokołów TCP/IP jest omówiona na przykładzie podstawowych protokołów tworzących poszczególne warstwy oprogramowania sieciowego. Druga część, obejmująca rozdziały 7 i 8, dotyczy różnych aspektów zapewniania bezpieczeństwa sieciom komputerowym i przesyłanym za ich pomocą danym.

Tematyka sieci komputerowych jest bardzo obszernym działem informatyki. Z konieczności, wynikającej z przeznaczonej na przedmiot liczby godzin, publikacja ta zawiera starannie wybrany zestaw tematów. Podany na końcu przegląd bibliograficzny ma ułatwić zainteresowanym poszczególnymi zagadnieniami poszerzenie wiadomości (w skrypcie znajdują się liczne odwołania do jego zawartości). Zawiera on osobno spis publikacji książkowych i artykułów z czasopism, uporządkowany według nazwisk autorów, oraz osobno spis publikacji zamieszczonych w Internecie na stronach WWW. Druga grupa obejmuje zarówno dokumenty RFC (będące oficjalnie zatwierdzonymi standardami), jak i artykuły z periodyków internetowych i stron prywatnych.

Autorka zakłada, że podstawowe terminy informatyczne są Czytelnikowi znane. Przyjęta w informatyce terminologia powstała w języku angielskim, dlatego wszystkie użyte terminy są także podane w tym języku w nawiasach i wyróżnione kursywą, a cyfry w nawiasach kwadratowych odwołują się do odpowiednich pozycji literatury umieszczonych w bibliografii na końcu książki.

1. Wprowadzenie

Szybki rozwój możliwości sieci komputerowych w zakresie rodzajów, szybkości i objętości przesyłanej informacji spowodował, że dziś są one niezbędnym narzędziem pracy w najróżniejszych instytucjach, organizacjach, a także w szkołach. Również użytkownicy indywidualni połączeni z Internetem korzystają, często nieświadomie, z tej największej na świecie sieci sieci (czyli sieci składającej się z wielu sieci – rozdz. 5.2.).

Chociaż pojęcie „sieć komputerowa" zapewne wydaje się oczywiste, to warto je jednak doprecyzować, jako termin podstawowy dla niniejszego skryptu. Definicje sieci komputerowych podawane w różnych źródłach nieco różnią się od siebie, zależnie od tego, jaki element sieci jest najważniejszy dla danego opracowania. Na potrzeby tego skryptu przyjęto wersję następującą: **sieć komputerowa** jest to system komunikacyjny, służący do przesyłania danych (czyli wymiany informacji), łączący dwa lub więcej autonomicznych komputerów i urządzenia peryferyjne (takie jak drukarki, skanery, pamięci masowe). Składa się on z zasobów obliczeniowych i informacyjnych, mediów transmisyjnych i urządzeń sieciowych.

1.1. Krótka historia Internetu

Hasło *sieć komputerowa* kojarzy się większości osób przede wszystkim z Internetem. Dlatego zapewne warto rzucić okiem na kilka najważniejszych faktów z niespełna półwiecznej historii jego burzliwego rozwoju [38]*:

1957 U początków Internetu pojawia się wątek rywalizacji technologicznej pomiędzy USA i ówczesnym Związkiem Radzieckim: w odpowiedzi na wystrzelenie przez ZSRR sputnika USA powołują agencję ARPA (*Advanced Research Project Agency*), która później walnie przyczynia się do powstania Internetu i jego początkowego rozwoju.

1969 Powstaje ARPANET, sieć złożona z czterech komputerów, aby do 1973 roku rozrosnąć się do 35 komputerów. Od początku jest ona wykorzystywana do komunikacji między naukowcami i wspólnej pracy nad projektami.

1971 Powstaje poczta elektroniczna, pozwalająca przesyłać wiadomości tekstowe.

1974 Po raz pierwszy pojawia się słowo *Internet* w opracowaniu badawczym dotyczącym protokołu TCP, napisanym przez Vintona Cerfa („ojciec Internetu") i Boba Kahna *A Protocol for Packet Intercommunication.*

* Patrz: bibliografia na końcu książki

1979 Powstaje Usenet, czyli tekstowe grupy dyskusyjne.

1981 Ted Nelson proponuje Xanadu – hipertekstową bazę danych, zawierającą informacje wszelkiego rodzaju, z płatnym dostępem. Koncepcja ta, co prawda, nigdy nie zostaje zrealizowana, ale za jej dalekiego potomka możemy uznać World Wide Web.

1989 Sieć ARPANET formalnie przestaje istnieć, mimo to Internet pomyślnie rozwija się dalej.

1990 Tim Berners-Lee tworzy World Wide Web, system pozwalający autorom publikacji internetowych na połączenie słów, zdjęć i dźwięku, początkowo pomyślany dla wsparcia naukowców zajmujących się fizyką w CERN. Projekt World Wide Web powstaje na komputerze NeXT, w pierwszej odsłonie umożliwia jednocześnie przeglądanie i edycję hipertekstowych dokumentów. Z CERN rozpowszechnia się na cały świat.

1991 Następuje pierwsza wymiana poczty elektronicznej pomiędzy Polską a zagranicą, uważana za początek Internetu w Polsce.

1993 Pojawia się Mosaic, pierwsza przeglądarka graficzna stron WWW.

W tym czasie równolegle opracowywano kolejne standardy sieciowe i protokoły wymiany danych, które będą sukcesywnie omawiane w ramach niniejszego skryptu.

1.2. Standaryzacja sieci

Istnieje tak wielu producentów sprzętu i dostawców usług sieciowych, że standaryzacja technologii sieciowych jest działaniem niezbędnym. Bez koordynacji działań zapanowałby kompletny chaos. Współdziałanie produktów różnych producentów wymaga, aby różne platformy wiedziały, w jaki sposób mogą się ze sobą komunikować. Wymogi te przyczyniły się do rozwoju uniwersalnych standardów, dotyczących każdego aspektu sieciowego przetwarzania danych.

Potrzeba standaryzacji przyczyniła się też do wyłonienia organizacji międzynarodowych, zajmujących się normalizowaniem. Ich celem jest ustanawianie jak najbardziej uniwersalnego zbioru standardów. Organizacje w miarę potrzeb współpracują ze sobą, choć dziedziny ich zainteresowań są różne.

Organizacja **ISO** (*International Standards Organization*) jest międzynarodową organizacją normalizacyjną, do której należą krajowe organizacje normalizacyjne krajów członkowskich (np. Polski Komitet Normalizacyjny). ISO wydaje standardy w ogromnej liczbie dziedzin. W dziedzinie sieci komputerowych organizacja ta opracowała model odniesienia OSI, omówiony szczegółowo w rozdz. 1.5. W kwestiach standardów telekomunikacyjnych ISO współpracuje często z ITU-T.

Standardy telekomunikacyjne (istotne m.in. dla sieci rozległych WAN – rozdz. 4) opracowywane są przez organizację **ITU** (*International Telecommunication Union*). Dzieli się ona na trzy główne sektory:

- ITU-R – sektor radiokomunikacji (*Radiocommunication Sector*),
- ITU-T – sektor standaryzacji telekomunikacji (*Telecommunications Standarization Sector*),
- ITU-D – sektor badań naukowych (*Development Sector*).

Z punktu widzenia sieci komputerowych najważniejsze są normy ITU-T, dotyczące systemów telefonicznych i transmisji danych. W latach od 1956 do 1993 organizacja ta znana była jako CCITT (od jej nazwy francuskiej) i ten skrót jeszcze często bywa używany w nazwach norm (choć już jest nieaktualny).

Instytut Elektryków i Elektroników **IEEE** (*The Institute of Electrical and Electronic Engineers*) jest odpowiedzialny za definiowanie standardów: telekomunikacyjnych oraz przesyłania danych. Wynikiem pracy tej organizacji są m.in. standardy sieci LAN i MAN, określane ogólnie jako seria standardów 802. W dalszej części skryptu zostaną bardziej szczegółowo opisane standardy: 802.3, dotyczący sieci Ethernet (rozdz. 3.4.) i 802.11, dotyczący sieci Wi-Fi (rozdz. 3.5.).

Standardy dotyczące pracy warstwy fizycznej oraz łącza danych (rozdz. 1.5.) opracowują także wspólnie organizacje **EIA** (*Electronic Industries Association*) oraz **TIA** (*Telecommunications Industry Association*). Kiedyś organizacja EIA oznaczała wszystkie swoje standardy przedrostkiem „RS" (*Recommended Standard*). Wielu inżynierów nadal używa tego oznaczenia, choć oficjalnie zostało ono zastąpione przez „EIA/TIA" w celu ułatwienia identyfikacji pochodzenia standardu. Przykładowo standard EIA/TIA-568 B definiuje parametry elektryczne i sposób tworzenia okablowania strukturalnego sieci LAN, co opisano szczegółowo w [1] oraz [7]. Standardy EIA/TIA-232 (dawniej RS-232) oraz EIA/TIA-449 określają m.in. szybkości transmisji oraz sposób przesyłania danych w sieciach WAN ([1], [14]).

Standardy dotyczące rozwoju sieci Internet są tworzone pod auspicjami organizacji **IETF** (*Internet Engineering Task Force*) wchodzącej w skład organizacji **IAB** (*The Internet Architecture Board*). IETF działa w tzw. grupach roboczych, które projektują i proponują do zatwierdzenia nowe standardy dotyczące pracy Internetu (lub rozszerzenia czy modyfikacje istniejących standardów). Wszystkie standardy internetowe są zdefiniowane za pomocą otwartych, ogólnodostępnych specyfikacji. Nadesłane do IETF lub tam opracowane dokumenty zyskują nazwę RFC (*Request for Comments*) i kolejny numer. W związku z tym rozszerzenia istniejących standardów mają numery zupełnie niezwiązane z numerami oryginalnych specyfikacji. Dokumentowi RFC, przedstawionemu do zatwierdzenia przez IETF, nadawane są kolejne stany.

Początkowo dokument zyskuje status „Internet-Draft", kiedy nie jest jeszcze oficjalną publikacją, ale jest przedstawiony wszystkim zainteresowanym do dyskusji. Na taki dokument nie można się powoływać w żadnych innych publikacjach ani informacjach technicznych.

Dokument, który zyskał odpowiednią aprobatę jako „Internet-Draft" może zostać dalej przedstawiony do zatwierdzenia jako Standard Internetowy. Następnie przechodzi przez trzy kolejne poziomy zatwierdzania:

- Proposed Standard
- Draft Standard
- Internet Standard

Aby mogła zostać przeniesiona na kolejny poziom, specyfikacja musi spełniać wiele ściśle określonych warunków.

Proces zatwierdzania standardów jest określony w dokumencie RFC 2026 *The Internet Standards Process – Revision 3* [23],

Informacje o dostępnych dokumentach RFC można znaleźć pod adresem organizacji IETF http://www.ietf.org.

1.3. Powody tworzenia sieci komputerowych

Pomimo że powody tworzenia sieci komputerowych wydają się oczywiste, warto najważniejsze z nich usystematyzować.

Współużytkowanie programów i plików – program i jego pliki danych przechowywane są na serwerze plików, a dostęp do nich ma wielu użytkowników sieci.

Współużytkowanie sprzętowych zasobów sieci – sprzętowe zasoby sieci to m.in. drukarki, plotery oraz urządzenia pamięci masowych, do których mają dostęp użytkownicy sieci.

Współużytkowanie programowych zasobów sieci, np. baz danych – system zarządzania bazami danych jest idealną aplikacją dla sieci. Możliwość blokowania rekordów pozwala wielu użytkownikom na jednoczesny dostęp do pliku bez niszczenia danych. Blokowanie rekordów umożliwia edycję rekordu wyłącznie jednemu użytkownikowi w danym momencie.

Grupy robocze – sieć pozwala na tworzenie grup użytkowników, niekoniecznie pracujących w tym samym wydziale. Umożliwia to tworzenie nowych struktur poziomych, w których osoby z różnych i odległych wydziałów uczestniczą w tym samym projekcie.

Wymiana informacji i wiadomości – poczta elektroniczna (*e-mail*) pozwala użytkownikom na łatwe komunikowanie się. Wiadomości umieszczone są w „skrzynkach pocztowych", umożliwiając ich odczyt w dowolnym czasie.

Ułatwienie zarządzania zasobami – sieć umożliwia zgrupowanie serwerów oraz przechowywanych na nich danych (zasobów). Udoskonalenie sprzętu, archiwizacja danych, utrzymanie systemu i jego ochrona są łatwiejsze w realizacji, gdy urządzenia zgrupowane są w jednym miejscu lub gdy można nimi zdalnie zarządzać z jednego miejsca.

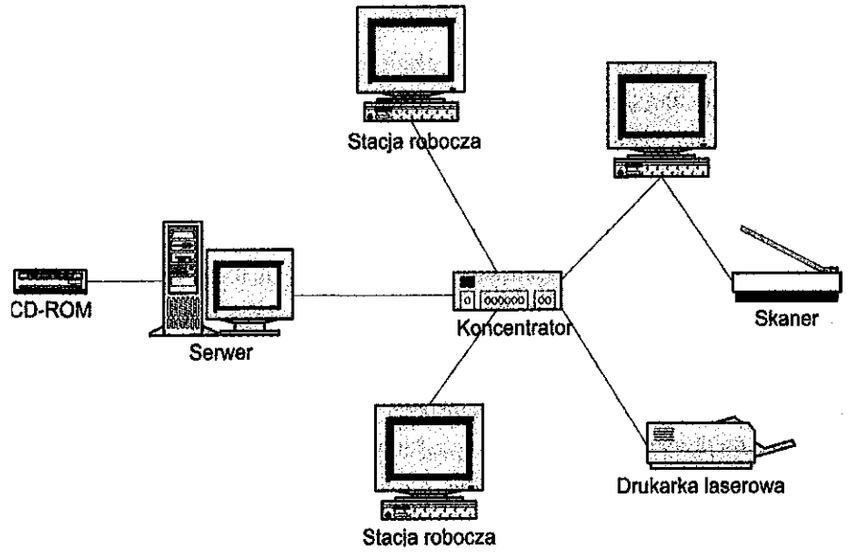
Rozwój organizacji – sieci mogą zmieniać strukturę organizacyjną firmy i sposób jej zarządzania. Użytkownicy pracujący w określonych wydziałach nie muszą przebywać w jednym miejscu. Ich biura mogą być ulokowane tam, gdzie jest to najbardziej uzasadnione. Sieć łączy ich ze współpracownikami z tego samego wydziału.

1.4. Zasięg sieci komputerowych

Ze względu na zasięg terytorialny sieci komputerowych dokonano ich podziału na trzy podstawowe kategorie:

Sieć lokalna LAN (*Local Area Network*) – sieć obejmująca swoim zasięgiem budynek lub kilka sąsiadujących budynków, charakteryzująca się dużą szybkością transmisji i niskim poziomem błędów. Umożliwia ona wielu użytkownikom dostęp do mediów komunikacyjnych, zapewnia połączenia z lokalnymi usługami, łączy sąsiadujące ze sobą urządzenia (stacje robocze i urządzenia peryferyjne: drukarki, skanery, pamięci masowe itp.) [1]. Zwykle jest własnością jednego operatora.

Rys. 1. Przykład sieci lokalnej, łączącej stacje robocze i urządzenia peryferyjne*

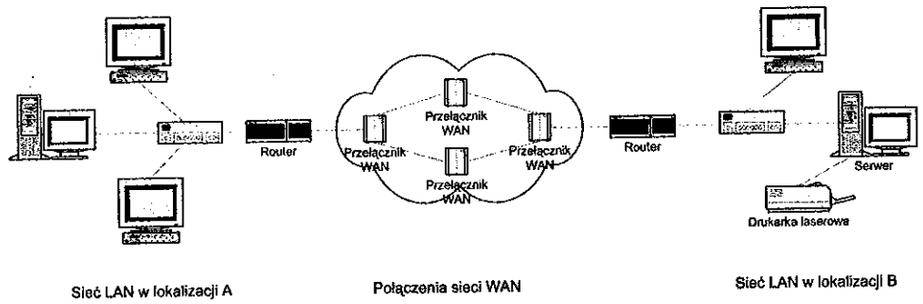


* Źródło: jeśli nie podano inaczej – opracowanie własne autorki.

Sieć metropolitalna MAN (*Metropolitan Area Network*) – sieć obejmująca całe miasto, często wykorzystująca do połączeń gotową infrastrukturę telekomunikacyjną na terenie danego miasta.

Sieć rozległa WAN (*Wide Area Network*) – sieć o zasięgu nawet ogólnoświatowym, przekraczająca granice miast, a często także państw i kontynentów, łącząca ze sobą sieci lokalne, co zapewnia dostęp do umieszczonych w nich plików i komputerów. Wykorzystuje istniejące rozwiązania telekomunikacyjne z systemem satelitarnym włącznie. Sieć o bardzo dużej niezawodności, osiąganej przez stosowanie wielu dróg komunikacyjnych (rezerwowych). Zwykle jest własnością wielu operatorów.

Rys. 2. Przykład sieci WAN, łączącej sieci LAN w różnych lokalizacjach



Z zasięgiem sieci ściśle związana jest m.in. wykorzystywana w nich technologia przesyłania danych. W kolejnych rozdziałach zostaną omówione podstawowe technologie stosowane w sieciach lokalnych (Ethernet, Wi-Fi) i rozległych (Frame Relay, ATM).

Z czasem do trzech podstawowych kategorii sieci dołączono następne, z których najważniejsze i najczęściej spotykane w literaturze to:

Sieć kampusowa (*Campus Network*) – sieć obejmująca swoim zasięgiem kampus uniwersytecki, wykorzystująca technologie sieci LAN i MAN.

Sieć korporacyjna (*Enterprise Network*) – sieć łącząca systemy komputerowe wewnątrz jednej organizacji (składającej się z wielu oddziałów) bez względu na położenie geograficzne, sprzęt komputerowy, systemy operacyjne i protokoły komunikacyjne. Może łączyć w sobie sieci LAN, MAN jak i WAN.

Sieć domowa (*Home Network*) – sieć łącząca urządzenia domowe zdolne do komunikacji z innymi urządzeniami, np. komputery, urządzenia peryferyjne, sprzęt audio/wideo, telefony, a w przyszłości także sprzęt AGD, mierniki mediów itp. Jest to odmiana sieci LAN, dedykowana do konkretnych zastosowań [15].

Sieć Intranet – sieć oparta na technologii sieci Internet, łącząca wszystkie sieci wewnętrzne firmy i służąca do udostępniania zasobów firmy (dokumentów, aplikacji itp.) [14]. Cechą charakterystyczną sieci Intranet jest ścisłe ograniczenie dostępu do niej uprawnionym użytkownikom, m.in. przez zastosowanie sposobów ochrony sieci i danych, opisanych w rozdz. 7 i 8.

Sieć Extranet – sieć selektywnie integrująca dwie lub więcej sieci Intranet w celu realizacji określonych działań gospodarczych [14]. Również w przypadku tych sieci muszą być ściśle przestrzegane zasady bezpieczeństwa (rozdz. 7 i 8).

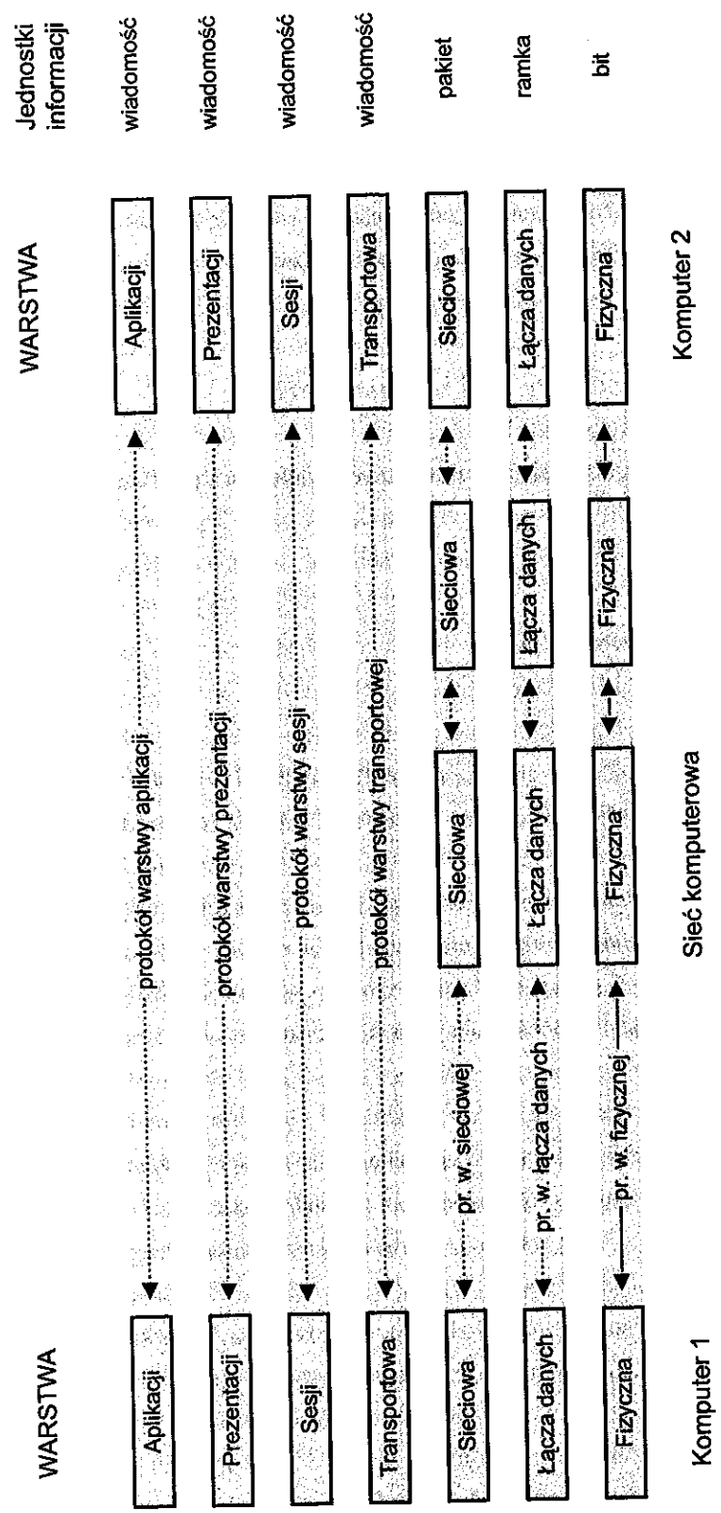
1.5. Model warstwowy sieci komputerowej

Poniżej zostaną przedstawione dwa najważniejsze modele warstwowe sieci komputerowej. Model OSI reprezentuje podejście teoretyczne, wynikające z przeprowadzonych badań naukowych. Model TCP/IP jest w pełni zaimplementowaną rodziną protokołów, będącą podstawą działania całego Internetu.

1.5.1. Model odniesienia OSI

Model odniesienia OSI (*Open Systems Interconnection*) został opracowany przez organizację ISO w ramach prac prowadzących do międzynarodowej standaryzacji rozmaitych protokołów sieci komputerowych [15]. Jest on zbiorem zasad komunikowania się urządzeń i programów sieciowych. Podzielony jest na siedem warstw (rys. 3), z których każda zbudowana jest na bazie warstwy poprzedniej. Model ten nie określa fizycznej budowy poszczególnych warstw, a koncentruje się na sposobach ich współpracy. Takie podejście do problemu sprawia, że każda warstwa może być implementowana przez producenta na swój sposób, a urządzenia sieciowe od różnych dostawców będą zgodnie współpracować. Poszczególne warstwy sieci stanowią niezależne całości i chociaż nie potrafią wykonywać żadnych widocznych zadań w odosobnieniu od pozostałych warstw, to z punktu widzenia oprogramowania są one odrębnymi poziomami.

Dla każdej warstwy powinien zostać stworzony własny **protokół komunikacyjny**, czyli zasady porozumiewania się nadawcy (oznaczonego na rys. 3 jako *Komputer 1*) i odbiorcy (oznaczonego na rys. 3 jako *Komputer 2*) danych. Komunikacja między komputerami odbywa się tylko na poziomie odpowiadających sobie warstw. Oznacza to, że oprogramowanie warstwy aplikacji nadawcy komunikuje się zawsze z oprogramowaniem warstwy aplikacji odbiorcy, oprogramowanie warstwy transportowej zaś z oprogramowaniem warstwy transportowej itd.



Rys. 3. Warstwy modelu OSI i komunikacja między nimi [15]

Rzeczywista komunikacja w sieci komputerowej odbywa się wyłącznie na poziomie pierwszej warstwy, czyli warstwy fizycznej. Na rysunku 3 obrazuje to linia ciągła, symbolizująca kanał komunikacyjny od nadawcy przez sieć komputerową do odbiorcy.

Pomiędzy wyższymi warstwami modelu OSI zachodzi tzw. komunikacja wirtualna, przedstawiona w postaci linii przerywanej. Oznacza to, że informacja wysyłana do sieci przez oprogramowanie dowolnej warstwy wyższej niż pierwsza, jest każdorazowo przekazywana do kolejnej sąsiedniej niższej warstwy aż dotrze do warstwy fizycznej. W warstwie fizycznej informacja jest przesyłana przez sieć. U odbiorcy informacja pokonuje drogę odwrotną. Z warstwy fizycznej jest przekazywana kolejno do sąsiedniej warstwy wyższej, aż dotrze do takiej samej warstwy, jaka informację nadała. Taka komunikacja nazywa się komunikacją wirtualną dlatego, że warstwa przyjmująca informację u adresata dostaje ją dokładnie w takim samym formacie, w jakim została ona wysłana przez nadawcę. Oprogramowanie każdej warstwy wyższej niż pierwsza „ma wrażenie" komunikacji bezpośredniej z odpowiadającą jej taką samą warstwą na zdalnym komputerze, chociaż rzeczywista komunikacja występuje tylko w warstwie fizycznej.

Pełny stos warstw oprogramowania sieciowego jest potrzebny tylko u końcowych odbiorców informacji. Do przesyłania danych przez sieć wystarczą pierwsze trzy warstwy modelu OSI, czyli warstwa fizyczna, sieciowa i łącza danych. Na rys. 3 przedstawia to środkowa część rysunku, symbolizująca urządzenia sieci komputerowej.

Poniżej zostaną krótko scharakteryzowane zadania poszczególnych warstw:

Warstwa fizyczna – odpowiada za transmisję sygnałów w sieci. W warstwie tej określa się m.in. fizyczny kształt i rozmiar łączy, parametry przesyłanego sygnału, sposoby nawiązywania połączenia i jego rozłączania po zakończeniu transmisji. Warstwa ta operuje najmniejszą jednostką informacji, czyli bitem.

Warstwa łącza danych – odpowiedzialna jest za odbiór i konwersję strumienia bitów pochodzących z urządzeń transmisyjnych w taki sposób, aby nie zawierały one błędów. Warstwa ta postrzega dane jako grupy bitów, zwane ramkami. Warstwa łącza danych tworzy i rozpoznaje granice ramki. Ramka tworzona jest przez dołączenie do jej początku i końca grupy specjalnych bitów. Kolejnym zadaniem tej warstwy jest eliminacja zakłóceń powstałych w trakcie transmisji danych kanałem łączności. Ramki, które zostały przekazane niepoprawnie, są przekazywane ponownie. Ponadto warstwa łącza danych zapewnia synchronizację szybkości przesyłania danych oraz umożliwia ich przesyłanie w obu kierunkach.

Warstwa sieciowa – steruje działaniem podsieci transportowej. Jej podstawowe zadanie to przesyłanie danych w pakietach pomiędzy węzłami sieci wraz z wyznaczeniem trasy przesyłu.

W najprostszym przypadku określanie drogi transmisji pakietu odbywa się na podstawie stałych tablic opisanych w sieci. Istnieje również możliwość dynamicznego określania trasy na bazie bieżących obciążeń linii łączności. Stosując drugie rozwiązanie, mamy możliwość uniknięcia przeciążeń na trasach, na których pokrywają się drogi wielu pakietów, oraz ominięcia uszkodzonych fragmentów sieci.

Warstwa transportowa – podstawową funkcją tej warstwy jest obsługa danych, przyjmowanych z warstwy sesji. Obejmuje ona opcjonalne dzielenie strumienia danych na mniejsze jednostki, przekazywanie zblokowanych danych warstwie sieciowej, otwieranie połączenia stosownego typu i prędkości, realizację przesyłania danych, zamykanie połączenia. Celem postawionym przy projektowaniu warstwy transportowej jest zapewnienie jej pełnej niezależności od zmian konstrukcyjnych sprzętu.

Warstwa sesji – podstawowym jej zadaniem jest wymiana wiadomości pomiędzy urządzeniami komunikującymi się przez sieć. Pełni ona także szereg funkcji zarządzających, związanych np. z taryfikacją usług w sieci. Przy projektowaniu tej warstwy zwraca się również uwagę na zapewnienie bezpieczeństwa przesyłanych danych. Przykładowo, jeśli zostanie zerwane połączenie, w czasie którego aktualizowana była baza danych, to stan tej bazy może okazać się niespójny. Warstwa sesji musi przeciwdziałać takim sytuacjom.

Warstwa prezentacji – jej zadaniem jest obsługa formatów danych. Odpowiada ona za kodowanie i dekodowanie zestawów znaków oraz wybór algorytmów, które do tego będą użyte. Przykładową funkcją realizowaną przez tę warstwę jest kompresja przesyłanych danych, pozwalająca na zwiększenie szybkości transmisji informacji. Ponadto warstwa udostępnia mechanizmy kodowania danych w celu ich utajniania oraz konwersję kodów dla zapewnienia mobilności danych.

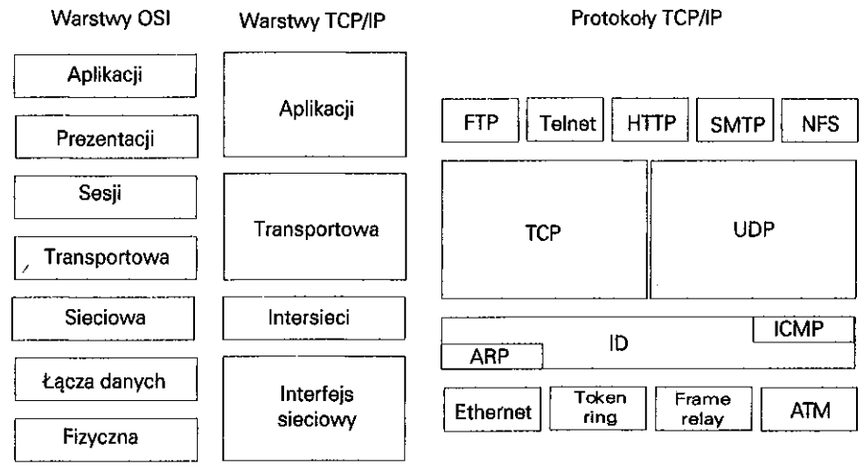
Warstwa aplikacji – zapewnia programom użytkowym usługi komunikacyjne. Określa ona formaty wymienianych danych i opisuje reakcje systemu na podstawowe operacje komunikacyjne. Warstwa stara się stworzyć wrażenie przezroczystości sieci, co jest szczególnie ważne przy korzystaniu np. z rozproszonych baz danych, kiedy użytkownik nie musi wiedzieć, gdzie zlokalizowane są wykorzystywane przez niego zasoby.

1.5.2. TCP/IP a model OSI

Rodzina protokołów TCP/IP (omawiana w rozdz. 5 i 6), na której oparte jest działanie Internetu, ma również strukturę warstwową i wykorzystuje paradygmat modelu OSI. Warstwy TCP/IP różnią się jednak nieco od warstw OSI (rys. 4). Sama rodzina protokołów TCP/IP obejmuje w zasadzie trzy warstwy: intersieci, transportową i aplikacji. Niektóre z nich łączą w sobie zadania dwóch warstw modelu OSI. Ponadto zwykle w jednej warstwie działa dwa lub

więcej protokołów, różniących się zakresem wykonywanych funkcji. Warstwa interfejsu sieciowego, obejmująca technologie sieci fizycznych (omawiane w rozdz. 3 i 4), jest niezbędna do tego, żeby protokoły rodziny TCP/IP mogły przekazywać dane.

Rys. 4. Warstwy protokołów TCP/IP w porównaniu z modelem OSI [8]



Warstwa interfejsu sieciowego (*Network Interface Layer*) – odbiera datagramy IP (rozdz. 5.3.2.) i po przesłaniu ich przez sieć przekazuje do oprogramowania IP odbiorcy. Obejmuje technologie sieci lokalnych (np. Ethernet, Token Ring – rozdz. 3) i rozległych (np. Frame Relay, ATM – rozdz. 4).

Warstwa intersieci (*Internet Layer*) – odpowiada za obsługę komunikacji między jedną maszyną a drugą. Przyjmuje ona pakiety z warstwy transportowej razem z informacjami identyfikującymi maszynę – odbiorcę, kapsułkuje pakiet w datagramie IP (rozdz. 5.3.2.), wypełnia jego nagłówek i przekazuje datagram do interfejsu sieciowego, czyli do warstwy łącza danych. Warstwa ta zajmuje się też datagramami przychodzącymi, sprawdzając ich poprawność i stwierdzając, czy należy je przesłać dalej, czy też przetwarzać na miejscu. Główne protokoły tej warstwy to IP (rozdz. 5.3.), ARP (rozdz. 5.4) i ICMP (rozdz. 5.5.).

Warstwa transportowa (*Host-to-Host Transport Layer*) – jej podstawowym zadaniem jest zapewnienie komunikacji między jednym programem użytkownika a drugim. Warstwa ta może kontrolować przepływ informacji. Może też zapewnić pewność przesyłania danych (przesyłanie danych z potwierdzeniem). Łączy zadania warstwy transportowej i warstwy sesji modelu OSI. Główne protokoły tej warstwy to UDP (rozdz. 5.6.) i TCP (rozdz. 5.7.).

Warstwa aplikacji (*Application Layer*) – na najwyższym poziomie użytkownicy uruchamiają programy użytkowe, które mają dostęp do usług TCP/IP. Programy użytkowe współpracują z jednym z protokołów z warstwy transportowej i wysyłają lub odbierają dane w postaci pojedynczych komunikatów lub strumieni bajtów. Najbardziej znane protokoły tej warstwy to HTTP (rozdz. 6.8.), FTP (rozdz. 6.5), Telnet (rozdz. 6.6.), SMTP (rozdz. 6.10.) i NFS (rozdz. 6.9.). Do warstwy aplikacji zaliczają się też protokoły ułatwiające użytkowanie i zarządzanie siecią TCP/IP, tzn. DNS (*Domain Name System* – rozdz. 6.3.), RIP (*Routing Information Protocol* – protokół do wymiany informacji pomiędzy routerami) oraz SNMP (*Simple Network Management Protocol* – używany do sterowania urządzeniami sieciowymi: routerami, mostami, inteligentnymi hubami).

2. Warstwa fizyczna i łącza danych

2.1. Ośrodki transmisji w sieciach komputerowych

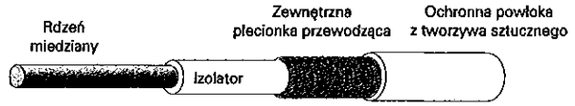
Każda komunikacja komputerowa wymaga na najniższym poziomie (czyli w warstwie fizycznej modelu OSI) kodowania danych do jakiejś postaci energii. Ta energia musi zostać przesłana za pomocą odpowiedniego **ośrodka transmisji** [6]. Prąd elektryczny może być użyty do przesłania danych kablem, a impulsy świetlne do przesłania ich światłowodem. Poniżej zostaną krótko omówione wybrane, najbardziej popularne ośrodki przesyłania danych.

2.1.1. Kable miedziane

Kable są podstawowym medium (nośnikiem) transmisyjnym, ponieważ są one stosunkowo tanie i łatwe do zainstalowania. Używa się głównie kabli miedzianych ze względu na małą oporność miedzi. Kabel, przez który jest przesyłany sygnał elektryczny, emituje fale elektromagnetyczne, czyli działa jak mała radiostacja. Dlatego typ okablowania musi być tak dobierany, aby zminimalizować ten efekt i jednocześnie zminimalizować zakłócenia, które wówczas powstają w kablach biegnących w pobliżu. Zakłócenia te nazywa się **interferencją sygnałów**.

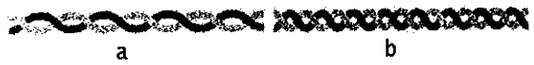
Aby zminimalizować interferencję, należy zminimalizować energię elektromagnetyczną emitowaną przez kabel. W tym celu sieci są budowane z użyciem jednego z dwóch podstawowych typów okablowania: **skrętki** (*twisted pair*) lub **kabla koncentrycznego**.

Rys. 5. Kabel koncentryczny [15]



Kabel koncentryczny (potocznie zwany koncentrykiem) daje lepsze zabezpieczenie przed interferencją niż skrętka. W kablu tym pojedynczy przewód jest otoczony osłoną z metalu, która stanowi ekran ograniczający interferencję. Kabel i osłona są rozdzielone izolacją. Osłonę stanowi elastyczna metalowa siatka, dlatego kabel, choć dosyć sztywny, może być zginany.

Rys. 6. Skrętka nieekranowana UTP [15]



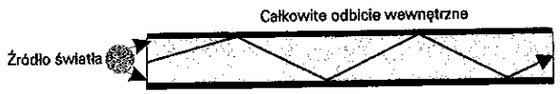
Skrętkę (rys. 6) tworzą dwa kable, zwykle o średnicy 1 mm, z których każdy jest otoczony materiałem izolacyjnym. Para takich przewodów jest skręcana, co powoduje zarówno zmniejszenie wypromieniowanej energii, jak i podatności na wpływ sygnału z innych kabli. Takie okablowanie jest również stosowane w telekomunikacji. W sieciach komputerowych spotykane są najczęściej dwa typy skrętki nieekranowanej **UTP** (*unshielded twisted pair*): skrętka kategorii 3 (rys. 6a) i skrętka kategorii 5 (rys. 6b) [15]. Pierwsza z nich składa się z przewodów lekko ze sobą skręconych i zapewnia pasmo przenoszenia o szerokości 16 MHz. Druga charakteryzuje się większą liczbą skrętów na centymetr, co daje lepszą jakość sygnału na dłuższe dystanse. Skrętka kategorii 5 ma pasmo przenoszenia 100 MHz, dzięki czemu kabel ten może być stosowany do szybkiej komunikacji komputerowej.

Pomysł użycia osłony zastosowano także do skrętki, tworząc skrętkę ekranowaną **STP** (*shielded twisted pair*). Składa się ona ze standardowej pary przewodów otoczonych dodatkowo metalową osłonką.

2.1.2. Światłowody

Do łączenia sieci komputerowych używa się również giętkich włókien szklanych (umieszczanych w plastikowych osłonach), przez które dane są przesyłane za pomocą światła. Na jednym końcu przewodu umieszcza się nadajnik z diodą świecącą (*light emitting diode* – LED) lub laser. Odbiornik umieszczony na drugim końcu wyposażony jest w światłoczuły tranzystor wykrywający impulsy świetlne.

Rys. 7. Zasada działania światłowodu [15]



W porównaniu z kablami światłowody mają kilka istotnych zalet:
- nie powodują interferencji elektrycznej,
- mają o wiele większy zasięg (impulsy świetlne, które są odbijane do wewnątrz włókna mogą docierać dalej niż sygnał elektryczny w kablu miedzianym),

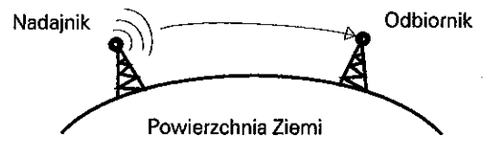
- mogą przenosić więcej informacji niż kable (za pomocą światła można zakodować więcej informacji),
- do przesyłania prądu niezbędna jest zawsze para przewodów, połączona w pełny obwód, a światło przemieszcza się przez pojedyncze włókno.

Ale są też i wady tego typu okablowania:
- do łączenia światłowodów potrzebne są specjalne urządzenia,
- kiedy włókno zostanie złamane wewnątrz plastikowej osłony, znalezienie tego miejsca jest bardzo trudne, a do ewentualnej naprawy również trzeba mieć specjalne urządzenia.

2.1.3. Radio

Fale elektromagnetyczne (radiowe) są coraz częściej używanym medium do transmisji danych komputerowych. Sieci używające tych fal nie wymagają bezpośredniego fizycznego połączenia między komputerami.

Rys. 8. Komunikacja za pomocą transmisji radiowej [15]



Podłączenie do takiej sieci wymaga zainstalowania anteny, która nadaje i odbiera fale. Rozmiar tej anteny zależy od wymaganego zasięgu, np. antena umożliwiająca komunikację wewnątrz budynku mieści się we wnętrzu komputera. Zasięg kilku kilometrów wymaga odpowiednio większej anteny.

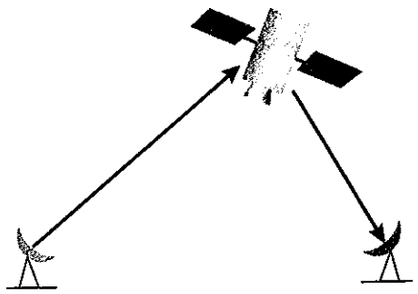
2.1.4. Komunikacja satelitarna

Zasięg komunikacji opartej na falach radiowych jest ograniczony krzywizną Ziemi. Ale technika radiowa pozwala na komunikację dalekosiężną za pomocą satelitów.

Satelita musi być wyposażony w **transponder** – urządzenie, które odbiera sygnały radiowe skierowane do satelity, wzmacnia je i wysyła ponownie w kierunku Ziemi pod odpowiednio zmienionym kątem. Często taka komunikacja odbywa się pomiędzy nadajnikiem i odbiornikiem, stojącymi na dwóch kontynentach. Zwykle jeden satelita wyposażony jest w kilka czy nawet kilkanaście niezależnych transponderów, co umożliwia prowadzenie jednocześnie wielu sesji komunikacyjnych.

Satelity dzielą się na geostacjonarne i umieszczone na niskich orbitach [6]. Wykorzystanie obu tych rodzajów nieco się różni, bo satelity na niskich orbitach okrążają Ziemię z dosyć dużą szybkością.

Rys. 9. Komunikacja satelitarna [15]



2.1.5. Podczerwień

Transmisja w podczerwieni jest ograniczona do małej przestrzeni i zwykle wymaga, aby nadajnik był nakierowany na odbiornik. Jednak sprzęt korzystający z podczerwieni jest niedrogi i nie wymaga anteny. Transmisja w podczerwieni może być użyta w sieciach komputerowych do przenoszenia danych w obrębie jednego pomieszczenia. Jest też często stosowana w urządzeniach peryferyjnych, takich jak myszki.

2.2. Warstwa łącza danych

Warstwa łącza danych pełni zadania związane z wysyłaniem i odbiorem danych przez łącza fizyczne (rozdz. 2.1). Jest ona odpowiedzialna za umieszczenie bloku przesyłanych danych w odpowiednio sformatowanej ramce. **Ramka** jest strukturą zawierającą taką ilość informacji, która wystarcza do przesłania danych za pomocą sieci (LAN lub WAN) z miejsca nadania do miejsca przeznaczenia. Dane powinny dotrzeć do odbiorcy bez uszkodzeń, dlatego ramka zawiera mechanizmy sprawdzania poprawności własnej zawartości. Warstwa łącza danych obsługuje też powiadamianie o błędach, topologię sieci i kontrolę przepływu. Węzeł nadający dane musi otrzymać potwierdzenie, że dane dotarły do miejsca przeznaczenia w stanie nieuszkodzonym. Ramki z błędami muszą zostać przesłane ponownie.

Ramka składa się z pól, czyli ciągów bitów służących do zapisania odpowiednich informacji. Protokoły warstwy łącza danych (np. opisane w rozdz. 3.4 i 3.5) określają minimalny i maksymalny rozmiar ramki oraz konkretny układ pól ramki. Najczęściej spotykane pola w ramkach to:

- ogranicznik początku ramki – sekwencja bitów jednoznacznie identyfikująca początek ramki,
- adres źródłowy – nadawcy,
- adres docelowy – odbiorcy,
- dane – czyli zasadnicza zawartość ramki,
- sekwencja kontrolna ramki – zazwyczaj suma kontrolna CRC (*Cyclic Redundancy Check*).

Adres źródłowy i adres docelowy jednoznacznie identyfikują w sieci komputer nadający i komputer odbierający ramki. Adres ten jest związany z zamontowaną w komputerze kartą interfejsu sieciowego (w skrócie kartą sieciową). Jest on nazywany adresem MAC (*Media Access Control*) lub też **adresem fizycznym**.

Karta interfejsu sieciowego przekształca otrzymane do wysłania bloki danych w ramki, a następnie ramki w sygnały, które są przesyłane siecią. Budowa karty sieciowej jest więc ściśle związana z technologią zastosowaną w danej sieci. Adres fizyczny jest zaprogramowany w układzie scalonym umieszczonym na karcie.

3. Sieci lokalne – LAN

W rozdziale tym zostaną omówione najczęściej spotykane w sieciach LAN urządzenia oraz typy i topologie sieci LAN, a następnie – wybrane standardy, implementujące pracę pierwszej i drugiej warstwy (tzn. fizycznej i łącza danych) modelu odniesienia OSI.

3.1. Urządzenia przyłączane do sieci LAN

Urządzenie podstawowe w sieci jest to takie urządzenie, które może uzyskać dostęp do innych urządzeń lub umożliwić innym urządzeniom dostęp do siebie. Trzema najbardziej rozpowszechnionymi podstawowymi urządzeniami przyłączanymi do sieci LAN są klienci, serwery i drukarki [14].

Serwer to dowolny komputer przyłączony do sieci LAN, zawierający zasoby udostępniane innym urządzeniom, przyłączonym do tej sieci. **Klient** natomiast to dowolny komputer, który za pomocą sieci uzyskuje dostęp do zasobów umieszczonych na serwerze. Drukarki to urządzenia wyjścia, tworzące wydruki zawartości plików. Wiele innych urządzeń przyłączanych do sieci, np. napędy CD-ROM, to **urządzenia dodatkowe**, przyłączone do urządzeń podstawowych i znajdujące się wobec nich w stosunku podporządkowania. Drukarki mogą pełnić rolę urządzenia podstawowego lub dodatkowego w zależności od konfiguracji sieci.

Pojęciem **serwer** określa się zwykle komputer wielodostępny (jednocześnie może z niego korzystać wielu użytkowników), wyspecjalizowany w wykonywaniu określonych funkcji. Funkcje te wskazuje przymiotnik, określający **typ serwera**. Do najczęściej spotykanych typów serwerów zalicza się:

Serwer plików – jest to scentralizowany mechanizm składowania plików, z których mogą korzystać grupy użytkowników. Składowanie plików w jednym miejscu, czyli na serwerze plików, pozwala na korzystanie z jednego ustalonego magazynu współdzielonych plików, ułatwia zabezpieczenie źródła zasilania na czas awarii zasilania podstawowego oraz zwiększa szybkość dostępu do plików.

Serwer wydruków – jest to mechanizm umożliwiający współdzielenie drukarek przez użytkowników sieci lokalnej. Zadaniem takiego serwera jest przyjmowanie żądania wydruków ze wszystkich urządzeń sieci, ustawianie ich w kolejki (każda drukarka ma własną kolejkę) i wysyłanie do odpowiedniej drukarki.

Serwer aplikacji – jest to mechanizm centralnego składowania i udostępniania oprogramowania użytkowego, czyli miejsce, gdzie znajdują się programy wykonywalne, dostępne dla użytkowników sieci. Aby uruchomić odpowiedni program, klient musi nawiązać w sieci połączenie z takim serwerem i tam jest uruchamiana aplikacja. Serwery aplikacji umożliwiają organizacji zmniejszenie kosztów zakupu oprogramowania.

Problematyka związana z przyłączaniem do sieci poszczególnych urządzeń (serwerów i stacji roboczych) oraz z przyłączeniem sieci LAN do sieci WAN i tworzeniem szkieletów sieci LAN jest szczegółowo omówiona w [14].

3.2. Typy sieci LAN

Typ sieci opisuje sposób, w jaki przyłączone do sieci zasoby (serwery, klienci czy inne urządzenia, pliki) są udostępniane. Rozróżnia się dwa sposoby udostępniania zasobów w sieci: równorzędny i serwerowy [14].

Sieć typu **każdy-z-każdym**, zwana inaczej **równorzędną**, obsługuje nieustrukturalizowany dostęp do zasobów sieci. W takiej sieci komputer może być zarówno serwerem, jak i klientem, czyli może pobierać dane, programy i inne zasoby od innego komputera. Inaczej mówiąc – każdy komputer jest równorzędny w stosunku do każdego innego, nie ma hierarchii komputerów.

Sieci oparte na serwerach, czyli typu **klient-serwer**, wprowadzają hierarchię, która ma na celu zwiększenie sterowalności różnych funkcji obsługiwanych przez sieć w miarę, jak zwiększa się jej skala. Zasoby często używane są w takich sieciach gromadzone na komputerach, zwanych serwerami.

W rzeczywistych zastosowaniach podział na typy sieci nie jest oczywiście tak ścisły. Najczęściej są zakładane sieci będące mieszanką sieci równorzędnych oraz serwerowych. Często spotykanym przykładem jest sieć o architekturze serwerowej (typu klient-serwer), grupująca centralnie zasoby, które powinny być ogólnodostępne. W ramach takiej sieci udostępnianie zasobów wewnątrz lokalnych grup roboczych może odbywać się na zasadzie dostępu równorzędnego.

3.3. Topologia sieci lokalnych

W pierwotnych sieciach komputerowych używano łączy typu **punkt-do-punktu**. Taki typ połączenia ma pewne zalety. Każde połączenie jest realizowane niezależnie, można więc użyć łącza o odpowiedniej wydajności. Komputery mają do tego połączenia wyłączny dostęp, mogą więc decydować, jak przesyłać dane tym połączeniem. Ponadto wyłączność dostępu daje także pewność bezpieczeństwa i prywatności. Główną wadą tego typu połączeń jest szybko rosnąca

liczba łączy (szybciej niż liczba komputerów). Liczba łączy *LP* potrzebnych do połączenia *N* komputerów jest proporcjonalna do N^2 [6]:

$$LP = (N^2 - N)/2$$

W miarę wzrostu liczby połączonych komputerów coraz droższe staje się dodanie kolejnego komputera do zestawu. W praktyce cena jest szczególnie wysoka, gdyż wiele połączeń jest dokonywanych tą samą fizyczną drogą, np. wzdłuż tego samego korytarza. Zaistniała więc potrzeba opracowania sposobu wykorzystywania jednego wspólnego ośrodka, do którego podłączonych jest wiele komputerów. Tak powstały sieci LAN. Z badań naukowych wynikło wiele sposobów ich rozwiązań, różniących się szczegółami technicznymi. Ich wspólną cechą jest wyeliminowanie dublujących się łączy, zastąpionych wspólnym ośrodkiem przesyłania danych. Techniki sieci lokalnych są niedrogie i dzięki temu stały się niezwykle popularne. Zgodnie z zasadą lokalności odwołań sieci LAN stały się najpopularniejszą formą sieci komputerowych.

Zasada lokalności odwołań

Komunikacja komputerowa rządzi się dwoma wzorcami:

- komputer częściej komunikuje się z jednostkami, które są fizycznie niedaleko, niż z jednostkami odległymi;
- komputer najczęściej wielokrotnie komunikuje się z tymi samymi maszynami.

Zasada lokalności odwołań staje się jasna, jeśli odniesiemy ją do komunikacji międzyludzkiej.

Ponieważ opracowano wiele technik LAN, często klasyfikuje się je na podstawie ich *topologii*, czyli ogólnego kształtu.

Topologia jest to sposób okablowania sieci na określonym obszarze, czyli połączenia komputerów w jeden zespół. Podczas projektowania sieci komputerowej należy uwzględnić liczne czynniki, wśród których zasadniczą rolę odgrywają trzy:

- koszty instalacji kablowej, kart sterujących i osprzętu sieciowego;
- elastyczność architektury sieci, dająca możliwość jej rekonfiguracji lub wprowadzenia dodatkowych węzłów (komputerów, urządzeń peryferyjnych);
- niezawodność realizacji zadań informatycznych w sieci komputerowej, uzyskiwana na drodze redundancji komunikacyjnej (nadmiarowości połączeń) między węzłami sieci i wprowadzania dodatkowych węzłów sieci, stosownie do wagi zadań informatycznych.

Poniżej zostaną omówione trzy podstawowe topologie sieci lokalnych: gwiazda, szyna i pierścień, które znalazły najszersze zastosowanie praktyczne. Każda z tych topologii jest ściśle związana z określoną technologią, np. topologia szyny i gwiazdy z technologią Ethernet

(rozdz. 3.4.). Wraz z postępem technologicznym dołączyła do nich czwarta topologia, zwana **topologią przełączaną**. Zostanie ona omówiona na podstawie sieci Ethernet (rozdz. 3.4.6.), chociaż może być stosowana także w sieciach Token Ring czy FDDI [14].

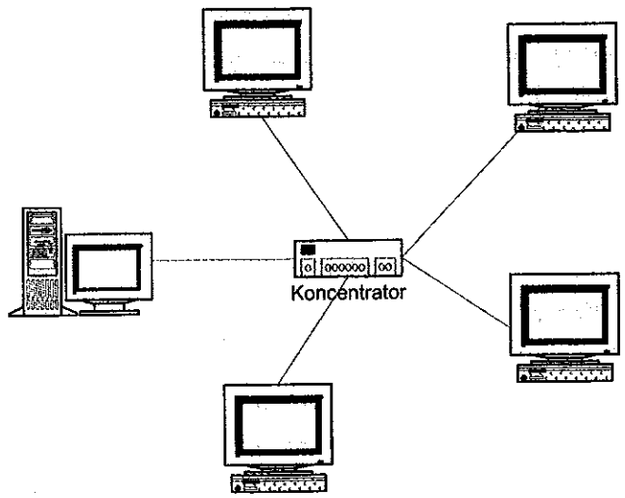
Topologie złożone (łańcuchy, hierarchie) są rozszerzeniami i/lub połączeniami podstawowych topologii fizycznych. Topologie podstawowe są odpowiednie jedynie dla bardzo małych sieci LAN, ponieważ ich skalowalność jest ograniczona. Topologie złożone tworzone są z elementów składowych umożliwiających uzyskanie topologii skalowalnych, odpowiednich dla konkretnego zastosowania. Topologie te zostały szczegółowo omówione w [14] w rozdz. 2.

Problematyka związana z projektowaniem sieci LAN została szeroko omówiona w [5].

3.3.1. Topologia gwiazdy

Jest to sieć zawierająca jeden centralny węzeł, zwany koncentratorem (*hub*), do którego przyłączone zostały elementy składowe sieci (rys. 10). Awaria jednego łącza nie powoduje unieruchomienia całej sieci. Stosowana jest do łączenia komputerów w jednej instytucji lub budynku. Większość zasobów sieci może znajdować się w komputerze centralnym, czyli serwerze. Wówczas pozostałe komputery mogą mieć niewielką moc obliczeniową. Okablowanie może stanowić skrętka.

Rys. 10. Topologia gwiazdy



Zalety topologii gwiazdy:

- łatwa konserwacja i lokalizacja uszkodzeń,
- prosta rekonfiguracja,
- proste i szybkie oprogramowanie użytkowe sieci,
- centralne sterowanie i centralna programowa diagnostyka sieci,
- możliwe duże szybkości transmisji.

Wady topologii gwiazdy:

- duża liczba kabli,
- ograniczona możliwość rozbudowy sieci,
- ograniczenie odległości komputera od huba,
- w przypadku awarii huba przestaje działać cała sieć.

W praktyce sieci gwiaździste rzadko mają kształt symetryczny, z koncentratorem usytuowanym w jednakowej odległości od wszystkich komputerów. Częściej może on znajdować się w innym pomieszczeniu.

Przykładem topologii gwiazdy jest często stosowana sieć Ethernet z przełącznikiem (rozdz. 3.4.6.).

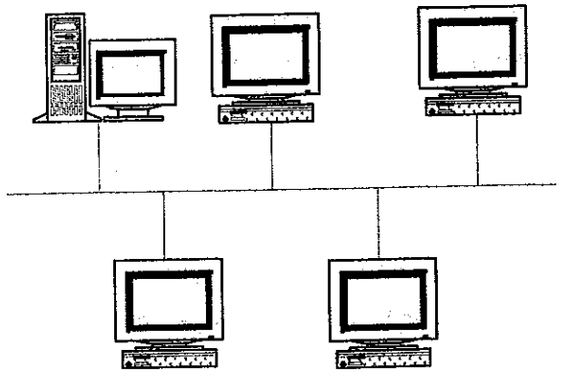
3.3.2. Topologia magistrali

Jest to swego rodzaju „autostrada", służąca do transmisji danych i łącząca stacje sieci (rys. 11). Dane, nim dotrą do stacji przeznaczenia, przechodzą po drodze przez wszystkie pozostałe stacje. W rozwiązaniu tym do wspólnego kabla transmisyjnego zostają podłączone komputery o dzielonym dostępie do medium transmisyjnego, a komputery konkurują ze sobą o dostęp do kabla. Sygnały nadawane z jednego komputera docierają do wszystkich stacji, ale pakiety odbierane są tylko przez tę stację, do której są adresowane. Każda stacja sprawdza, czy dane są adresowane do niej. Końce magistrali muszą być zakończone terminatorami, które zapobiegają odbiciu sygnału. **Terminator** jest to odpowiednio dobrany opornik. Bez niego sygnał elektryczny odbija się od końca przewodu jak światło od lustra. Powoduje to powstawanie kolizji, uniemożliwiając normalną pracę sieci. Topologia magistrali jest jedną z najbardziej popularnych konfiguracji sieci komputerowych.

Zalety magistrali:

- małe zużycie kabla,
- prosta instalacja,
- niska cena instalacji,
- bardzo prosta budowa sieci,
- łatwe łączenie segmentów sieci w jeden system,
- każdy komputer jest podłączony tylko do jednego kabla,
- pojedyncze uszkodzenie (kabla przyłączającego lub komputera) nie powoduje awarii całej sieci.

Rys. 11. Topologia magistrali



Wady magistrali:

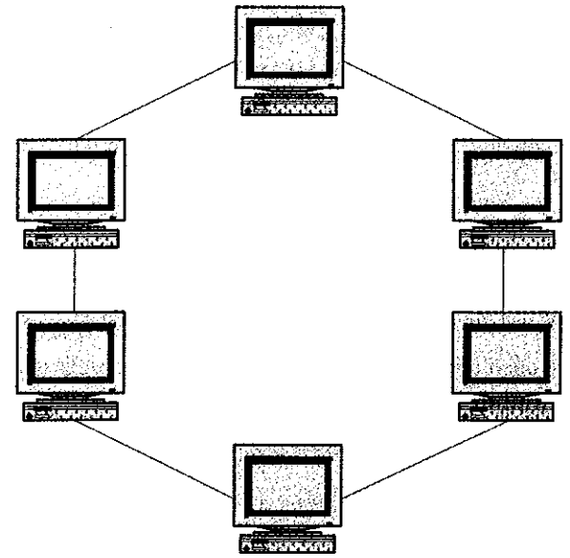
- konkurencja o dostęp (wszystkie komputery muszą dzielić się kablem transmisyjnym),
- utrudniona diagnostyka błędów z powodu braku centralnego zarządzania siecią,
- rozproszenie zadań zarządzających siecią, co może niekorzystnie wpływać na szybkość realizacji zadań informatycznych,
- zwykle dla uniknięcia zakłóceń sygnałów należy zachować pewną odległość między punktami przyłączenia poszczególnych stacji.

Typowym przykładem topologii magistrali jest tradycyjny **Ethernet** oparty na kablu koncentrycznym (rozdz. 3.4.5.).

3.3.3. Topologia pierścienia

W tej topologii każdy komputer dołączony do sieci bierze udział w procesie transmisji informacji i jest połączony z dwoma innymi sąsiadami (rys. 12). Węzły połączone w pierścień przekazują **komunikat sterujący** (*token*) do następnego. Węzeł aktualnie posiadający token może wysyłać komunikat. Dzięki temu wszystkie komputery mogą nadawać informacje po kolei. Informacja wędruje zawsze w jednym kierunku i po przejściu wszystkich węzłów wraca do miejsca nadania. Interfejs sieciowy każdego komputera musi odbierać dane od jednego sąsiada i przesyłać je do następnego. Odmianą tego układu jest podwójny pierścień, przy czym są one skonfigurowane tak, że sygnał jest w nich przesyłany w przeciwnych kierunkach. Drugi pierścień stanowi rezerwę na wypadek awarii pierwszego. Topologia pierścieniowa ma wiele zalet. Funkcjonowanie nie zostaje przerwane nawet w razie awarii głównego komputera, gdyż jego zadania może przejąć inna stacja. Dzięki zastosowaniu układów obejściowych (*by-pass*) można wyłączyć z sieci dowolną stację i tym sposobem uniknąć awarii sieci.

Rys. 12. Topologia pierścienia



Zalety pierścienia:

- małe zużycie kabla,
- łatwa koordynacja dostępu i wykrywanie niepoprawnego działania sieci,
- możliwość zastosowania złącz optoelektronicznych (światłowodowych), które wymagają bezpośredniego nadawania i odbierania sygnałów,
- możliwe wysokie osiągi, ponieważ kabel łączy dwa konkretne komputery.

Wady pierścienia:

- w przypadku pojedynczego pierścienia awaria kabla lub komputera powoduje przerwanie pracy całej sieci (ale istnieje możliwość zastosowania układów obejściowych),
- trudna rekonfiguracja sieci,
- wymagane specjalne procedury transmisyjne,
- dołączenie nowych stacji jest utrudnione, jeśli w pierścieniu jest wiele stacji.

Przykładem sieci o topologii pojedynczego pierścienia jest technologia **Token Ring**, opracowana w firmie IBM. W topologii podwójnego pierścienia działa sieć światłowodowa **FDDI** (*Fiber Distributed Data Interconnect*), pozwalająca przesyłać dane z bardzo dużymi prędkościami. Sieci te są szczegółowo opisane w [14].

3.4. Sieć Ethernet

Ethernet jest siecią lokalną najczęściej spotykaną w firmach, biurach i laboratoriach. Nazwa ta oznacza pewien zestaw standardów, określających na najniższym poziomie sposób przesyłania danych w sieci lokalnej, jak również praktyczne implementacje tych standardów.

Pod pojęciem najniższego poziomu rozumiemy samo okablowanie sieci oraz sposób przesyłania nią sygnału elektrycznego, odpowiednio kodującego dane. Standard Ethernet pracuje w pierwszej i drugiej warstwie modelu OSI.

3.4.1. Rozwój sieci Ethernet

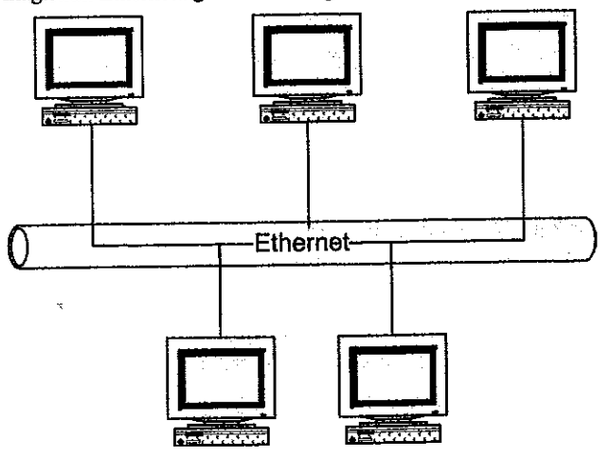
Ethernet ma już ponad ćwierć wieku. Powstał w roku 1976 w Xerox Palo Alto Research Center (Bob Metcalfe, David Boggs). Później rozwijały go firmy DEC, Intel i Xerox, opracowując standard DIX, o przepustowości 10 Mb/s. Następnie w 1983 r. stał się on standardem IEEE 802.3 [15].

Nazwa Ethernet nawiązuje do eteru: otóż w XIX wieku fizycy próbujący wyjaśnić zjawiska fal elektromagnetycznych, tłumaczyli je jako drgania hipotetycznej substancji, którą nazwali eterem. W przypadku Ethernetu takim eterem jest pojedynczy segment sieci.

Pierwszy Ethernet pozwalał na przesyłanie danych z przepustowością 10 Mb/s. Obecnie najszerzej stosowana jest wersja o przepustowości 100 Mb/s (*Fast Ethernet*), dla której norma powstała w 1995 r. Wersja o przepustowości 1 Gb/s (*Gigabit Ethernet*) jest opisana normami 802.3z i 802.3ab. Najnowsza wersja Ethernetu (prace nad normą 802.3ae ukończono w 2002 r. [9]) umożliwia przepływ danych z przepustowością 10 Gb/s. W zależności od wersji Ethernet pozwala na maksymalną odległość między stacjami od 100 m do 2,8 km.

Początkowo sieć lokalna Ethernet składała się z kabla koncentrycznego, do którego były podłączone komputery. Kabel taki określa się mianem segmentu. Jego długość była ograniczona do 500 m, a odległość między każdą parą połączeń musiała wynosić minimum 3 m. Stosowano sztywny kabel koncentryczny półcalowy (gruby – 10Base5). Sieć miała topologię typowej szyny (rys. 13).

Rys. 13. Topologia magistrali Ethernetu grubokablowego



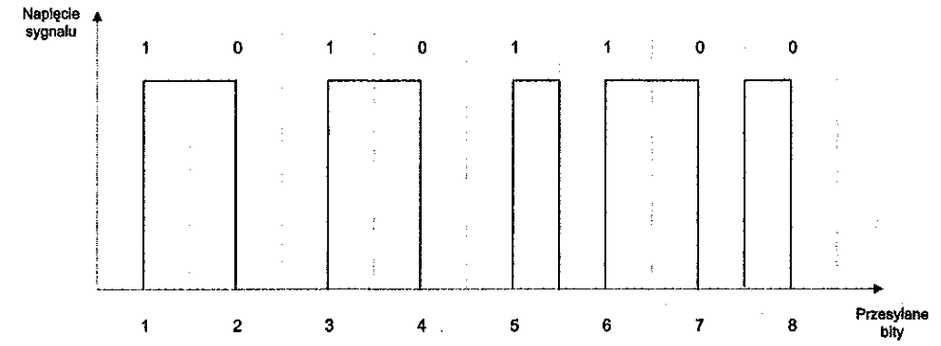
Następnie wprowadzono kabel koncentryczny cienki – ćwierćcalowy (10Base2), który był bardziej elastyczny od swojego poprzednika, a potem skrętkę (10BaseT), która jest obecnie najbardziej rozpowszechnionym medium dla sieci Ethernet. Najnowszy standard (10 Gb/s) powstał początkowo tylko dla światłowodów, dopiero dalsze prace nad jego rozwojem umożliwiły zastosowanie skrętki miedzianej.

Kabel Ethernetu w topologii szyny musi mieć na obu końcach zainstalowane urządzenia, wytłumiające sygnał elektryczny, czyli terminatory.

3.4.2. Kodowanie Manchester

Transmisja w Ethernecie tradycyjnym odbywa się z wykorzystaniem **kodowania Manchester**. Oznacza to, że każdy bit jest kodowany zboczem narastającym, oznaczającym 1, lub zboczem opadającym, oznaczającym 0 (rys. 14).

Rys. 14. Kodowanie Manchester [6]



Na rysunku 14 oś pozioma oznacza czas potrzebny na przesłanie 8 bitów, natomiast oś pionowa – napięcie. Kolejne odcinki czasu zaznaczono liniami przerywanymi. Zmiana napięcia następuje dokładnie w połowie odcinka czasu przeznaczonego na przesłanie jednego bitu, tak jak dla bitów od 1 do 4. Jeśli dwa kolejne bity mają tę samą wartość, to na brzegu odcinka pojawia się dodatkowa zmiana napięcia, tak jak dla bitów 5 i 6 oraz 7 i 8. Aby odbiorca dokładnie wiedział, kiedy w sygnale przychodzącym dany odcinek zaczyna się i kończy, używa się odpowiedniej preambuły (*Preamble*), która pozwala karcie odbierającej sygnał na synchronizację. Taka preambuła składa się z odpowiedniej liczby na przemian następujących po sobie zer i jedynek.

3.4.3. Ramka

W Ethernecie dane przesyłane są w **ramkach**. Są dwa bardzo podobne i niesprzeczne warianty struktury ramki Ethernetu (rys. 15).

Rys. 15. Format ramki Ethernetu [15]

IEEE 802.3

7	7	6	6	2	46–1500		4
PA	SFD	DA	SA	LEN	Dane	Wypełnienie	CRC

Ethernet właściwy

8	6	6	2	46–1500		4
PA	DA	SA	Typ	Dane	Wypełnienie	CRC

Liczby na rysunku określają wielkość poszczególnych elementów ramek wyrażoną w bajtach, a skróty literowe oznaczają kolejne pola ramki, opisane poniżej:

PA – 7 lub 8 bajtów synchronizacji – są to bajty o wartości dwójkowej 10101010 (*Preamble*),

SFD – bajt o wartości 10101011 na oznaczenie początku zawartości merytorycznej ramki (*Start of Frame Delimiter*),

DA – docelowy adres fizyczny (*Destination Address*),

SA – źródłowy adres fizyczny (*Source Address*),

LEN – liczba bajtów danych (*Length*),

Typ – kod protokołu wyższego poziomu, dla którego przeznaczone są dane,

Wypełnienie – bajty dodawane, gdy ramka przesyła mniej niż 46 bajtów danych,

CRC – suma kontrolna (*Cyclic Redundancy Check*).

Adres źródłowy i docelowy są czterdziestoośmiobitowymi adresami ethernetowymi. Każda karta sieciowa Ethernetu otrzymuje taki unikalny adres fabrycznie.

Suma kontrolna pozwala na wykrycie większości błędów transmisji. Odbierająca karta sieciowa wylicza ją sama na podstawie przesyłanych danych i porównuje z odebraną sumą kontrolną. Niezgodność oznacza błąd.

Suma długości danych i długości wypełnienia musi być zawarta w przedziale od 46 do 1500 bajtów. Wypełnienie potrzebne jest wtedy, kiedy ramka przesyła mniej niż 46 bajtów danych. Za krótka ramka może prowadzić do nieprawidłowego działania protokołu rozwiązywania konfliktów w dostępie do możliwości nadawania.

3.4.4. Protokół CSMA/CD

Sieć Ethernet charakteryzuje się podłączeniem komputerów do wspólnego kabla oraz brakiem centralnego ośrodka kontroli, który by wyznaczał, w jakiej kolejności komputery używają sieci. Oznacza to, że komputery konkurują ze sobą o dostęp do medium, ale odbywa się

to w pewien określony sposób. Wszystkie komputery biorą udział w metodzie rozproszonej koordynacji, znanej jako **protokół CSMA/CD**. Skrót CSMA/CD oznacza *Carrier Sense Multiple Access with Collision Detection* – wykrywanie fali nośnej przy wielodostępie z wykrywaniem kolizji. Carrier Sense to wykrycie fali nośnej – zanim stacja zacznie nadawać, musi posłuchać, aby przekonać się, że w jej segmencie nikt inny nie nadaje (eter jest wolny, tzn. nie ma w kablu sygnału elektrycznego). Multiple Access oznacza, że wszystkie stacje mają równe prawa, jednocześnie mogą słuchać, a może się zdarzyć, że jednocześnie zaczną nadawać. Każda stacja sprawdza eter w trakcie nadawania i przestaje nadawać natychmiast, kiedy tylko wykryje, że jednocześnie nadaje inna stacja. Sytuacja taka nazywana jest kolizją, stąd Collision Detection. W przypadku kolizji każda ze stacji biorących w niej udział odczekuje przez losowo wybrany (ze zmiennego zakresu – według standardu) czas, a następnie próbuje wznowić nadawanie. Jeśli komputery obiorą po kolizji prawie tę samą wartość opóźnienia, kolizja powtórzy się. Wówczas każdy komputer podwaja zakres, z którego wybierana jest długość opóźnienia. Proces powtarza się do skutku, czyli do momentu, kiedy kolizja już nie wystąpi.

3.4.5. Topologia sieci Ethernet

Jak już wspomniano, Ethernet grubokablowy (gruby kabel koncentryczny) ma topologię szyny (rys. 13). Dziś jest to technologia bardzo przestarzała. Zastąpiona została najpierw Ethernetem cienkokablowym, który także ma topologię szyny, chociaż kabel jest elastyczny i rozciąga się go od komputera do komputera, podłączając odpowiednią wtyczką (złącze BNC).

Obecnie sieci Ethernet tworzone są z użyciem kabla, zwanego skrętką (10BaseT). W najprostszym przypadku za centrum sieci służy tutaj urządzenie elektroniczne, zwane koncentratorem, do którego podłącza się każdy komputer za pomocą osobnego kabla. Układy elektroniczne koncentratora symulują fizyczny kabel, całość więc działa jak zwykły Ethernet. Chociaż fizycznie sieć ma topologię gwiazdy, logicznie nadal ma topologię szyny i działa jak szyna – rolę szyny pełni koncentrator.

Na tym przykładzie widać, że o topologii sieci można mówić w dwóch aspektach – **topologii fizycznej**, czyli jak połączone są kable, oraz **topologii logicznej**, czyli jak działa dana sieć.

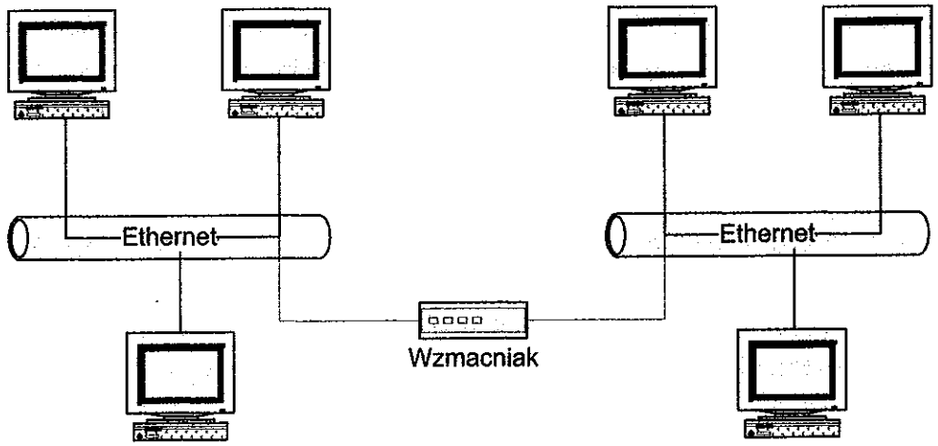
3.4.6. Urządzenia pomocnicze w sieci Ethernet

Jednym z powodów ograniczenia zasięgu sieci lokalnych jest tłumienie sygnałów elektrycznych w kablu miedzianym.

W sieci Ethernet oprócz stacji i okablowania mogą występować różne urządzenia, pozwalające zwiększyć możliwości sieci co do jej maksymalnej długości czy też wydajności:

Wzmacniak (*repeater*) – najprostsze urządzenie dodatkowe, które działa jak zwykły wzmacniacz sygnału. Łączy on kable sieci Ethernet, zwane segmentami i przekazuje kopie sygnałów elektrycznych z jednego do drugiego (rys. 16).

Rys. 16. Sieć Ethernet ze wzmacniakiem

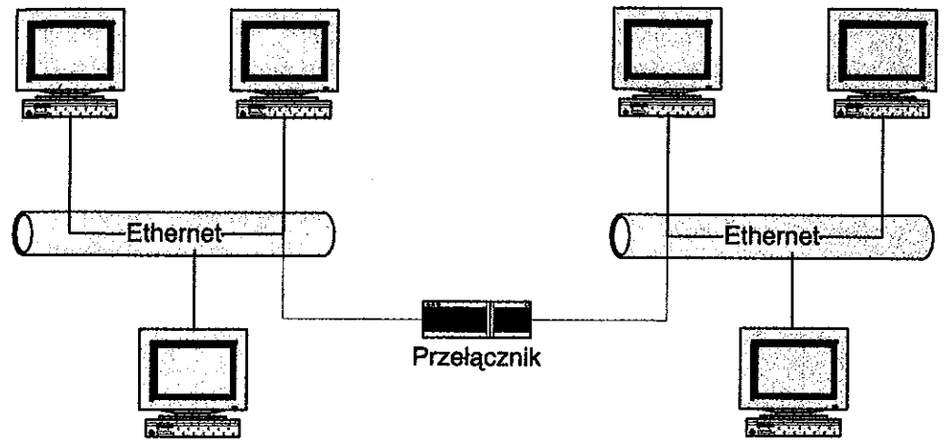


Wzmacniaki nie interpretują formatu ramek ani adresów fizycznych komputerów w sieci. Przekazują one wszystkie sygnały elektryczne między segmentami, komputery podłączone do różnych segmentów zatem mogą swobodnie się komunikować i nie muszą nic wiedzieć o istnieniu wzmacniaka. Zastosowanie wzmacniaków pozwala na zwiększenie maksymalnej odległości w sieci między dwiema stacjami: na przykład w przypadku grubego kabla koncentrycznego z 500 m do 2,8 km. Inną zaletą wzmacniaka jest podział sieci na oddzielne segmenty. Dzięki temu, jeśli w jednym z segmentów sieci wystąpi awaria (np. przecięcie kabla lub zwarcie), to nie będzie to miało wpływu na pracę pozostałych segmentów. Taki sposób powiększania zasięgu sieci jest ograniczony m.in. przez zastosowaną metodę wykrywania kolizji CSMA/CD, która działa poprawnie tylko w przypadku dostatecznie małych opóźnień sygnału. Dlatego standard Ethernet przewiduje, że sieć może nie działać poprawnie, jeśli dowolną parę komputerów dzielą więcej niż cztery wzmacniaki.

Przełącznik (*switch*) – pozwala na łączenie kilku, a w najprostszym przypadku dwóch segmentów sieci Ethernet, w jedną sieć lokalną (rys. 17).

Przełącznik, podobnie jak wzmacniak, przekazuje sygnały z jednego kabla do drugiego. Jego możliwości jednak są znacznie większe. Przełącznik przesyła z jednego segmentu do drugiego tylko te ramki, które są adresowane do stacji w tym segmencie. Proces ten nazywa się filtrowaniem ramek. Przełącznik nie wysyła poza segment ramek adresowanych tylko do stacji z tego segmentu. Nie przekazuje też zakłóceń elektrycznych. Takie działanie sprawia, że z jednej strony dwa segmenty sieci zachowują się jak jeden segment, ale jednocześnie każdy segment jest oddzielną domeną kolizji. W ten sposób zmniejsza się prawdopodobieństwo wystąpienia kolizji, przyspiesza jej rozwiązanie i w konsekwencji zwiększa efektywną przepustowość sieci. Dodatkowe przyspieszenie pochodzi stąd, że przełącznik może jednocześnie wysyłać różne dane do różnych segmentów sieci. Przełączniki mogą też łączyć w jedną całość sieci wykonane przy użyciu różnego rodzaju okablowania i/lub o różnych nominalnych przepustowościach.

Rys. 17. Sieć Ethernet z przełącznikiem



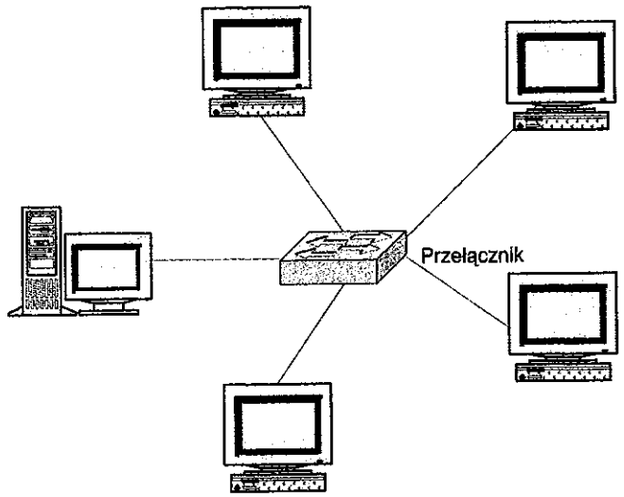
3.4.7. Topologia gwiazdy z przełącznikiem

Przełącznik fizycznie przypomina koncentrator, różni się jednak zasadniczo zasadą działania. W topologii gwiazdy koncentrator symuluje wspólne medium, natomiast przełącznik symuluje sieć lokalną z mostami, w której do każdego segmentu jest dołączony jeden komputer. Segment zajęty jest tylko wtedy, gdy dołączony do niego komputer wysyła lub odbiera informacje. Dzięki temu znacznie zwiększa się przepustowość sieci. Otrzymana w ten sposób topologia to topologia gwiazdy z przełącznikiem i jednostacyjnymi segmentami (rys. 18). Na rysunku pokazano często spotykany na schematach sieci symbol logiczny przełącznika, wprowadzony przez firmę Cisco [1].

W przypadku sieci lokalnych małych i średnich rozmiarów jest to rozwiązanie optymalne pod względem efektywnej przepustowości, niezawodności i łatwości diagnostyki. Jednak w celu ograniczenia kosztów wiele firm stosuje rozwiązanie kompromisowe: zamiast przyłączać jeden komputer do każdego portu przełącznika, podłącza się do przełącznika koncen-

tratory, a dopiero do nich komputery. Każdy koncentrator odpowiada tu jednemu segmentowi sieci. Komputery przyłączone do jednego koncentratora muszą dzielić między siebie czas nadawania. Komunikacja między komputerami przyłączonymi do różnych koncentratorów może odbywać się niezależnie.

Rys. 18. Sieć Ethernet o topologii gwiazdy



3.5. Sieci bezprzewodowe Wi-Fi

Technologia sieci bezprzewodowych Wi-Fi (*Wireless Fidelity*) wykorzystywana jest przez coraz szerszą grupę użytkowników. Są to nie tylko duże firmy. Malejące ceny urządzeń dostępowych spowodowały rosnącą popularność takich sieci wśród osób prywatnych i w małych biurach. Zdecydowaną zaletą sieci **WLAN** (*Wireless LAN*) jest prostota budowy i to, że nie trzeba dziurawić ścian i kłaść kabli tylko po to, żeby połączyć ze sobą komputery w różnych pomieszczeniach.

Technologia bezprzewodowa jest też coraz częściej wykorzystywana przez dostawców usług internetowych do rozwiązania tzw. problemu ostatniej mili podczas tworzenia stałego łącza do użytkownika.

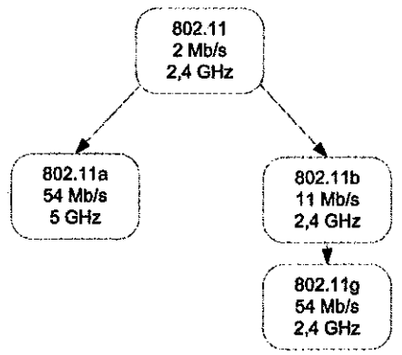
Podstawowe wady sieci radiowych to wolniejsza transmisja danych i ograniczenie zasięgu z powodu tłumienia sygnału. Trudniej też jest zapewnić bezpieczeństwo sieci niż w technologiach tradycyjnych. Sieć jest bardziej podatna na podsłuchiwanie danych czy na nieautoryzowany do niej dostęp. Są jednak mechanizmy pozwalające się przed tym zabezpieczyć.

3.5.1. Standardy IEEE 802.11

Symbol **802.11** odnosi się do rodziny specyfikacji opracowanych przez **IEEE** (*Institute of Electrical and Electronic Engineers*) dla potrzeb bezprzewodowych sieci lokalnych. W czerwcu 1997 r. zakończono prace nad inicjalnym standardem 802.11 (rys. 19). Standard ten określał działanie sieci w paśmie 2,4 GHz z prędkością transmisji 1 oraz 2 Mb/s. Ten inicjalny standard pozwalał na dwa sposoby modulacji: FHSS (*Frequency Hopping Spread Spectrum*) i DSSS (*Direct Sequence Spread Spectrum*). Od tego czasu Grupa Robocza IEEE 802.11 opracowała przez swoje grupy zadaniowe kilka jego rewizji. Zostaną tu omówione bardziej szczegółowo dwie rewizje najistotniejsze z punktu widzenia lokalnych sieci komputerowych: **IEEE 802.11b** i **IEEE 802.11g**.

Istnieje jeszcze standard **IEEE 802.11a.** Sieci w tym standardzie można spotkać głównie w USA. W Europie używana w nim częstotliwość jest zarezerwowana dla celów wojskowych i takiej sieci nie można używać bez ograniczeń. Na przykład w Polsce może ona być stosowana jedynie w obrębie budynków [4].

Rys. 19. Ewolucja omawianych standardów IEEE 802.11

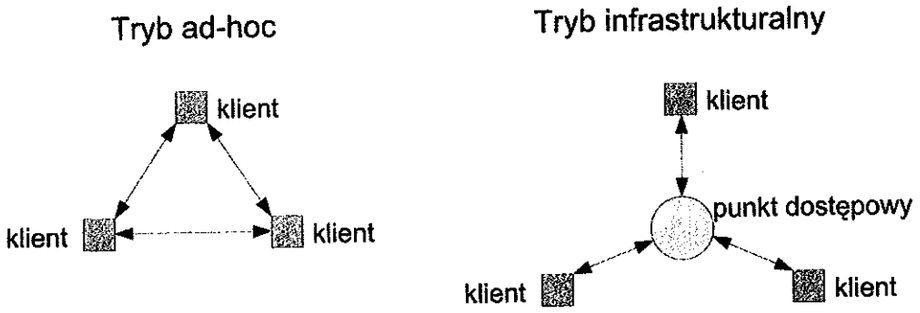


Wszystkie warianty sieci 802.11 określa się nazwą Wi-Fi. Nazwa ta jest propagowana przez Wi-Fi Alliance. Początkowo mianem Wi-Fi określało się tylko standard 802.11b, ale znaczenie tego terminu zostało rozszerzone.

3.5.2. Tryby pracy sieci

Sieci 802.11 mogą działać w dwóch trybach: w trybie **ad hoc** i w trybie **infrastrukturalnym** (rys. 20).

Rys. 20. Tryby pracy sieci IEEE 802.11



W trybie *ad hoc* indywidualne węzły tworzą sieć *peer-to-peer* (jak równy z równym). W tym trybie nie trzeba inwestować w drogie urządzenia dostępowe. Nadaje się on doskonale do połączenia komputerów znajdujących się w jednym pomieszczeniu. Nie może jednak w takiej sieci pracować zbyt wiele stacji roboczych (do pięciu) i odległości pomiędzy nimi nie mogą być zbyt duże.

W trybie infrastrukturalnym każdy z węzłów komunikuje się z innymi nie bezpośrednio, ale za pośrednictwem punktu dostępowego (*access point*). Oznacza to, że przez niego przechodzi cała transmisja danych, pełni on rolę bezprzewodowego koncentratora. Punkt dostępowy zarządza ruchem w sieci, co pozwala unikać kolizji pakietów.

3.5.3. Standard 802.11b

Standard 802.11b to rewizja inicjalnego standardu 802.11, która dotyczy sieci lokalnych i daje możliwość transmisji z prędkością 11 Mb/s (z możliwością zmniejszania w gorszych warunkach do 5,5, 2 i 1 Mb/s) w paśmie 2,4 GHz. Wprowadzono tu ciekawy mechanizm *dynamic rate shifting* pozwalający na dynamiczną zmianę prędkości transmisji w zależności od właściwości kanału w danym momencie. Wraz z pojawieniem się zakłóceń mechanizm ten zmniejsza prędkość transmisji, ale po ponownym poprawieniu się warunków prędkość transmisji zostaje przywrócona.

Standard ten został zatwierdzony we wrześniu 1999 roku. Wykorzystuje on tylko modulację DSSS. Zwiększenie prędkości transmisji w stosunku do inicjalnego 802.11 zostało osiągnięte dzięki zastosowaniu obok modulacji DSSS także CCK (*Complementary Code Keying*)

– techniki modulacji, która zwiększa wydajność wykorzystania pasma radiowego. Efektywna prędkość transmisji przy szyfrowaniu jest mniejsza od nominalnej.

Sieć 802.11b działa na odległość do 100 metrów, w budynku – do około 30 metrów, w obecności stropów żelbetowych – do około 10 metrów. Są to dane orientacyjne, bo wiele zależy od konstrukcji budynku. Ogólnie rzecz biorąc, budynki amerykańskie są zbudowane z lżejszych materiałów niż europejskie, a zwłaszcza – niż polskie, w naszych budynkach więc zasięg jest zazwyczaj mniejszy niż podają standardy.

Sieć 802.11b jest bezprzewodową alternatywą dla sieci Ethernet. Przez wyższe warstwy sieciowe ten standard jest „widziany" tak jak Ethernet, ale wolniejszy od niego. Pomimo tego, jeśli trzeba zainstalować niewielką sieć lokalną na przykład w mieszkaniu czy domu, to wybór technologii 802.11b pozwoli nam uniknąć konieczności układania kabli. Obecnie większość bezprzewodowych sieci lokalnych korzysta właśnie ze standardu 802.11b.

Standard 802.11b jest również używany w tak zwanych **publicznych lokalnych sieciach bezprzewodowych** (*public wireless LAN*). Są to lokalne sieci bezprzewodowe instalowane w miejscach publicznych, takich jak centra handlowe, lotniska czy centra konferencyjne.

3.5.4. Standard 802.11g

Standard 802.11g to rozszerzenie standardu 802.11b mające na celu zapewnienie prędkości transmisji do 54 Mb/s w ramach pasma 2,4 GHz. Standard 802.11g jest nadzbiorem standardu 802.11b, a więc klient 802.11g może połączyć się z serwerem 802.11b i odwrotnie (z prędkością do 11 Mb/s). Na początku 2002 roku podjęto decyzję, że 802.11g dla prędkości transmisji powyżej 20 Mb/s zamiast z modulacji DSSS i CCK, ma korzystać z modulacji OFDM (*Orthogonal Frequency Division Multiplexing*).

4. Sieci rozległe – WAN

Zgodnie z definicją podaną w rozdz. 1.4. sieci **WAN** (*Wide Area Network*) są to sieci rozległe, mogące łączyć miasta, kraje, a nawet kontynenty. Sieci WAN od sieci LAN różnią się przede wszystkim zasięgiem i rodzajem własności. Sieci LAN rozciągają się zwykle na niewielkiej powierzchni (np. wewnątrz budynku czy na terenie campusu uniwersyteckiego) i mają jednego właściciela. Sieci WAN natomiast mają dowolny zasięg i zazwyczaj korzystają z okablowania zarządzanego przez więcej niż jednego operatora. Jednak podstawową cechą, odróżniającą sieci WAN od sieci LAN, jest **skalowalność** [6]. Technologia WAN musi umożliwiać rozrastanie się sieci w miarę potrzeby. Sieć taka łączy zwykle wiele ośrodków znajdujących się w dużych odległościach od siebie, a w ośrodkach tych może znajdować się wiele komputerów (np. korporacja posiadająca oddziały w wielu krajach). Wobec tego sieć WAN musi mieć wystarczającą przepustowość, żeby umożliwić tym komputerom jednoczesną komunikację na odpowiednich warunkach. Musi też mieć możliwość rozrastania się w miarę wzrostu liczby jej użytkowników oraz ich wymagań.

4.1. Budowa sieci rozległych

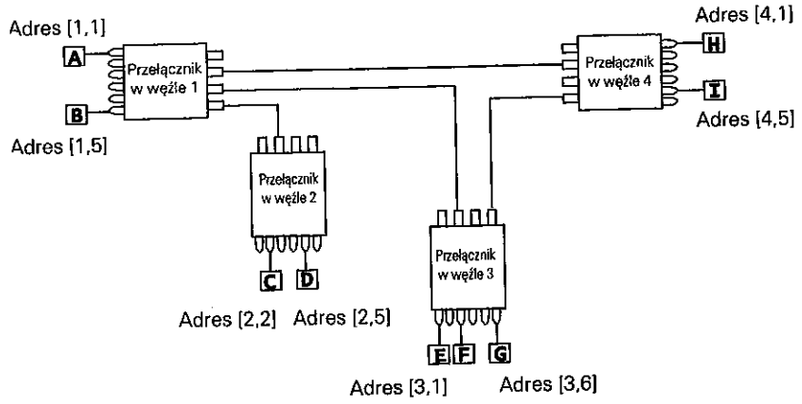
Podstawowym elementem elektronicznym służącym do budowy takich sieci jest przełącznik, przenoszący pakiety z jednego węzła sieci do drugiego. Jest on nazywany *przełącznikiem pakietów* (rys. 21). Ma dwa rodzaje urządzeń wejścia-wyjścia. Jeden z nich, działający z dużą szybkością, służy do łączenia przełącznika z innymi przełącznikami. Drugi rodzaj urządzenia wejścia-wyjścia jest używany do łączenia przełącznika z poszczególnymi komputerami.

Rys. 21. Przełącznik pakietów z dwoma rodzajami urządzeń wejścia-wyjścia [6]

Utworzenie sieci WAN polega na odpowiednim połączeniu zestawu przełączników pakietów. Taki przełącznik ma zwykle więcej niż jedno urządzenie wejścia-wyjścia, co pozwala na

wybór najlepszej topologii. Przykładowa sieć (rys. 22) pokazuje jeden ze sposobów połączenia komputerów i przełączników.

Rys. 22. Przykładowa sieć WAN [6]



Połączenia między przełącznikami i ich szybkości są dobrane tak, żeby odpowiadały spodziewanemu ruchowi oraz zapewniały redundancję w razie awarii. Oczywiście połączenia między przełącznikami działają zwykle z większymi szybkościami niż bezpośrednie połączenia z komputerami.

Sieć WAN jest zbudowana tak, aby umożliwiała jednoczesną wymianę ramek wielu parom komputerów. Stosowana jest tutaj zasada **zapisz i przekaż**. Oznacza to, że przełącznik pakietów, aby przekazać pakiet dalej, musi go najpierw buforować w pamięci. Kiedy pakiet przychodzi do przełącznika, odbywa się operacja *zapisz* – układ wejścia-wyjścia umieszcza kopię pakietu w pamięci przełącznika, a następnie informuje procesor, że przybył pakiet. Operacja *przekaż* polega na tym, że procesor analizuje pakiet, określa, przez który interfejs powinien być on wysłany i uruchamia wysyłanie odpowiednim urządzeniem wyjściowym. Jeśli wyjście jest zajęte wysyłaniem poprzedniego pakietu, procesor umieszcza wychodzący pakiet w kolejce związanej z tym urządzeniem wyjściowym.

4.2. Adresowanie i tablice tras

Z punktu widzenia komputera podłączonego do sieci, sieć WAN działa podobnie jak sieć LAN. Technika WAN określa dokładny format ramek używany do przesyłania danych. Każdy komputer dołączony do sieci WAN ma swój adres fizyczny. W wysyłanej ramce nadawca musi podać adres odbiorcy.

Sieci WAN używają zazwyczaj **adresowania hierarchicznego** w którym adres składa się z wielu części. W najprostszym przypadku adres składa się z identyfikatora przełącznika i identyfikatora komputera podłączonego do tego przełącznika (rys. 22). Użytkownicy i programy użytkowe traktują adres jako pojedynczą liczbę całkowitą.

Przełącznik pakietów, który musi wybrać dla pakietu odpowiednią ścieżkę, analizuje poszczególne części adresu. Jeśli pakiet jest skierowany do komputera dołączonego bezpośrednio do danego przełącznika, przełącznik przekazuje pakiet do tego komputera. Jeśli pakiet jest skierowany do komputera dołączonego do innego przełącznika pakietów, to musi on zostać przekazany przez jedno z szybkich połączeń, prowadzących do tego przełącznika.

Przełącznik pakietów nie przechowuje pełnej informacji o tym, jak dotrzeć do wszystkich możliwych odbiorców pakietów. Ma on tylko informację o następnym etapie, do którego ma być wysłany pakiet, skierowany do pewnego odbiorcy. Ten następny etap oczywiście zależy od ostatecznego miejsca przeznaczenia pakietu.

Informacja na temat następnego etapu zwykle jest przechowywana w postaci **tablicy tras**. Poniższa tabela przedstawia tablicę tras, przechowywaną w węźle numer 3 przykładowej sieci, pokazanej na rys. 22.

Tab. 1. Tablica przekazywania do następnego etapu dla węzła numer 3 dla sieci z rys. 22

Odbiorca	Następny etap
[1,1]	interfejs 2
[1,5]	interfejs 2
[2,2]	interfejs 2
[2,5]	interfejs 2
[4,1]	interfejs 4
[4,5]	interfejs 4
[3,1]	komputer E
[3,3]	komputer F
[3,6]	komputer G

Przełącznik przekazuje pakiet do następnego etapu niezależnie od tego, kto jest jego nadawcą oraz którą ścieżką pakiet przybył. Ten sposób przekazywania pakietu jest nazywany niezależnym od nadawcy i jest podstawową metodą stosowaną w sieciach. Niezależność od nadawcy znacznie upraszcza proces przesyłania danych.

Proces przekazywania pakietu do następnego etapu jest nazywany **wyznaczaniem trasy**. Wykorzystuje się w nim zalety adresowania hierarchicznego. W tablicy tras w tabeli 1 więcej niż jedna pozycja zawiera ten sam numer interfejsu, określający następny etap. Wszystkie pakiety na adres docelowy o identycznej pierwszej części są przekazywane do tego samego przełącznika pakietów, czyli mają taki sam następny etap. Wobec tego całą tablicę tras można tak uprościć, aby zawierała tylko po jednej pozycji na docelowy przełącznik pakietów, zamiast

po jednej pozycji na każdy docelowy komputer (tab. 2). Powoduje to, zwłaszcza w większych sieciach, znaczne zmniejszenie rozmiaru tablicy. Poza tym można zastosować indeksowanie tablicy zamiast przeszukiwania, co dodatkowo przyspiesza proces wyznaczania trasy.

Tab. 2. Skrócona wersja tablicy tras z tabeli 1

Odbiorca	Następny etap
[1, dowolny]	interfejs 2
[2, dowolny]	interfejs 2
[4, dowolny]	interfejs 4
[3, dowolny]	lokalny komputer

Jak wynika z powyższego rysunku, schemat adresowania dwustopniowego umożliwia używanie tylko pierwszej części adresu odbiorcy przy przekazywaniu pakietów we wszystkich przełącznikach oprócz ostatniego.

4.3. Wyznaczanie tras w sieciach WAN

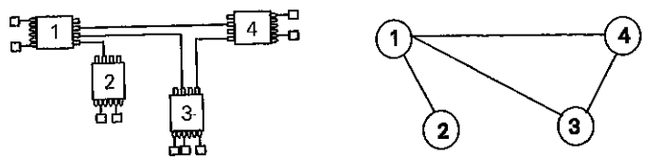
4.3.1. Modelowanie topologii

Budowanie sieci WAN o dużej pojemności jest możliwe dzięki temu, że do sieci można dodawać przełączniki, do których nie są dołączone żadne komputery. Są to tzw. **przełączniki wewnętrzne** (*interior switches*). Przełączniki, do których bezpośrednio są dołączone komputery, to **przełączniki zewnętrzne** (*exterior switches*). Każdy przełącznik musi mieć swoją tablicę tras, której zawartość zapewnia [6]:

- uniwersalne wyznaczanie tras – tablica tras w każdym przełączniku musi określać następny etap dla każdego możliwego odbiorcy;
- optymalne trasy – następny etap podany w przełączniku musi dawać najkrótszą ścieżkę do odbiorcy.

Na potrzeby wyznaczania tras topologia sieci modelowana jest za pomocą grafu. Każdy wierzchołek grafu odpowiada przełącznikowi pakietów. Krawędzie grafu występują tam, gdzie są bezpośrednie połączenia pomiędzy przełącznikami. Poniższy rysunek pokazuje przykładową sieć WAN i odpowiadający jej graf połączeń:

Rys. 23. Przykładowa sieć WAN oraz odpowiadający jej graf [6]



Graf przedstawia same przełączniki pakietów bez podłączonych komputerów, a zatem zasadniczą część sieci. Na podstawie układu krawędzi w grafie można dla każdego wierzchołka wyznaczyć tablicę tras. Pole następnego etapu w każdej pozycji zawiera parę (u,v), oznaczającą krawędź grafu od wierzchołka u do v.

Tab. 3. Tablice tras dla poszczególnych wierzchołków grafu [6]

Wierzchołek 1		Wierzchołek 2		Wierzchołek 3		Wierzchołek 4	
Odbiorca	**Następny etap**	**Odbiorca**	**Następny etap**	**Odbiorca**	**Następny etap**	**Odbiorca**	**Następny etap**
1	–	1	(2,1)	1	(3,1)	1	(4,1)
2	(1,2)	2	–	2	(3,1)	2	(4,1)
3	(1,3)	3	(2,1)	3	–	3	(4,3)
4	(1,4)	4	(2,1)	4	(3,4)	4	–

Chociaż rozmiar tablicy tras został skrócony przez zastąpienie tras do poszczególnych komputerów trasami do przełączników, w wielu wierszach zawartość pola *następny etap* powtarza się (tab. 3). Duplikaty wynikają m.in. z tego, że przełącznik pakietów zwykle nie ma połączeń ze wszystkimi innymi przełącznikami. Co więcej, czasem ma tylko jedno połączenie (jak *wierzchołek 2*), którym są wysyłane wszystkie wychodzące pakiety. Im większa jest sieć WAN, tym takich duplikatów w tablicy tras jest więcej. Zwiększa to oczywiście czasochłonność jej przeszukiwania.

W większości systemów WAN stosowana jest metoda służąca do eliminowania takich duplikatów tras, polegająca na używaniu **trasy domyślnej**. Pojedyncza pozycja w tablicy tras może zastępować listę pozycji z tą samą wartością następnego etapu. W każdej tablicy tras dopuszczalna jest tylko jedna wartość domyślnej pozycji i ma ona niższy priorytet niż inne wpisy. Jeśli algorytm wyszukiwania trasy nie znajdzie jawnego wpisu dla danego odbiorcy pakietu, to wtedy używa pozycji domyślnej. Dane z tab. 3 ulegają następującym zmianom, jeśli zastosuje się w nich pozycje domyślne, oznaczone gwiazdką:

Tab. 4. Tablica tras z zastosowaniem tras domyślnych [6]

Wierzchołek 1		Wierzchołek 2		Wierzchołek 3		Wierzchołek 4	
Odbiorca	**Następny etap**	**Odbiorca**	**Następny etap**	**Odbiorca**	**Następny etap**	**Odbiorca**	**Następny etap**
1	–	2	–	3	–	3	(4,3)
2	(1,2)	*	(2,1)	4	(3,4)	4	–
3	(1,3)			*	(3,1)	*	(4,1)
4	(1,4)						

Pozycja domyślna w tablicy tras (tab. 4) występuje wtedy, kiedy co najmniej dwóch odbiorców ma taką samą wartość następnego etapu. I tak tablica dla *wierzchołka 1* nie ma pozycji domyślnej, ponieważ ma on połączenia ze wszystkimi innymi wierzchołkami. Natomiast tablica dla *wierzchołka 2* ma tylko pozycję domyślną (oprócz komputerów dołączonych bezpośrednio), ponieważ wszystkie pakiety wychodzące są kierowane tym samym urządzeniem wejścia-wyjścia.

4.3.2. Wyliczanie tablicy tras

Do wyznaczania zawartości tablicy tras stosuje się oprogramowanie realizujące odpowiednie algorytmy. Pracują one na dwa różne sposoby: jako **statyczne** lub **dynamiczne** wyznaczanie tras [6]. Zaletą pierwszego jest prostota i mały narzut sieciowy, ponieważ program oblicza i zapisuje w pamięci trasy zaraz po włączeniu przełącznika. Potem trasy nie ulegają zmianie. Dużą wadą takiego podejścia jest brak elastyczności – trasy statyczne nie mogą być łatwo zmienione w przypadku przeciążenia lub awarii jakiegoś fragmentu sieci. Natomiast dynamiczne wyznaczanie tras polega na tym, że program buduje początkową tablicę tras po uruchomieniu przełącznika, ale potem tablica jest modyfikowana w miarę zmian sytuacji w sieci przez oprogramowanie, śledzące ruch w sieci i stan sprzętu sieciowego. Oprogramowanie może np. zmienić trasę, aby ominąć awarię.

Oprogramowanie obliczające trasy w sieci posługuje się modelem grafu sieci. Zwykle obliczana jest najkrótsza ścieżka w grafie (np. algorytm Dijkstry lub SPF [6]), przy czym *najkrótsza* nie zawsze musi oznaczać *prowadząca przez najmniejszą liczbę przełączników*. W niektórych metodach długość ścieżki może zależeć także od jej przepustowości lub innych wybranych czynników.

4.4. Przykładowe techniki WAN

4.4.1. Komutacja obwodów i komutacja pakietów

W komunikacji telefonicznej i podczas przesyłania danych wykorzystywane są dwie odmienne techniki komutacji (przełączania): **komutacja obwodów** i **komutacja pakietów**.

Komutacja obwodów polega na tym, że podczas tworzenia połączenia telefonicznego przez komputer lub abonenta sprzęt przełączający w systemie telefonicznym wyszukuje fizyczną ścieżkę od telefonu nadawcy do telefonu odbiorcy. Ta dedykowana ścieżka przechodzi odpowiednio przez kolejne centrale i jest aktywna aż do zakończenia rozmowy. Ważną właściwością techniki komutacji obwodów jest konieczność utworzenia ścieżki pomiędzy dwoma końcami połączenia, zanim będzie można zacząć rozmowę lub przesyłanie danych.

Alternatywną formą komunikacji jest komutacja pakietów, która polega na przesyłaniu

bloków danych w odpowiednim formacie i o limitowanym rozmiarze. W tego typu komunikacji nie jest zestawiane dedykowane połączenie pomiędzy dwiema lokalizacjami. Zamiast tego, gdy nadawca ma do wysłania blok danych, to jest on zapisywany w pierwszej centrali (tzn. zwykle w routerze), a następnie przekazywany dalej, po jednym przeskoku naraz. Komutacja pakietów nie wymaga zatem żadnej wstępnej konfiguracji. Sieci z komutacją pakietów dobrze sobie radzą z komunikacją interaktywną (żaden nadawca nie blokuje sieci na dłużej). Ponadto podczas wysyłania komunikatu złożonego z wielu pakietów każdy następny pakiet może być wysłany zanim poprzedni dotrze do adresata, co zmniejsza opóźnienia i zwiększa przepustowość. Z tego powodu technika komutacji pakietów dobrze się nadaje do tworzenia sieci komputerowych.

Sieci rozległe mają około 40 lat. Pierwszą na świecie siecią wymieniającą pakiety była ARPANET powstała na zlecenie Departamentu Obrony USA. Chociaż według dzisiejszych standardów była to sieć bardzo powolna, to badania nad jej rozwojem zaowocowały powstaniem algorytmów i standardów używanych do dzisiaj.

4.4.2. Sieci X.25

Standard **X.25** został opracowany przez instytucję ITU (*International Telecommunications Union*), ustanawiającą międzynarodowe standardy telefoniczne i był przez wiele lat oferowany przez firmy telefoniczne. Standard ten opisuje komunikację w pierwszych trzech warstwach modelu OSI (fizycznej, łącza danych i sieciowej) pomiędzy końcowym terminalem danych DTE (*Data Terminal Equipment*) a urządzeniem transmisji danych DCE (*Data Circuit -terminating Equipment*) [13]. Dane są przesyłane techniką komutowania pakietów.

Sieć składa się z co najmniej dwóch przełączników pakietów X.25, połączonych dzierżawionymi łączami. Początkowo sieci te były używane do podłączania terminali tekstowych do odległych komputerów. Komunikacja w sieci była dwukierunkowa. Interfejs sieciowy terminala rejestrował naciskane klawisze, umieszczał ich wartości w pakietach i wysyłał każdy pakiet siecią. Kiedy komputer wyświetlał wynik działania programu, to równocześnie przekazywał go do interfejsu sieciowego X.25. Interfejs wysyłał wynik w pakiecie do terminalu użytkownika [6]. Technika ta jest stosunkowo droga i nie zapewnia dobrych prędkości przesyłania danych, ponieważ protokół X.25 zaopatrzono w skuteczne, ale spowalniające jego pracę mechanizmy wykrywania i korekcji błędów. Wydajność została poświęcona na rzecz niezawodności.

4.4.3. Sieci Frame Relay

Frame Relay jest siecią z komutacją pakietów, powszechnie używaną jako łącze sieci WAN do przyłączania odległych stanowisk. Usługa ta jest wykorzystywana do łączenia sieci

LAN o różnych lokalizacjach. Na każdym końcu łącza znajdują się routery, które przyłączają poszczególne sieci do sieci Frame Relay. Specyfikacja Frame Relay opisuje komunikację sieciową w obrębie dwóch pierwszych warstw modelu OSI (fizycznej i łącza danych). Przesyłana siecią ramka może zawierać do 8K oktetów (bajtów) danych [6].

Komunikacja między użytkownikami realizowana jest przez zestawienie połączenia logicznego, zwanego obwodem wirtualnym [13], [14]. Połączenie to emuluje komutację obwodów. Stałe obwody wirtualne **PVC** (*Permanent Virtual Circuit*) zestawia się na stałe i nie rozłącza przy braku transmisji. Komutowane obwody wirtualne **SVC** (*Switched Virtual Circuit*) zestawia się na żądanie i rozłącza po określonym okresie bezczynności. Ponadto komunikacja kontrolna, związana z utrzymywaniem obwodów wirtualnych, jest przesyłana wydzielonymi kanałami logicznymi, innymi niż dane użytkownika. Węzły sieci Frame Relay sprawdzają tylko sumę kontrolną CRC, pozostawiając kontrolę przepływu danych oraz wykrywanie błędów wyższym warstwom modelu OSI. Wszystko to pozwala sieci Frame Relay działać z dość dużymi prędkościami (do 155 Mbit/s).

4.4.4. Sieci ATM

Przykładem techniki o rodowodzie telekomunikacyjnym jest **ATM** (*Asynchronous Transfer Mode*), czyli tryb przesyłania asynchronicznego [6]. Ponieważ plany szerokiego wykorzystania łączy ISDN (rozdz. 4.5.2.) nie powiodły się, firmy telekomunikacyjne opracowały następną, lepszą technikę, która miała odpowiadać na zapotrzebowanie przesyłania danych, obrazu i głosu jedną siecią. Technika ta obsługuje znacznie większe szybkości transmisji niż ISDN oraz oferuje więcej usług dodanych. ATM została zaprojektowana jako technika uniwersalna do działania we wszystkich rodzajach sieci komputerowych. Jednak, ponieważ jest dosyć kosztowna, wykorzystują ją przede wszystkim przez firmy telekomunikacyjne do tworzenia szybkich sieci szkieletowych.

Projektanci sieci ATM musieli spełnić sprzeczne wymagania. Przesyłanie siecią obrazu, głosu i danych wymaga zapewnienia różnych warunków przekazu. Przesyłanie dźwięku i obrazu wymaga małych opóźnień i małych niestabilności, co umożliwia dostarczanie dźwięku i obrazu bez przerw, opóźnień czy zakłóceń. Przesyłanie obrazu wymaga o wiele większej przepustowości niż przesyłanie dźwięku. Podstawowy problem stanowi więc dobranie odpowiedniego rozmiaru pakietu. Przepływność danych zwiększa się, gdy rozmiar pakietu jest duży, ponieważ narzut związany z nagłówkami jest proporcjonalnie mniejszy w stosunku do ilości przesyłanych danych. W związku z tym sieci opracowane pod kątem optymalizacji przepustowości mają pakiety o rozmiarze 4 kB lub większe. Niestety, pakiety tej wielkości nie nadają się do transmisji dźwięku, gdyż wprowadzają zbyt duże opóźnienia.

W technice ATM dane są rozdzielane do małych pakietów o stałym rozmiarze 53 bajty: 5 bajtów nagłówka (rys. 24) i 48 bajtów danych. Taki pakiet o stałym rozmiarze nazywany jest **komórką**.

Rys. 24. Pola nagłówka komórki ATM [6]

0	2	4	6
Sterowanie przepływem		VPI (pierwsze 4 bity)	
VPI (ostatnie 4 bity)		VCI (pierwsze 4 bity)	
VCI (środkowe 4 bity)			
VCI (ostatnie 4 bity)		Typ zawartości	PRIO
CRC			

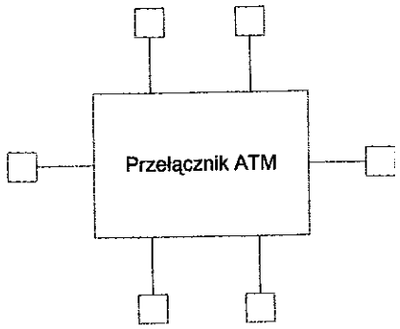
ATM pracuje w modelu **usług połączeniowych**. Zanim dwa komputery będą mogły się komunikować, muszą ustanowić połączenie: jeden zleca wykonanie połączenia, a drugi musi je zaakceptować. Połączenie ATM nazywane jest **kanałem wirtualnym VC** (*Virtual Channel*), ponieważ połączenia ATM są budowane przez umieszczanie danych w pamięci, a nie wyłącznie przez połączenie kabli. Inną stosowaną nazwą jest **obwód wirtualny VC** (*Virtual Circuit*). ATM przypisuje każdemu połączeniu VC 24-bitowy identyfikator, podzielony na dwie części: **VPI** (8 bitów) oraz **VCI** (16 bitów). VPI, czyli identyfikator ścieżki wirtualnej (*Virtual Path Identifier*), określa ścieżkę, którą VC biegnie przez sieć. Składnik VCI, identyfikator kanału wirtualnego (*Virtual Channel Identifier*), określa pojedynczy VC w ramach ścieżki. Komputer komunikujący się przez ATM nie rozróżnia tych dwóch pól, identyfikator ten określa się jako VPI/VCI.

Pole **CRC** zawiera sumę kontrolną, pozwalającą na stwierdzenie, czy komórka nie została uszkodzona w czasie transportu, natomiast pole **PRIO** określa, czy pakiet może być porzucony w razie przepełnienia sieci.

Przyjęty rozmiar komórki jest wynikiem kompromisu, zwolennicy techniki ATM więc twierdzą, że taki rozmiar zapewnia właściwą pracę sieci dla wszystkich obsługiwanych mediów. Krytycy ATM-u zaś uważają, że wybór taki oznacza, że nie jest ona dobra ani dla przesyłania dźwięku, ani dla danych. Ponadto zwracają uwagę na rozmiar nagłówka, który stanowi 10 proc. przesyłanych danych, twierdząc, że to zbyt dużo („podatek komórkowy"). Dla porównania w Ethernecie, gdzie pakiet może mieć do 1500 bajtów, nagłówek zajmujący 14 bajtów stanowi ok. 1 proc.

Sieć ATM jest zbudowana z jednego lub więcej **przełączników ATM** (rys. 25), każdy z nich ma wiele fizycznych punktów połączenia (**portów**).

Rys. 25. Przełącznik ATM z podłączonymi komputerami [6]



Do tych portów są dołączone komputery lub inne przełączniki. Komórki są przekazywane z komputera do najbliższego przełącznika, a następnie do kolejnych przełączników, aż dotrą do miejsca przeznaczenia. W każdym przełączniku znajduje się **tablica przekazywania**, w której określono sposób, w jaki będą przekazywane komórki. Pozycje w tej tablicy odpowiadają wszystkim możliwym parom VPI/VCI dla danego portu. Innymi słowy, każdy przełącznik wie, do którego następnego przełącznika ma przesłać pakiet. Przełącznik zmienia wartość VPI/VCI w każdej obsługiwanej przez siebie komórce, wymieniając otrzymaną wartość na nową, określającą dalszą trasę komórki. Ten proces nazywa się przepisywaniem etykiet lub **przełączaniem etykiet**. Dlatego ATM określa się mianem systemu przełączającego etykiety. W komórce nie umieszcza się adresu docelowego, ponieważ w ATM nie ma takiego pojęcia. Po ustanowieniu połączenia operuje się już tylko jego identyfikatorem VPI/VCI.

Obwód wirtualny można ustawić „na stałe", realizując np. usługę analogiczną do łącza dzierżawionego. Jednak większość ruchu w sieciach ATM odbywa się w sposób dynamiczny. Połączenia są zestawiane zgodnie z bieżącymi potrzebami na czas przesyłania danych.

Potrzeby użytkowników korzystających z usług głosowych i wideo są zaspokajane przez oferowaną w ramach ATM możliwość określania parametrów **jakości obsługi QoS (Quality of Service)** dla każdego połączenia. Specyfikacje jakości obsługi są podawane przy ustanawianiu połączenia i pozostają aktualne do jego zamknięcia. Po przyjęciu takiego zlecenia przełączniki muszą zarezerwować wynikające z niego zasoby. Jeśli przełącznik nie dysponuje odpowiednimi zasobami, to połączenie nie jest ustanawiane, generowany jest komunikat błędu.

Sieć ATM ma sporo wad. Do podstawowych należą wysoki koszt, spory narzut danych, przeznaczonych na nagłówek, złożoność związana z jakością obsługi, minimalne wsparcie do współpracy z innymi technikami itd. Te i inne przyczyny spowodowały, że obecnie sieci ATM są wykorzystywane głównie przez firmy telekomunikacyjne do tworzenia sieci szkieletowych. Jednym z najistotniejszych czynników wpływających na los ATM było pojawienie się Ethernetu gigabitowego, dzięki któremu firmy mogły unowocześniać swoje sieci bez ponoszenia aż takich dużych nakładów. Innym czynnikiem było pojawienie się możliwości przesyłania pakietów IP bezpośrednio przez łącza światłowodowe SONET/SDH (rozdz. 4.5.1.). ATM też często korzysta z tej sieci do transportu danych, możliwość przesyłania IP bezpośrednio przez SONET oznacza więc możliwość wyeliminowania jednej, kosztownej warstwy.

4.5. Łącza cyfrowe

Wraz z rozwojem Internetu i coraz większą ilością przesyłanych danych nastąpił znaczny rozwój sieci szkieletowych, zapewniających szybki i bardzo szybki przepływ danych na duże odległości, np. sieci SDH. Takie sieci jednak nie zapewniają transportu danych bezpośrednio do użytkownika. Obsługa lokalnego łącza abonenckiego wymaga nieco innej technologii, pozwalającej wykorzystać istniejącą infrastrukturę kabli miedzianych. Rozwiązania stosowane dla łączy abonenckich to modemy analogowe, o maksymalnej szybkości transmisji do 56 Kb/s lub łącza cyfrowe w technologii ISDN lub DSL, a zwłaszcza ADSL. Technologie cyfrowe zapewniają większą przepustowość niż modemy analogowe.

Alternatywną technologią, związaną z dostępem indywidualnych użytkowników, jest telewizja kablowa, która oferuje zwykle większą przepustowość niż linie telefoniczne i nie wymaga całkowicie nowej infrastruktury.

4.5.1. Synchroniczna sieć światłowodowa SDH

SONET jest to zdefiniowany w USA zestaw standardów do transmisji cyfrowej siecią światłowodową. Jest on używany przede wszystkim do realizowania sieci szkieletowych. SONET jest skrótem od angielskiej nazwy „synchroniczna sieć światłowodowa", czyli *Synchronous Optical Network*. W Europie jego odpowiednikiem jest standard **SDH** (*Synchronous Digital Hierarchy*). Nazwy te są czasem używane razem, oddzielone ukośnikiem lub nawet zamiennie. Działanie SONET jest opisane tylko na poziomie pierwszej warstwy modelu OSI (warstwa fizyczna). Dane przekazywane siecią SONET/SDH umieszczane są w ramkach, przy czym rozmiar ramki zależy od przepustowości łącza. Standard SONET/SDH określa sposób umieszczania danych w ramkach, sposób multipleksowania łączy o mniejszej pojemności w łączu o większej pojemności oraz sposoby przekazywania wraz z danymi informacji synchronizującej.

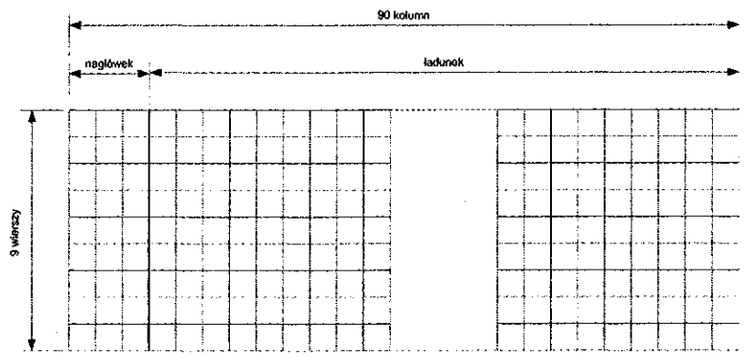
Przepustowości sieci synchronicznych zostały ujęte w standardy podane w tab. 5.

Tab. 5. Standardowe przepustowości sieci optycznych [6]

Standard optyczny	Przepustowość
OC-1	51,840 Mb/s
OC-3	155,520 Mb/s
OC-12	622,080 Mb/s
OC-24	1,244 Gb/s
OC-48	2,488 Gb/s
OC-192	9,5 Gb/s

Ramka musi mieć taką długość, aby można ją było przesłać w czasie 125 μs. Wartość 125 μs jest jedną z podstawowych stałych w telefonii cyfrowej i wynika z warunków koniecznych do przetwarzania mowy na sygnał cyfrowy [6]. I tak, w sieci o przepustowości OC-1 ramka ma długość 810 bajtów i składa się z 9 rzędów po 90 bajtów każdy. Pierwsze trzy bajty każdego rzędu są bajtami sterującymi.

Rys. 26. Ramka sieci SONET o przepustowości OC-1 (długości 810 bajtów) [6]



Dla łącza o przepustowości OC-3 ramka ma już 9 rzędów po 270 bajtów, czyli 2430 oktetów danych. Uzależnienie rozmiaru ramki od przepustowości łącza pozwala na synchroniczne przesyłanie danych (dane z taką samą prędkością przesuwają się na wejściu, jak na wyjściu).

Technologia SONET/SDH pozwala na budowanie połączeń punkt-do-punktu, ale umożliwia także tworzenie sieci złożonej z pierścieni przeciwbieżnych oraz sieci kratownicowych (siatkowych). Każda stacja w pierścieniu korzysta z urządzenia o nazwie **ADM** (*add/drop mux*), które oprócz przekazywania danych wzdłuż pierścienia może przyjmować dodatkowe dane z łącza lokalnego i dodawać je do ramek przekazywanych w pierścieniu lub też wydobywać przesyłane dane i dostarczać je do komputera lokalnego.

4.5.2. Łącze ISDN

W połowie lat osiemdziesiątych na świecie istniały trzy typy sieci komunikacyjnych: telefoniczna do prowadzenia rozmów telefonicznych, telewizyjna do przekazywania obrazu oraz nowo powstające sieci komputerowe do przesyłania danych. Ponieważ sieć komunikacji danych rozwijała się najbardziej intensywnie, firmy telekomunikacyjne doszły do wniosku, że powinny rozwijać swoje sieci w ten sposób, aby przenosiły nie tylko głos.

ISDN (*Integrated Services Digital Network*), czyli sieć cyfrowa z integracją usług, była jedną z pierwszych prób zapewnienia abonentom usług cyfrowych na dużą skalę [6]. W Europie obowiązuje standard EURO-ISDN. Usługa ta zapewnia użytkownikom w ramach konwencjonalnego okablowania łącza lokalnego zarówno możliwość przekazywania głosu, jak i danych, używając tego samego miedzianego kabla co analogowe systemy telefoniczne.

Podstawowy standard ISDN oferuje trzy rozłączne kanały cyfrowe oznaczane jako B, B i D (w skrócie 2B+D). Kanały B pracują z szybkością 64 Kb/s każdy. Służą one do przenoszenia zdigitalizowanego głosu, danych oraz skompresowanego obrazu. Kanał D działa z szybkością 16 Kb/s i służy jako kanał sterujący, czyli do porozumiewania się urządzeń między sobą i z centralą telefoniczną. Innymi słowy kanał ten służy do zestawiania połączeń i zarządzania nimi. Transmisją sterują odpowiednio zaprojektowane protokoły.

W standardzie podstawowym jeden kanał może służyć do przesyłania danych, a drugi do prowadzenia rozmowy. Jeśli oba kanały B zostaną przeznaczone do transmisji danych, to prędkość przesyłania wyniesie 128 Kb/s.

Rzadziej wykorzystywany standard ISDN, zwany głównym (*primary*), oferuje model transmisji 30 kanałów B o przepustowości 64 Kb/s każdy oraz jeden kanał sterujący D, działający z szybkością 64 Kb/s.

Standard ISDN został opracowany wiele lat temu przez firmy telekomunikacyjne. Wówczas szybkość transmisji 64 Kb/s wydawała się bardzo atrakcyjna w porównaniu z szybkością modemów telefonicznych. Jednak w miarę upływu czasu, zanim standard ten dopracowano i zatwierdzono, wydajność modemów telefonicznych znacznie wzrosła. Zostały też opracowane inne techniki przesyłania sygnałów z dużą przepustowością przy stosunkowo niskich kosztach. W wyniku tego ISDN jest obecnie uważany za usługę zbyt drogą w stosunku do oferowanych możliwości i wychodzi z użycia.

4.5.3. Łącze DSL

DSL (*Digital Subscriber Line*), czyli cyfrowe łącze abonenckie, jest jedną z najbardziej obiecujących technik usług cyfrowych w ramach łączy lokalnych. Ponieważ ta technika występuje w kilku wariantach, w odniesieniu do całej rodziny używa się skrótu **xDSL**.

Najbardziej interesującą techniką spośród xDSL jest technika **ADSL** (*Asymmetric Digital Subscriber Line*), czyli **łącze asymetryczne**. Usługa ta zapewnia wysyłanie i odbieranie danych z dużą szybkością, ale jak wynika z nazwy, szybkości te nie są jednakowe, czyli usługa działa asymetrycznie – dostępna przepustowość jest podzielona tak, że szybkość przesyłania danych w jedną stronę jest inna niż w drugą.

Skąd taka asymetria? Przeciętny użytkownik korzystając z Internetu odbiera o wiele więcej danych niż wysyła. Dane wysyłane to zwykle krótkie zlecenia, natomiast dane odbierane to całe strony HTML-owe, pliki wideo, obrazki itd., czyli tysiące, a nawet miliony bajtów. Aby rozróżnić te dwa kierunki wprowadzono pojęcia **strumienia wejściowego** (*downstream*) na określenie danych napływających do użytkownika z Internetu oraz **strumienia wyjściowego** (*upstream*) na określenie danych wysyłanych przez użytkownika. Technika ADSL optymalizuje łącze lokalne właśnie pod kątem takiego asymetrycznego ruchu, rezerwując większą przepustowość dla strumienia wejściowego.

Z punktu widzenia abonenta taka technologia zapewnia znacznie szybsze wyświetlanie stron WWW, niż przy rozwiązaniu symetrycznym. Oczywiście taka technologia nie jest przydatna dla tych użytkowników, którzy więcej przez łącza wysyłają niż odbierają, np. dla firm, które udostępniają swoje własne strony WWW.

Szybkość przesyłania danych jest ogromna, ponieważ maksymalna przepustowość do abonenta wynosi 6,144 Mb/s, a od abonenta – 640 Kb/s, z czego 64 Kb/s są wykorzystywane jako kanał sterujący, czyli efektywna przepustowość strumienia wyjściowego wynosi 576 Kb/s. Rzeczywista szybkość przesyłania może być znacznie mniejsza, zależy bowiem od długości kabla i jego przekroju, a także może być ograniczona przez operatora telekomunikacyjnego, dostarczającego usługę.

ADSL nie wymaga żadnych zmian w okablowaniu łącza lokalnego, posługuje się tą samą skrętką co analogowe łącze telefoniczne. Pozwala na dzielenie tego samego kabla pomiędzy rozmowę telefoniczną i dane. Modemy ADSL są podłączane zarówno u abonenta, jak i na centrali, równolegle z istniejącym analogowym sprzętem telefonicznym.

Idea techniki ADSL opiera się na tym, że do przesyłania danych cyfrowych można wykorzystać wyższe częstotliwości niż te stosowane w systemie telefonicznym. Jednak żadne dwa łącza lokalne nie mają identycznych charakterystyk elektrycznych, a zdolność do przenoszenia sygnałów zależy od odległości, rodzaju użytego okablowania oraz od poziomu zakłóceń elektrycznych. Dlatego projektanci nie mogli przyjąć żadnego określonego zestawu częstotliwości nośnych czy technik modulacji, który działałby we wszystkich przypadkach. ADSL jest **techniką adaptacyjną** – umożliwia dopasowanie sposobu przesyłania danych do możliwości danego łącza lokalnego. Modemy, znajdujące się na końcach łącza (u abonenta i w centrali), po włączeniu próbkują łącze, aby określić jego charakterystykę (czyli możliwości przesyłania sygnału) i na tej podstawie uzgadniają optymalne dla tego przypadku parametry komunikacji.

W ADSL stosuje się metodę modulacji **DMT** (*Discrete Multi Tone Modulation*), czyli dyskretnej modulacji wielotonowej. Pełna przepustowość pasma jest podzielona na podkanały. Każdy podkanał działa na innej częstotliwości nośnej, rozmieszczonej co 4,135 kHz, co zabezpiecza przed wzajemną interferencją sygnałów. Dodatkowo, aby nie zachodziła interferencja z sygnałem telefonu analogowego, ADSL nie korzysta z częstotliwości mniejszej niż 4 kHz. Większość podkanałów jest używana do transmisji wejściowej, a pozostałe do transmisji wyjściowej i jako kanały sterujące.

Jak już wspomniano, technika asymetryczna nie zawsze i nie dla każdego będzie najlepszym rozwiązaniem. Dlatego też powstały inne odmiany technik DSL.

SDSL, czyli **symetryczne łącze abonenckie** (*Symmetric Digital Subscriber Line*) zapewnia identyczną przepustowość w obu kierunkach. Może być ono wykorzystywane przez abonentów udostępniających informacje w Internecie, ponieważ łącze ADSL, ale o odwrotnej charakterystyce raczej nie jest oferowane przez firmy telefoniczne. SDSL używa innego schematu kodowania niż ADSL, w pewnych przypadkach więc może być wykorzystywane na łączach, na których ADSL nie może pracować.

HDSL, czyli **cyfrowe łącze abonenckie do szybkiej transmisji danych** (*High-Rate Digital Subscriber Line*) zapewnia w obu kierunkach przepustowość 1,544 Mb/s. Jego wadą jest ograniczona długość łącza lokalnego oraz to, że wymaga dwóch niezależnych kabli. Wariant tej techniki o nazwie HDSL2 pracuje na pojedynczej skrętce. Jedną z zalet HDSL jest możliwość pracy pomimo awarii, tzn. jeśli jeden z kabli zostanie uszkodzony, to transmisja odbywa się dalej, tylko z szybkością o połowę mniejszą.

4.5.4. Dostęp do Internetu przez telewizję kablową

Systemy telewizji kablowych mają prawie wszystkie udogodnienia potrzebne do przesyłania danych z dużą szybkością w kierunku użytkownika. Nośnikiem jest kabel koncentryczny, a więc medium o dużej pojemności i odporne na zakłócenia elektromagnetyczne. Zwykle taki system ma też niewykorzystaną przepustowość, ponieważ przystosowany jest do przesyłania o wiele większej liczby kanałów telewizyjnych, niż obecnie się przesyła. Teoretycznie możliwe jest więc takie rozszerzenie systemu kablowego, aby możliwe było przesyłanie informacji cyfrowych strumieniem wejściowym. W tym celu konieczne jest zamontowanie pary **modemów kablowych**: jednego w centrali telewizji kablowej, a drugiego u abonenta. Oba modemy muszą pracować na tej samej częstotliwości nośnej. Dane przekazywane z Internetu do modemu w centrali są kodowane za pomocą fali nośnej o ustalonej częstotliwości, innej dla

każdego odbiorcy. Modem po stronie abonenta „wyławia" swoją falę nośną ze strumienia danych, przesyłanych kablem, wydobywa zakodowane w niej informacje cyfrowe i przekazuje strumień danych do komputera odbiorcy.

Taki model działania nie wytrzymuje jednak próby skali, tzn. nie jest w stanie obsłużyć wystarczającej liczby odbiorców indywidualnych. Dlatego zwykle jedna częstotliwość fali nośnej jest przypisana do grupy odbiorców, a dodatkowo pakiety danych są adresowane. Modemy kablowe mają, podobnie jak karty ethernetowe, unikalne 48-bitowe adresy. Zanim pakiet zostanie przyjęty, modem sprawdza, czy adres odbiorcy w pakiecie zgadza się z adresem abonenta.

Opisana technika ma jeszcze jedną poważną wadę: cały system telewizji kablowej został zoptymalizowany pod kątem przesyłania danych strumieniem wejściowym. W konwencjonalnym systemie kablowym nie istniały żadne mechanizmy umożliwiające wysyłanie danych w kierunku wyjściowym.

Jednym z najwcześniejszych rozwiązań tego problemu była propozycja utworzenia **dualnej ścieżki**, tzn. okablowaniem telewizyjnym są przesyłane tylko dane wejściowe. Strumień danych wyjściowych jest przekazywany przez telefoniczne połączenie modemowe. W takim rozwiązaniu abonent musi mieć interfejs sieciowy, podłączony do dwóch modemów: modemu kablowego i standardowego modemu telefonicznego. Żeby podłączyć się do Internetu, użytkownik musi najpierw nawiązać połączenie telefoniczne ze swoim usługodawcą sieciowym. Wszystkie dane, które użytkownik wysyła, są przekazywane przez modem telefoniczny. Wszystkie dane odbierane przychodzą przez modem kablowy. Główną zaletą tego rozwiązania jest niski koszt z punktu widzenia usługodawcy, czyli operatora sieci kablowej, bo nie musi on ponosić żadnych kosztów związanych ze zmianą systemu okablowania. Główną wadą jest niezgrabny interfejs sieciowy oraz niewygoda użytkownika, któremu połączenie z Internetem blokuje linię telefoniczną.

Dalsze prace związane z dostępem do Internetu przez telewizję kablową skoncentrowały się jednak na takiej modyfikacji podstawowej infrastruktury, aby umożliwić komunikację dwustronną. W przyszłości taka ścieżka wyjściowa może być wykorzystana do dostarczania usług typu „wideo na żądanie" lub telewizji interaktywnej.

Jedną z najszerzej chyba wykorzystywanych technik komunikacji dwustronnej w telewizji kablowej jest system mieszany oparty na światłowodzie i kablu koncentrycznym **HFC** (*Hybrid Fiber Coax*). Światłowody są używane w urządzeniach centralnych, a kable koncentryczne w połączeniach z indywidualnymi użytkownikami. Inaczej mówiąc, światłowód występuje tam, gdzie wymagana jest większa przepustowość sieci, a kabel koncentryczny tam, gdzie przepustowość może być mniejsza.

Zastosowanie technologii HFC wymaga od firmy telewizji kablowej wymiany sporej części okablowania i wzmacniaczy (w tradycyjnej telewizji kablowej stosowano wzmacniacze jednokierunkowe, w tym rozwiązaniu niezbędne są wzmacniacze dwukierunkowe). Każdy z abonentów musi zaopatrzyć się w odpowiedni, dwukierunkowy modem kablowy. Odpowiednia część pasma przesyłania zarezerwowana jest oczywiście na kanały telewizyjne, pasmo o częstotliwościach wyższych jest wykorzystywane przez dane wejściowe, a pasmo o częstotliwościach niższych – przez dane wyjściowe. Cała technologia jest zaprojektowana z przyjęciem asymetrii przesyłu, czyli kanał danych wejściowych ma większą przepustowość niż kanał danych wyjściowych, podobnie jak w technice ADSL.

5. Protokoły TCP/IP

5.1. Definicje

TCP/IP (*Transmission Control Protocol/Internet Protocol*) jest zbiorem standardów określających szczegóły komunikacji w sieci, sposoby łączenia sieci i wyboru trasy w sieci. Nazwa pochodzi od głównych protokołów, a zbiór standardów obejmuje całą rodzinę protokołów i oprogramowania, udostępniającego szereg usług sieciowych. TCP/IP stanowi podstawowe rozwiązanie stosowane w światowej sieci, czyli Internecie. Umożliwia rozpatrywanie zagadnień dotyczących komunikacji niezależnie od sprzętu sieciowego.

Protokołem w sieci komputerowej nazywamy zbiór zasad syntaktycznych (składniowych) i semantycznych (znaczeniowych) sposobu komunikowania się jej elementów funkcjonalnych. Tylko dzięki nim urządzenia tworzące sieć komputerową mogą się porozumiewać. Podstawowym zadaniem protokołu jest identyfikacja procesu, z którym chce się komunikować proces rozpoczynający komunikację. W tym celu konieczne jest podanie sposobu określania właściwego adresata, sposobu rozpoczynania i kończenia transmisji, a także sposobu przesyłania danych. Przesyłana informacja może być porcjowana, protokół więc musi umieć odtworzyć ją w postaci pierwotnej. Informacja z różnych powodów może być przesłana niepoprawnie – protokół musi wykrywać i usuwać powstałe błędy, prosząc nadawcę o powtórną transmisję odpowiednich fragmentów informacji. Przesyłane fragmenty informacji nazywamy **pakietami**.

5.2. Intersieci

Sprzęt i technologie używane w sieciach lokalnych i rozległych różnią się znacznie między sobą. Każda konkretna technika sieciowa ma spełniać pewien ustalony zestaw zadań. Techniki sieci lokalnych mają zapewniać szybką komunikację na małe odległości, a techniki sieci rozległych mają umożliwić komunikację na duże odległości. W związku z tym żadna technika sieciowa nie jest w pełni odpowiednia do wszystkich zastosowań.

W układzie złożonym z wielu różnych sieci podstawowym pojawiającym się problemem jest zapewnienie użytkownikom możliwości komunikacji ze wszystkimi, niezależnie od tego, do której sieci są fizycznie podłączeni. Problem ten wystąpił w latach siedemdziesiątych, kiedy następował znaczny rozwój sieci komputerowych. Duże przedsiębiorstwa zaczęły tworzyć wiele sieci, z których każda była osobnym systemem. Każdy komputer był podłączony do

jednej sieci, a pracownicy musieli decydować, którego komputera użyć do wykonania danego zadania. Chcąc wysłać wiadomość do wielu odbiorców, użytkownik musiał niejednokrotnie korzystać kolejno z wielu komputerów. Taki system pracy nie wpływa dodatnio na jej wydajność. W związku z tym zaczęto rozwijać techniki umożliwiające komunikację między dwoma dowolnymi komputerami, niezależnie od tego, do której sieci są podłączone, przez odpowiednie połączenie tych sieci. Nazywamy to mechanizmem **jednolitych usług** [6]. Technikę umożliwiającą tworzenie jednolitego systemu komunikacyjnego, pozwalającego na łączenie wielu sieci fizycznych, nazywamy techniką intersieci.

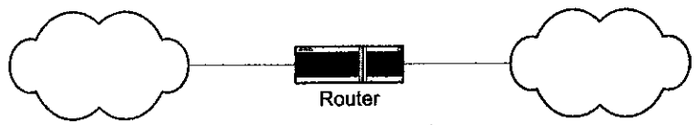
5.2.1. Technika intersieci

Dwie sieci używające różnych technik sieciowych nie mogą zostać ze sobą bezpośrednio połączone. Ramka z sieci używającej jednej techniki nie może zostać przesłana w sieci używającej innej techniki. Zwykle różne są w tych sieciach mechanizmy adresowania, inne parametry elektryczne itd. Utrudnia to zdecydowanie tworzenie mechanizmu jednolitych usług.

Musiały więc zostać opracowane metody, które pozwoliły na tworzenie takich usług w środowisku heterogenicznym, czyli pomimo różnic w technikach sieciowych. Systemy sieciowe tworzone z zastosowaniem tych metod opierają się na kombinacji odpowiedniego sprzętu i oprogramowania. Łączenie sieci fizycznych, korzystających z różnorodnych technik sieciowych, wymaga zastosowania specjalnego sprzętu. Oprogramowanie każdego z komputerów w takim systemie sieciowym musi umożliwić realizację jednolitych usług. Wynikowy system, łączący w całość wiele różnych sieci fizycznych, nosi nazwę **intersieci** (lub **internetu**). Jest to pojęcie bardzo ogólne. Rozmiar intersieci może być praktycznie dowolny, od kilku do tysięcy sieci. Dowolna może być także liczba składających się na tę sieć komputerów.

Podstawowym urządzeniem, pozwalającym na łączenie sieci heterogenicznych, jest **router**. Fizycznie jest on podobny do mostu, jest specjalistycznym komputerem przeznaczonym do wypełniania roli pomostu między sieciami. Router ma własny procesor i pamięć oraz osobny interfejs sieciowy dla każdej z sieci, do których jest podłączony. Z punktu widzenia sieci połączenie z routerem nie różni się niczym od połączenia z dowolnym komputerem (rys. 27).

Rys. 27. Sieci fizyczne w dowolnej technologii, połączone za pomocą routera [6]



Router może łączyć dwie sieci lokalne, dwie sieci rozległe lub sieć lokalną z siecią rozległą. Nawet kiedy obie sieci należą do tej samej kategorii, np. są to sieci lokalne, to nie muszą

być wykonane w tej samej technice. Router może łączyć lokalną sieć Ethernet z lokalną siecią FDDI. Chmurki na rysunku reprezentują sieci, które mogą stosować różne techniki.

Routery pełnią funkcję granicy między siecią LAN i WAN [14]. Ich podstawowym zadaniem jest komunikacja z innymi routerami o znanych adresach międzysieciowych. Adresy te są przechowywane w tablicach routingu, umożliwiających powiązanie adresów z fizycznym interfejsem routera, który należy wykorzystać w celu uzyskania połączenia ze wskazanym adresem.

Tematyka związana z konfiguracją i administrowaniem routerami stanowi osobny obszerny dział, opisany w [1].

5.2.2. Architektura intersieci

Routery umożliwiają wybór najwłaściwszej techniki sieciowej dla danego zastosowania, ponieważ za ich pomocą można potem połączyć tę sieć z innymi sieciami w jedną intersieć (rys. 28).

Rys. 28. Intersieć złożona z czterech różnych sieci fizycznych [6]

Na rysunku 28 przedstawiono trzy routery łączące cztery sieci w jedną intersieć. Każdy z tych routerów ma dwa połączenia, jednak komercyjne routery mogą łączyć więcej niż dwie sieci. W praktyce rzadko się zdarza, żeby wszystkie sieci jednej firmy były połączone jednym routerem. Wynika to z dwóch zasadniczych powodów:

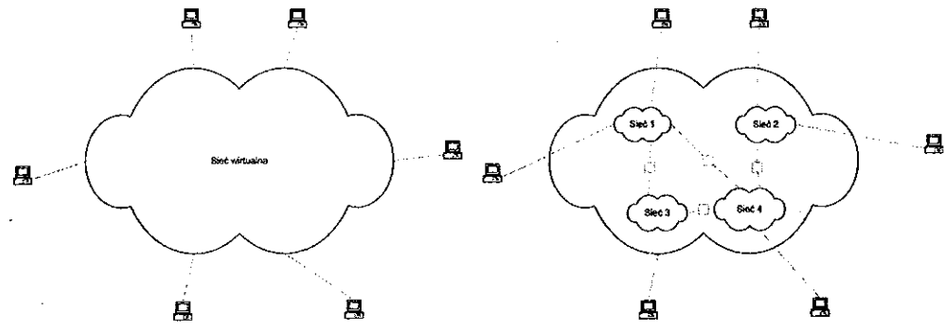
- przetworzenie każdego pakietu wymaga zaangażowania procesora i pamięci routera; połączenie kilku sieci jednym routerem mogłoby bardzo spowolnić pracę tych sieci;
- nadmiarowość zwiększa niezawodność działania sieci. oprogramowanie protokołów intersieci monitoruje stan połączeń i w razie awarii jednego z routerów kieruje pakiety inną trasą.

Projektując architekturę intersieci należy więc uwzględnić zarówno potrzeby dotyczące niezawodności pracy i wymaganych wydajności, jak również możliwe do zaakceptowania koszty tworzących daną intersieć urządzeń.

Cele budowy intersieci są proste: zapewnienie jednolitych usług w sieciach heterogenicznych. Realizacja takich usług wymaga, aby routery mogły przekazywać rozmaite schematy ramek i stosować schematy adresowania adekwatne do sytuacji. Wymaga to odpowiedniego oprogramowania routerów, jak i wsparcia ze strony oprogramowania komputerów. Z punktu widzenia użytkownika oprogramowanie intersieci stwarza wrażenie pracy w jednolitej,

wielkiej sieci, do której są przyłączone wszystkie komputery (rys. 29). System taki oferuje też jednolite usługi: każdy komputer ma unikalny adres, każdy też może wysyłać informacje do innego komputera. Oprogramowanie protokołów intersieci ukrywa szczegóły związane ze stosowanymi technikami fizycznymi, fizycznymi połączeniami, adresami i informacją o trasach – ani użytkownicy, ani programy użytkowe nie są świadome struktury fizycznych sieci składowych, ani łączących je routerów.

Rys. 29. Idea intersieci: sieć widziana przez użytkownika (A) oraz jej fizyczna budowa (B) [6]



A B

Mówimy, że intersieć jest **siecią wirtualną**, gdyż jej system komunikacyjny jest dla użytkowników tworem abstrakcyjnym. Choć odpowiednia kombinacja sprzętu i oprogramowania stwarza złudzenie pojedynczej sieci, sieć taka jednak w rzeczywistości nie istnieje. Rysunek 29 ilustruje pojęcie sieci wirtualnej i odpowiadającej jej sieci fizycznej.
Podstawowe protokoły intersieci to rodzina protokołów TCP/IP.

5.3. Protokół IPv4

Protokół **IP** jest protokołem warstwy sieciowej, należącym do rodziny protokołów TCP/IP. Jego zadaniem jest stworzenie użytkownikom Internetu wrażenia pracy w jednej wielkiej sieci. Protokół ten izoluje użytkowników i oprogramowanie warstw wyższych od warstwy fizycznej (będącej strukturą bardzo różnorodną) i od wszystkich zachodzących w niej zmian, związanych przede wszystkim z nieustannym rozwojem technologii.

Protokół IP w wersji 4 został opublikowany w 1981 r., kiedy Internet miał znacznie mniejszy zasięg niż obecnie. Do jego sukcesu przyczyniły się wszystkie podstawowe cechy protokołu i jest on wciąż podstawowym protokołem sieciowym. Jednak ze względu na pewne istotne ograniczenia (rozdz. 5.8.) została zaprojektowana i zaimplementowana nowa wersja, znana jako IPv6. Omówimy ją pod koniec tego rozdziału. Jeśli nie zostanie to jawnie wskazane, termin *protokół IP* będzie dotyczył jego wersji 4.

5.3.1. Zadania warstwy sieciowej

Warstwa sieci jest odpowiedzialna za określenie trasy transmisji między komputerem-nadawcą a komputerem-odbiorcą, czyli za tzw. trasowanie (*routing*). **Protokół trasowany**, innaczej – routowany (*routed protocol*) to dowolny protokół sieciowy, który umożliwia przenoszenie pakietu między komputerami na podstawie systemu adresowania. Definiuje on też format i zastosowanie pól wewnątrz pakietu. Przykłady protokołów trasowanych to IP, IPX, Apple Talk. **Protokół trasowania** – routingu (*routing protocol*) obsługuje trasowany protokół przez dostarczanie mechanizmów współdzielenia informacji o trasowaniu. Komunikaty protokołów trasowania przemieszczają się między routerami, a protokół trasowania umożliwia routerom komunikowanie się ze sobą w celu uaktualniania przechowywanych przez nie tablic. Przykładem protokołów trasowania są protokoły **RIP** (*Routing Information Protocol*) oraz **IGRP** (*Interior Gateway Routing Protocol*) [1].

Rozróżnia się dwa sposoby trasowania (routingu): statyczne i dynamiczne. **Trasowanie statyczne** jest zarządzane ręcznie, tzn. konfigurowane przez administratora sieci. Jeśli topologia sieci zmieni się, to administrator sieci musi ręcznie dokonać uaktualnienia poszczególnych pozycji w tablicy trasowania. Router zaprogramowany do trasowania statycznego przesyła pakiety przez z góry określone porty. **Trasowanie dynamiczne** polega tym, że informacje o trasach są automatycznie uaktualniane przez proces trasowania za każdym razem, kiedy z sieci są otrzymywane nowe informacje. Informacje o zmianach są wymieniane między routerami. Protokoły trasowania dynamicznego dzielą się na trzy kategorie: protokoły wektora odległości, protokoły zależne od stanu łącza oraz protokoły hybrydowe, opisane szczegółowo w [1] i [14].

5.3.2. Adresy IP

Internet składa się z wielu różnych sieci fizycznych, zbudowanych według różnych technologii, posługujących się różnymi schematami adresowania. Protokoły TCP/IP muszą zapewnić jedno wspólne adresowanie dla całego Internetu. Każdy komputer na całym świecie otrzymuje swój unikalny **adres IP**, różny od adresu fizycznego. Chociaż schemat adresowania w Internecie jest realizowany przez oprogramowanie, to adresy protokołu IP są stosowane jako punkty docelowe w Internecie, tak jak adresy sprzętowe są stosowane jako punkty docelowe w sieci fizycznej. Jednorodne adresowanie IP ukrywa szczegóły bazowych adresów fizycznych. Dwa programy użytkowe komunikują się ze sobą, nie znając swoich adresów fizycznych.

Standard IP określa, że każdy komputer ma przypisany 32-bitowy numer, czyli adres IP (32-bitowa liczba całkowita). Każdy adres jest podzielony na dwie części. Każdej sieci fizycznej przypisywana jest jednoznaczna wartość, zwana numerem sieci. Pojawia się ona zawsze

jako pierwsza część (prefiks) adresu IP każdego komputera, dołączonego do tej sieci. Każdy komputer musi mieć przypisany jednoznaczny numer w obrębie danej sieci (czyli sufiks). Daje to jednoznaczne adresowanie w obrębie całego Internetu.

Aby ułatwić ludziom posługiwanie się adresami IP, przyjęto ich zapisywanie w postaci czterech liczb dziesiętnych, oddzielonych kropkami. Każda liczba dziesiętna odpowiada ośmiu bitom adresu IP, np. adres: **10000000 00001010 00000010 00011110** zapisuje się jako: 128.10.2.30.

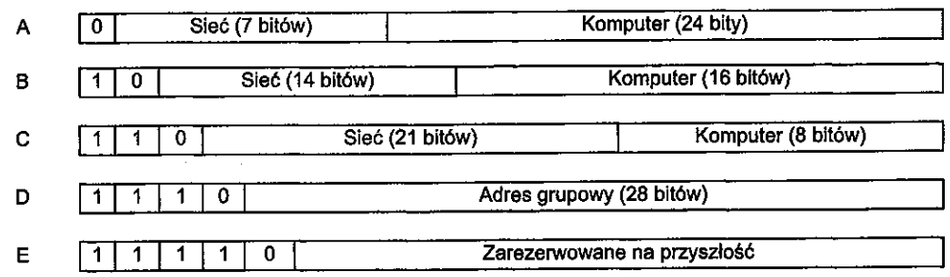
W celu zapewnienia jednoznaczności identyfikatorów w sieci wszystkie adresy przydzielane są przez jedną organizację: *Internet Assigned Number Authority* (IANA). Przydziela ona adresy sieciom instytucji, które chcą zostać podłączone do sieci Internet, oczywiście za pośrednictwem firm telekomunikacyjnych, działających jako dostawcy usług internetowych (ISP). Natomiast adresy maszyn administrator może przydzielać sam.

Klasy adresów IP

Projektanci protokołu IP, po obraniu rozmiaru adresu i zadecydowaniu o podziale każdego z nich na dwie części, musieli określić, ile bitów mieści się w każdej części. Adres sieci (prefiks) musi mieć wystarczającą liczbę bitów, aby umożliwiał przypisanie każdej sieci fizycznej jednoznacznego numeru w Internecie. Część adresu IP, adresująca komputer (sufiks), musi mieć wystarczającą liczbę bitów, aby umożliwić przypisanie jednoznacznego numeru każdemu komputerowi w sieci. Ponieważ Internet obejmuje dowolne techniki sieciowe i składa się z sieci o bardzo różnej wielkości, projektanci obrali kompromisowy schemat, działający i dla dużych sieci, i dla małych sieci. Znany on jest jako adresowanie z klasami, z których każda ma inny rozmiar prefiksu i sufiksu.

Adresy IP są podzielone na **klasy** (rys. 30), identyfikowane przez najstarsze bity.

Rys. 30. Klasy adresów IP [6]



Na podstawie czterech najstarszych bitów adresu można stwierdzić, do jakiej klasy należy dany adres, a zatem ile bitów będzie adresowało sieć, a ile sam komputer. W klasie A na adres

sieci jest przeznaczony pierwszy bajt adresu, w klasie B – 2 pierwsze bajty adresu, a w klasie C – 3 pierwsze bajty adresu. Te trzy klasy są nazywane klasami pierwotnymi, gdyż są przeznaczone na adresy komputerów. Adresów klasy A jest niewiele, bo 128, ale w każdej z tych sieci może się znajdować aż do 16 777 216 maszyn. W klasie B możemy umieścić 16 384 sieci po 65 536 komputerów, a w klasie C – 2 097 152 małych (do 256 maszyn) sieci. W praktyce liczba dostępnych adresów komputerów jest zmniejszona przez zarezerwowanie części z nich do określonych celów. Adres klasy D jest wykorzystywany, gdy ma miejsce jednoczesna transmisja do większej liczby urządzeń (czyli rozsyłanie grupowe), a klasa E jest zarezerwowana.

Zakresy adresów z poszczególnych klas są przedstawione w poniższej tabeli:

Tab. 6. Zakresy adresów z poszczególnych klas adresów [6]

Klasa	Adresy od	Adresy do
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.255

Adresy IP są nazywane samoidentyfikującymi się, gdyż klasa adresu może być określona na podstawie jego samego.

Adresowanie bezklasowe

Gdy Internet się rozrósł, pierwotny schemat adresowania z klasami okazał się niewystarczający. Przestrzeń adresów IP zaczęła się wyczerpywać, a z drugiej strony w wielu sieciach wiele adresów było nieużywanych. W związku z tym opracowano dwa nowe mechanizmy: **adresowanie z podsieciami** i jego następca, **adresowanie bezklasowe**. Uogólnienie polega na tym, żeby umożliwić podział adresu na prefiks i sufiks w dowolnym miejscu. Pozwala to znacznie lepiej dostosować adresy do rzeczywistych potrzeb. Nie trzeba wybierać między adresem sieci dla 256 lub 65 536 węzłów. Można podzielić adres tak, aby pozwalał adresować sieć o 1024 węzłach.

W adresowaniu bezklasowym z każdym adresem musi być związana dodatkowa informacja, która określa dokładnie granicę między częścią będącą adresem sieci a częścią będącą adresem komputera. Jest ona nazywana **maską adresową** lub inaczej **maską podsieci**. Jest to również liczba 32-bitowa, w której bity o wartości 1 oznaczają prefiks sieci, a bity o wartości 0 – adres komputera. Ta liczba musi być razem z adresem IP przechowywana w tablicach routerów. Przechowywanie jej w postaci maski bitowej przyspiesza obliczenia, które musi

wykonać router. Maska adresowa podczas operacji iloczynu logicznego pozwala z adresu IP odbiorcy (zawartego w pakiecie) obliczyć adres sieci, do której należy przesłać ten pakiet.

Jeśli mamy maskę podsieci 255.255.0.0

M = **11111111 11111111 00000000 00000000**

oraz adres odbiorcy 128.10.2.3

O = **10000000 00001010 00000010 00000011**,

to iloczyn logiczny O i M daje w wyniku adres sieci 128.10.0.0:

O & M = **10000000 00001010 00000000 00000000**.

W ten sposób dowolna liczba bitów może adresować samą sieć, np. maska 255.255.252.0

M = **11111111 11111111 11111100 00000000**

pozwala adresować sieć dla 1024 komputerów.

Notacja CDIR

We wnętrzu komputera zarówno adres IP, jak i maska bitowa są przechowywane jako 32- bitowe liczby binarne. Człowiek jednak niechętnie posługuje się taką reprezentacją. Zapis adresu w postaci czterech liczb dziesiętnych jest dużym ułatwieniem. Na potrzeby adresowania bezklasowego zaproponowano notację CDIR, pozwalającą łatwo uzupełnić go o maskę adresową. Dla podanego wyżej adresu 128.10.2.3 i maski 255.255.0.0 zapis ten ma postać:

128.10.2.3/16

Zapis ten oznacza, że adresowi towarzyszy maska, w której pierwszych 16 bitów jest ustawione na 1, czyli inaczej mówiąc – adres sieci zajmuje 16 bitów całego adresu. Drugi z powyższych przykładów to maska bitowa zawierająca 22 bity ustawione na 1.

5.3.3. Internet Protocol IP

Zasadniczo sieć TCP/IP udostępnia trzy zbiory usług:

- usługi bezpołączeniowego przesyłania pakietów;
- usługi niezawodnego przesyłania;
- usługi programów użytkowych.

Najbardziej podstawową usługą jest „bezpołączeniowe przenoszenie pakietów", nazywane *Internet Protocol*, czyli **protokół IP**. Protokół IP zawiera definicję podstawowej jednostki przesyłania danych (zwanej datagramem), definicję operacji trasowania (wybór drogi dla pakietu) oraz zbiór reguł, które służą realizacji operacji bezpołączeniowego przenoszenia pakietów.

Usługa ta jest określana jako zawodny system przenoszenia pakietów, gdyż nie ma gwarancji, że takie przenoszenie zakończy się sukcesem. Każdy pakiet jest przesyłany niezależnie od innych, a pakiety z jednego ciągu mogą podróżować różnymi drogami. Niektóre pakiety mogą

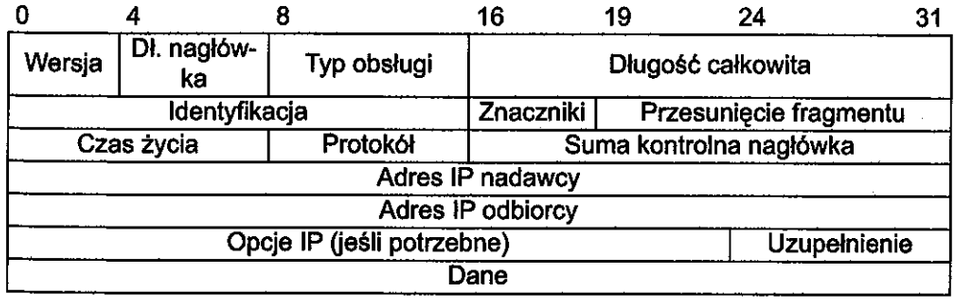
zostać zagubione, zduplikowane lub dostarczone z błędem. Oprogramowanie warstwy IP nie ma możliwości sprawdzenia, że taka sytuacja zaistniała, ani nie powiadomi o tym odbiorcy i nadawcy. Do obsługi tych błędów konieczne są dodatkowe warstwy protokołów (protokół ICMP).

Datagram IP

Datagram jest to podstawowa jednostka przesyłania danych za pomocą protokołu IP. Jest on podzielony na nagłówek i dane. Ponieważ przetwarzaniem datagramów zajmują się programy, ich zawartość i format nie są uwarunkowane sprzętowo. Gdy aplikacja ma zamiar wysłać informację przez sieć, konstruuje datagram zgodnie z wymogami lokalnej implementacji IP (konstruuje nagłówek, oblicza sumę kontrolną, umieszcza dane). Następnie datagram zostaje „wysłany". Każdy router otrzymujący datagram wykonuje na nim serię testów, np. obliczana jest suma kontrolna. W razie wykrycia błędów datagram jest odrzucany i do nadawcy wysyłany jest komunikat o błędzie (do tego celu służy protokół ICMP). Następnie zmniejszane jest i sprawdzane pole *Czas życia*. Jeśli limit czasu został przekroczony, to sygnalizowany jest błąd, jeśli nie, to zostaje obliczona nowa suma kontrolna i określony następny węzeł na drodze do odbiorcy. Czasami konieczny jest podział datagramu na mniejsze fragmenty. Wówczas każdy powstały datagram zostaje opatrzony nagłówkiem IP. Po dotarciu do celu fragmenty pierwotnego datagramu zostają scalone i wtedy dopiero przekazane właściwemu odbiorcy (np. właściwej aplikacji).

Protokół IP określa format datagramu IP (rys. 31), a szczególnie nazwy, położenie i rozmiar pól nagłówkowych (rozmiar tych pól jest podawany w bitach). Nagłówek składa się z co najmniej pięciu 32-bitowych słów, a długość poszczególnych pól, oprócz pola *Opcje* i *Uzupełnienie* jest stała.

Rys. 31. Format datagramu IP [6]



Datagram składa się z dwóch głównych części: nagłówka i danych. W nagłówku wyróżniamy następujące pola:

Wersja (4 bity) – informacja o wersji protokołu IP, która była użyta podczas tworzenia datagramu. Obecna wersja to 4, binarnie 0100.

Długość nagłówka (4 bity) – długość nagłówka mierzona w 32-bitowych słowach, najczęściej równa 5 (jeśli pola *Opcje* i *Uzupełnienie* nie są wypełnione).

Typ obsługi (8 bitów) – określa, w jaki sposób datagram powinien być obsłużony. Składa się z pięciu podpól (rys. 32). Część oprogramowania routerów ignoruje to pole. Poszczególne bity tego pola są następujące:

- **pierwszeństwo** (3 bity) – stopień ważności datagramu od 0 (normalny) do 7 (sterowanie siecią),
- bit **O** – oznacza prośbę o krótkie czasy oczekiwania,
- bit **S** – oznacza prośbę o przesyłanie szybkimi łączami,
- bit **P** – oznacza prośbę o dużą pewność przesyłania danych.

Długość całkowita (16 bitów) – długość całego datagramu, ale mierzona w bajtach. Rozmiar pola danych może być obliczony na podstawie danych zawartych w tym polu i w polu *Długość nagłówka*. Maksymalny możliwy rozmiar datagramu wynosi 65535.

Czas życia TTL (8 bitów) – określa, jak długo (w sekundach) datagram może pozostawać w systemie sieci (zwykle czas życia wynosi od 15 do 30 s). Każdy router podczas przetwarzania nagłówka zmniejsza wartość tego pola o co najmniej jeden. Gdy wartość tego pola zmaleje do zera, router porzuca datagram i wysyła do nadawcy komunikat o błędzie. Jest to mechanizm zabezpieczający przed krążeniem datagramu w sieci w nieskończoność.

Protokół (8 bitów) – zawiera numer identyfikacyjny protokołu transportowego, dla którego pakiet jest przeznaczony. Najbardziej popularnymi są ICMP (numer 1) i TCP (numer 6).

Suma kontrolna nagłówka (16 bitów) – służy do sprawdzenia poprawności nagłówka. Zawiera bitową negację sumy kolejnych 16-bitowych półsłów nagłówka. Obliczenie sumy kontrolnej nagłówka po otrzymaniu pakietu pozwala wykryć większość błędów (z wyjątkiem m.in. zagubienia zerowego półsłowa).

Opcje IP – opcje nie występują w każdym datagramie, a długość pola zależy od tego, jakie opcje zostały wybrane.

Uzupełnienie – zawartość tego pola zależy od wybranych opcji. Zawiera ono tyle zerowych bitów, aby długość nagłówka była wielokrotnością 32-bitowego słowa.

Rys. 32. Podział pola *Typ obsługi*

Pierwszeństwo	O	S	P	Nie używane

Pozostałe pola zostaną omówione w podrozdziale dotyczącym fragmentowania datagramu.

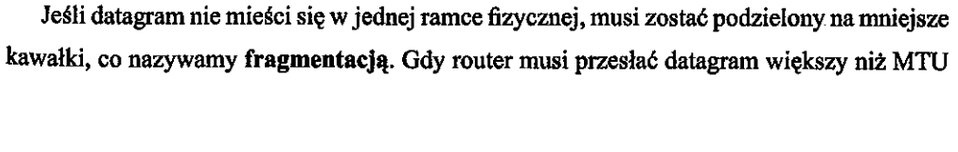
5.3.4. Przesyłanie datagramu przez sieć

Kapsułkowanie datagramu

Datagramy przenoszone są przez sieć fizyczną, która przesyła dane w ramkach. Każda technika sieciowa definiuje swój własny format ramek i mechanizm adresów fizycznych. Sprzęt przyjmuje i przesyła jedynie te ramki, których format jest zgodny z obowiązującym dla danej techniki sieciowej. Odpowiedzią na te problemy jest zapakowanie datagramu do ramki lub ramek sieci fizycznej.

Kapsułkowaniem (rys. 33) nazywamy sytuację, kiedy jeden datagram mieści się w jednej ramce sieciowej. Datagram jest przesyłany w części danych ramki. W polu typu ramki nadawca musi wpisać pewną ustaloną wartość wskazującą, że ramka zawiera datagram IP. Mechanizm odwzorowania adresów tłumaczy adres IP następnego etapu na odpowiadający mu adres fizyczny, który następnie zostaje umieszczony w nagłówku ramki.

Rys. 33. Kapsułkowanie datagramu IP [6]



U nadawcy każdy datagram kapsułkowany jest osobno. Po wyznaczeniu adresu następnego etapu nadawca kapsułkuje datagram w ramce i przesyła ją za pomocą sieci fizycznej pod wskazany adres. Po odebraniu ramki odbiorca odczytuje datagram i porzuca niosącą go ramkę. Jeśli datagram należy przekazać do innej sieci fizycznej, to tworzy się dla niego nową ramkę, zgodną ze specyfikacją tej drugiej sieci. Gdy zatem datagram dociera do ostatecznego odbiorcy ramka, w której jest zawarty, zostaje porzucona, a odbiorca otrzymuje taki sam datagram, jaki wysłał do niego nadawca.

Fragmentacja datagramu

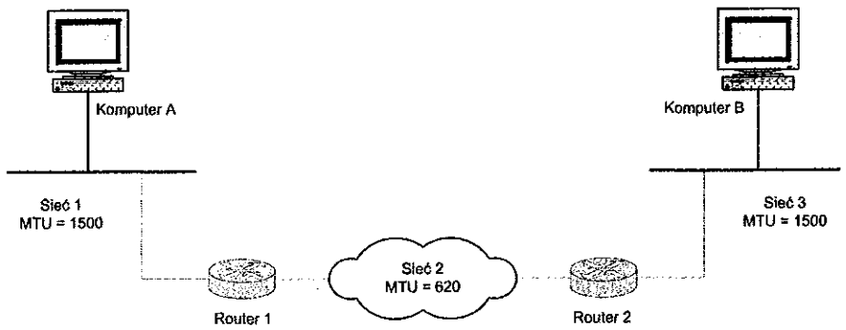
Każda z sieci fizycznych ma ustaloną górną granicę wielkości danych, które mogą być przesyłane w jednej ramce. Jest to maksymalna jednostka transmisyjna sieci **MTU** (*Maximum Transmission Unit*). Sprzęt sieciowy nie jest w stanie przekazywać ramek zawierających więcej danych niż określa MTU. W Internecie, złożonym z różnych sieci fizycznych, router może łączyć sieci o różnych wartościach MTU, może zatem dostać z jednej sieci datagram, którego nie będzie mógł przesłać do drugiej sieci.

Jeśli datagram nie mieści się w jednej ramce fizycznej, musi zostać podzielony na mniejsze kawałki, co nazywamy **fragmentacją**. Gdy router musi przesłać datagram większy niż MTU

sieci, najpierw dzieli taki datagram na mniejsze części, zwane fragmentami, po czym wysyła każdy z nich osobno.

W komputerze odbiorcy poszczególne fragmenty są składane w całość (defragmentacja).

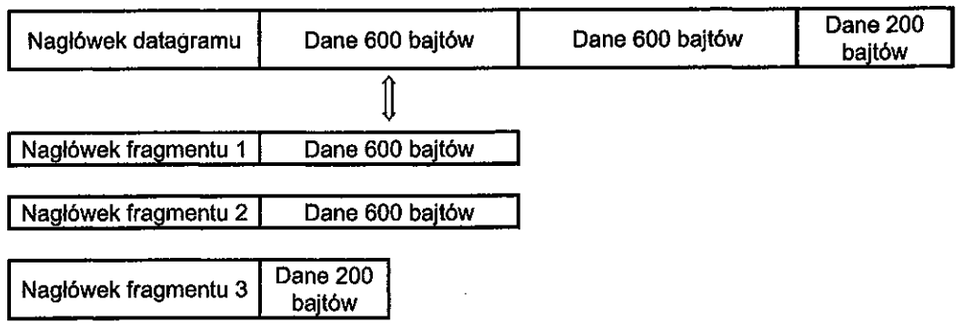
Rys. 34. Droga datagramu przez sieć [6]



Router 1 (rys. 34) musi fragmentować duże datagramy przesyłane od A do B. Router 2 musi fragmentować duże datagramy przesyłane z B do A. Przy fragmentacji datagramu przed wysłaniem przez sieć router, na podstawie MTU sieci i rozmiaru podanego w nagłówku datagramu, wyznacza maksymalną ilość danych, które można zmieścić w pojedynczym fragmencie oraz potrzebną liczbę takich fragmentów (rys. 35).

Każdy z fragmentów jest kompletnym datagramem i zawiera nagłówek, w którym powielona jest większość zawartości nagłówka pierwotnego. Wyjątek stanowi drugie 32-bitowe słowo nagłówka (rys. 31), którego trzy pola zawierają informacje dotyczące fragmentacji:

Rys. 35. Fragmentacja datagramu [6]



Identyfikacja (16 bitów) – zawiera liczbę całkowitą, jednoznacznie identyfikującą datagram, co zapobiega pomieszaniu fragmentów różnych datagramów.

Znaczniki (3 bity) – służy do kontroli fragmentacji. Pierwszy bit nie jest używany. Nadanie drugiemu wartości 1 oznacza zakaz fragmentacji. W tym przypadku, jeśli datagram nie może być przesłany w całości, to zostaje odrzucony i sygnalizowany jest błąd. Trzeci bit służy do oznaczenia ostatniego fragmentu datagramu – ma wartość zero. Inne fragmenty mają tutaj wartość jeden.

Przesunięcie fragmentu (13 bitów) – zawiera informację, w którym miejscu scalanego datagramu należy umieścić dane zawarte w tym fragmencie. Adres ten jest mierzony w jednostkach po 64 bajty.

Na podstawie zawartości tych pól odbiorca może sprawdzić, czy dany datagram jest całością czy fragmentem. Przy składaniu datagramów odbiorca na podstawie zawartości tych pól sprawdza, czy otrzymał już wszystkie fragmenty, nawet jeśli przyjdą one w niewłaściwej kolejności.

Składanie fragmentów u ostatecznego odbiorcy ma dwie zalety:

1. ogranicza się ilość informacji, jaką muszą przechowywać w pamięci routery (nie muszą one sprawdzać, czy datagram jest fragmentem i zbierać fragmentów w całość),
2. umożliwia to dynamiczne zmiany tras, tzn. w miarę potrzeb poszczególne fragmenty mogą podróżować różnymi trasami.

Ponieważ IP nie gwarantuje niezawodnego dostarczania pakietów, może w drodze zaginąć cały zakapsułkowany datagram lub niektóre jego fragmenty. Odbiorca może odtworzyć datagram tylko wtedy, gdy otrzyma wszystkie fragmenty. Musi więc przechowywać otrzymane fragmenty przez określony czas, czekając, aż dotrą wszystkie. Jeśli wszystkie fragmenty nie dotrą w tym czasie, zgromadzone fragmenty są usuwane (żeby nie zajmowały pamięci). Nie ma mechanizmu, który pozwoliłby odbiorcy na powiadomienie nadawcy, który fragment nie dotarł. Nadawca przecież nic nie wie o fragmentacji datagramu. Ponadto przy powtórnym wysłaniu datagram może wędrować inną trasą i jego podział na fragmenty może być zupełnie inny.

5.4. Odwzorowanie adresów IP

Oprogramowanie użytkowe wysyłające dane przez sieć dysponuje adresem IP odbiorcy jako adresem jednoznacznie identyfikującym komputer w Internecie. Dane są umieszczane w pakiecie IP. Oprogramowanie routerów, przez które wędruje pakiet, używa adresu IP odbiorcy do wyznaczenia adresu następnego etapu dla tego pakietu (jest to również adres IP). Następnie pakiet jest kapsułkowany w ramkę sieciową i wysyłany siecią fizyczną.

Niestety, sprzęt sieciowy nie potrafi interpretować adresów IP. Ramka sieci fizycznej musi zawierać adres fizyczny interfejsu sieciowego komputera, będącego następnym węzłem na drodze pakietu. Przed wysłaniem ramki zatem adres protokołu IP należy przetłumaczyć na adres fizyczny. Proces ten nazywa się **odwzorowaniem adresu** (*address resolution*).

Odwzorowywanie adresów jest lokalne dla danej sieci. Jeden komputer może odwzorować adres IP drugiego komputera na adres sprzętowy tylko wówczas, gdy oba są przyłączone do tej samej sieci fizycznej. Nie ma nigdy potrzeby odwzorowywania adresu IP komputera przyłączonego do innej sieci [6]. Router, do którego wysyłany jest pakiet, jest zawsze przyłączony do tej samej sieci fizycznej co komputer wysyłający. Odbiorca pakietu jest zawsze przyłączony do tej samej sieci fizycznej co ostatni router na drodze pakietu.

Trzy najczęściej używane mechanizmy odwzorowania adresu to:

- metoda opierająca się na tablicy odwzorowań adresów, używana najczęściej w sieciach rozległych,
- metoda odwzorowania adresów przez obliczanie, stosowana w sieciach z konfigurowalnymi adresami fizycznymi,
- metoda odwzorowania adresów przez wymianę komunikatów, stosowana m.in. w sieciach Ethernet.

Pierwsze dwa mechanizmy można zrealizować, mając do dyspozycji pojedynczy komputer: instrukcje i dane potrzebne do wykonania zadania są dostępne dla systemu operacyjnego danej maszyny. Metoda trzecia reprezentuje podejście rozproszone: komputer w celu odwzorowania adresu IP rozgłasza w sieci zapytanie i oczekuje odpowiedzi. Komunikat z pytaniem zawiera adres IP, odpowiedź – odpowiadający mu adres sprzętowy. Na pytanie odpowiada albo specjalizowany serwer (rozwiązanie scentralizowane) albo komputer, którego adres sprzętowy odpowiada podanemu adresowi IP (wówczas specjalizowane serwery nie są potrzebne).

Zestaw protokołów TCP/IP zawiera **protokół odwzorowywania adresów ARP** (*Address Resolution Protocol*) [6]. W modelu warstwowym nie tworzy on osobnej warstwy, ale należy do warstwy sieciowej (rys. 4). Standard ten definiuje dwa podstawowe rodzaje komunikatów: pytania i odpowiedzi. Pytanie zawiera adres IP, dla którego poszukiwany jest odpowiadający mu adres sprzętowy. Standard ARP określa, że pakiet z pytaniem ARP jest umieszczany (kapsułkowany) w ramce sieci fizycznej i wysyłany na adres rozgłaszania w danej sieci. Każdy komputer porównuje swój adres IP z otrzymanym w pytaniu. Pakiet z odpowiedzią nie jest już rozgłaszany, ale wysyłany bezpośrednio do komputera, który wysłał pytanie.

Aby nie powodować nadmiernego ruchu w sieci, oprogramowanie ARP zapamiętuje otrzymane informacje i używa ich podczas wysyłania kolejnych pakietów. Do tego celu służy przechowywana w pamięci operacyjnej niewielka tablica odwzorowań, ponieważ te informacje nie mogą być przechowywane przez zbyt długi czas. W razie przepełnienia się tablicy usuwa się z niej najstarsze informacje.

5.5. Protokół ICMP

Usługa przesyłania datagramów przez sieć, czyli protokół IP, nie ma wbudowanego mechanizmu powiadamiania o błędach. W rodzinie protokołów TCP/IP służy do tego protokół **ICMP** (*Internet Control Message Protocol*), czyli protokół komunikatów sterujących intersieci. Jeśli np. datagram podczas swojej wędrówki przez sieć zostanie uszkodzony lub gdy zostanie przekroczona dopuszczalna liczba etapów (pole *Czas życia*), to router może wygenerować komunikat o błędzie, używając protokołu ICMP. Komunikaty ICMP są także używane do testowania stanu sieci.

Na czas przesyłania przez sieć komunikaty ICMP są kapsułkowane w datagramy IP, podobnie jak datagramy UDP (rys. 37). Jednak w modelu warstwowym sieci protokół ICMP nie tworzy osobnej warstwy (rys. 4).

Podstawowe typy komunikatów przedstawia tabela 7 [15].

Tab. 7. Podstawowe typy komunikatów ICMP

Typ komunikatu	Znaczenie
Destination unreachable	Cel nieosiągalny – nie można odnaleźć urządzenia docelowego o podanym adresie
Time exceeded	Przekroczony limit czasu – wartość w polu czas życia (TTL) spadła do zera
Parameter problem	Problem z parametrem – w polu nagłówka została wykryta niedopuszczalna wartość
Source quench	Tłumienie nadawcy – kiedyś był używany do hamowania komputerów, które wysyłały zbyt wiele pakietów; obecnie kontrola przeciążeń w Internecie odbywa się głównie w warstwie transportowej
Redirect	Przekierowanie – wysyłany, kiedy router zauważa pakiet kierowany złą trasą; informuje komputer nadający o możliwym błędzie
Echo	Żądanie echa – sprawdzenie, czy dane urządzenie docelowe jest osiągalne i działa; urządzenie powinno odpowiedzieć komunikatem *Echo reply*
Echo reply	Odpowiedź echa – odpowiedź urządzenia na komunikat *Echo*
Timestamp request	Żądanie znacznika czasowego – komunikat taki, jak *Echo*, lecz ze znacznikiem czasowym
Timestamp reply	Odpowiedź ze znacznikiem czasowym – komunikat taki, jak *Echo reply*, lecz ze znacznikiem czasowym. Pozwala mierzyć szybkość działania sieci

Pełną listę komunikatów ICMP można znaleźć pod adresem [15]: http://www.iana.org/assignments/icmp-parameters.

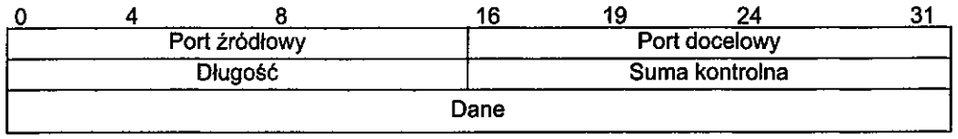
5.6. Protokół UDP

Protokół **UDP** (*User Data Protocol*) należy do warstwy transportowej, czyli leży ponad protokołem IP [15]. Jest protokołem bezpołączeniowym, co oznacza, że nie daje żadnych gwa-

rancji, iż wysyłany pakiet nie zostanie zagubiony czy zduplikowany oraz iż pakiety przyjdą do odbiorcy w takiej samej kolejności, w jakiej zostały wysłane. Aplikacja korzystająca z takiej metody transportu jest odpowiedzialna za właściwe przetworzenie odebranych danych.

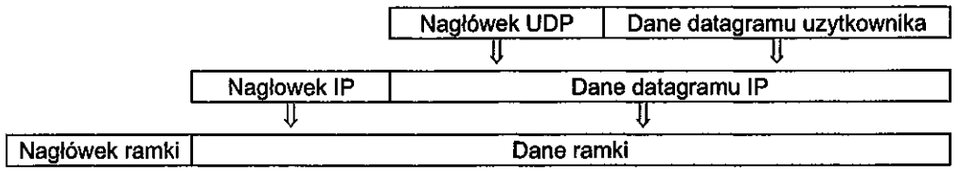
Komunikat UDP nazywa się **datagramem użytkownika** i ma format przedstawiony na rys. 36. W pierwszym słowie nagłówka mieszczą się port źródłowy i docelowy (mechanizm portów omówiono poniżej). Port źródłowy jest wykorzystywany jako adres pakietu, będącego odpowiedzią na dany pakiet. Pole określające długość zawiera długość całego datagramu, czyli nagłówka i danych w bajtach. Suma kontrolna przesyłanych danych (całego pakietu) jest opcjonalna. Jej umieszczanie jest zalecane, ponieważ pozwala kontrolować poprawność otrzymanych pakietów.

Rys. 36. Pola nagłówkowe datagramu UDP [15]



Komunikat UDP jest przed wysłaniem kapsułkowany w datagram IP, podobnie jak datagram IP jest kapsułkowany w ramkę sieci fizycznej. Powoduje to oczywiście przenoszenie przez sieć dodatkowych danych w postaci kolejnych nagłówków, ale zysk wynikający z „podziału pracy" pomiędzy poszczególne warstwy jest o wiele większy.

Rys. 37. Kapsułkowanie datagramu UDP



Systemy operacyjne większości komputerów zapewniają wielozadaniowość. Mechanizm adresowania IP i przesyłania datagramów nie zapewnia rozróżniania użytkowników ani programów użytkowych, do których jest skierowany datagram. Adresowanie wprost do konkretnego procesu nie jest możliwe, ponieważ m.in. procesy są tworzone i likwidowane dynamicznie, nadawca więc ma za mało informacji, aby stworzyć poprawny adres. Wprowadzono zatem abstrakcyjne punkty docelowe procesów nazwane **portami** protokołów. Każdy port jest identyfikowany dodatnią liczbą całkowitą. Nadawca musi znać adres IP i numer docelowego portu na maszynie odbiorcy, a komunikat musi zawierać numer portu nadawcy, żeby umożliwić przesłanie potwierdzeń i komunikatów o błędach. Port można wyobrazić sobie jako kolejkę pakietów.

Jako protokół bezpołączeniowy (a więc działający szybciej niż połączeniowy TCP) UDP jest stosowany wtedy, gdy ważniejsza jest eliminacja opóźnień niż integralność danych, np. przy rozmowach internetowych.

5.7. Protokół TCP

Programy użytkowe potrzebują do normalnej pracy niezawodnego przesyłania danych. Gdyby każda aplikacja musiała sprawdzać, czy pakiety przychodzą poprawnie (czy nie są zduplikowane, zgubione, dostarczone nie po kolei), byłoby to ogromne marnotrawstwo czasu pracy programistów oraz użytkowników korzystających z tych aplikacji. Aby aplikacja sieciowa mogła działać poprawnie, oprogramowanie sieci musi gwarantować szybką i niezawodną komunikację. Dane muszą być dostarczane dokładnie w takiej samej kolejności, w jakiej zostały wysłane, i nie może być strat ani duplikowania. Za niezawodność transmisji odpowiada protokół transportowy, z którego korzystają programy użytkowe przy wysyłaniu i odbieraniu danych. W rodzinie protokołów TCP/IP taką usługę zapewnia protokół **TCP** (*Transmission Control Protocol*).

TCP jest protokołem sekwencyjnym, pracującym w trybie połączeniowym. TCP wykorzystuje podczas przesyłania danych protokół IP, tzn. pakiety TCP (zwane segmentami) są kapsułkowane w datagramy IP (podobnie jak pakiety UDP – patrz rys. 37). Z punktu widzenia projektowania sieci TCP tworzy nad IP kolejną warstwę.

Połączeniowe sesje komunikacyjne TCP umożliwiają:

- sterowanie przepływem, co pozwala obu stronom aktywnie uczestniczyć w transmisji pakietów, chroniąc je przed zagubieniem,
- potwierdzenie odbioru pakietów, co daje nadawcy pewność, że pakiety dotarły do odbiorcy,

zachowanie kolejności przesyłania pakietów,

- sprawdzenie sumy kontrolnej, co pozwala upewnić się, że integralność pakietów nie została naruszona,
- powtórne przesłanie – dotyczy pakietów uszkodzonych lub zagubionych i realizowane jest szybko i efektywnie.

TCP organizuje dwukierunkową współpracę między warstwą IP a warstwami wyższymi, uwzględniając wszystkie aspekty priorytetów i bezpieczeństwa. Musi prawidłowo obsłużyć niespodziewane zakończenie aplikacji, do której właśnie wędruje datagram, musi też izolować warstwy wyższe (szczególnie aplikacje użytkownika) od skutków awarii w warstwie

protokołu IP. Scentralizowanie tych wszystkich funkcji w jednej warstwie umożliwia znaczną oszczędność nakładów na projektowanie oprogramowania.

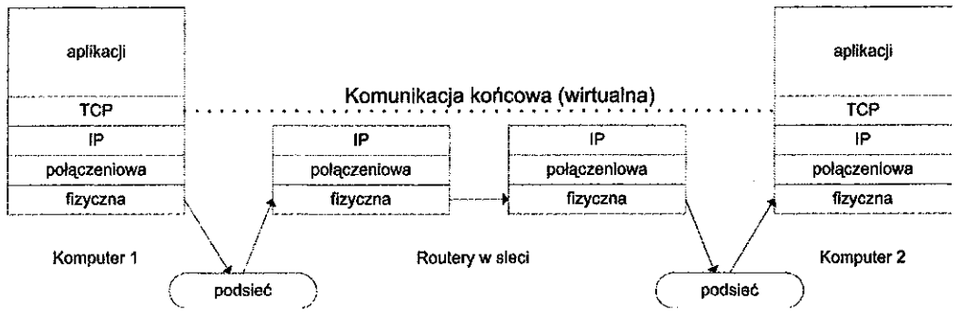
5.7.1. Kanał wirtualny TCP

Protokół TCP nazywa się protokołem „obsługi końców połączenia" (*end-to-end*), gdyż zapewnia on połączenie bezpośrednio między programem na jednym komputerze a programem na drugim. Program użytkowy może poprosić, żeby oprogramowanie TCP utworzyło połączenie, wysyłało i odbierało nim dane, a następnie je zamknęło. Połączenia udostępniane przez TCP są nazywane połączeniami wirtualnymi, gdyż uzyskuje się je tylko za pomocą oprogramowania. Moduły oprogramowania TCP na dwu maszynach wymieniają komunikaty, uzyskując złudzenie połączenia.

Rozpatrując TCP z punktu widzenia funkcjonalności można jego pracę potraktować jako ustanowienie wirtualnego kanału realizującego komunikację między konkretnymi „końcówkami" w sieci – tak to wygląda z punktu widzenia użytkownika (rys. 38). Rzeczywisty przepływ danych oczywiście odbywa się w warstwie fizycznej za pośrednictwem warstwy IP.

Z punktu widzenia TCP cała sieć jest systemem komunikacyjnym, który może przyjąć i dostarczyć komunikaty, nie zmieniając ani nie interpretując ich zawartości.

Rys. 38. Komunikacja wirtualna, nawiązywana za pomocą protokołu TCP [6]

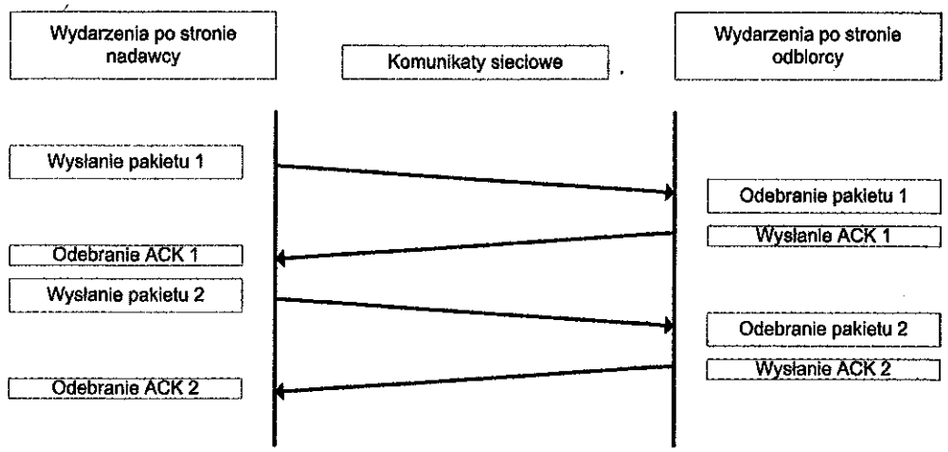


Oprogramowanie protokołu TCP jest niezbędne tylko tam, gdzie zachodzi przetwarzanie przesyłanych danych, czyli tam, gdzie działają aplikacje. Protokół TCP zatem pracuje u odbiorców końcowych. W urządzeniach sieciowych, np. routerach, protokół TCP nie jest potrzebny. Do przesyłania danych przez sieć wystarczy oprogramowanie protokołu IP.

5.7.2. Realizacja niezawodnego połączenia

Niezawodny protokół TCP jest zrealizowany na podstawie zawodnego przesyłania pakietów za pomocą protokołu IP. Mechanizmem, który umożliwia taką komunikację, jest metoda pozytywnego potwierdzania z retransmisją. Gwarantuje ona, że dane przesyłane w sieci nie są duplikowane ani tracone. Wymaga się, aby odbiorca komunikował się z nadawcą, przesyłając mu w momencie otrzymania danych komunikat potwierdzenia **ACK**. Podstawowa idea jest następująca: nadawca zapisuje sobie informacje o każdym wysłanym pakiecie i przed wysłaniem następnego czeka na potwierdzenie. Ponadto nadawca w momencie wysłania pakietu uruchamia zegar i wysyła ten pakiet ponownie, jeśli potwierdzenie nie nadejdzie w określonym czasie (rys. 39).

Rys. 39. Przesyłanie pakietów TCP z potwierdzeniem [6]

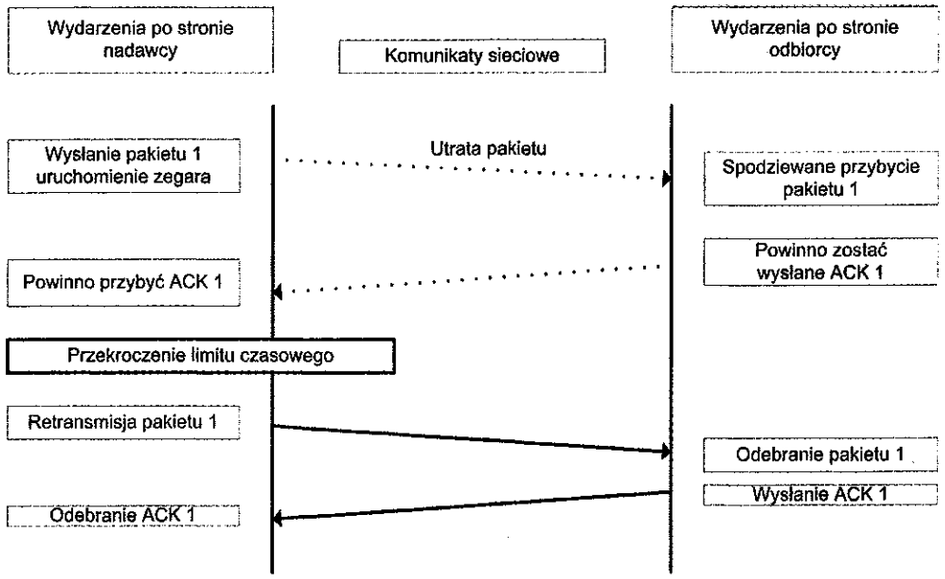


Gdy pakiet został zagubiony lub przekroczono limit czasu, nadawca przyjmuje, że pakiet się zagubił i wysyła go ponownie (rys. 40).

Pozostaje do rozwiązania problem, jak długo czekać z wykonaniem retransmisji. Potwierdzenie od komputera z sieci lokalnej przyjdzie zapewne w ciągu kilku milisekund. Za to taki czas retransmisji jest zupełnie nieodpowiedni dla dalekosiężnego połączenia satelitarnego. Zbyt wczesna retransmisja powoduje niepotrzebne zapychanie łącza, a zbyt późna – niepotrzebne opóźnienia w przesyłaniu danych.

Jako rozwiązanie tych problemów opracowano metodę retransmisji z adaptacją. Oprogramowanie TCP śledzi czasy opóźnień dla każdego połączenia i dostosowuje do nich czas retransmisji, używając do tego odpowiednich oszacowań.

Rys. 40. Mechanizm retransmisji [6]

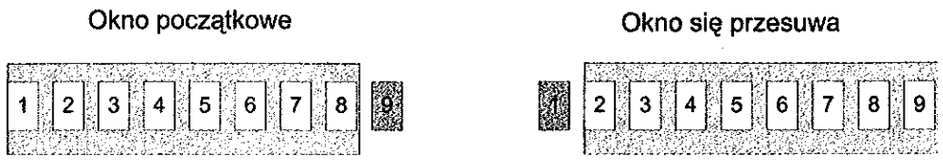


5.7.3. Technika przesuwającego się okna

Omówiony powyżej mechanizm przesyłania jest jednak mało efektywny, dane płyną tylko w jednym kierunku, a w momentach weryfikacji pakietów (np. obliczanie sum kontrolnych) – nie są w ogóle przesyłane. Powoduje to marnowanie przepustowości sieci. Usprawnieniem tego mechanizmu jest technika **przesuwającego się okna** (*sliding window*). Technika ta znacznie lepiej wykorzystuje przepustowość sieci, gdyż umożliwia wysyłanie wielu pakietów przed otrzymaniem potwierdzenia. Polega ona na umieszczeniu na ciągu pakietów okna ustalonego rozmiaru i przesyłaniu wszystkich pakietów, które znajdują się w jego obrębie. Pakiet, który został wysłany, a nie nadeszło jego potwierdzenie, nazywamy pakietem niepotwierdzonym. Maksymalna liczba pakietów niepotwierdzonych jest wyznaczona przez rozmiar okna.

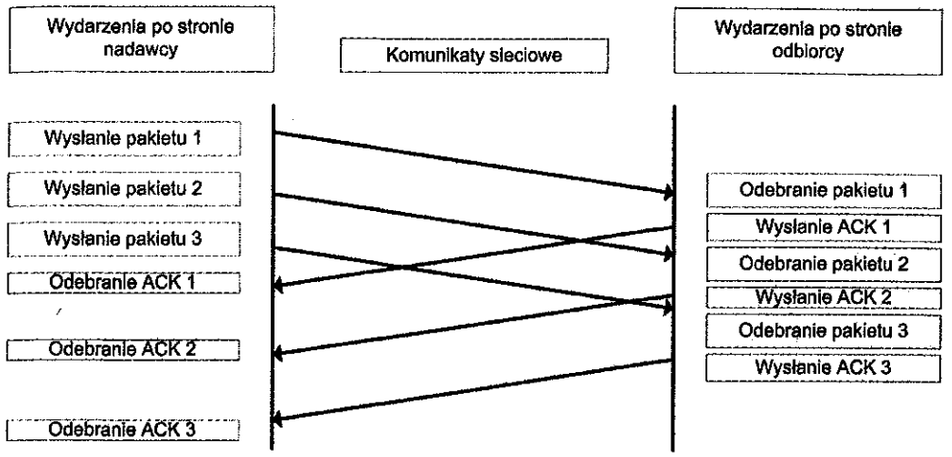
Jeśli rozmiar okna wynosi 8 (rys. 41), to nadawca ma możliwość wysłania kolejno ośmiu pakietów przed otrzymaniem potwierdzenia dla wysłanego pakietu. Gdy nadawca odbierze potwierdzenie dla pierwszego pakietu, okno przesuwa się i zostaje wysłany następny pakiet. Okno przesuwa się dalej, gdy nadchodzą kolejne potwierdzenia.

Rys. 41. Okno wyznaczające liczbę pakietów wysyłanych bez potwierdzenia



Pakiet dziewiąty może zostać wysłany, gdy przyszło potwierdzenie dla pierwszego pakietu. Retransmitowane są tylko te pakiety, dla których nie było potwierdzenia. Oczywiście protokół musi pamiętać, które pakiety zostały potwierdzone, i utrzymywać oddzielny zegar dla każdego niepotwierdzonego pakietu.

Rys. 42. Komunikacja z wykorzystaniem metody przesuwającego się okna [6]

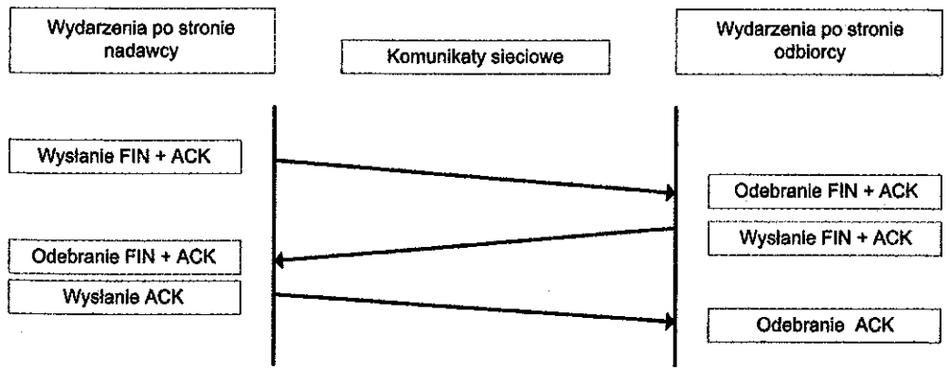


Skrócenie czasu przesyłania danych, uzyskiwane przy protokołach z przesuwającymi się oknami, zależy od rozmiaru okna i od szybkości, z jaką sieć akceptuje pakiety. Rozmiar okna wynoszący 1 (jeden) jest tym samym co podstawowy protokół z potwierdzeniem. Zwiększając rozmiar okna można w ogóle wyeliminować momenty nieaktywności sieci. Oznacza to, że w stabilnej sytuacji nadawca może wysyłać pakiety tak szybko, jak szybko sieć jest w stanie je przesyłać.

5.7.4. Trójetapowa wymiana komunikatów

Podczas rozpoczynania i kończenia połączenia TCP stosuje trójetapową wymianę komunikatów (rys. 43), która gwarantuje jednoznaczną zgodę obu stron na te zdarzenia.

Rys. 43. Komunikaty wymieniane na zakończenie połączenia [6]

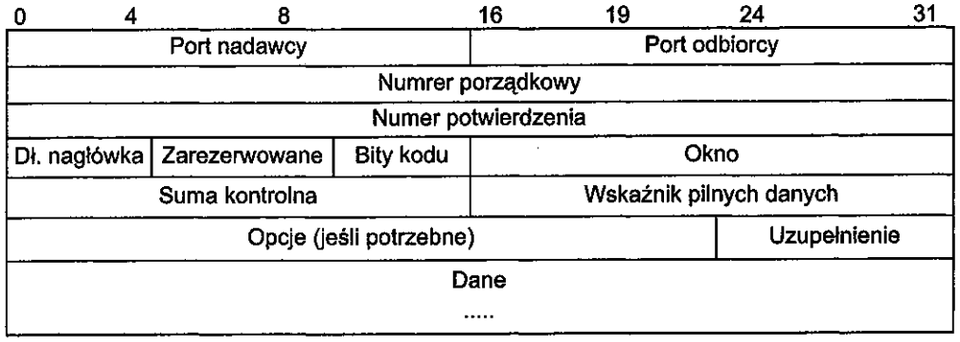


Segment SYN używany jest do nawiązania połączenia, a FIN – do jego zakończenia. Tak jak inne segmenty TCP, także SYN i FIN podlegają retransmisji. Podczas rozpoczynania połączenia generowana jest 32-bitowa liczba losowa, jednoznacznie identyfikująca to połączenie.

5.7.5. Segment TCP

Segment jest to jednostkowa porcja danych przesyłana pomiędzy oprogramowaniem TCP na różnych maszynach. Format segmentu jest następujący:

Rys. 44. Format segmentu TCP [6]



Nagłówek segmentu TCP, podobnie jak nagłówek datagramu IP, składa się z co najmniej pięciu 32-bitowych słów, podzielonych na opisane poniżej pola:

Port nadawcy, port odbiorcy (po 16 bitów) – numery portów TCP, które identyfikują programy użytkowe na końcach połączenia.

Numer porządkowy (32 bity) – wyznacza porcję danych segmentu w strumieniu bajtów nadawcy. Odbiorca stosuje to pole do porządkowania segmentów, które przyszły, oraz do obliczania numeru potwierdzenia.

Numer potwierdzenia (32 bity) – określa numer porządkowy danych, które zostały potwierdzone. *Numer porządkowy* odnosi się do strumienia płynącego w tym samym kierunku co segment, *numer potwierdzenia* zaś odnosi się do strumienia danych płynących w kierunku przeciwnym.

Długość nagłówka (4 bity) – liczba całkowita, określająca liczbę 32-bitowych słów nagłówka (pole *Opcje* ma zmienną długość).

Zarezerwowane (6 bitów) – pozostawione do wykorzystania w przyszłości.

Bity kodu (6 bitów) – informacja o zawartości segmentu, np. dane, potwierdzenie, prośba o ustanowienie lub zamknięcie połączenia itp.

Okno (16 bitów) – rozmiar bufora TCP – określa, ile danych oprogramowanie odbiorcy może przyjąć.

Suma kontrolna (16 bitów) – suma kontrolna obliczana dla całego segmentu, która powinna być obliczona także przez odbiorcę w celu sprawdzenia poprawności otrzymanego pakietu.

Wskaźnik pilnych danych (16 bitów) – adres początku obszaru, w którym znajdują się pilne dane.

5.7.6. Porty i połączenia

Protokół TCP umożliwia wielu programom użytkowym działającym na jednej maszynie jednoczesne komunikowanie się przez sieć oraz rozdziela między te programy użytkowe przybywające pakiety TCP. Podobnie jak UDP, TCP używa numerów portów protokołu do identyfikacji w ramach komputera końcowego odbiorcy. Każdy z portów ma przypisaną małą liczbę całkowitą, która go jednoznacznie identyfikuje.

TCP działa wykorzystując połączenia i te połączenia są identyfikowane przez parę punktów końcowych. TCP identyfikuje punkt końcowy jako parę liczb całkowitych (węzeł, port), gdzie węzeł oznacza adres IP węzła, a port jest portem TCP w tym węźle. Na przykład punkt końcowy (128.10.2.3,25) oznacza port 25 maszyny o adresie IP 128.10.2.3. W efekcie może istnieć połączenie pomiędzy np.: (18.26.0.36, 1069) oraz (128.10.2.3, 25) i w tym samym czasie może też istnieć połączenie (128.9.2.3, 1184) oraz (128.10.2.3, 25). TCP identyfikuje połączenie za pomocą pary punktów końcowych, dany numer portu może więc być przypisany do wielu połączeń w danej maszynie.

5.8. Protokół IPv6

Do sukcesu protokołu IP w wersji 4 przyczyniły się wszystkie jego podstawowe cechy. Pozwolił on obsłużyć różnorodność sieci fizycznych, składających się na Internet, przez:

- zdefiniowanie jednolitego formatu pakietów i mechanizmu ich przesyłania,
- zdefiniowanie jednolitego systemu jednoznacznych adresów,
- skalowalność w dosyć dużym zakresie.

Twórcy protokołu IP4 nie przewidzieli jednak tak ogromnego rozwoju Internetu. Dynamiczny wzrost liczby przyłączanych komputerów doprowadził do sytuacji, w której sieć „pęka w szwach", a pula dostępnych adresów jest na wyczerpaniu. Teoretycznie 32-bitowy adres pozwala zaadresować ponad 4 miliardy (4 294 967 296) maszyn. Praktycznie, przez podział przestrzeni adresowej na klasy i z powodu innych ograniczeń przestrzeń adresowa obejmuje ok. 250 milionów adresów [27]. Także zadania, jakim mają sprostać współcześnie i w przyszłości łącza internetowe, są zupełnie inne niż 20 lat temu. Podstawowe ograniczenia protokołu IP w wersji 4, poza zbyt małą przestrzenią adresową, są następujące:

- bezpieczeństwo, np. IPSec (rozdz. 8.3.4), jest opcjonalnym dodatkiem,

- usługi jakości obsługi QoS są trudne do zrealizowania; np. programy, które przekazują dźwięk i obraz, powinny dostarczać dane w regularnych odstępach, oprogramowanie IP powinno więc w tym przypadku unikać częstego zmieniania tras,
- tablice routowania są zbyt duże i przetwarzanie ich trwa zbyt wolno,
- utrudniona jest rzeczywista mobilność komputerów,
- automatyczna konfiguracja urządzeń dołączonych do sieci, np. za pomocą usługi DHCP (rozdz. 6.4), nie działa poza granicami danej organizacji.

Rozwiązaniem wielu z tych problemów jest protokół IP w wersji 6, czyli IPv6, zwany początkowo protokołem IP następnej generacji **IPng** (*Internet Protocol Next Generation*) [6], [10].

Protokół IP w wersji 6 został zaprezentowany w dokumencie RFC 1752 *The Recomendation for the IP Next Generation Protocol* i w listopadzie 1994 roku został przyjęty jako *Proposed Standard*. Zaprojektowano go tak, aby pozwolić użytkownikom na stopniową zmianę stosowanego standardu.

Protokół IPv6 zachował najważniejsze cechy swojego poprzednika: jest protokołem bezpołączeniowym i każdy datagram ma wyznaczaną trasę niezależnie od innych. Najważniejsze zmiany, jakie zaimplementowano w IPv6, są następujące:

- dłuższe adresy – dotychczasowe adresy 32-bitowe zostały zastąpione adresami 128-bitowymi, tym samym przestrzeń adresowa została zwiększona do ok. $3{,}4\times10^{38}$;
- większa elastyczność i nowe struktury adresowe – nastąpiło odejście od adresowania bazującego na klasach; IPv6 definiuje trzy formaty adresów: unicast, multicast (mające swoje odpowiedniki w IPv4) i nowy anycast; łatwa autokonfiguracja adresu; tworzenie hierarchii adresów;
- uproszczony i bardziej elastyczny format nagłówków datagramów;
- zwiększenie bezpieczeństwa datagramów – wprowadzono elementy zapobiegania najczęściej spotykanym atakom oraz wbudowano opcje szyfrowania, identyfikacji hostów (węzłów sieci) i kontroli integralności danych, co zapewnia bezpieczeństwo na całej długości połączenia;
- lepsze wsparcie dla QoS przez etykietowanie pakietów, ustanawianie odpowiedniej jakości ścieżki i powiązanie tych pakietów ze ścieżką;
- przestrzeń dla przyszłych rozszerzeń protokołu;
- wsparcie dla komputerów i urządzeń przenośnych;
- pełna kompatybilność z IPv4, tzn. oprogramowanie IPv6 zawiera mechanizmy pozwalające na współpracę komputerów wykorzystujących oba standardy protokołu IP.

Protokół IP w wersji 6 został zaimplementowany we wszystkich ważniejszych systemach

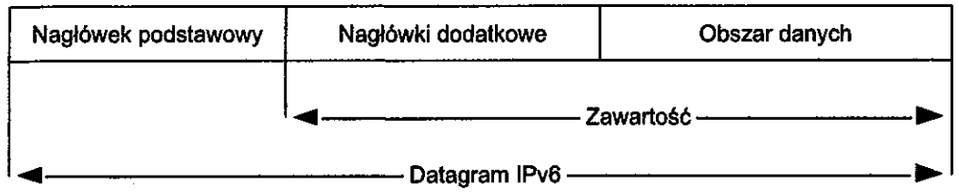
operacyjnych. Przewiduje się, że większość użytkowników Internetu stopniowo zacznie go wykorzystywać, choć na pewno jeszcze długo będzie można korzystać także z wersji 4.

5.8.1. Struktura datagramu

Tym razem datagram IP składa się z trzech części (rys. 45):

- nagłówek podstawowy pakietu w wersji 6 (*IPv6 Header*);
- nagłówki dodatkowe (*Extension Headers*) – dodatkowe opcje pakietu; większość z nich nie jest przetwarzana przez routery (w przeciwieństwie do pakietów IPv4), co znacznie przyspiesza przesyłanie pakietów; są to m.in. opcje dotyczące fragmentacji, autentyzacji czy kapsułkowania danych;
- obszar danych (*Upper Layer Protocol Data Unit*) – dane przenoszone w tym pakiecie, czyli zazwyczaj pakiet pochodzący z warstwy wyższej, np. z TCP lub UDP.

Rys. 45. Budowa datagramu protokołu IPv6 [10]

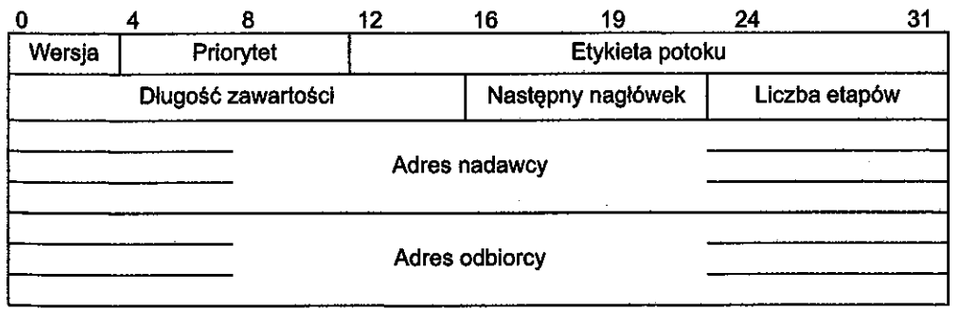


Minimalny datagram składa się z nagłówka podstawowego i danych. Obszar za nagłówkiem podstawowym, czyli nagłówki dodatkowe i dane, nazywa się ładunkiem datagramu.

5.8.2. Struktura nagłówka datagramu

Nagłówek podstawowy datagramu IPv6 ma stałą długość (40 bajtów). Liczbę pól nagłówka ograniczono do minimum, pozostawiając osiem niezbędnych do przesłania datagramu przez sieć (rys. 46).

Rys. 46. Format nagłówka datagramu IPv6 [10]



Poszczególne pola nagłówka datagramu niosą następujące informacje:

Wersja (4 bity) – pole zawierające numer wersji protokołu IP (wartość 6, binarnie `0110`).

Priorytet (8 bitów) – pole pozwalające na nadanie datagramowi wymaganego dla niego priorytetu przesyłania. Pozwala ono określać klasę przepływu, na podstawie której wybierana jest trasa, np. z odpowiednio małym opóźnieniem dla dźwięku lub obrazu.

Etykieta potoku (20 bitów) – etykieta strumienia pakietów, czyli sekwencji pakietów przesyłanych z konkretnego źródła do konkretnego odbiorcy typu *unicast* lub *multicast*. Pakiety są przesyłane ustaloną wcześniej ścieżką. Oznacza ona strumień pakietów specjalnego przeznaczenia lub wymagający przetwarzania w czasie rzeczywistym. Wraz z polem *Priorytet* pozwala wprowadzać usługi typu *Quality of Service*.

Długość zawartości (16 bitów) – długość datagramu bez nagłówka, podawana w bajtach.

Następny nagłówek (8 bitów) – identyfikuje typ nagłówka następujący bezpośrednio za nagłówkiem podstawowym datagramu IP. Jeśli nie ma żadnego nagłówka dodatkowego, to w tym polu jest określony typ danych przenoszonych w datagramie.

Liczba etapów (8 bitów) – pole zmniejszane o 1 przez każdy węzeł sieci, przetwarzający pakiet. Kiedy to pole osiągnie wartość zero, pakiet jest niszczony.

Adres nadawcy (128 bitów) – adres nadawcy datagramu.

Adres odbiorcy (128 bitów) – adres odbiorcy/odbiorców datagramu.

Nagłówki dodatkowe mogą mieć różną długość i dlatego każdy z nich musi mieć pole *długość nagłówka*. Każdy z nich ma też pole *następny nagłówek*. Ostatni nagłówek dodatkowy ma tam wpisaną wartość określającą typ danych przenoszonych w datagramie. Powody używania wielu nagłówków są dwa: ekonomia i rozszerzalność. W protokole IPv6 zaprojektowano wiele różnych możliwych nagłówków dodatkowych, jednak większość datagramów wykorzystuje niewielką ich część. Pola związane z pozostałymi możliwymi opcjami nie powiększają niepotrzebnie rozmiaru nagłówka (np. pola związane z fragmentacją są obecne zawsze w nagłówku IPv4). Taki mechanizm pozwala też na dodawanie do protokołu nowych możliwości w postaci nowych nagłówków dodatkowych, czyli zapewnia rozszerzalność bez konieczności przeprojektowania nagłówka podstawowego.

5.8.3. Fragmentacja w protokole IPv6

Nagłówek podstawowy nie zawiera żadnych pól związanych z fragmentacją. Jeśli datagram musi zostać podzielony, to informacja na ten temat jest umieszczana w dodatkowym nagłówku [6]. W nagłówku poprzednim należy wówczas odpowiednio zmienić wartość pola *następny nagłówek*.

Za fragmentację odpowiedzialny jest komputer wysyłający. Musi on dowiedzieć się, jakie jest *MTU ścieżki* (czyli minimalne MTU wzdłuż ścieżki od nadawcy do odbiorcy), próbkując ją. Oznacza to konieczność wysłania ciągu datagramów o różnej wielkości i sprawdzenia, czy dotarły do odbiorcy bez błędów. Następnie komputer wysyłający musi dobrać rozmiar datagramu tak, aby pasował do określonego MTU.

5.8.4. Adresowanie w protokole IPv6

128-bitowe adresy IP w wersji 6 są adresami interfejsów, a nie węzłów. Ponieważ jednak każdy interfejs należy do jednego węzła sieci, to adres typu *unicast* tego interfejsu może być używany jako identyfikator tego węzła. Podobnie jak w wersji 4, każdy adres jest podzielony na część adresującą sieć (prefiks) i część identyfikującą komputer (sufiks). Jednak granica między tymi częściami może występować w dowolnym miejscu.

IP wersja 6 definiuje trzy typy adresów [25]:

- adres jednostkowy (*unicast*) – adres identyfikujący pojedynczy interfejs, datagram wysyłany pod takim adresem jest wysyłany najkrótszą ścieżką;
- adres rozsyłania grupowego (*multicast*) – adresy tego typu identyfikują grupę interfejsów w ten sposób, że pakiet wysłany na adres multicast jest dostarczany do wszystkich interfejsów z tej grupy, członkostwo w grupie można zmieniać w dowolnym momencie; takie adresowanie pozwala obsłużyć między innymi techniki konferencyjne;
- adres grona (*anycast*) – adresy tego typu identyfikują grupę interfejsów w ten sposób, że pakiet wysłany na adres *anycast* jest dostarczany do jednego interfejsu z tej grupy. Taki sposób rozsyłania jest wykorzystywany wtedy, kiedy istnieje wiele kopii danej usługi, a pakiet jest wysyłany do tej z nich, która jest najbliższa nadawcy.

Przykładowy adres w formie binarnej wygląda następująco:

`00100001110110100000000011010011000000000000000000010111100111011`
`00000011101010100000000011111111111111110001010001001110001011010`

Można go podzielić na 16-bitowe bloki:

`0010000111011010 0000000011010011 0000000000000000 0010111100111011`
`0000001110101010 0000000011111111 1111111000101000 1001110001011010`

Każdy z takich bloków zapisywany jest szesnastkowo, a bloki są oddzielane dwukropkami (*colon hex*): 21DA:00D3:0000:2F3B:02AA:00FF:FE28:9CAA

Pewne skrócenie (kompresję) adresu osiąga się przez usunięcie zer wiodących w kolejnych grupach: 21DA:D3:0:2F3B:2AA:FF:FE28:9CAA

Dalszą kompresję uzyskuje się zastępując grupy zawierające same zera dwoma dwukropkami „::": 21DA:D3::2F3B:2AA:FF:FE28:9CAA

Jeśli w adresie występuje dłuższy ciąg zer, to kompresja jest znaczna – adres:

FE80:0:0:0:2AA:FF:FE9A:4CA2

jest zastępowany przez: FE80::2AA:FF:FE9A:4CA2

Natomiast: FF02:0:0:0:0:0:0:2

można skompresować do: FF02::2

Taka kompresja dotyczy tylko zer początkowych w grupach.

Adresu: FF02:30:0:0:0:0:0:5

nie można skompresować do: FF02:3::5 !!!

Istniejące adresy IPv4 zostały także umieszczone w nowej przestrzeni adresowej następująco: adres rozpoczynający się od 96 bitów zerowych zawiera w młodszych 32 bitach adres IPv4.

5.9. Wybrane narzędzia do rozwiązywania problemów TCP/IP w MS Windows XP

W systemie operacyjnym MS Windows XP zaimplementowano cały zestaw narzędzi do konfigurowania protokołu TCP/IP. Ich szczegółowy opis znacznie wykracza poza ramy niniejszego opracowania. Zostaną tu natomiast krótko omówione najważniejsze narzędzia (tab. 8) służące do uzyskiwania informacji o aktualnej konfiguracji oraz do wykrywania podstawowych problemów, które mogą wystąpić podczas prób przesyłania danych przez sieć. Narzędzi tych używa się jako poleceń z linii komend.

Tab. 8. Wybrane narzędzia TCP/IP systemu MS Windows XP

Nazwa	Opis polecenia
hostname	wyświetla bieżącą nazwę komputera
ipconfig	wyświetla bieżącą konfigurację TCP/IP
ping	weryfikuje poprawność konfiguracji TCP/IP oraz dostępność zdalnego komputera o podanym adresie
pathping	śledzi drogę do komputera o wskazanym adresie oraz podaje raport pakietów traconych na każdym z napotkanych routerów
tracert	podaje znalezioną drogę (adresy kolejnych routerów) do komputera o podanym adresie
netstat	podaje statystykę aktualnych połączeń internetowych (używany protokół, adres i port lokalny, adres i port zdalny oraz stan dla połączeń TCP; adres i numer portu oddzielone są dwukropkiem)

W systemach operacyjnych z rodziny Unix zaimplementowano analogiczny zestaw narzędzi o podobnych nazwach. Niektóre z podanych wyżej narzędzi systemowych korzystają z komunikatów ICMP podczas realizowania swoich zadań. Przykładem mogą być narzędzia *ping* i *tracert* [6].

Narzędzie *ping* korzysta z komunikatów *ICMP Echo* i *Echo reply*. Zgodnie z protokołem oprogramowanie sieciowe odbiorcy ma obowiązek na komunikat *Echo* odpowiedzieć komunikatem *Echo reply*. Po wydaniu przez użytkownika polecenia *ping adres*, program wysyła komunikat Echo do wskazanego adresata. Następnie program czeka krótki czas na odpowiedź. Jeśli odpowiedź nie przyszła, to wysyła następny taki sam komunikat. Jeśli na niego też nie otrzyma odpowiedzi lub otrzyma odpowiedź *Destination unreachable* (odbiorca nieosiągalny), to wypisuje informację, że żądany komputer jest nieosiągalny.

Program narzędziowy *tracert* (*traceroute* w systemach Unix) korzysta z właściwości pola *TTL* (*Czas życia* – rozdz. 5.3.2.) datagramów IP: po osiągnięciu wartości 0 w tym polu odsyłany jest komunikat ICMP *Time exceeded*. Program *tracert* wysyła ciąg datagramów i czeka na odpowiedź dotyczącą każdego z nich. W pierwszym datagramie wartość pola *TTL* jest ustawiana na 1, już pierwszy router więc porzuci ten datagram i wyśle komunikat o błędzie. Komunikaty ICMP wysyłane są w datagramach IP, program narzędziowy *tracert* może zatem sprawdzić w ten sposób adres pierwszego routera na trasie. Następnie *tracert* wysyła datagram z czasem życia równym 2, ustalając w ten sposób adres drugiego routera itd.

Tracert musi także obsłużyć przypadek, kiedy czas życia datagramu jest wystarczająco duży i komunikat dociera do odbiorcy, ale należy to ostatecznie potwierdzić. Program korzysta w tym celu z komunikatu, na który odbiorca musi wysłać odpowiedź, czyli z *Echo* i otrzymuje komunikat *Echo reply*. Pozwala to programowi *tracert* uzyskać pełną listę adresów, którą może wypisać użytkownikowi.

6. Protokoły warstwy aplikacji

6.1. Model klient-serwer

Mechanizm komunikacji w intersieci jest trochę podobny do komunikacji w sieci telefonicznej. Żeby nawiązać połączenie telefoniczne, potrzebnych jest dwóch użytkowników: jeden wybiera numer, a drugi akceptuje połączenie, podnosząc słuchawkę. W sieci komputerowej program użytkowy na jednym komputerze próbuje nawiązać łączność z programem użytkowym na drugim komputerze, a ten program na to nawiązanie połączenia odpowiada. Jednak zasadnicza różnica pomiędzy siecią telefoniczną a komputerową jest taka, że w tej drugiej nie ma odpowiednika dzwonka telefonu. Nie ma sygnału, którym oprogramowanie protokołu komunikacyjnego powiadamiałoby program użytkowy o nadejściu połączenia. W związku z tym znaleziono inne rozwiązanie. Jeden z programów uruchamiany jest wcześniej, powiadamia o tym protokoły sieciowe lokalnego komputera i czeka, aż inne programy się z nim połączą. Drugi program musi znać miejsce, w którym czeka pierwszy. Program biernie czekający na połączenie nazywamy **serwerem**, natomiast program, który aktywnie inicjuje połączenie, to **klient**. Klient musi znać lokalizację komputera oraz identyfikator programu, czyli serwera, z którym się chce połączyć. Mianem **model klient-serwer** określa się taki właśnie model współpracy między programami. W tym modelu pracują omawiane w tym rozdziale protokoły użytkowe.

Zazwyczaj oprogramowanie klienta nosi podane cechy:

- jest dowolnym programem użytkowym, który staje się klientem tymczasowo, ale wykonuje również obliczenia lokalne,
- jest wywoływane bezpośrednio przez użytkownika,
- działa lokalnie na komputerze osobistym użytkownika,
- aktywnie inicjuje kontakt z serwerem,
- może w razie potrzeby kontaktować się z wieloma serwerami, jednak na raz kontaktuje się z jednym serwerem,
- nie wymaga specjalnego sprzętu ani wyrafinowanego systemu operacyjnego.

Oprogramowanie serwera nosi zwykle następujące cechy:

- jest specjalizowanym, uprzywilejowanym programem, którego zadaniem jest świadczenie konkretnej usługi, i może obsługiwać wielu klientów,
- jest uruchamiane automatycznie przy uruchamianiu systemu i działa przez wiele kolejnych sesji,

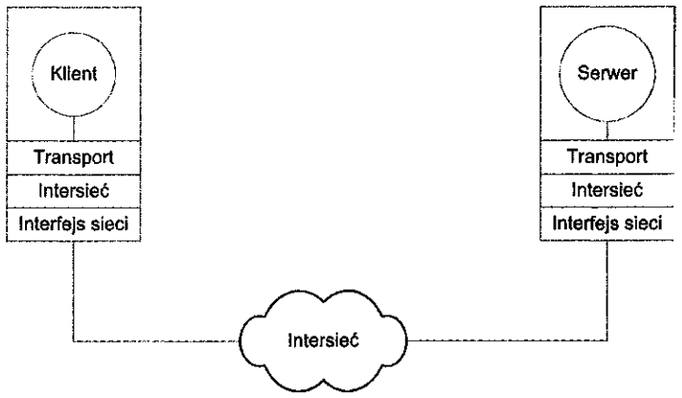
- działa na ogólnie dostępnym komputerze,
- czeka biernie na zgłoszenia od dowolnych klientów,
- przyjmuje połączenia od dowolnych odległych klientów, ale pełni jedną konkretną usługę,
- wymaga wydajnego i specjalistycznego systemu operacyjnego,
- wymaga bardziej wydajnego sprzętu (szybszy procesor lub procesory oraz więcej pamięci operacyjnej).

Informacja przesyłana pomiędzy klientem i serwerem może płynąć w jednym kierunku lub w obu jednocześnie. Zwykle klient wysyła żądanie, a serwer odsyła klientowi odpowiedź, jak w przypadku komunikacji za pomocą protokołu HTTP (rozdz. 6.8) pomiędzy przeglądarką a serwerem WWW. Czasami klient wysyła ciąg żądań, a serwer odsyła ciąg odpowiedzi. Przykładem może być połączenie ze zdalną bazą danych, kiedy w czasie jednej sesji poszukuje się wielu różnych informacji. W pewnych przypadkach serwer zapewnia stały strumień danych bez potrzeby wysyłania jakichkolwiek żądań – od momentu, kiedy klient połączy się z serwerem, ten natychmiast zaczyna do niego wysyłać informacje. Przykładem takiego działania może być serwer podający informacje o pogodzie, który po nawiązaniu połączenia nieustannie wysyła informacje o bieżącej temperaturze i ciśnieniu.

Serwery mogą zarówno odbierać, jak i wysyłać informacje. Większość serwerów udostępnia klientom zestaw plików. Na żądanie klienta serwer wysyła mu zawartość pliku o podanej nazwie. Serwer może jednak także przyjmować pliki od klientów, tzn. zapamiętywać na dysku przysłane kopie plików.

Klient i serwer potrzebują do przesyłania informacji protokołu transportowego. Zwykle jest tu wykorzystywany protokół TCP.

Rys. 47. Warstwy oprogramowania sieciowego w komunikacji klient-serwer [6]



Klient i serwer korzystają bezpośrednio z protokołu warstwy transportowej w celu zestawienia połączenia i przesyłania informacji. Z kolei protokół transportowy używa protokołów

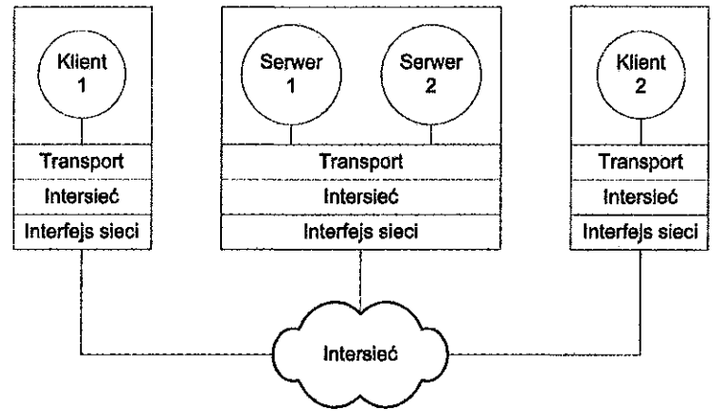
niższych warstw do wysyłania poszczególnych komunikatów. Do wykonywania zarówno programu klienta, jak i serwera komputer musi być zatem wyposażony w pełen stos protokołów sieciowych.

Odpowiednio wydajny komputer może jednocześnie wykonywać programy wielu różnych klientów i serwerów. W tym celu muszą być jednak spełnione pewne warunki:

- komputer musi być odpowiednio wyposażony sprzętowo,
- komputer musi mieć zainstalowany system operacyjny, umożliwiający jednoczesne wykonywanie wielu różnych programów.

Na takim komputerze może pracować jednocześnie wiele serwerów – po jednym dla każdej wykonywanej usługi.

Rys. 48. Świadczenie wielu usług przez jeden komputer [6]



Rysunek 48 pokazuje przykładową sytuację tego rodzaju. Programy klientów z dwóch różnych komputerów korzystają z dwóch różnych serwerów zainstalowanych na jednym komputerze.

Możliwość wykonywania wielu programów serwerów na jednym komputerze oczywiście znacznie zmniejsza koszty sprzętu i administracji, gdyż wymaga administrowania mniejszą liczbą maszyn. Zwykle też żądania nie są wysyłane do jednego serwera przez cały czas. Uruchomienie wielu serwerów na jednym komputerze pozwala wydajniej wykorzystywać jego zasoby.

Protokoły transportowe zapewniają mechanizm umożliwiający jednoznaczne wskazanie potrzebnej usługi. Polega on na przydzieleniu każdej usłudze jednoznacznego numeru, który musi być używany zarówno przez klienta, jak i przez serwer. Po uruchomieniu serwer rejestruje swój numer w protokołach sieciowych komputera, na którym działa. Gdy klient kontaktuje się z odległym serwerem, podaje numer żądanej usługi. Przykładem takiej identyfikacji są numery portów TCP.

System komputerowy umożliwiający jednoczesne wykonywanie wielu programów nazywamy systemem współbieżnym. Program używający więcej niż jednego wątku sterowania nazywamy programem współbieżnym. Współbieżność jest zasadniczą częścią modelu klient-serwer, gdyż współbieżny serwer oferuje swoją usługę wielu klientom jednocześnie, nie wymagając, aby klient zaczekał na zakończenie obsługi poprzedniego klienta, ponieważ taka obsługa może trwać dość długo. Dla każdego nowego klienta serwer tworzy nowy wątek sterowania. Taki serwer składa się zwykle z dwóch części: jedna jest odpowiedzialna za przyjęcie żądania i utworzenie nowego wątku, a druga zawiera kod obsługi pojedynczego żądania. Uruchomienie serwera uruchamia jedynie jego pierwszą część. Główny wątek czeka na żądania. Po odebraniu żądania główny wątek tworzy nowy wątek, który je obsługuje. Po obsłużeniu jednego żądania wątek ten kończy działanie. W tym czasie główny wątek czeka na następne żądania, a kiedy nadejdą, tworzy nowe wątki do ich obsługi. Wobec tego, jeśli dany serwer obsługuje w danym momencie *N* klientów, to jednocześnie działa *N+1* jego wątków.

Jeśli na komputerze działa wiele kopii danego serwera, to musi istnieć jakaś metoda, żeby żądania jednego klienta były kierowane zawsze do tej samej kopii serwera. Przykładem takiego rozwiązania jest przydzielanie portów TCP. Serwer oczekuje na żądania klientów pod konkretnym portem TCP, czyli wszystkie kopie tego serwera używają tego samego numeru portu TCP. Każdy klient, odwołujący się do tego serwera, podaje taki sam numer portu. Wobec tego należy jakoś rozróżniać tych klientów. Rozwiązaniem tego problemu jest podawanie przez klienta takiego numeru portu, który jednoznacznie go identyfikuje pośród wszystkich korzystających z danego serwera. Klient wybiera sobie taki numer portu, który nie odpowiada żadnej usłudze na jego komputerze, i zawsze podaje ten numer portu jako port nadawcy. Na komputerze serwera kopia serwera przydzielona danemu klientowi jest więc identyfikowana dwoma numerami portów: numerem portu TCP, pod którym działa serwer, i numerem portu identyfikującym klienta.

Na zakończenie można jeszcze dodać, że serwery nie zawsze muszą korzystać z transportu TCP. Mogą także korzystać z transportu bezpołączeniowego UDP albo z obu metod jednocześnie, pozostawiając wybór klientowi. Jeśli serwer jest tak przygotowany, że może korzystać z wielu metod transportu, to zawsze odpowiada na żądanie klienta za pomocą tego samego protokołu, za pośrednictwem którego to żądanie otrzymał.

6.2. Interfejs gniazd

Programy klienta i serwera używają protokołów transportowych do komunikacji. Program użytkowy musi podać oprogramowaniu protokołu czy jest klientem, czy serwerem, tzn. czy będzie komunikację inicjował aktywnie, czy będzie czekał biernie na połączenia. Następnie

musi określić dalsze szczegóły komunikacji, np. adresy i porty protokołów. Nadawca musi podać dane do wysłania, a odbiorca – określić, gdzie powinny być umieszczane informacje przychodzące. Wszystkie te informacje są podawane do oprogramowania warstwy transportowej za pomocą interfejsu programu użytkowego **API** (*Application Program Interface*) [6]. Interfejs API określa zestaw operacji, które program użytkowy może wykonywać w ramach interakcji z oprogramowaniem protokołów. Standardy protokołów komunikacyjnych nie określają zwykle konkretnego interfejsu API, tylko ogólne operacje, które powinny być udostępnione programom użytkowym. Szczegóły implementacyjne zależą od konkretnego systemu operacyjnego. Większość systemów operacyjnych udostępnia obecnie tzw. **interfejs gniazd**. Został on zaprojektowany w University of California w Berkeley dla systemu Unix BSD. Z czasem stał się podstawowym standardem tego rodzaju interfejsów. Nazwa interfejsu pochodzi od nazwy **gniazdo** (*socket*) przyjętej dla określania punktu komunikacyjnego.

Podstawowe procedury interfejsu gniazd przedstawia tabela 9.

Tab. 9. Podstawowe procedury interfejsu gniazd [15]

Procedura	Przeznaczenie
socket	Utworzenie nowego punktu komunikacyjnego (gniazda).
bind	Przypisanie gniazdu lokalnego adresu sieciowego.
listen	Sygnalizacja gotowości do przyjmowania połączeń; parametr wywołania określa maksymalny rozmiar kolejki połączeń.
accept	Wprowadzenie w stan oczekiwania na nadchodzące żądania połączeń.
connect	Aktywna próba nawiązania połączenia.
send	Wysłanie danych w ramach połączenia.
receive	Odebranie danych w ramach połączenia.
close	Zwolnienie połączenia.

Cztery pierwsze z wymienionych w tabeli procedur muszą zostać wykonane przez serwer w podanej kolejności. Procedura *socket* powoduje utworzenie nowego punktu połączeniowego, czyli gniazda i daje w wyniku uchwyt (*handle*) – jego identyfikator, nadany mu przez system operacyjny. Jej parametry określają rodzinę protokołów (np. TCP/IP), rodzaj komunikacji (np. komunikacja strumieniowa, czy bezpołączeniowa) oraz konkretny protokół, który będzie stosowany w danym gnieździe (np. TCP).

Utworzonemu gniazdu należy nadać adres sieciowy korzystając z procedury *bind*. Argumenty tej procedury zawierają m.in. otrzymany poprzednio uchwyt gniazda oraz numer portu, pod którym serwer oczekuje na połączenia. Od tego momentu zdalny klient może już połączyć się z serwerem. Procedura *listen* służy do przydzielenia pamięci niezbędnej do kolejkowania

nadchodzących połączeń. Liczba połączeń, jakie mogą być jednocześnie przyjęte do obsługi, ograniczona jest maksymalnym rozmiarem kolejki, podawanym jako parametr procedury *listen*. Jeśli kolejka jest pełna w chwili nadejścia nowego połączenia, system je odrzuca. Wykonanie procedury *accept* powoduje przejście serwera w stan oczekiwania na przyjście połączenia. Jeśli nie przybyły żadne zgłoszenia, to system zawiesza program serwera do chwili, gdy jakiś klient utworzy połączenie. Jeśli zgłoszenie znajduje się w kolejce, to następuje natychmiastowy powrót z procedury *accept*. Podczas wykonywania tej procedury oprogramowanie protokołu tworzy nowe gniazdo dla tego połączenia i zwraca jego uchwyt. Serwer może wówczas utworzyć nowy proces (lub wątek) realizujący połączenie i powrócić do stanu oczekiwania na kolejne żądania, kierowane do gniazda oryginalnego.

W przypadku programu klienta sekwencja wywoływania procedur jest nieco inna. Podobnie jak serwer, klient musi utworzyć gniazdo za pomocą procedury *socket*, ale nie jest konieczne przypisywanie mu adresu, nie korzysta więc z procedury *bind*. Aktywna próba nawiązania połączenia z serwerem dokonywana jest za pomocą procedury *connect*. Powoduje ona wysłanie odpowiedniego komunikatu (adres serwera jest parametrem procedury) i wejście w stan oczekiwania na odpowiedź. Po nadejściu odpowiedzi program klienta zostaje odblokowany i połączenie staje się ostatecznie nawiązane. Obydwie strony mogą teraz korzystać z procedur *send* i *receive* w celu wysyłania i przyjmowania danych.

Zamknięcie połączenia następuje w wyniku wykonania procedury *close* dla wykorzystywanego gniazda. Powinny ją wykonać zarówno klient, jak i serwer.

Funkcjonalność interfejsu gniazd jest znacznie szersza [6], niż opisano wyżej. Można za pomocą odpowiednich procedur uzyskiwać np. pełne nazwy komputera serwera czy klienta, a także dokonywać odwzorowania nazwy komputera w jego adres IP i odwrotnie. Gniazdo ma wiele parametrów i opcji, interfejs zapewnia więc także procedury do ustawiania wartości dla nich.

Szczegółowo opisane przykłady zarówno programu klienta, jak i serwera z wykorzystaniem interfejsu gniazd znajdują się w [6] w rozdz. 28 oraz w [15] w rozdz. 6.1.4.

6.3. System DNS

System nazw domen **DNS** (*Domain Name System*) służy do zarządzania adresami symbolicznymi w Internecie [3]. Internet musi wewnętrznie posługiwać się adresami w formie liczbowej, bo działa w oparciu o protokół IP, który takich adresów wymaga. Z drugiej strony jednak użytkownicy Internetu powinni mieć możliwość korzystania z adresów symbolicznych, łatwych do zapamiętania dla człowieka, czyli adresów w postaci *www.wsei.pl*. Aby pogodzić te wymagania powstał właśnie DNS. Służy on do uzyskania wzajemnego odwzorowania między adresami liczbowymi a symbolicznymi. Dzięki DNS, mając adres symboliczny można uzyskać adres liczbowy i odwrotnie. Ponadto DNS, obok obu adresów, udostępnia też w Internecie inne informacje o serwerach i ich grupach (zwanych domenami), wykorzystywane przez różne protokoły sieciowe.

6.3.1. Powstanie systemu DNS

W początkach swojego rozwoju, kiedy Internet składał się z kilkuset zaledwie serwerów, do zarządzania odwzorowaniem pomiędzy adresami symbolicznymi a liczbowymi używano pojedynczego pliku tekstowego HOSTS.TXT, zawierającego w parach adresy liczbowe i odpowiadające im adresy symboliczne [3]. Plik ten był utrzymywany przez centrum informacji sieciowej NIC (*Network Information Center*) i udostępniany wszystkim serwerom. Administratorzy Internetu najczęściej przesyłali do NIC swoje zmiany pocztą elektroniczną (kiedy, na przykład, chcieli dodać nowe serwery). NIC wcielał ich zmiany do kopii-matki pliku HOSTS.TXT raz albo dwa razy w tygodniu i publikował aktualną wersję na serwerze FTP (*File Transfer Protocol*). Następnie administratorzy Internetu pobierali tę wersję do podległych im serwerów.

Ten sposób aktualizacji okazał się zupełnie niewystarczający, kiedy lawinowo zaczęła rosnąć liczba serwerów w Internecie. Po pierwsze, plik HOSTS.TXT rósł proporcjonalnie do liczby serwerów i zapanowanie nad nim stawało się coraz bardziej utrudnione. Przy każdej jego aktualizacji trzeba było sprawdzać, czy dwóch administratorów nie wybrało dla swoich serwerów identycznych adresów. Przeszukiwanie dużego pliku w procesie odwzorowywania adresów było też czasochłonne. Jego częste rozsyłanie po całym Internecie zapychało łącza. Wreszcie rozpatrzenie setek próśb o modyfikacje pliku HOSTS.TXT stawało się także coraz trudniejsze wraz z ciągłym powiększaniem się Internetu. W pewnym momencie stało się jasne, że do rozwiązania tych problemów trzeba opracować nowy system.

Uznano, że system ten powinien pozwalać na podzielenie odpowiedzialności za odwzorowywanie adresów pomiędzy wiele ośrodków. Z drugiej strony powinien umożliwiać też szybkie znalezienie ośrodka odpowiedzialnego za odwzorowanie danego adresu. Wreszcie, jak cały Internet, powinien móc przetrzymać większość awarii dzięki zastosowaniu zasobów zapasowych, a w ostateczności przynajmniej ograniczać jej skutki.

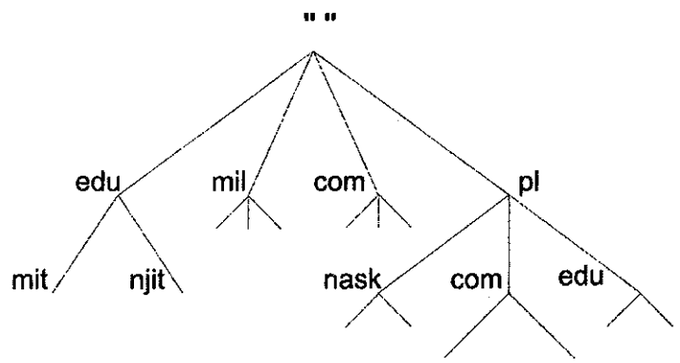
Paul Mockapetris, pracujący wtedy w Instytucie Informatyki Uniwersytetu Południowej Karoliny, był odpowiedzialny za opracowanie architektury nowego systemu. W 1984 roku opublikował on dokumenty RFC 882 i 883, opisujące DNS. Publikacje te zostały później zastąpione przez RFC 1034 i 1035, które stanowią aktualną specyfikację DNS-u. Dokumenty te uzupełniono o dodatkowe publikacje RFC, które dotyczą kwestii związanych z DNS.

6.3.2. Domeny DNS

DNS jest **rozproszoną bazą danych**, to znaczy, że żaden serwer nie posiada wszystkich danych. Nie byłoby to zresztą możliwe. DNS pozwala na lokalne zarządzanie poszczególnymi fragmentami tej bazy danych. Dane każdego fragmentu są jednak dostępne dla każdej maszyny w Internecie. Odporność na awarie i odpowiednia wydajność osiągane są przez powielanie i buforowanie danych.

Baza danych DNS ma **strukturę drzewiastą**. Jest ona podobna do struktury katalogów (folderów) na dysku (rys. 42).

Rys. 49. Drzewo domen DNS [3]



Korzeń drzewa zwyczajowo bywa rysowany u góry. Każdy węzeł drzewa jest oznaczony etykietą. Etykieta pusta " " zarezerwowana jest dla korzenia, czyli węzła głównego. Pozostałe węzły mogą mieć dowolne inne etykiety, jednak podwęzły umieszczone bezpośrednio pod dowolnym węzłem muszą mieć różne etykiety. Jest to podobne do systemu plików, gdzie korzeń drzewa ma specyficzne oznaczenie zależne od systemu operacyjnego (np. / w systemach uniksowych, czy *C:* w systemach firmy Microsoft), a w żadnym katalogu nie może być dwóch podkatalogów o tej samej nazwie.

W systemie plików każdy katalog ma swoją unikalną pełną nazwę, która składa się z oznaczenia korzenia i oddzielonych odpowiednimi separatorami nazw katalogów nadrzędnych. Podobnie w DNS każdy węzeł drzewa ma unikalną nazwę złożoną z oddzielonych kropkami nazw węzłów. Tak jak na dysku może być katalog o nazwie *C:\Users\Jola\Dokumenty\Streszczenia*, tak w DNS węzeł drzewa, znajdujący się w lewym dolnym rogu powyższego rysunku, jest oznaczany jako *mit.edu*. Różnica jest taka, że o ile przy tworzeniu pełnej nazwy katalogu wymieniamy nazwy węzłów drzewa począwszy od korzenia, to w przypadku DNS kierunek jest odwrotny: najpierw umieszczamy etykietę danego węzła, a potem etykiety jego kolejnych „rodziców" (węzłów nadrzędnych).

W DNS każdy węzeł drzewa nazywamy **domeną**. W systemie plików składnikami katalogu są jego wszystkie podkatalogi wraz ze swoimi podkatalogami itd. W DNS domena to węzeł drzewa wraz z poddrzewem, czyli węzłami potomnymi na wszystkich niższych poziomach.

DNS jest bazą danych indeksowaną nazwami domen. Oznacza to, że nazwa domeny jest w tej bazie kluczem, według którego można znaleźć różne informacje o domenie. Są one skojarzone z nazwą domeny i zawarte w rekordach zasobów. Podobnie jak katalog zawiera podkatalogi i pliki, tak domena zawiera subdomeny i serwery. Każdy serwer w sieci należy do domeny, która zawiera informacje o nim. Informacja ta może zawierać adres IP (czyli liczbowy), wskazówki na temat przekazywania poczty itd.

Ta z pozoru dość skomplikowana struktura została przyjęta, aby rozwiązać problemy, które stwarzał plik HOSTS.TXT. Hierarchiczne nazwy domen eliminują problem kolizji nazw. Każda domena ma unikalną nazwę, prowadząca ją organizacja może więc nadawać dowolne nazwy znajdującym się w niej subdomenom i serwerom. Nazwa domeny *delfin.kowalski.pl* nie koliduje w żaden sposób z nazwą *delfin.malinowski.pl*. Podobnie katalog *C:\Users\Kowalski\Docs* nie koliduje z katalogiem *C:\Users\Malinowski\Docs*.

Domena jest poddrzewem przestrzeni nazw domen. Nazwa domeny jest taka sama, jak nazwa węzła na szczycie domeny. Nazwy domen przy liściach drzewa reprezentują pojedyncze serwery i mogą wskazywać (w DNS) na adresy sieciowe, informację sprzętową i o przekazywaniu poczty. Nazwy domen we wnętrzu drzewa mogą (ale nie muszą) być nazwami serwerów, a ponadto wskazywać na informację o domenie.

6.3.3. Domeny najwyższego poziomu

Pierwotnie przestrzeń nazw domen Internetu dzieliła się na siedem domen najwyższego poziomu TLD (*top level domains*), utworzonych w 1980 r. [3]:

com – organizacje komercyjne, takie jak np. IBM (*ibm.com*), Sun Microsystems (*sun.com*), czy Inter-Design (*interdesign.com.pl*)

edu – organizacje oświatowe, takie jak Uniwersytet Purdue (*purdue.edu*), czy Uniwersytet Warszawski (*uw.edu.pl*)

gov – organizacje rządowe, takie jak NASA (*nasa.gov*) czy National Science Foundation (*nsf.gov*)

mil – organizacje wojskowe, takie jak Armia Amerykańska (*army.mil*) czy Amerykańska Marynarka Wojenna (*navy.mil*)

net – organizacje tworzące sieci, takie jak NSFNET (*nsf.net*)

org – organizacje niekomercyjne, takie jak Electronic Frontier Foundation (*eff.org*), czy Polska Akcja Humanitarna (*pah.org.pl*)

int – organizacje międzynarodowe, takie jak NATO (*nato.int*)

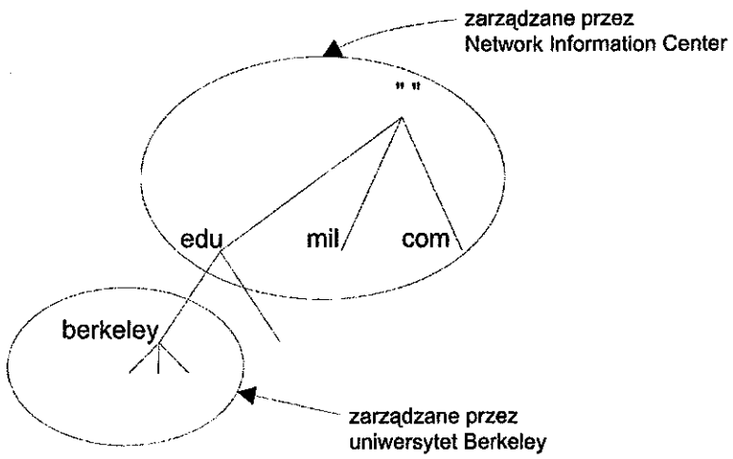
Odkąd jednak Internet stał się międzynarodowy, dodano po jednej domenie najwyższego poziomu dla każdego kraju. Dla domen związanych z krajem innym niż USA domena najwyższego poziomu określa ten kraj (np.: *.pl* dla Polski, czy *.de* dla Niemiec), a dopiero następny poziom odpowiada wyżej przedstawionemu podziałowi na siedem głównych domen.

Następnie w listopadzie 2000 r. zostały dodane kolejne domeny: *biz, info, name, pro, aero, coop, museum* [28].

6.3.4. Delegacja

Jednym z głównych celów projektu systemu zarządzania nazwami domen była jego decentralizacja [3]. Została ona osiągnięta za pomocą **delegacji**. Delegowana domena działa podobnie jak zadania delegowane (przydzielane) w pracy. Dyrektor może podzielić wielki projekt na mniejsze zadania i delegować odpowiedzialność za każde z nich do różnych pracowników.

Rys. 50. Przykład delegacji subdomeny *berkeley* w domenie *edu* [3]



Podobnie organizacja zarządzająca domeną może podzielić ją na subdomeny, a każda z nich może być delegowana do innej organizacji, która staje się odpowiedzialna za utrzymywanie wszystkich danych w tej subdomenie. Oznacza to, że taka organizacja może sama przydzielać i odbierać nazwy serwerów w ramach zarządzanej przez siebie domeny. Może też wydzielić ze swojej domeny subdomeny i zarządzanie nimi przekazać kolejnym organizacjom. W ten sposób ciężar zarządzania nazwami domen w Internecie jest podzielony między wiele organizacji. Na przykład domena *edu* jest zarządzana przez organizację InterNIC. InterNIC przekazał zarządzanie subdomeną *berkeley.edu* uniwersytetowi w Berkeley (rys. 50).

Przekazywanie odpowiedzialności, czyli delegacja uprawnień, może iść dalej. W niniejszym przykładzie uniwersytet przekazuje poszczególnym wydziałom zarządzanie ich domenami, np. Wydział Informatyki zarządza domeną *cs.berkeley.edu*.

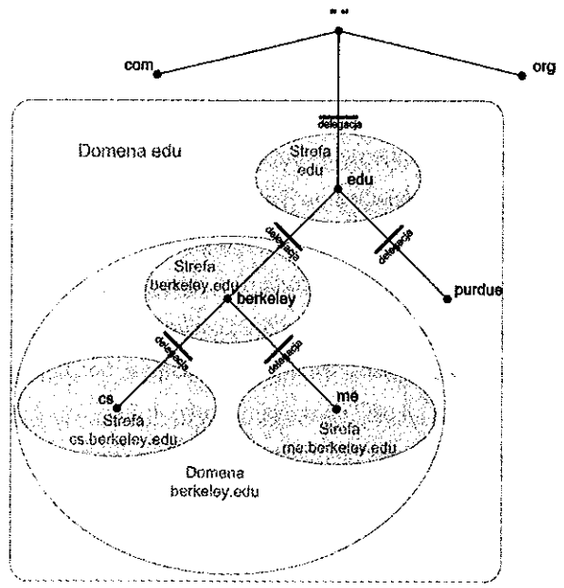
Nie wszystkie organizacje delegują całą domenę, podobnie jak nie wszyscy przełożeni delegują całą pracę. Domena może mieć kilka subdomen i zawierać również serwer, który do żadnej z nich nie należy.

6.3.5. Serwery nazw i strefy

Programy, które zapamiętują informacje o przestrzeni nazw domeny, nazywane są **serwerami nazw**. Zasadniczo serwery nazw dysponują całkowitą informacją o pewnej części przestrzeni nazw domeny, nazywanej **strefą**. Informacje te pobierają one z pliku lub z innego serwera nazw – mają wówczas pozwolenie (*authority*) dla tej strefy. Serwery nazw mogą mieć pozwolenia dla wielu stref jednocześnie.

Różnica między strefą a domeną jest taka, że domena obejmuje wszystkie swoje subdomeny, natomiast strefa to informacje o delegacjach danej domeny plus ta część domeny, która nie została delegowana gdzie indziej (rys. 51). W przypadku gdy wszystkie subdomeny danej domeny zostały delegowane, strefa obejmuje informacje o delegacjach oraz o swoich serwerach nie należących do subdomen. Jeśli subdomena domeny nie została nikomu delegowana, to strefa zawiera nazwy domen i dane subdomeny. Serwery nazw zarządzają strefami, a nie domenami.

Rys. 51. Podział przestrzeni domen na strefy nazw [3]



Istnieją dwa rodzaje serwerów nazw: serwery podstawowe (*primary masters*) i serwery drugorzędne (*secondary masters*). Podstawowy serwer nazw czyta dane na temat strefy z pliku na komputerze, na którym jest uruchomiony. Serwer drugorzędny pobiera dane strefy od innego serwera (podstawowego lub drugorzędnego), który ma uprawnienia do danej strefy. Robi to okresowo. Dzięki temu serwer podstawowy nie musi rozsyłać danych strefy. Pobieraniem danych strefy zajmują się serwery drugorzędne. Są one ważne, ponieważ ich zastosowanie pozwala na pracę DNS w sytuacji, kiedy połączenie się z serwerem podstawowym jest, z takich czy innych powodów, niemożliwe. Ich zastosowanie pozwala także rozłożyć obciążenie zapytaniami o dane strefy na kilka serwerów, co jest szczególnie ważne w przypadku stref związanych z domenami, które mają wiele subdomen (np. *com*). Kolejny zysk wynika z tego, że połączenie się z serwerem drugorzędnym może być o wiele szybsze. Na przykład szybciej i taniej można się skontaktować z serwerem drugorzędnym zlokalizowanym w Szwecji niż z serwerem podstawowym w USA, kiedy pytamy o dane strefy *com* przy połączeniu z serwerem *tessel.com*.

6.3.6. Rozwiązywanie nazw

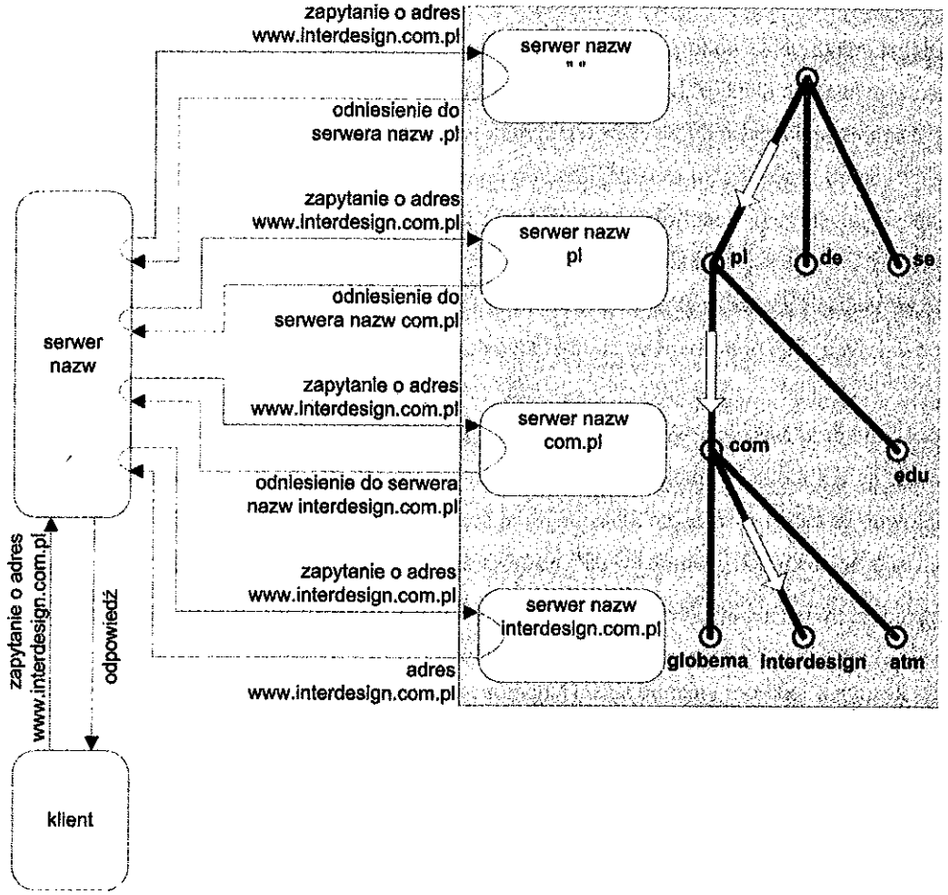
Resolwer to klient, który ma dostęp do serwerów nazw. Jest on pośrednikiem między aplikacją (np. przeglądarką stron WWW, czy raczej stosem TCP/IP w systemie operacyjnym), a serwerami nazw. Resolwer obsługuje:

- zapytania skierowane do serwera nazw,
- interpretację odpowiedzi (może ona być rekordem danych lub błędem),
- przekazanie informacji do programów, które o nią prosiły.

Serwery nazw służą do udostępniania danych z przestrzeni nazw domen. Nie tylko przekazują dane o strefach, dla których mają uprawnienia. Mogą też przekazywać dane o strefach, dla których nie mają uprawnień. Ten proces nazywa się **rozwiązywaniem nazw**.

Główny serwer nazw to taki, który ma uprawnienia dla strefy oznaczanej pustym łańcuchem znaków (" "), odpowiadającej korzeniowi (węzłowi głównemu) drzewa przestrzeni nazw. Główny serwer nazw wie, gdzie znajdują się wiarygodne serwery nazw dla każdej z domen najwyższego poziomu. Wie, gdzie się one znajdują – to znaczy zna ich adresy IP (liczbowe). Pytane o dane jakiejś domeny, główne serwery nazw mogą dostarczyć co najmniej nazwy i adresy IP serwerów nazw wiarygodnych dla domeny najwyższego poziomu, zawierającej daną domenę. Z kolei jeden z takich serwerów udzieli bardziej szczegółowych danych albo odeśle do serwera nazw dla kolejnego poziomu drzewa domen.

Rys. 52. Zamiana adresu symbolicznego na adres IP [3]



Rozwiązywanie nazw odbywa się w ten sposób, że klient (np. stacja robocza) zwraca się do swojego (o adresie IP zapisanym w swoich parametrach konfiguracyjnych) serwera nazw na przykład o adres IP serwera *www.interdesign.com.pl* (rys. 52). Zapytany przez klienta, serwer nazw musi udzielić mu odpowiedzi, dlatego po kolei:

- pyta główny serwer nazw o adres IP serwera *www.interdesign.com.pl*
- otrzymuje od niego adres IP serwera nazw domeny *pl*
- pyta serwer nazw domeny *pl* o adres IP serwera *www.interdesign.com.pl*
- otrzymuje od niego adres IP serwera nazw domeny *com.pl*
- pyta serwer nazw domeny *com.pl* o adres IP serwera *www.interdesign.com.pl*
- otrzymuje od niego adres IP serwera nazw domeny *interdesign.com.pl*
- pyta serwer nazw domeny *interdesign.com.pl* o adres IP serwera *www.interdesign.com.pl*
- otrzymuje od niego adres IP serwera *www.interdesign.com.pl*

Po przeprowadzeniu takiego wypytywania, serwer nazw, zapytany przez klienta, może mu wreszcie udzielić odpowiedzi. W praktyce, dzięki temu, że każdy serwer nazw przechowuje przez pewien czas odpowiedzi na pytania, które uzyskał od innych serwerów nazw, może okazać się, że odpowiedź na pytanie da się otrzymać bez wypytywania o nią tylu serwerów nazw. Na przykład jeśli klient zapyta najpierw swój serwer nazw o adres IP serwera *www.interdesign.com.pl*, a niedługo potem o adres serwera *www.globema.com.pl*, to serwer nazw skorzysta z zapamiętanego adresu serwera nazw domeny *com.pl*, co oszczędzi mu dwóch zapytań.

6.4. Protokół DHCP

Protokół DHCP [32] jest standardem opracowanym przez organizację IETF, przeznaczonym do dynamicznego przydzielania adresów IP wielu stacjom roboczym, znajdującym się w jednej sieci. Protokół DHCP zapewnia bezpieczną, niezawodną i łatwą konfigurację sieci TCP/IP, zapobiega występowaniu konfliktów adresów i ułatwia optymalne wykorzystanie adresów IP klientów w sieci.

Protokół DHCP wykorzystuje model typu klient-serwer, w którym serwer DHCP zajmuje się centralnym zarządzaniem adresami używanymi w sieci. Przechowuje on bazę dostępnych adresów IP, może też udostępniać klientom dodatkowe informacje konfiguracyjne, jak adresy serwerów DNS, adresy bram, maskę podsieci i in. Klienci obsługujący protokół DHCP mogą żądać i uzyskać dzierżawiony adres IP. Adres jest przydzielany klientowi na określony w serwerze czas, zwany **dzierżawą**. Dzierżawa może być okresowo odnawiana przez klienta. Jeśli tak się nie stanie, to jego adres IP jest zwracany do bazy danych, dzięki czemu staje się dostępny dla innych klientów DHCP.

Proces przydzielania adresów IP w systemie MS Windows 2000 za pomocą serwera DHCP ilustruje schemat na rys. 53.

1. Podczas pierwszego połączenia się z siecią klient żąda przydzielenia adresu IP, rozgłaszając w lokalnej sieci komunikat *DHCPDiscover*.

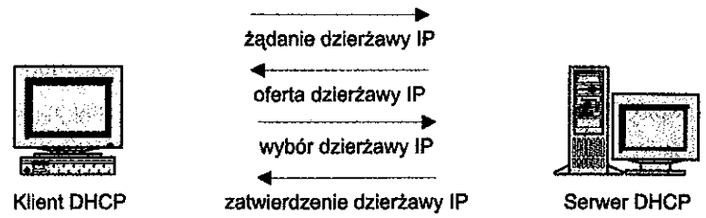
2. Serwer DHCP oferuje klientowi adres, odpowiadając mu komunikatem zawierającym adres IP i inne informacje konfiguracyjne (komunikat *DHCPOffer*).

3. Klient informuje serwer o zaakceptowaniu dzierżawy wybierając oferowany adres i odpowiadając serwerowi za pomocą komunikatu *DHCPRequest*.

4. Adres jest przydzielony, a serwer odsyła komunikat potwierdzający *DHCPAck*, zezwalając na dzierżawę. W komunikacie tym mogą być zawarte opcjonalnie takie informacje, jak brama domyślna lub adres serwera DNS.

5. Po odebraniu potwierdzenia klient konfiguruje swoje parametry TCP/IP, a następnie kończy inicjalizację protokołu TCP/IP.

Rys. 53. Działanie usługi DHCP [32]



6.5. Protokół FTP

FTP (*File Transport Protocol*) jest to prosty protokół, wchodzący w skład rodziny protokołów TCP/IP, służący do przesyłania plików dowolnego typu pomiędzy komputerami, tzn. do pobierania plików z odległej maszyny do lokalnej i odwrotnie. Protokół ten działa w architekturze klient-serwer i jest najpopularniejszą metodą kopiowania plików przez Internet. Dzięki niemu nie istnieje konieczność uzyskiwania pełnego dostępu do zasobów zdalnego komputera. Protokół FTP jest znacznie starszy niż stosowany w sieci WWW protokół HTTP. Jego początki sięgają 1971 roku. FTP, poczta elektroniczna i Telnet są to trzy podstawowe usługi zaprojektowane dla sieci ARPANet, poprzedniczki Internetu.

Komputer, który udostępnia użytkownikom pliki przez sieć, musi mieć zainstalowane oprogramowanie, zdolne rozpoznawać i realizować tego typu żądania, nadchodzące z Internetu. Maszyna spełniająca te warunki jest nazywana **serwerem FTP**.

Użytkownik łączy się z serwerem FTP za pomocą specjalnego programu, zwanego **klientem FTP**. Klient FTP, dostarczany wraz z systemem Windows, jest prostym programem, działającym w trybie tekstowym, uruchamianym z linii komend: *ftp.exe*. Listę poleceń tego programu można uzyskać poleceniem *help*. Istnieje też wiele płatnych i bezpłatnych wersji graficznych klientów FTP.

Sesja FTP składa się z trzech części. Pierwsza z nich to zalogowanie się na odległy komputer przez podanie nazwy użytkownika i hasła. Na ich podstawie serwer FTP identyfikuje użytkownika i przydziela mu na czas sesji określone prawo dostępu do swoich zasobów. Posiadanie konta i związanych z nim odpowiednich uprawnień na komputerze zdalnym jest warunkiem koniecznym do rozpoczęcia transmisji danych. Konto takie jest przydzielane przez administratora systemu. Na serwerach FTP z dostępem publicznym (tzw. anonimowym FTP) identyfikatorem użytkownika jest zazwyczaj nazwa *anonymous*, a hasłem – własny adres poczty elektronicznej użytkownika.

Druga część sesji to „właściwa" praca, podczas której wykonuje się operacje na odległych plikach i katalogach: kopiowanie, przenoszenie, zmiana nazwy itp. Etap ostatni to zamknięcie sesji poleceniem *close*. Niektóre serwery wysyłają na zakończenie statystykę, w której zawarte są takie informacje jak czas trwania sesji i ilość przesłanych danych.

Protokół FTP zapewnia dwa tryby transmisji danych: tryb binarny i tryb ASCII (*American Standard Code for Information Interchange*). Pierwszy z nich przesyła plik „tak, jak jest", bez dokonywania w nim żadnych zmian. Drugi natomiast przeznaczony jest do transmitowania zbiorów tekstowych i modyfikuje sposób kodowania niektórych bajtów. Plik binarny (program, grafika, skompresowane archiwum), przesłany w trybie ASCII, może zostać trwale uszkodzony, natomiast zbiory tekstowe można przesyłać również w trybie binarnym.

Każdy serwer FTP posiada swój własny, unikalny adres, pozwalający jednoznacznie go zidentyfikować. Nazwa domenowa często (choć nie jest to wymagane) rozpoczyna się od członu ftp, po którym następuje domena instytucji bądź firmy, utrzymującej dany serwer, np.:

```
ftp.wsei.pl
```

Pełny adres internetowy rozpoczyna się identyfikatorem protokołu i wygląda następująco:

```
ftp://ftp.wsei.pl
```

Dalsze człony adresu (ścieżka dostępu do katalogu lub pliku) są już budowane podobnie do adresów w sieci WWW:

```
ftp://ftp.wsei.pl/pub/wyklady/siecikomputerowe/ftp.txt
```

Powyższe adresy wskazują na serwer anonimowy. Jeżeli dostęp do serwera wymaga podania nazwy użytkownika i hasła, należy dołączyć te dane do adresu następująco:

```
ftp://uzytkownik:haslo@ftp.wsei.pl
```

Serwer FTP, na którym zgromadzono większą ilość plików udostępnianych określonej grupie użytkowników, jest określany mianem archiwum FTP. Publiczne (anonimowe) archiwa słyną przede wszystkim z bogatych zbiorów oprogramowania i zwykle są bardzo starannie prowadzone.

6.6. Protokół Telnet

Telnet jest jedną z najbardziej elementarnych i podstawowych usług, jakie mogą być realizowane w sieciach opartych na protokole TCP/IP [39]. Technicznie jest to najprostsza z usług sieciowych, służąca do nawiązania interaktywnego połączenia terminalowego ze wskazanym komputerem w sieci, a dokładniej z zainstalowanym tam serwerem Telnetu. Po ustanowieniu takiego połączenia znaki wpisywane na klawiaturze komputera użytkownika są przesyłane przez sieć do maszyny, z którą nawiązano połączenie. Przesyłane w odwrotną stronę odpowiedzi są wyświetlane na ekranie użytkownika. Nawiązanie połączenia przez Telnet zwykle uruchamia standardową procedurę identyfikacji użytkownika (logowanie), a po jej pomyślnym zakończeniu rozpoczyna się normalna praca w systemie operacyjnym zdalnego komputera, tak samo jak w przypadku bezpośredniego połączenia terminalowego z tym komputerem. Telnet pozwala więc na zdalną pracę na dowolnym komputerze w sieci, na którym użytkownik

ma udostępnione konto. Komputer ten musi pracować pod systemem operacyjnym pozwalającym na wielodostępną pracę.

Podstawowym zastosowaniem Telnetu jest korzystanie z własnego konta na innym komputerze. Przez Telnet również udostępniano w sieci szereg usług informacyjnych, przede wszystkim rozmaite bazy danych: katalogi biblioteczne, bazy danych o lotach sond kosmicznych, trzęsieniach ziemi, notowaniach giełdowych, wynikach sportowych itd. Część z tych baz była ogólnie dostępna, a część odpłatna – logowanie było możliwe po wykupieniu subskrypcji, z którą przydzielano konto użytkownika i hasło. Wraz z rozwojem sieci WWW zainteresowanie bazami telnetowymi wygasło.

Program klienta Telnetu, uruchamiany w celu nawiązania i obsługi zdalnego połączenia, nosi zwykle nazwę *telnet*, a jego uruchomienie wygląda następująco:

telnet *adres*

gdzie *adres* jest symbolicznym (domenowym) lub numerycznym adresem zdalnego komputera. Po wydaniu takiej komendy pojawi się kilka komunikatów informujących o przebiegu procesu nawiązywania połączenia i jeśli komputer jest dostępny, wyświetlona zostanie zachęta do rozpoczęcia pracy.

Pierwszy wiersz zawiera informację o systemie operacyjnym zdalnego komputera, nazwę komputera (często jest to nazwa taka sama jak pierwsza część adresu domenowego) oraz opcjonalnie numer *wirtualnego terminala*, przypisanego do właśnie rozpoczętej sesji. W drugim wierszu komputer zwraca się o podanie nazwy konta użytkownika, a w następnym o hasło, które ostatecznie potwierdza prawo użytkownika do pracy na tej maszynie. Po pozytywnej weryfikacji tych danych można rozpocząć normalną pracę.

Telnet jest usługą przeznaczoną przede wszystkim dla osób posiadających konta na komputerze, z którym się łączą. Nieliczne komputery posiadają konto „gościnne", anonimowe, na które można się logować bez podania hasła, ale są to przypadki coraz rzadsze.

Znacznie częściej można spotkać sytuację, kiedy tworzone jest konto ogólnodostępne, ale przeznaczone do wykonywania jednego konkretnego zadania. Można się na nie zalogować bez hasła (lub hasło jest ogólnie znane), ale potem nie uzyskuje się dostępu do linii poleceń systemu operacyjnego. Uruchamiany jest natomiast z góry określony program i tylko z nim można pracować. Zakończenie pracy z tym programem powoduje automatycznie zakończenie sesji.

Jeśli jakaś usługa, do której można się podłączyć za pomocą Telnetu, jest udostępniana pod innym numerem portu niż 23 (domyślny numer portu Telnetu), to trzeba ten numer podać jawnie razem z adresem komputera:

telnet *adres port*

6.7. Protokół Secure Shell

Wykorzystanie protokołu Telnet zostało bardzo ograniczone nie tylko dlatego, że bazy informacyjne zaczęto publikować w sieci WWW. Podstawowym powodem znacznego spadku zainteresowania Telnetem jest to, że nie zapewnia on bezpieczeństwa przesyłanych danych. Cała zawartość dialogu ze zdalnym komputerem (łącznie z hasłem) jest przesyłana siecią bez żadnego zabezpieczenia, co umożliwia jego podsłuchanie. Z tego powodu korzystanie z Telnetu nie jest zalecane i powinno zostać ograniczone do sytuacji, kiedy nie ma możliwości zastosowania innego protokołu.

Telnet został zastąpiony przez protokół **ssh** (*Secure Shell*), bezpieczny protokół zdalnej sesji. Działa on podobnie do protokołu Telnet, ale oferuje zarówno szyfrowanie nazwy konta i hasła, jak i całej prowadzonej zdalnie sesji [28]. Obecnie zalecana do używania jest jego wersja 2, choć programy obsługujące wersję 1 są cały czas dostępne. SSH2 jest standaryzowane w grupie roboczej *SecSh Internet Engineering Task Force* (IETF). Jest to otwarty i dobrze udokumentowany standard.

Zasada działania protokołu ssh [28] opiera się na kryptograficznej technologii szyfrowania asymetrycznego **RSA** (patrz rozdz. 8.1.3.). Każdy z komputerów, na którym zainstalowane jest oprogramowanie ssh, posiada parę kluczy: **klucz prywatny**, dostępny tylko dla administratora komputera (i oczywiście dla oprogramowania ssh) oraz **klucz publiczny**, dostępny dla użytkowników sieci. Informację zaszyfrowaną kluczem prywatnym można odszyfrować tylko za pomocą klucza publicznego i odwrotnie – informację zaszyfrowaną kluczem publicznym można odszyfrować tylko za pomocą klucza prywatnego. Klucze są ze sobą powiązane, ale żadnego z nich nie można odtworzyć na podstawie znajomości drugiego.

Połączenie ssh jest inicjowane przez klienta ssh. Łączy się on z serwerem i otrzymuje od niego jego klucz publiczny. Klucz ten jest porównywany z kluczem przechowywanym w wewnętrznej bazie danych klienta, z poprzednich połączeń. W przypadku wykrycia niezgodności kluczy wyświetlane jest specjalne ostrzeżenie, umożliwiające przerwanie połączenia. Następnie klient przekazuje serwerowi swój klucz publiczny, generuje 256-bitową liczbę losową, szyfruje ją za pomocą swojego klucza prywatnego oraz klucza publicznego serwera. Serwer po otrzymaniu tak zakodowanej liczby rozszyfrowuje ją za pomocą swojego klucza prywatnego i klucza publicznego klienta. Tak otrzymana liczba jest losowa i znana tylko klientowi i serwerowi. Jest ona używana jako klucz do kodowania podczas dalszej komunikacji.

W trakcie prac nad protokołem ssh opracowano jego liczne udoskonalenia i rozszerzenia. Jednym z nich jest możliwość autoryzacji użytkowników za pomocą pary kluczy RSA (jak przy nawiązywaniu połączenia ssh). Jest to znacznie bezpieczniejszy sposób logowania niż przy użyciu tradycyjnych haseł. Aby używać autoryzacji RSA należy wygenerować parę kluczy

(prywatny i publiczny). W systemach unixowych do tego celu służy program **ssh-keygen** (jego opis można uzyskać poleceniem *man ssh-keygen*). Wygenerowany klucz publiczny należy umieścić w pliku *identity.pub* w podkatalogu .ssh swojego katalogu domowego. Klucz prywatny należy przenieść w bezpieczne miejsce. Dla zwiększenia bezpieczeństwa można podczas tworzenia pary kluczy RSA zażądać dodatkowego zaszyfrowania klucza prywatnego wymyślonym przez siebie hasłem. Wówczas przed każdym wykorzystaniem swojego klucza prywatnego do uzyskania połączenia z serwerem konieczne jest podanie hasła, rozkodowującego klucz. Wiele programów klientów ssh pozwala na wykorzystanie autoryzacji RSA zamiast wprowadzania tradycyjnych haseł.

Innym rozszerzeniem oprogramowania realizującego połączenia ssh jest protokół **scp** umożliwiający bezpieczne przesyłanie plików. Powoli wypiera on opracowany w początkach istnienia sieci Internet protokół FTP, którego wadą (podobnie jak Telnetu) jest możliwość podsłuchania zarówno hasła, jak i zawartości transmisji. Protokół ssh może również tunelować dowolne sesje TCP przez pojedyncze zaszyfrowane połączenia. Tunelowanie ma szerokie zastosowanie w zabezpieczaniu komunikacji innych aplikacji i protokołów bez modyfikowania samych aplikacji. Używając tuneli użytkownik może dalej używać bezpiecznie aplikacji, które same w sobie nie są bezpieczne, takich jak poczta elektroniczna.

6.8. Protokół HTTP

HTTP (*Hypertext Markup Language*) jest protokołem używanym do porozumiewania się klientów (zwykle przeglądarek) i serwerów pracujących w sieci WWW [31]. Należy on do rodziny protokołów TCP/IP i jest to protokół warstwy aplikacji (tak jak FTP i Telnet). Obecnie stosowana jest wersja *HTTP 1.1*, przedstawiona w dokumencie RFC 2068 *Hypertext Transfer Protocol – HTTP 1.1* wraz z wieloma późniejszymi rozszerzeniami, uwzględniającymi m.in. bezpieczną transmisję z wykorzystaniem kluczy publicznych.

Proces wymiany danych używa modelu klient-serwer, jak większość protokołów sieciowych. Zawsze jest on inicjowany przez przeglądarkę, która zestawia połączenie z wybranym serwerem, a następnie wysyła do niego zapytanie HTTP. Serwer analizuje zapytanie i wysyła odpowiedź. Na tym komunikacja się kończy, tzn. następną odpowiedź serwera można uzyskać zadając następne zapytanie, a pomiędzy zapytaniami nie są przechowywane na serwerze żadne dane na temat tej sesji. Dlatego protokół HTTP nazywamy protokołem bezstanowym.

6.8.1. Nawiązanie połączenia HTTP

Przeglądarka może nawiązać połączenie z serwerem, jeśli zna adres **URL** (*Uniform Resource Locator*), który jednoznacznie określa lokalizację interesującego nas zasobu w Interne-

cie. Zasobem takim może być dokument HTML, udostępniany przez serwer HTTP, jak i plik na serwerze FTP czy artykuł przechowywany na serwerze grup dyskusyjnych.

Po otrzymaniu URL'a o postaci:

```
http://www.wsei.pl/pub/wyklady/siecikomputerowe/http.html
```

przeglądarka dzieli go na części składowe. Po oznaczeniu protokołu HTTP znajduje się nazwa serwera, a następnie ścieżka i nazwa dokumentu do pobrania. W pierwszej kolejności przeglądarka analizuje część URL'a zawierającą nazwę serwera, ponieważ na tej podstawie musi określić adres IP, niezbędny do nawiązania połączenia. Do tego celu wykorzystuje omawiany wcześniej protokół DNS (rozdz. 6.3). Jeśli adres serwera jest od razu podany w postaci IP, na przykład:

```
http://214.45.7.18/plik.html
```

przeglądarka nie traci czasu na konwersję nazwy serwera na adres IP.

Po ustaleniu adresu i skontaktowaniu się z serwerem przeglądarka nawiązuje połączenie TCP. Połączenie to działa jak bezpieczny kanał transmisyjny, która dba o poprawność przesyłanych danych. Teraz rozpoczyna się komunikacja na poziomie protokołu HTTP: przeglądarka wysyła **zapytanie** (*request*), na które serwer wysyła **odpowiedź** (*response*). Odpowiedź zwykle zawiera zasób „zamawiany" w zapytaniu.

Aby połączyć się z serwerem HTTP przeglądarka musi znać także numer portu, na którym serwer oczekuje na zapytania HTTP. Standardowo jest to port TCP numer 80, ale może to zostać zmienione, aby uchronić się przed nieproszonymi gośćmi. Pełny adres zasobu WWW może zawierać jeszcze nazwę użytkownika i hasło, potrzebne do uzyskania dostępu do tego zasobu. Ma on wtedy postać:

```
adres_serwera[:numer_portu[@identyfikator_użytkownika[:hasło]]]
```

Elementy podane w nawiasach prostokątnych są opcjonalne.

6.8.2. Format zapytania HTTP

Format zapytania i odpowiedzi jest podobny i składa się z następujących elementów:

- wiersz żądania/statusu
- wiersze pól nagłówkowych
- pusty wiersz (CRLF)
- część zasadnicza (opcjonalna)

Zapytanie (żądanie) lub odpowiedź wygląda następująco (przy czym pól nagłówkowych może być więcej):


```
<wiersz początkowy, inny dla zapytania i odpowiedzi>
Nagłówek1: wartość1
Nagłówek2: wartość2
Nagłówek3: wartość3
<opcjonalna część zasadnicza, jak zwracana strona http, wynik
zapytania do bazy danych itp.>
```

Wiersz początkowy zapytania składa się z następujących części:

```
GET /ścieżka/do/pliku.html HTTP/1.1
```

gdzie:

1. **GET** to standardowa metoda, służąca do pobierania zasobów w sieci WWW, stosowana przez przeglądarki, gdy użytkownik wybiera URL; żądania GET nie zawierają części zasadniczej;
2. ścieżka do żądanego zasobu (część URL'a występująca po adresie serwera, zwana także **URI** [*Uniform Resource Identifier*]);
3. wersja protokołu HTTP, zawsze w formie HTTP/x.x

Pozostałe metody żądań HTTP to:

- **HEAD** – serwer wysyła identyczną odpowiedź, jak na żądanie GET, ale bez części zasadniczej;
- **POST** – służy do wysyłania informacji do serwera; żądania POST zawsze zawierają część zasadniczą. Metodę POST stosuje się najczęściej w połączeniu z formularzami HTML.

Nazwy metod pisze się zawsze wielkimi literami.

Ważniejsze pola nagłówkowe w żądaniach HTTP:

- *Host* – nazwa (adres domenowy) docelowego serwera WWW;
- *User-Agent* – informacje o kliencie (nazwa, adres, platforma systemowa itp.);
- *Referer* – URL dokumentu zawierającego odnośnik, za pomocą którego nastąpiło żądanie bieżącego zasobu;
- *Content-Type* – typ danych przesyłanych przez klienta (np. w żądaniu POST) w części zasadniczej;
- *Content-Length* – rozmiar w bajtach części zasadniczej;
- *Accept* – lista akceptowanych formatów plików, podanych jako tzw. typy MIME (*Multipurpose Internet Mail Extensions*).

Poniższy przykład krótko ilustruje sposób tworzenia zapytania HTTP. Plikiem pobieranym jest tu *index.html*, umieszczony w katalogu *test* na serwerze *www.wsei.pl*. Następnie użyte są cztery z wymienionych wyżej pól nagłówkowych. Za nimi obowiązkowo musi być umieszczony pusty wiersz.

Przykład 1:

```
GET /test/index.html HTTP/1.1
Referer: http://mypage.com.pl
User-Agent: Mozilla/4.76
Host: www.wsei.pl
Accept: image/gif, image/jpeg
```

6.8.3. Format odpowiedzi HTTP

Wiersz początkowy odpowiedzi zwany jest **linią statusu**:

```
HTTP/1.1 200 OK
```

Składa się on z trzech części:

- wersji protokołu HTTP w formacie takim jak dla żądania: HTTP/x.x,
- kodu numerycznego statusu odpowiedzi,
- opisu odpowiedzi.

Kod statusu odpowiedzi jest trzycyfrowy, a jego pierwsza cyfra identyfikuje kategorię odpowiedzi:

1xx – żądanie jest w trakcie przetwarzania,

2xx – żądanie powiodło się,

3xx – przekierowanie żądania pod inny URL,

4xx – oznacza błąd po stronie klienta,

5xx – oznacza błąd po stronie serwera.

Ważniejsze pola nagłówkowe w odpowiedziach HTTP:

- *Content-Type* – typ danych przesyłanych przez serwer, podanych jako typy MIME, np. `text/html, text/plain, image/jpg`;
- *Content-Length* – rozmiar w bajtach części zasadniczej odpowiedzi;
- *Date* – data i czas wysłania odpowiedzi;
- *Location* – obowiązkowe w odpowiedziach z kodem stanu 3xx, określa nowy URL zasobu;
- *Server* – nazwa i wersja oprogramowania serwera WWW;
- *Expires* – data i czas wygaśnięcia ważności zasobu; po tym czasie zasób może zostać zmieniony lub zawarte w nim informacje staną się nieważne.

Poniższy przykład krótko ilustruje sposób tworzenia odpowiedzi HTTP. W pierwszym wierszu znajduje się wiersz statusu, mówiący, że żądanie się powiodło. Następnie użyte są cztery z wymienionych wyżej pól nagłówkowych. Za nimi obowiązkowo musi być umieszczony pusty wiersz. Za wierszem pustym znajduje się część zasadnicza, zawierająca zamawianą stronę html.

Przykład 2:

```
HTTP/1.1 200 OK
Date: Sat, 9 Aug 2003 22:00:00 GMT
Server: Apache/1.3.23 (Unix)
Content-Lenght: 1778
Content-Type: text/html
<HTML>
....
</HTML>
```

6.9. Protokół NFS

Protokół **NFS** (*Network File System*) powstał w firmie SUN Microsystems w latach osiemdziesiątych. Po udostępnieniu opisu szybko stał się standardowym protokołem dzielenia plików w sieci komputerowej pomiędzy systemami operacyjnymi różnych producentów, w tym: UNIX, IBM mainframe, VMS, OS/2. Powstała także implementacja tego protokołu dla systemów Microsoftu.

Protokół jest zaprojektowany w architekturze klient-serwer. Serwer **eksportuje** katalogi w systemie plików, natomiast klient **montuje** wyeksportowane katalogi i używa ich jak każdego innego systemu plików. Protokół ten jest bezstanowy, tzn. nie zachowuje stanu po stronie serwera, co w praktyce oznacza, że serwer NFS może być zrestartowany bez utraty zapisanych danych. Po restarcie kontynuuje zapis, nie przerywając pracy. NFS pracuje w oparciu o protokół UDP.

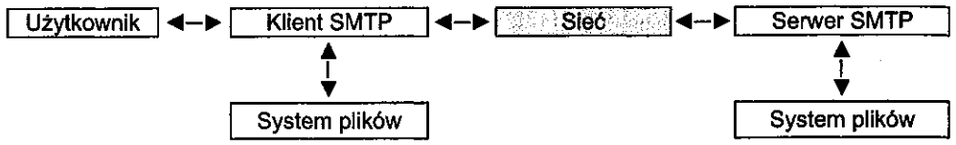
6.10. Protokół pocztowy SMTP

Protokół pocztowy **SMTP** (*Simple Mail Transfer Protocol*), prosty protokół przesyłania poczty, powstał ponad 20 lat temu i był wielokrotnie unowocześniany. Pierwszym dokumentem RFC (rozdz. 1.2) opisującym ten standard był RFC 821 z 1982 roku. Kolejne RFC precyzowały dodatkowe kwestie związane z przesyłaniem poczty. Dokumentem, który podsumowuje ten temat, jest RFC 2821 z roku 2001 [24]. Dokumenty RFC z lat następnych, związane z tym protokołem, dodają do niego nowe możliwości w miarę rozwoju technologii internetowych.

SMTP został zaprojektowany tak, aby przesyłał pocztę pomiędzy różnymi sieciami. Przesyłka na swej drodze od nadawcy do adresata może przechodzić przez kolejne bramki międzysieciowe. Protokół ten jest także niezależny od mechanizmu przesyłania danych i wymaga jedynie kanału pozwalającego na skuteczne przesyłanie uporządkowanego strumienia danych. Najczęściej SMTP wykorzystuje w tym celu protokół TCP/IP. SMTP określa, jak system przesyłania poczty przesyła wiadomości przez łącze między jedną maszyną a drugą. Standard nie określa natomiast, jak system poczty przyjmuje wiadomości od użytkownika, ani jak przechowywane są wiadomości.

Działanie protokołu pocztowego SMTP ilustruje rys. 54.

Rys. 54. Działanie protokołu pocztowego SMTP



Kiedy klient SMTP ma do wysłania przesyłkę, otwiera dwukierunkowy kanał transmisyjny do serwera SMTP. Sposób wymiany komunikatów za pomocą protokołu SMTP jest bardzo prosty. Można nawet połączyć się przez Telnet z portem TCP numer 25 maszyny, która obsługuje SMTP, i wysłać pocztę wpisując ręcznie odpowiednie polecenia w postaci tekstu i otrzymując odpowiedzi także w postaci czytelnego tekstu. Jest to zresztą jedna z metod diagnostyki problemów z konfiguracją poczty. Typową wymianę komunikatów opisano poniżej.

Po otwarciu odpowiedniego kanału transmisji do serwera, klient oczekuje na otrzymanie od niego komunikatu *220 READY FOR MAIL*. Jeśli serwer jest obciążony, może opóźnić wysłanie tego komunikatu. Po otrzymaniu tego komunikatu, klient wysyła do serwera komunikat *HELO* (jest to skrót od *HELLO*). Serwer odpowiada podając swoją nazwę. Następnie klient może wysłać jedną lub więcej przesyłek i zakończyć połączenie, albo zażądać zamiany ról między klientem a serwerem, żeby pobrać przesyłki. Odbiorca musi potwierdzić otrzymanie każdej przesyłki. Może również zerwać całe połączenie albo przesyłanie bieżącej przesyłki.

Transakcje pocztowe rozpoczynają się poleceniem *MAIL FROM:*, które zawiera adres wysyłającego, na jaki należy przesyłać informacje o błędach. Po otrzymaniu tego polecenia odbiorca przygotowuje się do odebrania przesyłki, a następnie odpowiada wysyłając *250 OK*. Odpowiedź ta oznacza, że polecenie zostało odebrane i wykonane poprawnie. Następnie wysyłający przesyła ciąg poleceń *RCPT*, określających docelowych odbiorców przesyłki. Serwer musi potwierdzić każde polecenie *RCPT* wysyłając *250 OK*, albo komunikat o błędzie *550 No such user here*.

Po przesłaniu wszystkich poleceń *RCPT* klient wysyła polecenie *DATA*. Serwer odpowiada komunikatem *354 Start mail input* i określa ciąg znaków, który ma być użyty do oznaczenia końca przesyłki. Najczęściej jest to sekwencja

<CR><LF>.<CR><LF>

Para znaków <CR><LF> oznacza przejście do nowego wiersza. Są to znaki o kodach ASCII 10 i 13.

Poniższy przykład ilustruje sytuację, kiedy nadawca *smith@alpha.edu* wysyła przesyłkę do trzech odbiorców: *jones@beta.gov, green@beta.gov i brown@beta.gov*. Program klienta SMTP na serwerze *alpha.edu* łączy się z serwerem SMTP na komputerze *beta.gov* i rozpoczyna wymianę komunikatów, mającą na celu wysłanie listów. Puste wiersze widoczne pomiędzy komunikatami są dodane, aby uczynić przykład bardziej czytelnym.

Przykład 3:

```
Serwer: 220 beta.gov Simple Mail Transfer Service Ready
Klient: HELO alpha.edu
S: 250 beta.gov

K: MAIL FROM:<smith@alpha.edu>
S: 250 OK

K: RCPT TO:<jones@beta.gov>
S: 250 OK

K: RCPT TO:<green@beta.gov>
S: 550 No such user here

K: RCPT TO:<brown@beta.gov>
S: 250 OK

K: DATA
S: 354 Start mail input; end with <CR><LF>.<CR><LF>
K: ... tu jest treść przesyłki ...
K: ... tyle linii, ile potrzeba ...
K: <CR><LF>.<CR><LF>
S: 250 OK

K: QUIT
S: 221 beta.gov Service closing transmission channel
```

W powyższym przykładzie serwer odrzucił adresata *green@beta.gov*, nie uznając go za poprawnego, bo na przykład nie ma takiego użytkownika ani listy adresowej. Protokół SMTP nie określa, jak klient powinien postąpić w przypadku takiego błędu. Chociaż klient może

całkowicie przerwać transmisję, większość klientów kontynuuje przesyłanie do wszystkich potwierdzonych przez serwer adresatów, a następnie informuje nadawcę o błędzie. Najczęściej robi to, wysyłając mu informację pocztą elektroniczną.

Po tym, jak klient wyśle wszystkie przesyłki, które ma do przesłania, może wysłać polecenie *TURN*, które oznacza propozycję zamiany ról między klientem a serwerem. Jeśli serwer się zgodzi, będzie mógł przesyłać pocztę do klienta.

Powyższy opis protokołu SMTP dotyczy jego podstawowych możliwości. Jego możliwości są oczywiście większe. Na przykład, jeśli użytkownik zmienił adres, serwer może znać jego nowy adres i odpowiednio przekierować pocztę. Kolejne rozszerzenia (np. MIME) obsługują przesyłanie plików binarnych jako załączników listu elektronicznego.

6.11. Protokół pocztowy POP3

Protokół **POP** (*Post Office Protocol*) jest używany do pobierania poczty elektronicznej z serwera do skrzynki pocztowej użytkownika. Większość klienckich aplikacji pocztowych używa trzeciej wersji protokołu, czyli **POP3**, zdefiniowanego w RFC 1939 [22] z późniejszymi rozszerzeniami. Do tych samych celów używany jest również bardziej nowoczesny protokół IMAP, omawiany w kolejnym rozdziale.

Różnicę między protokołem SMTP a protokołem POP3 można zilustrować następującą analogią do rzeczywistej poczty. Protokół SMTP służy do przesyłania poczty pomiędzy poszczególnymi urzędami pocztowymi, czyli serwerami wyposażonymi w możliwość przesyłania poczty dalej. Może służyć też do dostarczania poczty przez klienta do urzędu pocztowego. Natomiast protokół POP3 służy do pobierania poczty ze skrytki, w której umieścił ją listonosz.

Serwer protokołu POP3 oczekuje na polecenia klientów, „słuchając" na porcie TCP o numerze 110. Kiedy klient chce użyć protokołu POP3, otwiera połączenie TCP z komputerem aplikacji serwera. Po tym, jak połączenie zostaje otwarte, serwer POP3 wysyła powitanie. Następnie klient wysyła polecenia, a serwer – odpowiedzi, dopóki połączenie nie zostanie zamknięte.

Polecenia POP3 składają się ze słowa kluczowego, po którym może wystąpić jeden lub więcej argumentów. Wszystkie polecenia zakończone są parą znaków CRLF (*carriage return, line feed* – znaki o kodach ASCII 10 i 13). Odpowiedzi w POP3 składają się ze wskaźnika statusu i słowa kluczowego, po których mogą występować dodatkowe informacje. Wszystkie odpowiedzi zakończone są parą znaków CRLF. Obecnie są dwa możliwe wskaźniki statusu: pozytywny +OK i negatywny –ERR.

Sesja POP3 przechodzi w ciągu swojego przebiegu przez kilka stanów. Po otwarciu połączenia TCP i wysłaniu powitania przez serwer, sesja wchodzi w stan AUTORYZACJI. W tym stanie klient musi przedstawić się serwerowi POP3, na przykład przez podanie swojego identyfikatora i hasła. Kiedy klient to zrobi, serwer dostaje się do jego skrytki pocztowej i sesja wchodzi w stan TRANSAKCJI. W tym stanie klient zleca serwerowi POP3 różne działania. Kiedy klient wyda polecenie QUIT, sesja wchodzi w stan AKTUALIZACJI. W tym stanie serwer usuwa listy, których usunięcie zlecił klient w stanie TRANSAKCJI, zamyka skrytkę pocztową klienta i żegna się z nim. Następnie połączenie TCP zostaje zamknięte.

W stanie AUTORYZACJI klient może wydawać serwerowi polecenia, z których dwa najważniejsze to:

USER – obowiązkowy argument tego polecenia to identyfikator użytkownika, którego skrytkę pocztową klient chce otworzyć; w odpowiedzi serwer informuje, czy podano prawidłowy identyfikator użytkownika,

PASS – obowiązkowy argument tego polecenia to hasło; w odpowiedzi serwer informuje, czy podano prawidłowe hasło dla użytkownika podanego uprzednio poleceniem USER.

W stanie TRANSAKCJI klient może wydawać serwerowi polecenia, z których najważniejsze są następujące:

STAT – serwer zwraca liczbę listów w skrytce i ich sumaryczną wielkość (w bajtach),

LIST – jeśli polecenie podane jest z argumentem, to argument ten jest numerem listu w skrytce; serwer w odpowiedzi zwraca wielkość tego listu; jeśli klient wydał polecenie bez argumentu, to serwer zwraca odpowiedź w wielu liniach tekstu – w każdej linii jest wielkość kolejnego listu ze skrytki,

RETR – obowiązkowy argument tego polecenia oznacza numer listu w skrytce; serwer zwraca treść tego listu,

DELE – obowiązkowy argument tego polecenia oznacza numer listu w skrytce; serwer zaznacza ten list do usunięcia; zostanie on usunięty w stanie AKTUALIZACJI (w trakcie wykonywania przez serwer polecenia QUIT),

RSET – jeśli jakieś listy zostały zaznaczone do usunięcia (przy pomocy polecenia DELE), to wykonanie polecenia RSET usuwa to zaznaczenie,

QUIT – wydanie tego polecenia powoduje opuszczenie stanu TRANSAKCJI i przejście do stanu AKTUALIZACJI, w którym usuwane są listy zaznaczone do usunięcia.

Poniżej przedstawiono przebieg przykładowej sesji protokołu POP3 (S oznacza serwer, a K – klienta).

Przykład 4:

```
S: <oczekuje na połączenie z portem 110 TCP>
K: <otwiera połączenie>
S: +OK POP3 server ready <4856.7892365823456@sikorka.ptaki.pl>
K: USER bogatka
```

```
S: +OK bogatka
K: PASS secret-latania
S: +OK bogatka's maildrop has 2 messages (320 octets)
K: STAT
S: +OK 2 320
K: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: <CR><LF>.<CR><LF>
K: RETR 1
S: +OK 120 octets
S: <serwer POP3 wysyła pierwszy list>
S: <CR><LF>.<CR><LF>
K: DELE 1
S: +OK message 1 deleted
K: RETR 2
S: +OK 200 octets
S: <serwer POP3 wysyła drugi list>
S: <CR><LF>.<CR><LF>
K: DELE 2
S: +OK message 2 deleted
K: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
K: <zamyka połączenie>
S: <oczekuje na następne połączenie>
```

6.12. Protokół pocztowy IMAP

Ograniczenia protokołu POP3 spowodowały, że w 1986 roku rozpoczęto pracę nad jego następcą – protokołem **IMAP** (*Internet Message Access Protocol*). Odtąd opracowywano kolejne jego wersje, aż w 1994 r. powstała wersja 4, IMAP4 [21]. W roku 2003 pojawiła się rewizja 1 tego standardu, opisana w RFC 3501 [26].

Protokół POP3 jest prosty i skuteczny, ale w wielu sytuacjach ujawniają się jego wady. Po pierwsze – przystosowany jest do modelu pracy, w którym klient okresowo łączy się z serwerem, pobiera listy usuwając je z serwera, a całe przetwarzanie otrzymanej korespondencji odbywa się lokalnie. Powoduje to kłopoty, jeśli ta sama osoba chciałaby przeglądać odebrane komunikaty używając kolejno różnych stacji roboczych. Kolejnym ograniczeniem POP3 jest brak z jego strony wsparcia dla przesyłania wybranych elementów zestrukturalizowanej wiadomości pocztowej. Jeśli zatem do małej wiadomości dołączony jest duży załącznik, to użytkownik musi odebrać całą przesyłkę, żeby poznać treść tej wiadomości.

Ograniczenia POP3 widać o wiele lepiej, kiedy wymieni się te cechy IMAP, których POP3 nie posiada:

IMAP

- możliwość zapamiętywania atrybutów listu (np. przeczytany, usunięty, odpowiedziano na niego itd.);
- udostępnienie wielu skrytek pocztowych jednemu użytkownikowi; może on je także tworzyć i usuwać;
- możliwość umieszczania przez klienta listów w skrytkach pocztowych, a nie tylko pobierania ich stamtąd;
- wybrane skrytki pocztowe mogą być wspólne dla wielu użytkowników;
- umożliwienie jednoczesnego dokonywania różnych operacji przez wielu użytkowników na współdzielonej skrytce pocztowej;
- możliwość dostępu także do danych nie będących e-mailami, np. list dyskusyjnych, czy też dokumentów;
- możliwość selektywnego pobierania z serwera wybranych elementów złożonych komunikatów pocztowych;
- możliwość dokonywania zapytań wyszukiwawczych po stronie serwera na zawartości jednej lub wielu skrytek pocztowych (bez pobierania ich zawartości z serwera).

6.12.1. Stany sesji IMAP

Podobnie jak w przypadku POP3 sesja IMAP może znajdować się w jednym z czterech stanów:

NIE AUTORYZOWANO – w tym stanie klient musi dostarczyć informacji uwierzytelniających. W przeciwnym razie większość poleceń nie będzie działać. Wejście w ten stan następuje po otwarciu połączenia, chyba że jest to połączenie z wcześniejszą autoryzacją.

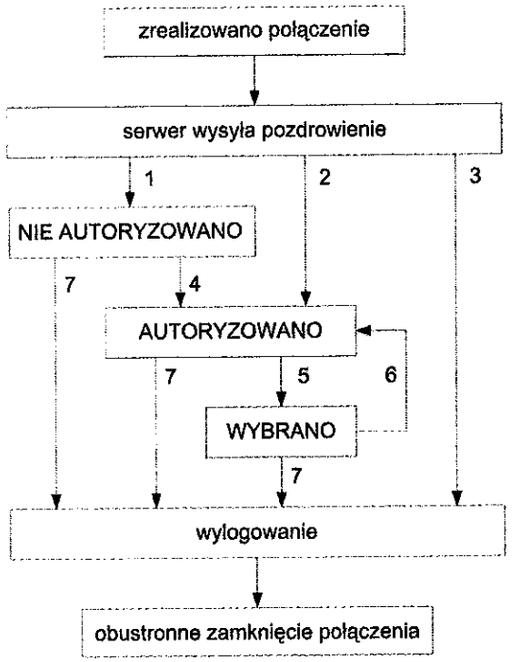
AUTORYZOWANO – w tym stanie klient jest autoryzowany i musi wybrać skrytkę pocztową, żeby skorzystać z poleceń wpływających na wiadomości. Wejście w ten stan następuje po otwarciu uprzednio autoryzowanego połączenia, po prawidłowym podaniu informacji uwierzytelniających, po błędnym wybraniu skrytki pocztowej albo po udanym wykonaniu polecenia CLOSE (które zamyka wybraną skrytkę pocztową).

WYBRANO – sesja znajduje się w tym stanie, kiedy klient określi, na której skrytce pocztowej chce aktualnie pracować.

WYLOGOWANIE – w tym stanie następuje zamknięcie połączenia. Wejście w ten stan może nastąpić w wyniku żądania ze strony klienta (za pomocą polecenia LOGOUT) albo w wyniku jednostronnej akcji czy to klienta, czy serwera.

Początkowy stan jest zaznaczony w powitaniu wysyłanym przez serwer. Przejścia między poszczególnymi stanami obrazuje poniższy diagram.

Rys. 55. Schemat stanów sesji IMAP i przejść pomiędzy nimi [26]



Numery na rysunku oznaczają przejścia pomiędzy poszczególnymi stanami:

1. Połączenie bez wcześniejszej autoryzacji (powitanie OK).
2. Połączenie z wcześniejszą autoryzacją (powitanie PREAUTH).
3. Połączenie odrzucone (powitanie BYE).
4. Wykonanie polecenia LOGIN lub AUTHENTICATE zakończone powodzeniem.
5. Wykonanie polecenia SELECT lub EXAMINE zakończone powodzeniem.
6. Polecenie CLOSE albo nieudane wykonanie polecenia SELECT lub EXAMINE.
7. Polecenie LOGOUT, zamknięcie serwera albo połączenia.

Jak wynika z rys. 55, niektóre stany sesji mogą zostać w pewnych przypadkach pominięte. Sesja może także kilkakrotnie powracać do stanu WYBRANO.

6.12.2. Polecenia IMAP

Zostaną tu omówione najważniejsze polecenia protokołu IMAP4. Większość poleceń działa tylko w określonych stanach sesji. Polecenia są pogrupowane według najniższego stanu, w jakim są dostępne.

Polecenia dostępne we wszystkich stanach:

CAPABILITY – polecenie pozwala klientowi dowiedzieć się, jakie dodatkowe funkcje związane z IMAP może realizować serwer. Lista tych funkcji nie zależy od stanu sesji ani od użytkownika.

NOOP – polecenie to nic nie robi. Ale skoro w wyniku każdego polecenia klient może otrzymać od serwera listę informacji o zmianie statusu wiadomości lub o nadejściu nowych wiadomości, to polecenie NOOP może służyć do okresowego uzyskiwania takich informacji.

LOGOUT – polecenie informuje serwer, że klient chce zakończyć sesję.

Polecenia dostępne w stanie NIE AUTORYZOWANO:

AUTHENTICATE – polecenie pozwala klientowi na wybranie sposobu autoryzacji i rozpoczęcie z serwerem wymiany informacji (seria pytań serwera i odpowiedzi klienta), która ma doprowadzić do uwierzytelnienia klienta.

LOGIN – jako argumenty tego polecenia klient podaje nazwę użytkownika i hasło.

Polecenia dostępne w stanie AUTORYZOWANO:

SELECT – argumentem tego polecenia jest nazwa skrytki pocztowej, na której użytkownik chce pracować. Poprawne wykonanie tego polecenia kończy się przejściem do stanu WYBRANO. Serwer zwraca do klienta listę flag dostępnych do oznaczania wiadomości znajdujących się w danej skrytce pocztowej. Zwraca także całkowitą liczbę wiadomości w danej skrytce oraz liczbę wiadomości nowych, czyli tych, które pojawiły się od momentu, kiedy ostatnio odczytywano informacje o skrytce.

EXAMINE – polecenie to jest takie samo jak polecenie SELECT z jedną różnicą, że otwiera skrytkę pocztową w trybie tylko do odczytu.

CREATE – polecenie tworzy skrytkę pocztową o danej nazwie. Skrytki pocztowe mogą tworzyć hierarchię analogiczną do katalogów (folderów) w systemie plików.

DELETE – polecenie usuwa skrytkę pocztową o danej nazwie.

RENAME – polecenie pozwala zmienić nazwę skrytki pocztowej na inną. Jeśli zmienimy nazwę domyślnej skrytce *Inbox* (w systemach polskojęzycznych zwanej *Odebrane*), to automatycznie zostanie utworzona nowa pusta skrytka *Inbox*.

SUBSCRIBE – polecenie powoduje rozpoczęcie przez serwer prenumeraty listy dyskusyjnej (*news*) o danej nazwie. Otrzymane z listy dyskusyjnej wiadomości będą dostępne w skrzynce o takiej samej nazwie jak nazwa listy dyskusyjnej poprzedzona znakiem #.

UNSUBSCRIBE – polecenie powoduje zakończenie przez serwer prenumeraty listy dyskusyjnej o danej nazwie.

LIST – polecenie zwraca listę nazw wszystkich skrytek pocztowych, które pasują do podanego w argumencie wzorca i są dostępne bieżącemu użytkownikowi. Jest analogiczne do DOS'owego polecenia *dir*, które pokazywałoby tylko nazwy podkatalogów, a nie zwracało nazw plików.

LSUB – polecenie zwraca listę nazw wszystkich grup dyskusyjnych, które zaprenumerował bieżący użytkownik i których nazwy pasują do wzorca podanego w argumencie tego polecenia.

APPEND – polecenie umieszcza dany tekst jako nową wiadomość w danej skrytce pocztowej.

W danej sesji może być wybrana tylko jedna skrytka pocztowa. Aby pracować na wielu skrytkach, trzeba otworzyć wiele sesji.

Polecenia dostępne w stanie WYBRANO:

CLOSE – polecenie ostatecznie usuwa z bieżącej skrytki pocztowej wiadomości zaznaczone do usunięcia, a następnie powoduje przejście ze stanu WYBRANO do stanu AUTORYZOWANO.

EXPUNGE – polecenie ostatecznie usuwa z bieżącej skrytki pocztowej wiadomości zaznaczone do usunięcia.

SEARCH – polecenie zwraca listę wszystkich wiadomości z bieżącej skrytki pocztowej, które spełniają warunki podane w argumentach tego polecenia. Warunki te mogą dotyczyć atrybutów wiadomości – binarnych (np. *nowy*) lub złożonych (np. *data*). Mogą też wymagać obecności danego tekstu w danym polu wiadomości (np. w jej treści).

FETCH – polecenie zwraca treść danej wiadomości: całość lub jej fragment określony w argumencie. Na przykład, jeśli wiadomość składa się z tekstu i z binarnego załącznika, to poleceniem FETCH można pobrać całą wiadomość albo też tylko jedną z tych dwu części.

PARTIAL – polecenie jest rozszerzeniem polecenia FETCH o możliwość podania przedziału numerów bajtów danych. Na przykład można nim zlecić pobranie pierwszych 100 bajtów załącznika.

STORE – polecenie działa tak jak polecenie FETCH z tą różnicą, że dane przesyłane są w drugą stronę. Można za jego pomocą dokonać modyfikacji elementu wiadomości znajdującej się w skrytce pocztowej na serwerze.

COPY – polecenie służy do kopiowania wiadomości z jednej skrytki pocztowej do drugiej.

7. Bezpieczeństwo sieci LAN

Sieci komputerowe pozwalają na wymianę informacji między komputerami – Internet daje nieograniczony dostęp do informacji i możliwości ich publikowania. Jednocześnie zastosowanie tych sieci zwiększa zagrożenia bezpieczeństwa komputerów, ponieważ umożliwia fałszowanie i niszczenie informacji w zupełnie nowy sposób. W niniejszym rozdziale zostaną omówione problemy bezpieczeństwa komputerowego ze szczególnym uwzględnieniem lokalnych sieci komputerowych [20].

7.1. Zasoby podlegające ochronie

Bezpieczeństwo komputerowe polega na ochronie następujących zasobów [20]:

- danych,
- sprzętu,
- reputacji (dobrego imienia lub wizerunku firmy albo osoby).

Zagrożenia dla bezpieczeństwa danych lub sprzętu często stanowią jednocześnie zagrożenia dla reputacji, ale są też takie zagrożenia dla reputacji, które nie wynikają z zagrożeń dla danych lub sprzętu, i dlatego zostaną omówione osobno.

7.2. Dane

Ochrona danych to zapewnienie ich [20]:

- tajności,
- integralności,
- dostępności.

Dzięki tajności mamy pewność, że nikt niepowołany nie odczyta chronionych danych. Integralność, czyli spójność, jest zapewniona, kiedy nikt niepowołany nie zmodyfikuje chronionych danych. Dostępność oznacza, że nikt niepowołany nie ogranicza dostępu do chronionych danych osobom uprawnionym.

W pierwszej chwili z bezpieczeństwem kojarzy się zwykle tylko zapewnienie tajności, jednak nie jest to wystarczające. Przypuśćmy, że firma X publikuje w Internecie cennik swoich produktów. Widać, że tajność nie ma w tym przypadku zastosowania. Konieczne jest natomiast zapewnienie integralności, aby nikt niepowołany nie mógł zmienić tego cennika.

Konieczne jest też zapewnienie dostępności, aby każdy uprawniony mógł odczytać ten cennik przez sieć w możliwie krótkim czasie.

Jak pokazuje powyższy przykład, priorytety w dziedzinie ochrony danych mogą być różne. W przypadku banków i sklepów internetowych tajność i integralność są na pierwszym miejscu, natomiast w przypadku informacyjnych portali internetowych na pierwszym miejscu jest dostępność.

Jeśli chodzi o integralność danych, to paradoksalnie: drobne zmiany w danych wprowadzone przez intruza mogą być o wiele bardziej dotkliwe w skutkach niż całkowite usunięcie danych. Usunięte dane można przecież odtworzyć z kopii zapasowej, a ich zniknięcia trudno nie zauważyć. Natomiast niewielkie z pozoru zmiany danych mogą długo pozostać niezauważone, a skutkiem tego wywołać więcej problemów.

7.1.2. Sprzęt

Głównym zagrożeniem dla bezpieczeństwa sprzętu komputerowego jest używanie go przez niepowołane osoby. Może ono mieć następujące skutki [20]:

- spowolnienie pracy systemu,
- zmniejszenie objętości wolnej przestrzeni dyskowej,
- awarie sprzętu.

Spowolnienie pracy systemu może w dzień pozostać niezauważone, ale kiedy w nocy komputer zacznie sporządzać comiesięczny bilans, może okazać się, że nie zdąży wykonać go w założonym terminie.

Awarie sprzętu w wyniku nieprawidłowego użycia przez sieć należą być może do przeszłości, ale warto pamiętać, że są możliwe. Na przykład oryginalny terminal VT100 firmy DEC wybuchał, kiedy zbyt często wysyłano do niego sekwencje sterujące zmianą częstotliwości odświeżania obrazu.

Warto tu wspomnieć także o niebezpieczeństwie fizycznej kradzieży sprzętu: na przykład w wyniku włamania do budynku firmy, wyrwania komuś z ręki komputera przenośnego na ulicy albo wyciągnięcia go z samochodu.

7.1.3. Reputacja

Jeśli napastnik zaatakuje dane lub sprzęt należące do X i sprawa zostanie upubliczniona, to reputacja X dozna uszczerbku. Ale reputacja X może też być narażona bez zagrażania jego danym lub sprzętowi: na przykład kiedy ktoś będzie się pod X podszywał, rozpowszechniając nieprawdziwe informacje. Internet jest niestety ułatwieniem dla tego rodzaju oszustw. Można wśród nich wymienić następujące [20]:

- wysyłanie e-maili wyglądających, jakby wysłał je kto inny,
- zmiana treści publikowanych na stronach internetowych na treści narażające na szwank reputację posiadacza tych stron,
- używanie zaatakowanego komputera jako odskoczni do atakowania innych komputerów.

Pierwsze dwa z powyższych problemów mogą wiązać się z publikowaniem w czyimś imieniu treści obraźliwych, kłamliwych lub, co gorsza, prawdziwych, lecz bardzo dla ofiary niewygodnych.

Reputacja i wizerunek firmy są najczęściej jej największym zasobem. Przypuszcza się, że wielkie banki są skłonne ponosić ogromne koszty, aby o włamaniach do ich systemów nikt się nie dowiedział.

7.2. Rodzaje ataków

Znanych jest wiele różnych rodzajów ataków na systemy komputerowe. Można je podzielić na trzy kategorie [20]:

- włamania,
- blokady usług,
- kradzieże informacji.

Włamanie

Włamania są najczęstszym rodzajem ataku. Włamanie występujące w czystej postaci nie przeszkadza w dalszym korzystaniu z komputera jego uprawnionym użytkownikom. Większość włamywaczy chce używać zaatakowanego komputera tak, jak jego uprawnieni użytkownicy.

Blokada usług

Blokada usług **DoS** (*Denial of Service*) to atak, którego celem jest uniemożliwienie korzystania z zaatakowanego komputera. Na przykład napastnik włamuje się do kilku komputerów i instaluje na nich program, który wysyła nieustanne żądania do komputera będącego celem ataku DoS. Żądania te są tak dobrane, że atakowany komputer nie może nadążyć z ich obsługiwaniem, co prowadzi do zajęcia stu procent czasu jego procesora. W ten sposób dochodzi do tego, że zaatakowany komputer nie może normalnie działać. Inny rodzaj blokady usług polega na takiej zmianie konfiguracji „systemu kierowania ruchem" (na przykład DNS), że legalne żądania nie docierają do zaatakowanego komputera.

Uniknięcie takich ataków jest praktycznie niemożliwe, ponieważ w Internecie znajdują się tysiące serwerów *zombie* (opanowanych przez włamywaczy), czego administratorzy tych serwerów nie są świadomi. Wobec tego najważniejsze jest takie skonfigurowanie usług, aby

Konieczne jest też zapewnienie dostępności, aby każdy uprawniony mógł odczytać ten cennik przez sieć w możliwie krótkim czasie.

Jak pokazuje powyższy przykład, priorytety w dziedzinie ochrony danych mogą być różne. W przypadku banków i sklepów internetowych tajność i integralność są na pierwszym miejscu, natomiast w przypadku informacyjnych portali internetowych na pierwszym miejscu jest dostępność.

Jeśli chodzi o integralność danych, to paradoksalnie: drobne zmiany w danych wprowadzone przez intruza mogą być o wiele bardziej dotkliwe w skutkach niż całkowite usunięcie danych. Usunięte dane można przecież odtworzyć z kopii zapasowej, a ich zniknięcia trudno nie zauważyć. Natomiast niewielkie z pozoru zmiany danych mogą długo pozostać niezauważone, a skutkiem tego wywołać więcej problemów.

7.1.2. Sprzęt

Głównym zagrożeniem dla bezpieczeństwa sprzętu komputerowego jest używanie go przez niepowołane osoby. Może ono mieć następujące skutki [20]:

- spowolnienie pracy systemu,
- zmniejszenie objętości wolnej przestrzeni dyskowej,
- awarie sprzętu.

Spowolnienie pracy systemu może w dzień pozostać niezauważone, ale kiedy w nocy komputer zacznie sporządzać comiesięczny bilans, może okazać się, że nie zdąży wykonać go w założonym terminie.

Awarie sprzętu w wyniku nieprawidłowego użycia przez sieć należą być może do przeszłości, ale warto pamiętać, że są możliwe. Na przykład oryginalny terminal VT100 firmy DEC wybuchał, kiedy zbyt często wysyłano do niego sekwencje sterujące zmianą częstotliwości odświeżania obrazu.

Warto tu wspomnieć także o niebezpieczeństwie fizycznej kradzieży sprzętu: na przykład w wyniku włamania do budynku firmy, wyrwania komuś z ręki komputera przenośnego na ulicy albo wyciągnięcia go z samochodu.

7.1.3. Reputacja

Jeśli napastnik zaatakuje dane lub sprzęt należące do X i sprawa zostanie upubliczniona, to reputacja X dozna uszczerbku. Ale reputacja X może też być narażona bez zagrażania jego danym lub sprzętowi: na przykład kiedy ktoś będzie się pod X podszywał, rozpowszechniając nieprawdziwe informacje. Internet jest niestety ułatwieniem dla tego rodzaju oszustw. Można wśród nich wymienić następujące [20]:

- wysyłanie e-maili wyglądających, jakby wysłał je kto inny,
- zmiana treści publikowanych na stronach internetowych na treści narażające na szwank reputację posiadacza tych stron,
- używanie zaatakowanego komputera jako odskoczni do atakowania innych komputerów.

Pierwsze dwa z powyższych problemów mogą wiązać się z publikowaniem w czyimś imieniu treści obraźliwych, kłamliwych lub, co gorsza, prawdziwych, lecz bardzo dla ofiary niewygodnych.

Reputacja i wizerunek firmy są najczęściej jej największym zasobem. Przypuszcza się, że wielkie banki są skłonne ponosić ogromne koszty, aby o włamaniach do ich systemów nikt się nie dowiedział.

7.2. Rodzaje ataków

Znanych jest wiele różnych rodzajów ataków na systemy komputerowe. Można je podzielić na trzy kategorie [20]:

- włamania,
- blokady usług,
- kradzieże informacji.

Włamanie

Włamania są najczęstszym rodzajem ataku. Włamanie występujące w czystej postaci nie przeszkadza w dalszym korzystaniu z komputera jego uprawnionym użytkownikom. Większość włamywaczy chce używać zaatakowanego komputera tak, jak jego uprawnieni użytkownicy.

Blokada usług

Blokada usług **DoS** (*Denial of Service*) to atak, którego celem jest uniemożliwienie korzystania z zaatakowanego komputera. Na przykład napastnik włamuje się do kilku komputerów i instaluje na nich program, który wysyła nieustanne żądania do komputera będącego celem ataku DoS. Żądania te są tak dobrane, że atakowany komputer nie może nadążyć z ich obsługiwaniem, co prowadzi do zajęcia stu procent czasu jego procesora. W ten sposób dochodzi do tego, że zaatakowany komputer nie może normalnie działać. Inny rodzaj blokady usług polega na takiej zmianie konfiguracji „systemu kierowania ruchem" (na przykład DNS), że legalne żądania nie docierają do zaatakowanego komputera.

Uniknięcie takich ataków jest praktycznie niemożliwe, ponieważ w Internecie znajdują się tysiące serwerów *zombie* (opanowanych przez włamywaczy), czego administratorzy tych serwerów nie są świadomi. Wobec tego najważniejsze jest takie skonfigurowanie usług, aby

podczas gdy jedna jest atakowana (zalewana żądaniami), pozostała część systemu funkcjonowała do czasu rozwiązania problemu.

Kradzież informacji

Kradzież informacji może być poprzedzona włamaniem do komputera. Nie zawsze jest to jednak konieczne. Niektóre ataki z tej grupy wykorzystują usługi internetowe, przeznaczone do pobierania informacji, i zmuszają je do podawania większej ilości informacji, niż im wolno, albo podawania ich nieuprawnionym osobom. Wyżej wymienione rodzaje kradzieży to kradzież aktywna.

Kradzież pasywna polega natomiast na podsłuchiwaniu w oczekiwaniu, aż interesujące informacje nadejdą same. W ten sposób najłatwiej zdobyć informacje, które są przesyłane w prosty do przewidzenia sposób. Na przykład nazwa użytkownika i hasło najczęściej przekazywane są wkrótce po nawiązaniu połączenia sieciowego. Jednocześnie są to informacje bardzo istotne, bo umożliwiają dostanie się do komputera. Podsłuchiwanie sieci (*sniffing*) nie jest zadaniem trudnym ze względu na jej budowę. Na przykład w sieci Ethernet pakiety wysyłane do sieci trafiają do wszystkich przyłączonych do niej komputerów.

7.3. Rodzaje napastników

Zawsze warto znać swojego wroga. Dlatego zostaną tu omówione pokrótce najważniejsze kategorie napastników komputerowych:

- my sami,
- amatorzy wrażeń,
- wandale,
- zdobywcy,
- szpiedzy.

My sami

Jeśli nasz komputer wskutek naszego niedbalstwa zostanie zaatakowany przez wirusa i będzie go rozsiewał po sieci, to sami możemy zaliczyć się (i zostać zaliczeni) do napastników komputerowych. Najczęściej wirusy korzystają ze znanych błędów, które mogą być szybko naprawione przez zastosowanie łaty (*patch*), którą zazwyczaj można łatwo uzyskać od producenta oprogramowania.

Amatorzy wrażeń

Włamują się z ciekawości i dla rozrywki. Swoją działalność mogą traktować jako formę samokształcenia. Nie są złośliwi. Jeśli wyrządzają szkody, to nieumyślnie, próbując zatrzeć ślady. Włamują się do komputerów, które wydają się im z takich czy innych powodów ciekawe.

Wandale

Wandale mają zdecydowanie destrukcyjne intencje. Czasem jest to agresja sama w sobie, czasem działanie na zlecenie, a czasem wynik negatywnych uczuć skierowanych na przykład pod adresem konkretnej firmy, osoby lub państwa. Wandale są dość niebezpieczni i trudno się przed nimi bronić. Zatrzymanie zdeterminowanego wandala jest prawie niemożliwe. Na szczęście spotyka się ich rzadko.

Niektóre rodzaje ataków są atrakcyjne tylko dla nich, np. blokada usług raczej nie zainteresuje amatora wrażeń czy zdobywcy.

Zdobywcy

Zdobywcy są trochę podobni do amatorów wrażeń. O ile jednak ci ostatni włamują się z ciekawości i dla rozrywki, o tyle zdobywcy włamują się, ponieważ uważają włamanie za osiągnięcie, którym chcą się pochwalić. Dlatego za cele wybierają komputery w jakiś sposób wyróżniające się. Może zależeć im zarówno na liczbie, jak i na jakości zaatakowanych maszyn. Zdobywcy nie zawsze są bardzo wykwalifikowani. Często są to dzieci używające do włamywania się programów, które napisał ktoś inny i udostępnił swoje „dzieło" w Internecie. Zwykle próbują na zaatakowanej maszynie zostawić sobie drogę powrotną, żeby potem użyć jej jako platformy do kolejnych ataków.

Szpiedzy

Szpiedzy kradną informacje, które są dla nich interesujące, bo na przykład mogą je sprzedać. Wykrycie szpiega jest bardzo trudne. Kradzież informacji może nie pozostawić żadnych śladów. Można tylko postarać się, aby szpiegowanie naszego komputera było czasochłonne, kosztowne i trudne.

7.4. Usługi internetowe

Podłączenie sieci lokalnej do Internetu ma na celu udostępnienie jej użytkownikom całego zestawu usług internetowych. Żadna z nich jednak nie jest zupełnie bezpieczna, każda ma swoje słabości i niejednokrotnie została złamana przez hakerów. Każda usługa internetowa pociąga więc za sobą pewne ryzyko dla zasobów sieci lokalnej. Warto zatem zastanowić się, czy wszystkie usługi są użytkownikom niezbędne. Z drugiej strony nadmierne ograniczanie dostępności Internetu stawia pod znakiem zapytania sensowność wykonywania takiego podłączenia, powodując poza tym frustrację użytkowników. Zanim dana usługa zacznie być udostępniana użytkownikom, należy upewnić się, czy rzeczywiście jest im potrzebna, i czy można ją zabezpieczyć. Menedżerowie i administratorzy systemu wspólnie muszą zadecydować, które usługi i do jakiego stopnia powinny być udostępnione w sieci lokalnej. Jest to proces ciągły, bo zmieniają się zarówno usługi internetowe, jak i potrzeby użytkowników.

Czasem używa się pojęcia **usługa zabezpieczona**, co oznacza, że daje ona dwa rodzaje gwarancji:

- nie można użyć usługi niezgodnie z jej przeznaczeniem i/lub
- nikt nie może podejrzeć lub sfałszować transakcji.

Nie oznacza to jednak, że tej usługi można używać dowolnie i zawsze będzie bezpieczna. Można pobrać plik przez protokół scp, pozwalający na bezpieczne przesyłanie plików tak, że nikt go nie podejrzy po drodze. Ale sam protokół nie daje gwarancji, że ten plik nie zawiera złośliwego wirusa.

Oceniając zabezpieczenia usług, należy kierować się ich wpływem na bezpieczeństwo własnego środowiska w jego konfiguracji – abstrakcyjne określenia „zabezpieczenie" i „bezpieczeństwo" niewiele mówią.

Sieć WWW

Sieć WWW jest to zbiór serwerów HTTP w Internecie, natomiast HTTP jest podstawowym protokołem w WWW, dającym użytkownikom dostęp do plików, które tworzą zasoby WWW. Pliki te mogą być w różnych formatach (tekst, obrazki, dźwięk, film), ale formatem zapewniającym połączenia między plikami jest HTML (*HyperText Markup Language*). Przeglądarki internetowe zapewniają użytkownikom rozbudowany interfejs graficzny do zasobów Internetu. Niestety, zarówno przeglądarki, jak i serwery WWW trudno jest zabezpieczyć. Użyteczność WWW opiera się głównie na elastyczności, która z trudem jednak poddaje się kontrolowaniu. Przeglądarki WWW natomiast posługują się tzw. wtyczkami (*plug-ins*), które można pobrać z sieci. Trzeba jednak przy takim pobieraniu bardzo uważać, skąd ten program pochodzi, żeby nie zainstalować sobie wirusa albo programu udostępniającego innym wszystkie zasoby naszego komputera. Coraz większe zagrożenie niosą także stosowane na stronach HTML elementy aktywne. Za ich pomocą także można dokonać skutecznego ataku.

Poczta elektroniczna

Jedną z najbardziej popularnych usług sieciowych jest poczta elektroniczna. Powoduje ona coraz większe ryzyko dla zasobów sieci lokalnych, ale nie tylko. Przede wszystkim sfabrykowanie listu elektronicznego jest bardzo łatwe i może służyć naruszeniu czyjejś reputacji lub manipulowaniu ludźmi. Poza tym przyjmowanie poczty elektronicznej naraża na ataki blokowania usług.

Najgroźniejsze są jednak załączniki, mogące zawierać bardzo niebezpieczne programy, np. konie trojańskie (programy z pozoru użyteczne, a tak naprawdę wykonujące niebezpieczne operacje), robaki czy wirusy.

Grupy dyskusyjne Usenet

Ryzyko związane z grupami dyskusyjnymi jest podobne jak w przypadku poczty elektronicznej – użytkownicy mogą bezpodstawnie zaufać otrzymanej tam informacji lub opublikować swoje poufne dane. Można także zostać zalanym informacjami, środowisko więc powinno być skonfigurowane tak, żeby takie zalanie nie zakłóciło innych usług.

Transfer plików

Standardowym protokołem do transferu plików jest FTP. Potrafi go obsługiwać większość przeglądarek. Teoretycznie FTP niesie ze sobą ryzyko nie większe niż pobieranie prostych plików HTTP. W praktyce istnieje duże ryzyko pobrania koni trojańskich czy wirusów, nie mówiąc już o nielegalnym oprogramowaniu czy pornografii. Nie należy więc ufać żadnemu oprogramowaniu pobranemu przez FTP.

Zdalny dostęp

Często przydaje się możliwość uruchomienia programu na komputerze innym niż ten, na którym aktualnie pracujemy. Zawsze jednak istnieje możliwość podsłuchania takiej zdalnej sesji czy przejęcia kontroli nad zdalnym komputerem. Przez długi czas usługą zdalnego dostępu był Telnet, który przesyła wszystkie informacje w postaci jawnej (niezaszyfrowanej), wobec czego jego poziom bezpieczeństwa jest bardzo niski. Dlatego został on zastąpiony programem szyfrującym transmisję (*ssh*). Innym rozwiązaniem jest utworzenie szyfrowanego połączenia sieciowego, czyli wirtualnej sieci prywatnej (*VPN*) i używanie w niej Telnetu w zwykły sposób.

7.5. Modele ochrony sieci komputerowej

Można wymienić następujące modele ochrony sieci komputerowej:

- brak zabezpieczeń,
- zabezpieczenie przez utajnianie,
- zabezpieczenie hosta,
- zabezpieczenie sieci.

Brak zabezpieczeń

Jest tylko jedno usprawiedliwienie dla stosowania tego modelu. Otóż najcenniejsze dane umieszczamy na komputerach, które nie są podłączone do Internetu. Bardzo rzadko jednak można sobie na to pozwolić. Trzeba przy tym pamiętać, że taki komputer może być zaatakowany na przykład przez wirusa, przyniesionego przez pracownika na dyskietce.

Zabezpieczenie przez utajnianie

W tym modelu liczymy na to, że nikt nas nie zaatakuje, bo o nas nie wie. Wiele osób zakłada, że mały domowy czy nawet firmowy komputer nikogo nie zainteresuje. Stosowanie tego modelu nie ma żadnego usprawiedliwienia. Po pierwsze: napastnicy mają wiele możliwości dowiedzenia się o istnieniu naszego komputera. Po drugie: nierzadko atak przeprowadza automat

(np. wirus) i dla niego nasz komputer jest tak samo dobry jak każdy inny. Po trzecie: dla wielu napastników liczy się ilość oraz możliwość zapewnienia sobie platformy do dalszych ataków.

Zabezpieczenie hosta

Model zabezpieczenia hosta polega na tym, że każdy komputer z danej sieci jest zabezpieczany osobno. Model ten ma kilka wad. Przy dużej liczbie komputerów instalowanie i konfigurowanie zabezpieczeń może być bardzo uciążliwe. Problem ten nasila się jeszcze bardziej, jeśli różne komputery mają różne konfiguracje, lub co gorsza – systemy operacyjne. Istnieje niebezpieczeństwo, że łaty zabezpieczające nie zostaną zainstalowane na czas. Poza tym bezpieczeństwo komputerów zależy od dobrej woli i umiejętności wszystkich, którzy mają do nich uprzywilejowany dostęp. Przy zwiększeniu liczby komputerów zwiększa się również liczba tych osób. Może się też zdarzyć, że ktoś podłączy niezabezpieczony komputer do sieci bez wiedzy administratora.

Ten model bezpieczeństwa może być odpowiedni dla małych firm. Poza tym pewien poziom zabezpieczenia hostów powinien być stosowany razem z modelem zabezpieczenia sieci.

Zabezpieczenie sieci

Zabezpieczenie sieci polega na kontrolowaniu dostępu z zewnątrz do sieci komputerowej w punktach, w których sieć ta połączona jest ze światem zewnętrznym. Punktów tych jest z reguły o wiele mniej niż komputerów w sieci, co ułatwia egzekwowanie bezpieczeństwa w porównaniu z modelem zabezpieczenia hosta. Model zabezpieczenia sieci obejmuje zastosowanie firewalli do ochrony wewnętrznych systemów i sieci, używanie dobrych mechanizmów uwierzytelniających oraz szyfrowania do ochrony szczególnie cennych danych podczas ich przesyłania przez sieć.

Model zabezpieczenia sieci jest bardzo efektywny. Pojedynczy firewall może chronić bardzo wiele maszyn dołączonych do sieci. Ponieważ jednak w tym modelu kontroluje się punkty dostępu do Internetu, w bardzo dużych lub bardzo rozproszonych ośrodkach więc może się to okazać niewykonalne. Trzeba wówczas użyć zabezpieczeń warstwowych, łączących różne podejścia do bezpieczeństwa.

7.6. Firewall

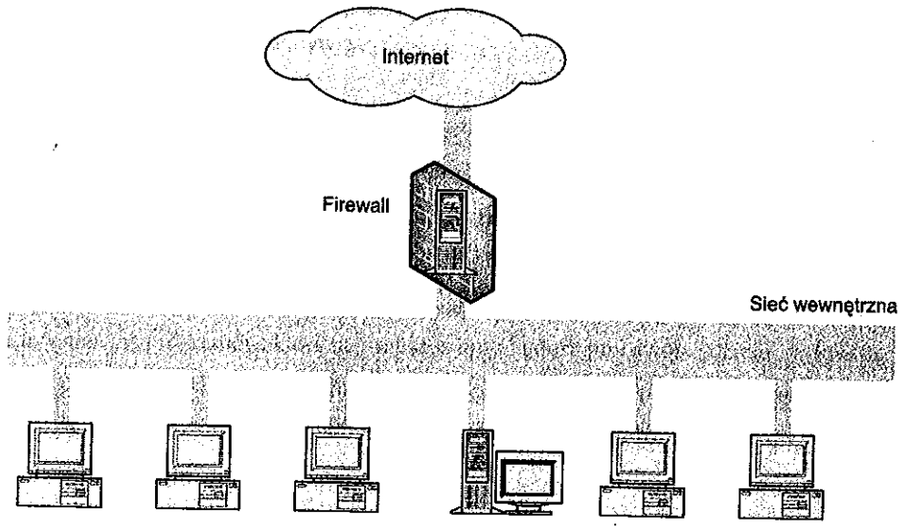
Firewall jest to urządzenie pośredniczące w przesyłaniu danych między chronioną siecią a światem zewnętrznym (może to być Internet albo np. sieć organizacji macierzystej), które ma kilka zadań:

- zmusza do wchodzenia do systemu w starannie kontrolowanym punkcie,

- zmusza do opuszczania systemu w starannie kontrolowanym punkcie,
- zapobiega zbliżaniu się napastników do chronionych maszyn,
- ogranicza możliwości przesyłania danych do takich, które uznane zostaną za dozwolone.

W praktyce więc firewall bardziej przypomina fosę w średniowiecznym zamku niż ścianę ogniową w nowoczesnym budynku (rys. 56). Ale jest to bardzo efektywne zabezpieczenie sieciowe.

Rys. 56. Firewall [20]



O tym, jakie operacje przesyłania danych są dozwolone, decyduje polityka bezpieczeństwa firmy. Firewall jest logicznym separatorem, ogranicznikiem i analizatorem.

Firewallem może być jeden komputer. wyposażony w dwie karty sieciowe, albo zestaw urządzeń – router, host lub ich kombinacja. Konieczne jest też odpowiednie oprogramowanie, które należy odpowiednio skonfigurować.

Firewall nie jest panaceum na wszystkie problemy bezpieczeństwa sieci komputerowej, ale jest najbardziej efektywnym sposobem zapewnienia tego bezpieczeństwa.

Firewall zapewnia:

- skupienie w jednym miejscu wszystkich środków bezpieczeństwa – cały ruch do i z Internetu musi przejść przez ten pojedynczy, wąski punkt, co powoduje znaczne zwiększenie bezpieczeństwa,
- egzekwowanie polityki bezpieczeństwa – firewall dopuszcza tylko dozwolone usługi i to według określonych dla nich reguł,

- efektywne rejestrowanie użycia Internetu – jest to dobre miejsce do zbierania informacji o wykorzystaniu sieci.

Firewall nie zabezpiecza jednak przed szeregiem zagrożeń:

- nie chroni przed złośliwymi użytkownikami wewnętrznymi,
- nie chroni przed połączeniami, które przez niego nie przechodzą,
- nie chroni przed całkowicie nowymi zagrożeniami, ponieważ ludzie cały czas odkrywają nowe możliwości i metody ataku,
- nie chroni całkowicie przed wirusami – podstawowe znaczenie ma tutaj instalacja na komputerach roboczych programów antywirusowych i edukacja użytkowników,
- nie potrafi sam siebie skonfigurować, a źle skonfigurowany – będzie dawał iluzoryczne poczucie bezpieczeństwa.

No i ostatnia, podstawowa wada firewalli – zakłócają pracę z Internetem, nakładając na użytkowników zbyt wiele ograniczeń.

7.7. Strategie zabezpieczeń

Strategie zabezpieczeń są stosowane do tworzenia firewalli i wymuszania bezpieczeństwa w sieci lokalnej.

Minimalne przywileje

Zasada minimalnych przywilejów (*least privilege*) jest chyba najbardziej podstawową zasadą bezpieczeństwa. Mówi ona, że każdy obiekt (np. użytkownik albo program) otrzymuje tylko te uprawnienia, które są mu niezbędne do wykonywania swoich obowiązków. Innych uprawnień nie otrzymuje. Pozwala to minimalizować szkody. Na przykład, kiedy ktoś złamie hasło zwykłego użytkownika, to nie uzyska uprawnień administracyjnych, a tylko ograniczony zestaw uprawnień tego użytkownika.

Inny przykład stosowania zasady minimalnych przywilejów to takie skonfigurowanie firewalla, aby nie pozwalał na połączenia przez Telnet (ponieważ w tym protokole hasła przesyłane są w postaci jawnej).

Większość systemów operacyjnych i urządzeń sieciowych jest dostarczana przez producentów z pominięciem zasady minimalnych przywilejów. Dlatego zadaniem administratora systemu jest takie ich skonfigurowanie, aby tej zasady przestrzegać.

Dogłębna obrona

Zasada dogłębnej obrony (*defense in depth*) mówi, że nie można polegać na jednym mechanizmie ochronnym, ale trzeba stosować wiele rodzajów zabezpieczeń, które wspierają się nawzajem na różnych poziomach. Chodzi o uniknięcie sytuacji, w której złamanie jednego rodzaju zabezpieczeń uczyni system całkowicie niezabezpieczonym. Można stosować zabezpieczenie sieci (firewall), zabezpieczenia hostów, edukację użytkowników, kopie zapasowe, rejestrowanie zdarzeń systemowych i połączeń sieciowych, skanery antywirusowe, filtry pocztowe oraz regularnie pobierać poprawki do systemu operacyjnego. Wszystkie te mechanizmy są ważne, żaden z nich samodzielnie nie wystarczy, ale zastosowane razem znacznie ograniczają ryzyko włamania. Stosowanie wielu poziomów zabezpieczeń jest bardzo istotne w przypadku awarii jednego z nich.

Wąskie przejście

Wąskie przejście (*choke point*) zmusza napastników do używania kanału, który można kontrolować i monitorować. W sieciach takim wąskim przejściem jest firewall między np. siecią lokalną a Internetem. Każdy włamywacz sieciowy musi przejść przez firewall, bo innej drogi po prostu nie ma. Dlatego firewall powinien być zabezpieczony przed atakami i umożliwiać ich wykrywanie. Zastosowanie wąskiego przejścia umożliwia skoncentrowanie obrony, co znacznie ją ułatwia. Oczywiście, jeśli chce się tym sposobem chronić sieć lokalną, to należy pozamykać wszystkie „drzwi kuchenne", czyli połączenia modemowe.

Najsłabszy punkt

Planując zabezpieczenie sieci trzeba być świadomym, co jest najsłabszym punktem zabezpieczeń, ponieważ tam najprawdopodobniej nastąpi atak. Zabezpieczenie sieci jest dokładnie tak silne jak jego najsłabszy punkt. Na przykład najlepszy firewall nie pomoże, jeśli hasło administratora będzie puste, albo nie ma sensu zabezpieczać Telnetu, jeśli FTP pozostaje niezabezpieczone (ryzyko jego używania jest również wysokie). Najsłabsze punkty należy eliminować do momentu, kiedy najsłabszy punkt obrony będzie wystarczająco silny.

Bezpieczeństwo w razie awarii

Zasada bezpieczeństwa w razie awarii mówi, że w przypadku uszkodzenia zabezpieczeń ich działanie powinno być bardziej restrykcyjne niż w przypadku poprawnej pracy. Na przykład jeśli firewall ulegnie awarii, to nie powinien w ogóle przesyłać danych, a nie przesyłać je bez kontroli.

Domyślny zakaz

Zasada domyślnego zakazu brzmi: co nie jest dozwolone, jest zabronione. Należy się nią kierować przy opracowywaniu strategii bezpieczeństwa. Zasada ta wynika z tego, że o ile można przygotować listę bezpiecznych operacji, a inne uznać za potencjalnie niebezpieczne, to nie sposób utworzyć kompletnej listy niebezpiecznych operacji, tak żeby niewymienione na niej operacje były rzeczywiście bezpieczne. To tak, jakby chcieć stworzyć listę wszystkich wirusów – wiadomo, że nie będzie ona długo aktualna. Można ją zaktualizować, ale zanim to zrobimy, będziemy znowu podatni na atak. Innym przykładem zastosowania zasady domyślnego zakazu jest takie konfigurowanie firewalla, że określamy, z jakimi portami wolno się łączyć, a pozostałych połączeń zabraniamy.

Powszechna współpraca

Powszechna współpraca jest bardzo pomocna dla zapewnienia bezpieczeństwa. Użytkownicy sieci powinni wspierać administratora w jego wysiłkach zmierzających do tego celu, a nie starać się mniej lub bardziej skutecznie obejść niewygodne zakazy. Na przykład system, który nie pozwala ustawić zbyt prostego hasła (jak imię żeńskie), może nie zorientować się, że ktoś wybrał sobie takie samo hasło jak numer rejestracyjny jego samochodu. Takie hasło jest zbyt łatwo zgadnąć. Część odpowiedzialności za bezpieczeństwo spada tu na użytkowników. Na pewno bezpieczeństwo zwiększy się także wtedy, gdy użytkownicy informują administratora, że dzieje się coś podejrzanego. Jeśli administrator nie potrafi współpracować z użytkownikami i skutecznie ich przekonywać do stosowanej strategii, to zaczyna się „wyścig zbrojeń" między nimi.

Zróżnicowana obrona

Zgodnie z tą zasadą trzeba nie tylko wielu warstw obrony, ale także różnych jej rodzajów. Najprostszym przykładem może być instalacja dwóch różnych filtrów pakietów od różnych dostawców. Oczywiście, muszą one działać jeden po drugim, a nie równolegle na różnych pakietach.

Teoretycznie zasada ta jest bardzo pomocna w zwiększaniu odporności systemu na ataki. W praktyce jednak powoduje wiele kłopotów administracyjnych i konfiguracyjnych (konieczność obsługi różnych systemów), jej wpływ na bezpieczeństwo zaś jest ograniczony, bo może się okazać, że produkty różnych firm posiadają te same jądra oprogramowania i powodują te same lub podobne błędy.

Zabezpieczanie przez utajnianie

Tę zasadę można przedstawić następująco: „jeśli coś jest ukryte, to jest bezpieczne". Bardzo łatwo jest stosować tę zasadę błędnie. Komputer podłączony do Internetu bez żadnych zabezpieczeń nie jest bezpieczny, chociaż nikt nie został poinformowany o jego podłączeniu. Nowy, tajny algorytm szyfrujący nie jest bezpieczny, dopóki nie przebadają go specjaliści od kryptoanalizy. Usługa pracująca na porcie innym niż domyślny nie jest bezpieczna, a tylko trochę schowana.

Można jednak tę zasadę stosować w sposób rzeczywiście zwiększający bezpieczeństwo. Nie powinno się nikomu udostępniać informacji o:

- dokładnej budowie i konfiguracji firewalla,
- wewnętrznych nazwach hostów i użytkowników,
- używanych rodzajach wykrywania włamań.

Im mniej tego typu informacji wydostaje się na zewnątrz, tym trudniejsze zadanie ma napastnik i tym więcej potrzebuje czasu, żeby tę ochronę złamać.

7.8. Technologie stosowane w firewallach

Podstawowe pojęcia dotyczące firewalli:

Firewall – komponent lub zestaw komponentów, które ograniczają wymianę danych między siecią chronioną a inną (najczęściej Internetem).

Host – system komputerowy podłączony do sieci.

Host bastionowy – system komputerowy, który musi być wyjątkowo dobrze zabezpieczony, ponieważ bardziej niż inne jest narażony na ataki, zwykle dlatego, że jest widoczny z Internetu i jednocześnie kontaktują się przezeń użytkownicy sieci wewnętrznej (chronionej).

Host dwusieciowy – host, który ma przynajmniej dwa interfejsy sieciowe.

Translacja adresów sieciowych NAT (*Network Address Translation*) – ma ona miejsce, kiedy router zmienia dane w pakietach, modyfikując adresy sieciowe. Dzięki temu prawdziwe adresy hostów chronionej sieci mogą pozostać ukryte. Technika ta umożliwia ponadto podłączenie do Internetu wielu hostów z użyciem mniejszej liczby przydzielonych adresów internetowych.

Filtrowanie pakietów – przesyłanie jednych pakietów między sieciami i nieprzesyłanie innych na podstawie reguł określonych w konfiguracji i na podstawie analizy treści pakietów. Czasami nazywane jest ekranowaniem (*screening*)

Sieć peryferyjna (*perimeter network*) – sieć utworzona pomiędzy siecią chronioną a zewnętrzną w celu zapewnienia dodatkowej warstwy zabezpieczeń. Zwana także strefą zdemilitaryzowaną DMZ (*demilitarized zone*).

Pośrednik (*proxy*) – program, który komunikuje się z zewnętrznymi serwerami „w imieniu" wewnętrznych klientów. Oprócz kontroli uprawnień może zapewniać przyspieszenie obsługi klientów dzięki zapamiętywaniu odpowiedzi serwerów zewnętrznych.

Wirtualna sieć prywatna VPN (*Virtual Private Network*) – sieć, w której pakiety sieci prywatnej przechodzą przez sieć publiczną – najczęściej po zaszyfrowaniu. Odbywa się to w sposób niezauważalny dla hostów sieci prywatnej (więcej na ten temat w rozdz. 8.4).

7.8.1. Filtrowanie pakietów

Filtrowanie pakietów można uznać za rozszerzenie działania zwykłego routera. Zwykły router sprawdza tylko adres docelowy danego pakietu, wybiera najlepszą ze znanych dróg i wysyła nią pakiet do celu. Albo router wie, gdzie przesłać dany pakiet i wtedy go przesyła, albo nie wie i wtedy pakietu nie wysyła.

Filtr pakietowy oprócz sprawdzenia, czy wie gdzie przesłać pakiet, sprawdza, czy wolno mu to zrobić. Sprawdzenia dokonuje na podstawie porównania ze swoją konfiguracją wielu różnych parametrów pakietu, takich jak: adres i port źródłowy, adres i port docelowy, rodzaj

protokołu. Wymienione informacje znajdują się w nagłówku pakietu, ale filtr pakietowy może także odczytać pewne informacje spoza nagłówka i odpowiednio je zinterpretować.

Przesyłanie lub blokowanie pakietów odbywa się zgodnie z polityką bezpieczeństwa, stosowaną w danej sieci, bo taka powinna być konfiguracja routera.

Bardziej zaawansowane filtry pakietowe, oprócz zignorowania „nielegalnego" pakietu mogą być tak skonfigurowane, że w przypadku wykrycia pewnego rodzaju pakietów zaalarmowany zostanie administrator i wszystkie pakiety z tego samego źródła zaczną być blokowane.

Jedną z głównych zalet filtrowania pakietów jest to, że jeden strategicznie umieszczony router może pomóc chronić całą sieć. Niestety, filtrowanie pakietów znacznie zwiększa obciążenie routera.

Usługi pośredniczenia

Pośrednicy (*proxy*) to specjalne aplikacje lub serwery, które pobierają żądania użytkowników dotyczące usług internetowych i przekazują je do właściwych usług. Służą jako bramki do tych usług. Mieszczą się pomiędzy wewnętrznym użytkownikiem a usługą na zewnątrz. Zamiast komunikować się bezpośrednio, każdy komunikuje się z pośrednikiem, ale pośrednik musi być usługą przezroczystą.

Usługa pośredniczenia wymaga dwóch komponentów: *serwera proxy*, umieszczanego zwykle w hoście z wieloma interfejsami sieciowymi, i *klienta proxy*, będącego specjalną wersją zwykłego programu klienckiego, który komunikuje się z serwerem proxy, zamiast z prawdziwym serwerem w Internecie. Serwer proxy, poza przekazywaniem żądań do prawdziwych serwerów, może także kontrolować, co którym użytkownikom wolno zrobić. Zależnie od przyjętej polityki bezpieczeństwa może przyjąć żądanie lub odmówić.

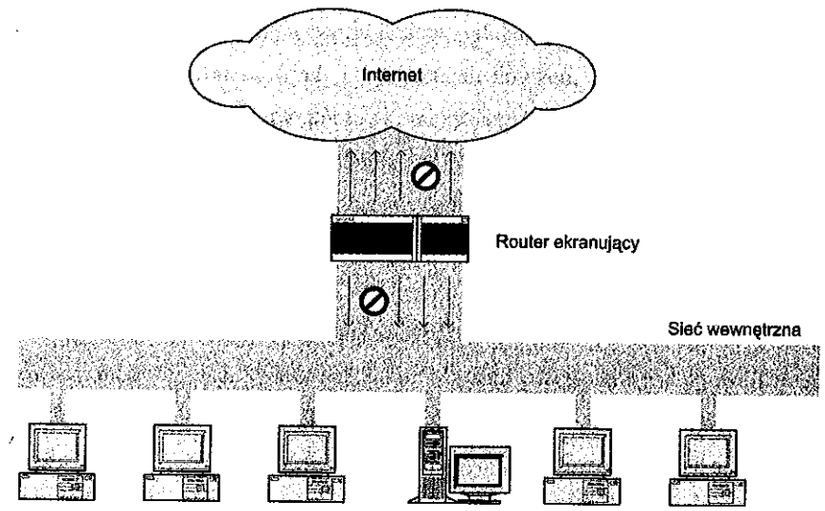
7.9. Architektura firewalli

7.9.1. W jednym pudełku

Najprostsza architektura to pojedynczy obiekt (*single-box architecture*), działający jako firewall. Zaletą jego jest to, że wszystkie elementy zabezpieczające są umieszczone w jednym miejscu, łatwiej więc się upewnić, czy zostały poprawnie skonfigurowane. Wadą natomiast – że od tego jednego miejsca zależy całe bezpieczeństwo sieci lokalnej. Nie ma tu wielowarstwowych zabezpieczeń, ale łatwiej znaleźć np. najsłabsze ogniwo. Zwykle o zastosowaniu takiej architektury decydują względy praktyczne – jest ona tańsza i łatwiej zrozumiała, jest też dobra do zabezpieczenia małych sieci.

Stosowane są dwa rodzaje tego typu architektury: **router osłaniający** (inaczej *ekranujący*) oraz **host dwusieciowy**.

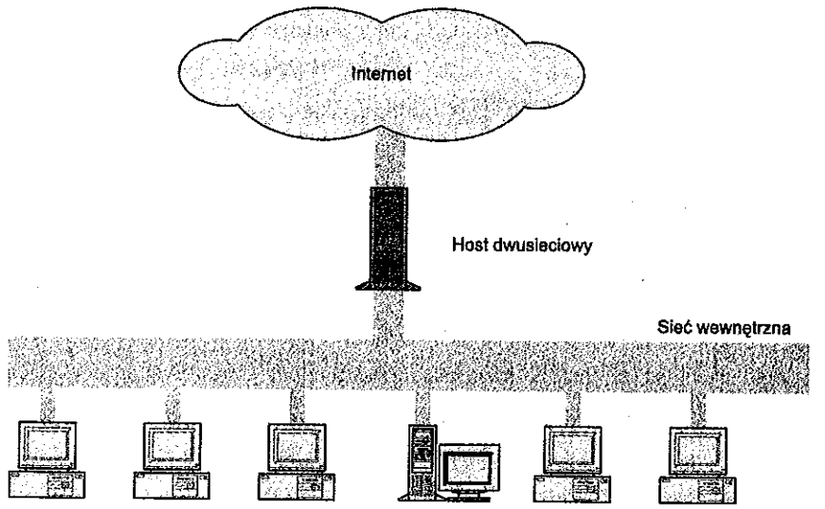
Rys. 57. Router osłaniający [20]



W architekturze z routerem osłaniającym jako firewall jest używany sam router filtrujący pakiety (rys. 57). Jest to rozwiązanie tanie, ale niezbyt elastyczne (np. można tylko zezwolić lub nie na używanie danego protokołu, ale nie można pozwolić na niektóre operacje, ale zabraniać innych w ramach jednego protokołu). Jeżeli router zostanie złamany, to nie ma dalszych zabezpieczeń.

Tak chroniona sieć musi mieć dobrze zabezpieczone hosty, a używane protokoły powinny być ograniczone do prostych.

Rys. 58. Host dwusieciowy [20]



W architekturze z hostem dwusieciowym jako firewall służy komputer, który ma przynajmniej dwa interfejsy sieciowe i jest umieszczony pomiędzy Internetem a siecią wewnętrzną (rys. 58). Trzeba w nim wyłączyć funkcję trasowania, aby pakiety IP z jednej sieci (np. z Internetu) nie były bezpośrednio przesyłane do drugiej (wewnętrznej, chronionej). Systemy chronione takim firewallem mogą się komunikować tylko z nim. Tak samo tylko z nim mogą się komunikować systemy wewnętrzne.

Zabezpieczenia takiego hosta muszą być bardzo dobre, bo jego opanowanie daje pełny dostęp do sieci wewnętrznej. Jeśli napastnik zawiesi taki host, to odcina sieć wewnętrzną od Internetu.

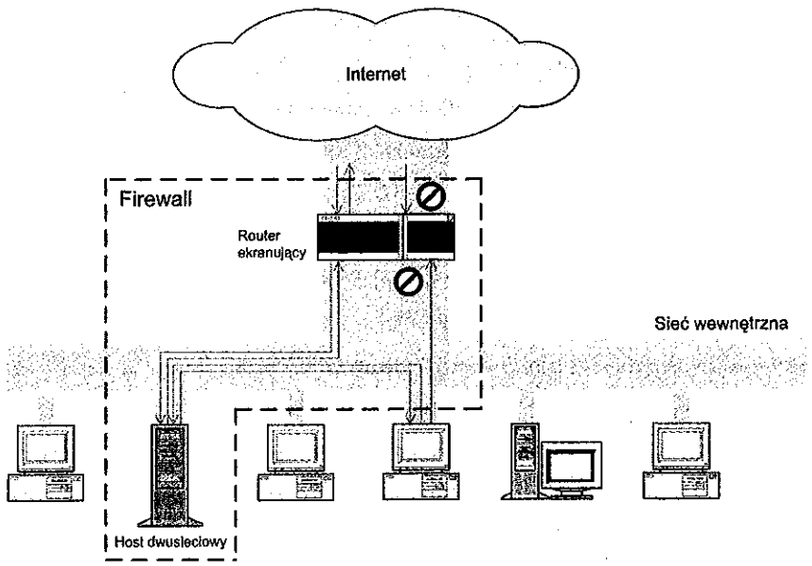
Host dwusieciowy może świadczyć usługi tylko przez pośredniczenie. Możliwe jest też logowanie się bezpośrednio na nim użytkowników, ale nie jest to bezpieczna technika. Pośredniczenie dużo lepiej współpracuje z usługami wychodzącymi niż z przychodzącymi, dlatego taka architektura nie jest odpowiednia dla firm oferujących usługi w Internecie.

Wiele firewalli typu „w jednym pudełku" oferuje pewną kombinację usług ekranowania i pośredniczenia, ale cały czas pozostaje problemem to, że jest to jedyne zabezpieczenie sieci wewnętrznej. Te urządzenia także nie są polecane tym, którzy udostępniają usługi w Internecie.

7.9.2. Architektura z ekranowanym hostem

Firewall zbudowany w architekturze z ekranowanym hostem (*screened host*) składa się z hosta przyłączonego do sieci wewnętrznej, zwanego hostem bastionowym i z oddzielnego routera (rys. 59).

Rys. 59. Firewall w architekturze z ekranowanym hostem [20]



Głównym zabezpieczeniem jest tutaj filtrowanie pakietów (zabezpiecza np. przed obchodzeniem serwerów proxy i tworzeniem bezpośrednich połączeń). Filtr pakietów jest tak skonfigurowany, aby host bastionowy był jedynym hostem w sieci wewnętrznej, z którym mogą się łączyć hosty z Internetu. Dozwolone typy połączeń są określone w polityce bezpieczeństwa.

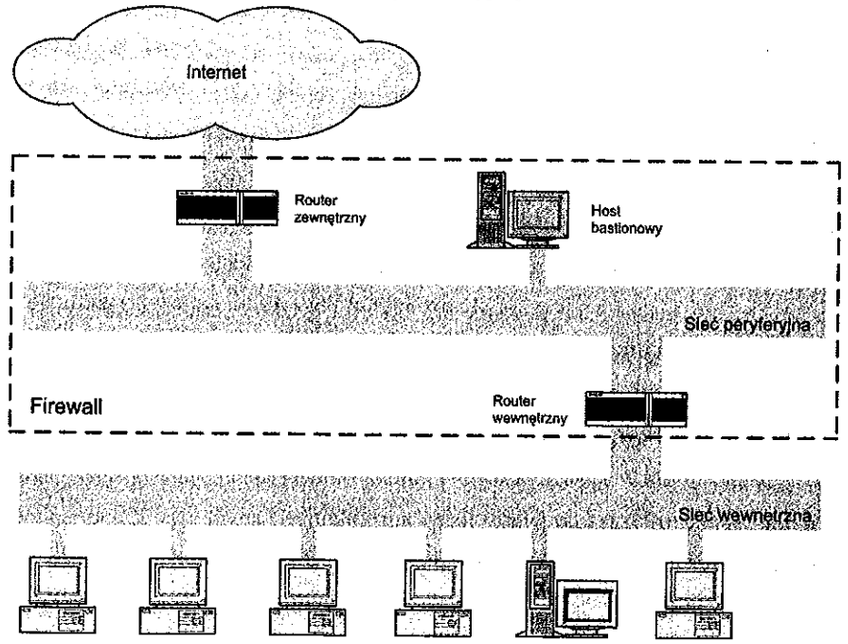
Router ekranujący zwykle udostępnia część usług bezpośrednio (przez filtr pakietów), a część blokuje, zmuszając do korzystania z systemu pośredniczącego.

W większości zastosowań taka architektura zapewnia lepsze zabezpieczenie i większą użyteczność niż architektura z hostem wielosieciowym.

7.9.3. Architektura ekranowanej podsieci

Najbardziej zaawansowana architektura firewalla to architektura z ekranowaną podsiecią. Polega ona na stworzeniu dodatkowej sieci peryferyjnej (*perimeter network*), izolującej sieć wewnętrzną od Internetu i będącej dodatkową warstwą zabezpieczeń.

Rys. 60. Firewall w architekturze z ekranowaną podsiecią [20]



W najprostszym przypadku są to dwa routery ekranujące (rys. 60): jeden łączący sieć wewnętrzną z siecią peryferyjną, drugi zaś sieć peryferyjną z Internetem. Host bastionowy umieszcza się pomiędzy nimi, czyli w sieci peryferyjnej. Większość jego czynności polega na działaniu jako serwer proxy dla różnych usług.

Aby włamać się do sieci wewnętrznej napastnik musi przejść oba routery. Nie ma tu więc jednego słabego miejsca, narażającego całą sieć.

Architekturę firewalla z ekranowaną podsiecią można w zależności od potrzeb dalej rozbudowywać, dodając kolejne podsieci, zwiększając liczbę hostów bastionowych, umieszczając w niej serwery usług internetowych itd. Zależy to od różnych zdarzeń zachodzących w sieci zewnętrznej i mających wpływ na sieć wewnętrzną. Taka architektura jest odpowiednia dla sieci wymagających wysokiego poziomu bezpieczeństwa i wielowarstwowej ochrony.

7.10. Polityka bezpieczeństwa

Polityka bezpieczeństwa jest to dokument obowiązujący w firmie [20], który określa na najwyższym poziomie najważniejsze decyzje dotyczące tego, co i jak należy chronić. Jego przyjęcie pozwala na wyznaczenie ogólnych kierunków zapewniania bezpieczeństwa i podjęcie na odpowiednio wysokich szczeblach kluczowych decyzji co do kompromisów, jakie w dziedzinie bezpieczeństwa są nieuchronne. Polityka to sposób porozumiewania się specjalisty od bezpieczeństwa komputerowego z użytkownikami i z kierownictwem. Powinna ich informować o wszystkim, co powinni wiedzieć, aby podjąć odpowiednie decyzje dotyczące bezpieczeństwa. Jest ono bowiem zbyt ważną kwestią, aby pozostawić je doraźnym czy nieuporządkowanym działaniom.

Większość ekspertów uważa, że pojedyncza, zunifikowana i opublikowana polityka bezpieczeństwa jest zasadniczo pożądana, ale zarazem wie – na podstawie własnych doświadczeń – że opracowanie dobrej polityki będzie bardzo trudne. Na pewno jednak się opłaci, bo poświęcony na to czas będzie zdecydowanie krótszy niż czas, który trzeba byłoby poświęcić na usuwanie skutków braku takiej polityki.

Polityka nie może zakładać bezpieczeństwa za wszelką cenę, bo prowadzi to do nieuzasadnionych wydatków albo do ograniczeń użytkowania, których nie można zaakceptować. Polityka powinna zalecać zabezpieczenia, które są do zaakceptowania pod względem ceny i ograniczeń, jakie nakładają na użytkowników.

Co powinna zawierać polityka bezpieczeństwa

Wyjaśnienia – jest ważne, aby polityka w sposób zrozumiały dla wszystkich zainteresowanych uzasadniała sens stosowania danych środków bezpieczeństwa. Po pierwsze większość ludzi nie stosuje się do zasad, jeśli nie wie po co. Po drugie istnieje potrzeba udokumentowania przyjętych rozwiązań, na przykład na wypadek odejścia z firmy ich twórców.

Odpowiedzialność wszystkich zainteresowanych – najczęściej ceną za bezpieczeństwo jest nałożenie obowiązków i ograniczeń nie tylko na administratorów sieci, ale i na zwykłych użytkowników. Wszyscy powinni zdawać sobie sprawę, że nie tylko oni podlegają ograniczeniom. Dzięki temu łatwiej będzie im poddać się stosownym rygorom.

Zwykły język – polityka powinna być zapisana językiem zrozumiałym dla wszystkich, których ona obowiązuje. Nie jest naprawdę konieczne używanie zaawansowanych pojęć technicznych ani prawniczego języka. Niezrozumiała polityka na pewno nie będzie realizowana.

Władza wykonawcza – polityka powinna określać, kto jest odpowiedzialny za poszczególne jej aspekty, i przyznawać tym osobom odpowiednie uprawnienia. Na przykład, w polityce może znaleźć się zapis, który mówi, że administratorom pewnych usług przyznaje się prawo do blokowania dostępu użytkownikom wykonującym nielegalne operacje.

Uwzględnianie wyjątków – polityka powinna określać, w jakim trybie będą rozpatrywane sprawy proponowanych wyjątków od reguł zapisanych w polityce. Na przykład, czy jakaś osoba albo komisja ma prawo zezwolić na taki wyjątek.

Uwzględnianie rewizji – polityka bezpieczeństwa powinna zmieniać się wraz z firmą. Inne są potrzeby bezpieczeństwa, kiedy zatrudnia się 10 osób, inne – kiedy 100, a jeszcze inne – kiedy firma udostępnia pewne usługi sieciowe swoim kontrahentom albo ma wiele oddziałów połączonych wirtualną siecią prywatną.

Omówienie konkretnych kwestii bezpieczeństwa – lista konkretnych kwestii w dużym stopniu zależy od danej firmy. Można tu podać kilka przykładów:

- czy kilka osób może używać tego samego konta,
- co należy zrobić, zanim podłączy się komputer do sieci,
- jakie warunki powinny spełniać hasła,
- jak chronić dane osobowe,
- jak pracownicy podróżujący w interesach bądź pracujący w domu mogą uzyskać dostęp do sieci.

Czego nie powinna zawierać polityka bezpieczeństwa

Szczegóły techniczne – opisywanie w polityce bezpieczeństwa szczegółów technicznych uniemożliwia zrozumienie jej przez wszystkich. Ponadto polityka wyznacza raczej kierunek, a konkretne rozwiązanie techniczne może zależeć na przykład od aktualnie dostępnych produktów i ich cen.

Problemy innych – polityki bezpieczeństwa dla jednej sieci nie da się skopiować z polityki dla innej sieci, ponieważ różne sieci mają różną budowę, różne uwarunkowania biznesowe i innych użytkowników.

Problemy niezwiązane z bezpieczeństwem komputerów – polityka bezpieczeństwa nie musi i nie powinna regulować kwestii, które do niej nie należą, czyli takich, które nie mają bezpośredniego związku z bezpieczeństwem zasobów sieci komputerowej.

7.11. Reagowanie na incydenty

Niektóre organizacje zaczynają myśleć o reagowaniu na incydenty dopiero po włamaniu. Powinno się jednak mieć opracowany plan postępowania na wypadek naruszenia bezpieczeństwa w sieci i należy opracować taki plan jeszcze przed wystąpieniem jakiegokolwiek incydentu.

Podstawowe czynności reagowania na incydent są następujące:

- ocena sytuacji,
- dokumentowanie przebiegu zdarzeń,
- odłączenie komputerów od sieci (lub ich wyłączenie),
- dokonanie analizy sytuacji i podjęcie odpowiednich działań,
- powiadamianie o „incydencie w toku",
- sporządzenie kopii systemu,
- odtworzenie danych i przywrócenie systemu do zwykłego stanu.

Kolejność wykonania powyższych czynności zależy od sytuacji (niektóre mogą być w danej sytuacji nieadekwatne).

Ocena sytuacji

Sytuację należy ocenić na podstawie odpowiedzi na co najmniej dwa pytania:

1. Czy napastnik zdołał się dostać do systemu?
2. Czy atak jest w toku?

Jeśli atak wygląda na agresywny, może być konieczne nawet odcięcie systemu od Internetu. Jeśli jednak incydent nie wygląda groźnie, np. ktoś przez Telnet wypróbowuje różne kombinacje nazw użytkowników i haseł, jest trochę czasu na podjęcie decyzji; można spokojnie poobserwować działania napastnika. Podstawowa reguła podczas wykonywania tych czynności brzmi: nie panikować!

Dokumentowanie przebiegu zdarzeń

Nie muszą to być żadne wymyślne raporty, po prostu notatki o kolejnych czynnościach z podanym bieżącym czasem ich wykonania.

Odłączenie komputerów od sieci

Po dokonaniu oceny sytuacji bardzo ważną czynnością jest zapewnienie sobie czasu na reakcję bez dalszego narażania swojego systemu komputerowego. Najmniej radykalnym posunięciem jest odcięcie atakowanego komputera od wszystkich sieci. Jeśli inne maszyny są podatne na ten sam rodzaj ataku albo zachodzi podejrzenie, że już padły łupem napastnika, to należy także je wszystkie odłączyć od sieci. Może to prowadzić do odcięcia połączenia z Internetem, ale pozwoli uporać się z ewentualnymi zniszczeniami i zabezpieczyć przed dalszą wrogą działalnością.

W niektórych sytuacjach należy wyłączyć cały system, ale powinno to być robione tylko w ostateczności, bo może zniszczyć potrzebne informacje.

Analiza i podjęcie odpowiedniego działania

Po pierwsze należy działać możliwie spokojnie i z dużym namysłem, żeby nie zrobić dodatkowych szkód. Ryzyko pomyłki można zmniejszyć pracując z partnerem, każdy sprawdza polecenia drugiego po wpisaniu, a przed ich wykonaniem. Pracując samemu, można czytać polecenia na głos i sprawdzać dwa razy wszystkie argumenty.

Użytkownicy nie powinni przeszkadzać w tym czasie osobom naprawiającym system. Można oddelegować kogoś tylko do odpowiadania na pytania użytkowników. Ponadto powinien być bardzo jasny podział zadań w zespole, żeby nie wchodzić sobie nawzajem w drogę.

Powiadamianie o „incydencie w toku"

Zwykle należy powiadamiać kierownictwo i pozostały personel techniczny o zaistniałej sytuacji. Inaczej istnieje ryzyko, że będą oni nieświadomie działali przeciwko naprawiającym, np. podłączając „niedziałający Internet". Poza tym kierownictwo może uznać z jakichś powodów, że przewidywane szkody są mniej istotne niż utrata połączenia z Internetem. Warto także powiadamiać użytkowników, żeby wiedzieli, dlaczego muszą znosić niewygody. Czasami z różnych względów należy powiadamiać inne działy firmy, np. prawny, nadzoru czy public relations.

Sporządzenie kopii systemu

Powinna zostać sporządzona kopia zapasowa każdego systemu, który padł łupem napastnika. Przede wszystkim, jeśli w czasie naprawy zostanie popełniony jednak jakiś błąd, to można będzie powtórzyć cały proces. Kopie mogą służyć również jako materiał dowodowy w ewentualnym dochodzeniu. Mogą też zostać na miejscu zbadane w celu dokładnego sprawdzenia, co się stało i ewentualnie – dlaczego.

Odtworzenie danych i przywrócenie stanu sprzed ataku

1. Jeśli napastnik nie zdołał wtargnąć do systemu, jest szansa, że nie ma wiele do zrobienia. Może się okazać, że atak był niegroźny albo że nie warto reagować na niedbałe próby włamania.
2. Jeśli atak był szczególnie zaciekły, należy na jakiś czas nasilić monitorowanie.
3. Jeśli napastnik rzeczywiście włamał się do systemu, to należy uszczelnić lukę w zabezpieczeniach, przez którą się przedostał. Następnie należy upewnić się, czy nic nie zostało zniszczone i czy napastnik nic po sobie nie zostawił. Większość napastników zaczyna od zapewnienia sobie ponownego wejścia do systemu. Może to oznaczać, że system należy odbudować od podstaw z pakietów instalacyjnych i kopii zapasowych. Przed tym należy sprawdzić, czy sprzęt działa bez zarzutu.

Zadania po incydencie

1. Ustalenie, co się stało i jak zapobiegać temu w przyszłości.
2. Analiza własnej reakcji na incydent, co zostało zrobione dobrze, a co źle, jakie inne narzędzia i zasoby byłyby pomocne, jak reagować lepiej następnym razem itd.
3. Poinformowanie o zakończeniu akcji wszystkich, którzy zostali powiadomieni o „incydencie w toku".

8. Metody i narzędzia bezpiecznej komunikacji elektronicznej w sieciach komputerowych

W poprzednim rozdziale omówione zostały zagrożenia dla danych przechowywanych w sieciach lokalnych organizacji oraz sposoby zabezpieczania się przed nimi, a tym samym sposoby chronienia zasobów firm i innych organizacji. Obecnie jednak komunikacja pomiędzy komputerami nie może ograniczać się tylko do jednej sieci lokalnej. Głównym medium komunikacyjnym jest Internet. Ze swojej definicji jednak jest on medium otwartym. Nie potrafi zapewnić, że nasze dane i informacje dotrą do odbiorcy bezpieczne ani że osoba próbująca dostać się zdalnie do zasobów sieciowych jest do tego rzeczywiście uprawniona. Dlatego opracowano różne rodzaje zabezpieczeń komunikacji w Internecie. W niniejszym rozdziale zostaną omówione następujące zagadnienia związane z tą problematyką:

- kryptografia – zabezpieczająca dane przed odczytaniem przez niepowołane osoby,
- system uwierzytelniający Kerberos – zapewniający uwierzytelnianie użytkowników systemu informatycznego na odpowiednio wysokim poziomie bezpieczeństwa,
- protokoły pośrednie – pozwalające korzystać m.in. z kryptografii do zabezpieczania przesyłanych danych na różnych poziomach komunikacji sieciowej,
- wirtualne sieci prywatne – zabezpieczające komunikację wewnętrzną w firmach wielooddziałowych.

8.1. Kryptografia

Bardzo szybki rozwój Internetu nastąpił po części dzięki możliwości udostępniania na odległość wielu różnych usług: zakupy i bankowość to tylko podstawowe przykłady. Wykonywane operacje wymagają jednak często przesyłania danych poufnych, jak numer karty kredytowej. Korporacje wymieniają pocztą elektroniczną tajne dane dotyczące prowadzonych przedsięwzięć. Internetu do przesyłania danych używa wojsko i wszystkie służby wywiadowcze.

Podstawowa struktura Internetu właściwie uniemożliwia zagwarantowanie, że wysyłanej przez nas informacji nikt poza adresatem nie przeczyta. Po drodze przebywa ona w pamięci kolejnych komputerów, na których pełny dostęp do przesyłanych danych ma administrator.

Istnieje też wiele różnych sposobów na przechwycenie czy podsłuchanie przepływającej informacji i techniki te są stale rozwijane.

Przed tymi zagrożeniami możemy chronić dane, korzystając z dostępnych narzędzi kryptograficznych. Efektywne narzędzia mogą się okazać zaskakująco tanie, choć są też rozwiązania drogie i bardzo drogie.

8.1.1. Definicje

Kryptografia – nazwa ta wywodzi się od greckiego słowa „kryptos", oznaczającego „ukryty" [34]. Jej celem jest więc ukrywanie informacji, tak aby mógł ją poznać tylko uprawniony odbiorca. Ukrywanie nosi miano szyfrowania, a ujawnianie – deszyfrowania. W tym celu wykorzystywany jest szyfr. Przed zaszyfrowaniem informacja nosi nazwę tekstu jawnego, a po zaszyfrowaniu nazywana jest szyfrogramem. Szyfrowanie wiadomości przez nadawcę ma na celu uniknięcie jej podejrzenia podczas transportu. Po odebraniu jej przez odbiorcę jest ona deszyfrowana za pomocą odpowiedniego klucza do postaci tekstu jawnego. Algorytmy szyfrujące zwykle opierają się na wyższej matematyce.

Kryptologia to nauka o pismach szyfrowanych, a **kryptoanaliza** – to dział kryptologii, zajmujący się łamaniem szyfrów.

Innymi słowy kryptografia jest sztuką skutecznej i bezpiecznej wymiany informacji. Bezpieczeństwo oznacza tu precyzyjnie sformułowane wymaganie co do trudności, jaką będzie miał przeciwnik, aby skutecznie przeprowadzić atak na dany kryptosystem. Pojęcie to jest zawsze względne. Należy sprecyzować, jakie środki ma przeciwnik do dyspozycji, jakiego rodzaju dostęp ma przeciwnik do systemu, a także, jaki jest cel ataku.

Zadania kryptografii są następujące [34]:

Poufność (*confidentiality*) to problematyka przesyłania wiadomości w obecności podsłuchu i pomimo niego. Posługuje się takimi metodami jak kryptografia symetryczna, kryptografia asymetryczna, zagadnienia doboru klucza, generacja liczb losowych (a właściwie – pseudolosowych).

Uwierzytelnienie (*authentication*) to tematyka związana z pojęciem dowodu. Mamy tu do czynienia z kontrolą dostępu i identyfikacją (*identification*) – należy dowieść, że jest się osobą uprawnioną do wejścia do danego systemu oraz z autentyzacją (*authentication*), pozwalającą zagwarantować, że dana osoba, a nie kto inny, jest autorem wiadomości. Kolejnym tematem jest podpis cyfrowy, który także gwarantuje pochodzenie wiadomości. Powinien on przekonać w sposób niezbity, np. sąd, o swojej autentyczności. Algorytm sprawdzania podpisu jest publiczny, ale wymaga się, żeby nikt inny (poza

właścicielem podpisu) nie był w stanie wygenerować ważnej pary „wiadomość, podpis". Nazywa się to niezaprzeczalnością (*non repudiation*).

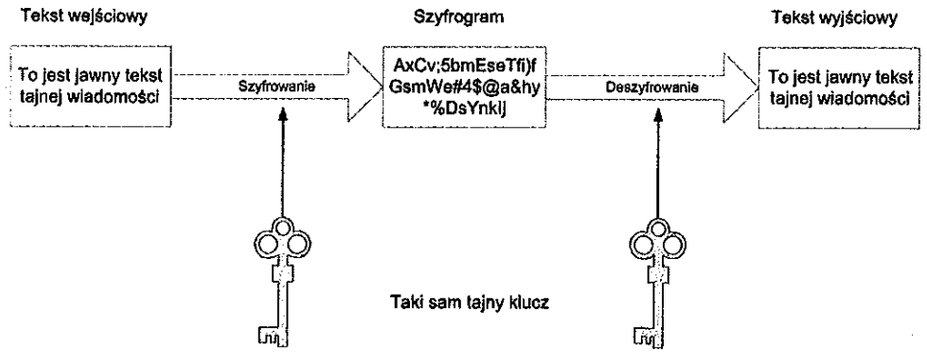
Integralność (*integrity*) pozwala wykryć modyfikację danych, albo wręcz taką modyfikację uniemożliwia. Wykorzystuje się tu funkcje jednokierunkowe i kryptograficzne funkcje mieszające (*hash functions*).

Zdecydowanie zwiększa bezpieczeństwo korzystanie ze znanych i sprawdzonych algorytmów szyfrowania. Należy unikać ofert proponujących superbezpieczny i supertajny algorytm szyfrujący. Najbezpieczniejsze jest wybranie algorytmu szyfrującego jawnego i od dłuższego czasu używanego, systemu, który już był atakowany przez hakerów.

8.1.2. Szyfrowanie symetryczne

Szyfrowanie symetryczne (kryptografia symetryczna) to taki rodzaj szyfrowania, w którym tekst jawny ulega przekształceniu w tekst zaszyfrowany za pomocą pewnego klucza, natomiast do odszyfrowania niezbędna jest znajomość tego samego klucza oraz algorytmu zastosowanego do szyfrowania (rys. 61). Operacje szyfrowania i deszyfrowania są takie same.

Rys. 61. Szyfrowanie symetryczne [11]



Bezpieczeństwo takiego szyfrowania zależy od kilku czynników. Po pierwsze, algorytm użyty do szyfrowania powinien czynić trudnym rozszyfrowanie kryptogramu bez znajomości klucza użytego do szyfrowania. Po drugie, bezpieczeństwo szyfrowania symetrycznego zależy od tajności klucza. Oznacza to, że algorytm może być jawny, jedynie klucz musi być bezpiecznie ukryty.

Najstarsze przykłady szyfrów symetrycznych, których użycie sięga bardzo dawnych czasów, to szyfry *podstawieniowe* i *przestawieniowe* [37]. Musiały one umożliwiać wykonanie szyfrowania i deszyfrowania przez człowieka, a więc opierać się na bardzo prostych operacjach.

Szyfry **podstawieniowe** to szyfry, których działanie polega na podstawianiu pod poszczególne znaki tekstu jawnego znaków alfabetu szyfrowego. Najstarszym przykładem takiej metody jest *szyfr Cezara*, używany przez Gajusza Juliusza Cezara: szyfrogram uzyskuje się przez zastąpienie każdej litery alfabetu literą leżącą o trzy miejsca dalej. Oznacza to, że w tekście jawnym każdą literę *a* należy zastąpić przez *d*, każdą literę *b* należy zastąpić przez *e* itd. Zatem wynikiem zaszyfrowania słowa

KRYPTOGRAFIA

będzie słowo

NUBTYSJUDILD.

Kluczem jest w tej metodzie liczba definiująca przesunięcie, czyli 3. Deszyfrowanie polega na wykonaniu zastępowania liter szyfrogramu literami leżącymi o trzy miejsca bliżej.

Szyfry **przestawieniowe** (permutacyjne) charakteryzują się tym, że w tekście zaszyfrowanym występują wszystkie znaki tekstu jawnego, tylko w innej kolejności. Szyfry należące do tej grupy zmieniają kolejność liter w tekście według określonego schematu. Najczęściej przestawienia liter dokonuje się za pomocą figury geometrycznej. Przykładem takiej metody jest *szyfr płotkowy*, polegający na tym, że z tekstu jawnego tworzy się „płotek" o zadanej głębokości:

```
K     T     A
  R P O R F A
    Y     G     I
```

Głębokość płotka jest kluczem w tej metodzie. Aby otrzymać szyfrogram należy tekst „płotka" odczytać wierszami:

KTARPORFAYGI.

Zaletą algorytmów szyfrowania symetrycznego jest to, że są proste i szybkie. Mogą nawet być realizowane sprzętowo. Ich podstawową wadą jednak jest konieczność przekazania klucza odbiorcy w sposób bardzo bezpieczny, bo musi on pozostać tajny.

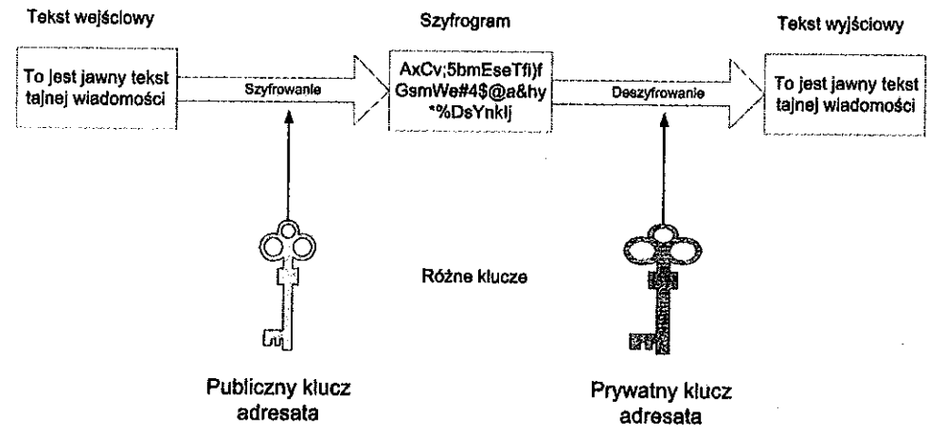
Najbardziej znanym algorytmem szyfrowania symetrycznego jest DES (*Data Encryption Standard*). Opis algorytmu szyfrującego został opublikowany. DES jest szyfrem blokowym, dane są szyfrowane blokami 64-bitowymi, przy użyciu 56-bitowego klucza. W wyniku szyfrowania otrzymuje się 64-bitowe bloki danych. Niestety, wadą tego algorytmu jest zbyt krótki klucz, co znacznie obniża jego odporność na ataki. Znaczną poprawę bezpieczeństwa można osiągnąć, stosując algorytm 3DES, będący odmianą algorytmu DES z trzykrotnym powtórzeniem procesu szyfrowania na poprzednich wynikach z różnymi kluczami. Inne algorytmy to: IDEA (*International Data Encryption Standard*), RC2, RC5, AES (*Advanced Encryption Standard*).

8.1.3. Szyfrowanie asymetryczne

Szyfrowanie asymetryczne (kryptografia asymetryczna) to taki rodzaj szyfrowania, w którym podczas operacji szyfrowania i deszyfrowania używane są dwa różne klucze: klucz szyfrujący i klucz deszyfrujący. Klucze te stanowią nierozłączną parę. Nie istnieje jednak sposób, aby na podstawie jednego klucza odtworzyć drugi. Ponadto znajomość klucza szyfrującego nie pomoże w odszyfrowaniu wiadomości.

W celu otrzymywania zaszyfrowanych wiadomości należy zatem utworzyć parę kluczy (czyli dwa). Klucz szyfrujący staje się **kluczem publicznym**, dostępnym dla każdego, kto chce przysłać zaszyfrowaną wiadomość. Oznacza to, że dowolna osoba może przysłać taką wiadomość. Klucz deszyfrujący staje się **kluczem prywatnym** i powinien być bardzo dobrze chroniony. Ponieważ klucz prywatny jest w posiadaniu adresata, tylko on może przeczytać zaszyfrowaną wiadomość.

Rys. 62. Szyfrowanie asymetryczne [11]



Wadami algorytmów szyfrowania asymetrycznego jest ich wysoki stopień skomplikowania, a co za tym idzie, znaczna powolność (nawet tysiąc razy wolniejsze niż symetryczne). Zwykle też klucze tych algorytmów są dłuższe niż klucze algorytmów symetrycznych o podobnym poziomie bezpieczeństwa. Rozwiązują one jednak problem przekazywania tajnego klucza, służącego do szyfrowania i deszyfrowania symetrycznego.

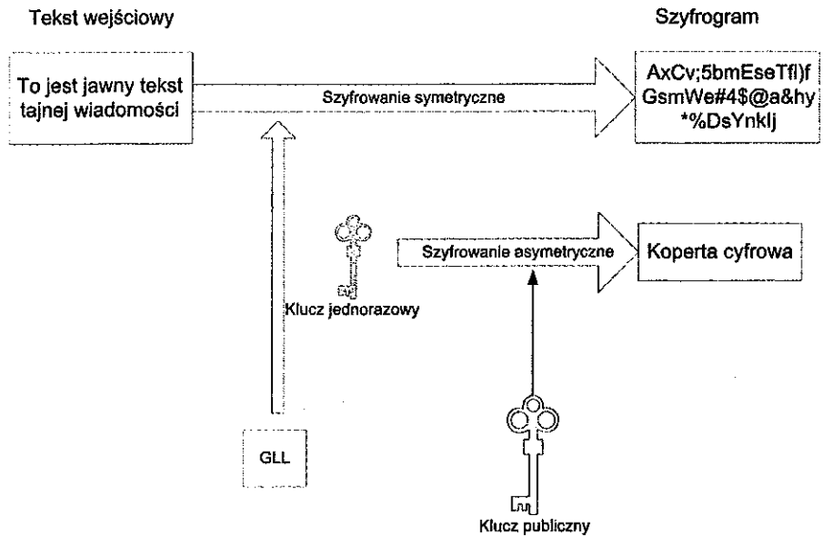
Najbardziej znanym szyfrem asymetrycznym jest RSA, algorytm opracowany w 1977 r. przez trzech matematyków, profesorów MIT (*Massachusetts Institute of Technology*) i nazwany od pierwszych liter ich nazwisk. Algorytmy RSA, mające klucze długości do 512 bitów, są stosunkowo proste do złamania. Obecnie stosuje się 1024, 2048 lub nawet 4096-bitowe klucze [11]. Algorytm ten, aby był skuteczny, musi mieć klucze będące liczbami pierwszymi.

RSA może być wykorzystywany do zarówno do szyfrowania, jak i do realizacji algorytmów podpisów cyfrowych.

8.1.4. Szyfrowanie hybrydowe

Ze względu na bardzo długi czas wykonania szyfrowania asymetrycznego w zastosowaniach praktycznych stosuje się szybsze algorytmy hybrydowe, czyli metody łączące szyfrowanie symetryczne i asymetryczne (rys. 63).

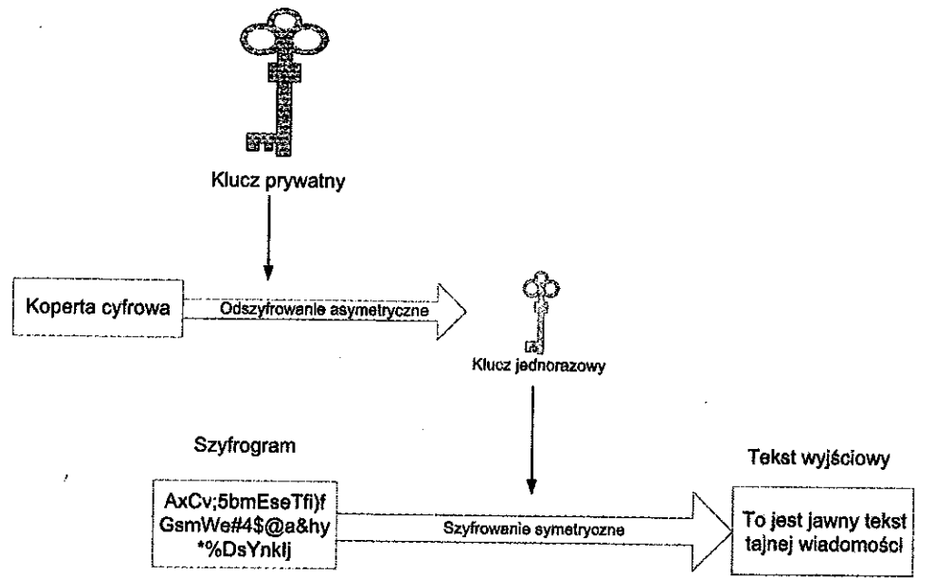
Rys. 63. Szyfrowanie hybrydowe [11]



Wiadomość jest szyfrowana algorytmem symetrycznym za pomocą jednorazowego klucza. Jako jednorazowy klucz wykorzystywana jest liczba losowa, generowana przez generator liczb losowych **GLL** tuż przed rozpoczęciem szyfrowania. Następnie ten jednorazowy klucz jest szyfrowany algorytmem asymetrycznym za pomocą klucza publicznego adresata, tworząc tzw. **kopertę cyfrową**, która wraz z szyfrogramem jest wysyłana do adresata. Dzięki temu rozwiązany zostaje problem długiego czasu trwania szyfrowania asymetrycznego (tą metodą jest szyfrowany tylko klucz) oraz problem przekazywania tajnego klucza, używanego w szyfrowaniu symetrycznym. Newralgicznym elementem tego sposobu szyfrowania jest generator liczb losowych: im lepszy (tzn. dający bardziej losowe wyniki), tym szyfrowanie bezpieczniejsze.

U adresata następuje proces odwrotny. Najpierw, algorytmem asymetrycznym za pomocą klucza prywatnego odbiorcy, odszyfrowywana jest *koperta cyfrowa*. Następnie za pomocą otrzymanego w ten sposób klucza jednorazowego odszyfrowywana jest algorytmem symetrycznym sama wiadomość (rys. 64).

Rys. 64. Deszyfrowanie hybrydowe [11]

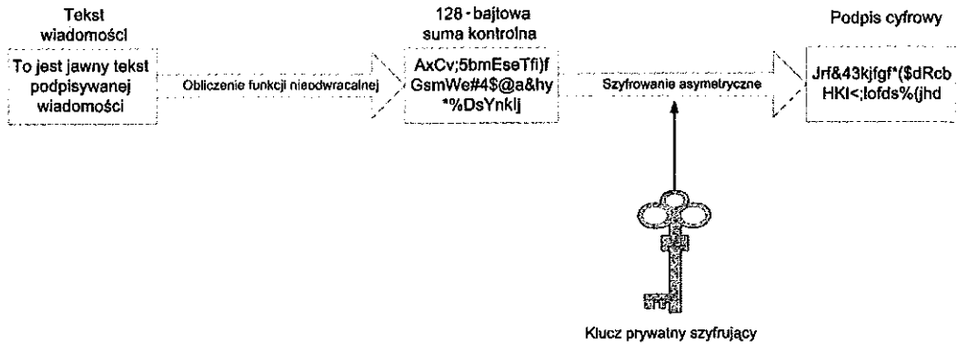


8.1.5. Podpis cyfrowy

W omówionej metodzie szyfrowania asymetrycznego (rozdz. 8.1.3.) klucz szyfrujący stawał się kluczem publicznym (każdy może wysłać zaszyfrowaną wiadomość do właściciela tego klucza), natomiast klucz deszyfrujący stawał się pilnie strzeżonym kluczem prywatnym, aby tylko jego właściciel mógł odszyfrować wiadomość. Jeśli odwrócimy tę sytuację, tzn. klucz szyfrujący zostaje prywatny, a publikuje się klucz deszyfrujący, to wówczas ten sam algorytm szyfrowania można wykorzystać jako podpis elektroniczny. Ogólna zasada jest następująca: właściciel klucza może do dowolnej osoby wysłać wiadomość zaszyfrowaną prywatnym kluczem. Jeśli ta osoba może odszyfrować wiadomość jego kluczem publicznym, to wtedy ma pewność, kto jest autorem wiadomości. Podpis elektroniczny jest o wiele trudniejszy do sfałszowania niż podpis tradycyjny, na papierze. W zasadzie można uznać, że jeśli klucz prywatny jest bardzo dobrze strzeżony i jest on odpowiedniej długości, to wówczas podpis cyfrowy jest nie do podrobienia.

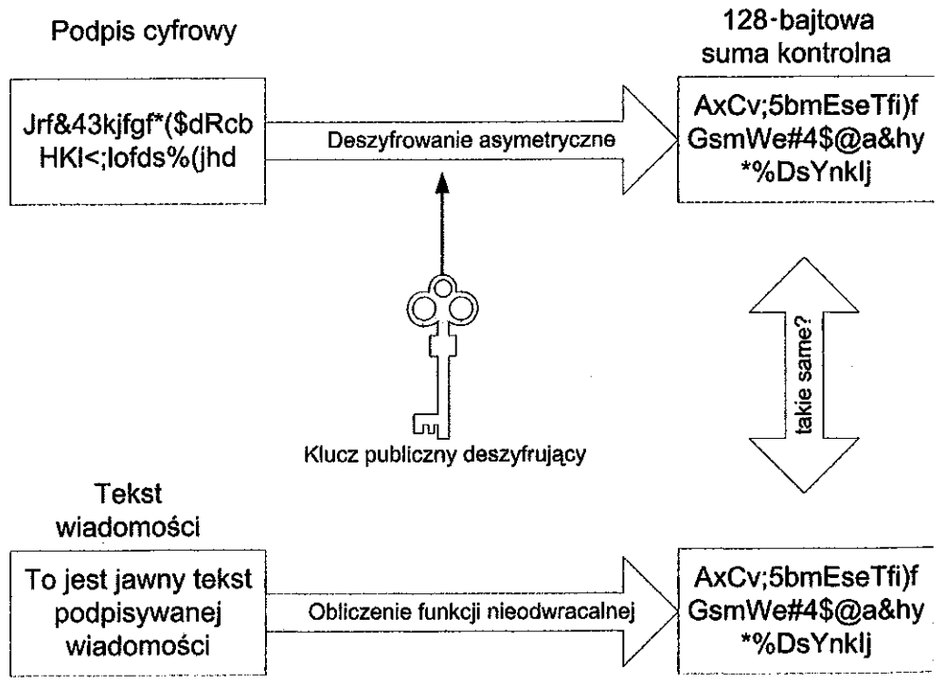
Jak już wspomniano, algorytmy szyfrowania asymetrycznego są czasochłonne. W praktyce więc stosuje się podpis cyfrowy z użyciem funkcji nieodwracalnej (*hash function*), inaczej jednokierunkowej (rys. 65). Funkcja nieodwracalna pozwala przetworzyć zadany ciąg znaków w ciąg znaków o określonej długości, ale w taki sposób, że jest on niepowtarzalny dla danego wejściowego ciągu znaków. Jeśli na wejściu zmienimy choć jeden znak, to suma kontrolna też się zmieni.

Rys. 65. Tworzenie podpisu cyfrowego [12]



Dla wiadomości, która ma być podpisana cyfrowo, oblicza się sumę kontrolną za pomocą funkcji nieodwracalnej, a następnie tę sumę szyfruje się kluczem prywatnym nadawcy i dołącza do oryginalnej wiadomości jako podpis cyfrowy.

Rys. 66. Weryfikacja podpisu cyfrowego [12]



Odbiorca oblicza sumę kontrolną otrzymanej wiadomości, następnie za pomocą klucza publicznego nadawcy odszyfrowuje dołączoną sumę kontrolną i jeśli wartości się zgadzają, to ma pewność, że wiadomość pochodzi od osoby, która posiada klucz szyfrujący, czyli od osoby właściwej (rys. 66). Ale czy na pewno?

8.1.6. Infrastruktura klucza publicznego

Jak już wiemy, podpisanie cyfrowe wiadomości i weryfikacja tego podpisu nie jest sprawą skomplikowaną.

Pozostaje jednak do rozwiązania problem tożsamości osoby podpisującej, tzn. stwierdzenia, czy rzeczywiście wiadomość otrzymana od p. Kowalskiego została podpisana przez niego samego, czy też może to jego sąsiad, p. Nowak, opublikował w Internecie swój klucz pod nazwiskiem Kowalski, żeby przejąć jego bardzo ważną pocztę. Czyli inaczej, jak można zweryfikować czyjś publiczny klucz służący do odszyfrowania podpisu?

W tym celu tworzy się tzw. infrastrukturę klucza publicznego **PKI** (*Public Key Infrastructure*), która może mieć dwa modele [12]:

1. Model nieformalny, oparty na wzajemnym zaufaniu (*web-of-trust*), w którym użytkownicy nawzajem podpisują sobie klucze publiczne. Oznacza to, że kiedy znana mi osoba sygnuje podpis osoby X, to ja mam zaufanie, że to jest rzeczywiście podpis osoby X, choć osobiście jej nie znam. Ten model jest wykorzystywany np. przez program *PGP*, służący do szyfrowania i podpisywania wiadomości. Model ten ma „wysoki współczynnik zaufania", ale jest czasochłonny, wymaga sporo pracy, żeby potwierdzić podpis cyfrowy.
2. Model instytucjonalny, w którym istnieje nadrzędny **Autorytet Certyfikujący** (*Certificate Authority*), do którego wszyscy mają zaufanie. Instytucja CA podpisuje (certyfikuje) klucz publiczny każdemu, kto wystarczająco uwiarygodni tradycyjnymi metodami swoją tożsamość. CA może też delegować swój przywilej certyfikowania kluczy publicznych innym instytucjom, które automatycznie stają się też „autorytetami certyfikującymi". Powstaje wtedy tzw. **ścieżka zaufania** (*path of trust*). Choć ustalenie jednej instytucji, której wszyscy muszą zaufać z góry, wydaje się może trochę trudne do przyjęcia, to jednak system ten jest łatwiejszy do wdrożenia i zarządzania od poprzedniego.

W dużych przedsiębiorstwach nigdy nie powinno się używać oryginalnego klucza prywatnego, tylko na jego podstawie tworzy się klucze pochodne i nimi podpisuje dokumenty. Zgubienie jednego klucza pochodnego nie unieważnia wtedy posiadanego certyfikatu, a tylko ten jeden klucz.

Na wypadek utraty klucza prywatnego (tajnego) tworzy się bazy certyfikatów nieaktualnych. Do tych baz należy zgłaszać, że dany certyfikat nie jest już ważny.

8.2. System uwierzytelniający Kerberos

W celu zwiększenia bezpieczeństwa działania wiele usług sieciowych identyfikuje użytkowników i na tej podstawie określa ich uprawnienia. Do uwierzytelnienia można użyć informacji pochodzących z różnych źródeł, jak pliki lokalne, czy uwierzytelnienie zwykłymi metodami systemu operacyjnego (jak w Windows NT). Starsze sposoby uwierzytelnienia używają powtarzalnych haseł. Za każdym razem, kiedy podajemy takie hasło, ktoś może je zarejestrować i potem podawać się za nas, uzyskując uwierzytelnienie.

Poprawą jakości uwierzytelniania są mechanizmy haseł jednorazowych. Lista haseł jednorazowych może być wygenerowana losowo, przy czym jedną kopię otrzymuje użytkownik a drugą system, w którym następuje uwierzytelnienie, albo konkretna pozycja listy może być generowana na żądanie przez użytkownika i zatwierdzana przez system. Oczywiście taka lista haseł jednorazowych musi być odpowiednio zabezpieczona.

Najbardziej zaawansowane metody korzystają ze scentralizowanego mechanizmu uwierzytelniającego, niezależnego od konkretnej usługi czy konkretnego komputera, w którym ta usługa działa. Mechanizm ten jest częścią systemu, nazywanego często **serwerem AAA** [20].

Serwer AAA udostępnia trzy rodzaje usług:

uwierzytelnianie (*authentication*) – proces weryfikowania i udowadniania tożsamości; dzięki uwierzytelnieniu można sprawdzić, kim lub czym jest dany użytkownik lub komputer;

autoryzacja (*authorization*) – proces ustalania, co komuś wolno zrobić; nie należy mylić uwierzytelnienia i autoryzacji, ponieważ uwierzytelnienie jest to warunek wstępny autoryzacji (chyba że każdy jest autoryzowany do robienia tego samego, jak na anonimowym serwerze FTP);

ewidencjonowanie (*auditing*) – dostarcza informacji o pomyślnym lub niepomyślnym przebiegu uwierzytelniania i autoryzacji.

Zadanie uwierzytelniania może być bardzo łatwe, jeśli nie ma to większego znaczenia lub znajdujemy się w godnym zaufania środowisku (wewnętrzne, zaufane sieci), albo bardzo trudne i skomplikowane, kiedy istnieje możliwość, że ktoś będzie chciał oszukiwać.

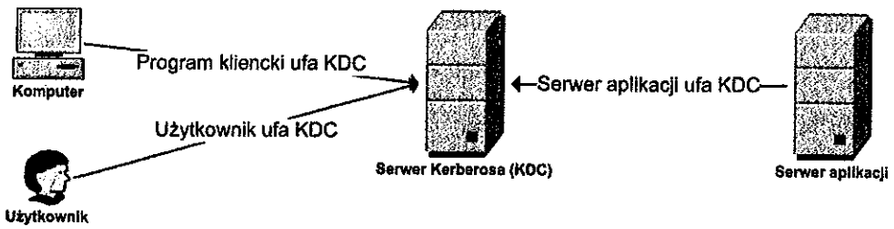
Najbardziej znanym mechanizmem scentralizowanego uwierzytelniania jest system Kerberos, opracowany w MIT w ramach projektu *Athena*. Kerberos zapewnia silne uwierzytelnianie w środowisku rozproszonym. Jest on podstawowym sposobem uwierzytelniania w środowisku Windows 2000 i XP. Obsługuje go również wiele wersji Unixa. Kerberos próbuje rozwiązać problem porozumiewania się klienta z serwerem przez niezaufaną sieć. Klient i serwer nie muszą ufać sobie nawzajem ani sieci, ale muszą ufać serwerowi Kerberosa, czyli zaufanej stronie trzeciej. Zalecana wersja Kerberosa to wersja 5, natomiast Windows 2000/XP używa pewnych rozszerzeń w stosunku do niej. Ponadto w implementacji Microsoftu wiele

tradycyjnych pojęć Kerberosa zostało nazwanych inaczej, aby ułatwić życie doświadczonym administratorom Windows.

8.2.1. Działanie systemu Kerberos

Kerberos zapewnia bezpieczne uwierzytelnienie klientów i serwerów w niezaufanej sieci, jeśli tylko same serwery Kerberosa są odpowiednio zabezpieczone przed obcą ingerencją. System opiera się na przedstawionych poniżej jednostronnych relacjach zaufania (rys. 67).

Rys. 67. Relacje zaufania w systemie Kerberos [33]



Wszystkie komputery korzystające z tego rodzaju uwierzytelnienia muszą mieć poprawnie zsynchronizowany czas. Kerberos używa następującej terminologii:

Królestwo (*realm*) – obszar podległy serwerowi Kerberosa, Windows 2000 i XP używają tu terminu **domena**;

Aktor (*principal*) – każda ze stron, zaangażowana w transakcję, czyli jednostka, która ma zostać uwierzytelniona; aktorami są zwykle ludzie i usługi; każdy aktor ma sekret (hasło), który dzieli tylko z serwerem uwierzytelniającym;

KDC (*Key Distribution Center*) – centrum dystrybucji kluczy, czyli serwer uwierzytelniający (w Windows 2000 jest to część kontrolera domeny); na serwerze KDC udostępniane są dwie usługi: **AS** (*Authentication Service*), czyli usługa uwierzytelniania, oraz **TGS** (*Ticket-Granting Service*), czyli usługa przyznawania biletów;

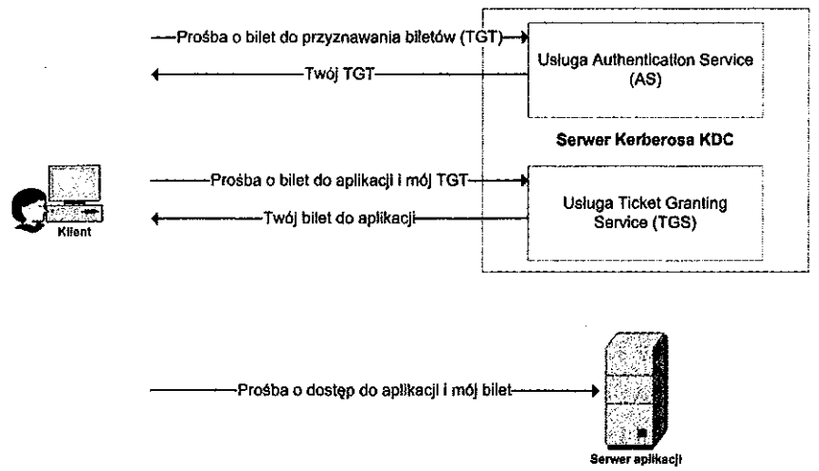
TGT (*ticket granting ticket*) – nadrzędny bilet przydzielany przez AS uwierzytelnionemu aktorowi, służący do otrzymania biletu od TGS;

Bilet (*ticket*) – identyfikator wręczany serwerowi przez użytkownika, który chce skorzystać z aplikacji pracującej na tym serwerze, specyficzny dla użytkownika i serwera.

Przyznawanie użytkownikowi dostępu do serwera aplikacji, pokazane na rys. 68, wygląda następująco. Podczas inicjalnego logowania do sieci użytkownik musi negocjować dostęp, przedstawiając swoje hasło usłudze uwierzytelniającej AS na serwerze Kerberosa KDC. Hasło to nigdy nie jest przesyłane przez sieć otwartym tekstem. Po pozytywnej weryfikacji otrzymuje on bilet nadrzędny TGT ważny dla danego królestwa (w Windows 2000 i XP dla danej domeny). Bilet ten, przechowywany w komputerze użytkownika, jest ważny cały dzień (w Windows 2000 i XP ważny 10 godzin). Chcąc skorzystać z następnej usługi użytkownik nie musi

więc ponownie podawać hasła. Oprogramowanie klienckie przedstawia otrzymany bilet TGT usłudze przyznawania biletów TGS na serwerze KDC i prosi o dostęp do konkretnej aplikacji, z której chce skorzystać użytkownik. Po pozytywnej weryfikacji TGT serwer generuje bilet do żądanej aplikacji dla danego użytkownika. W końcu użytkownik po przedstawieniu serwerowi aplikacji wygenerowanego biletu, może zacząć z niej korzystać.

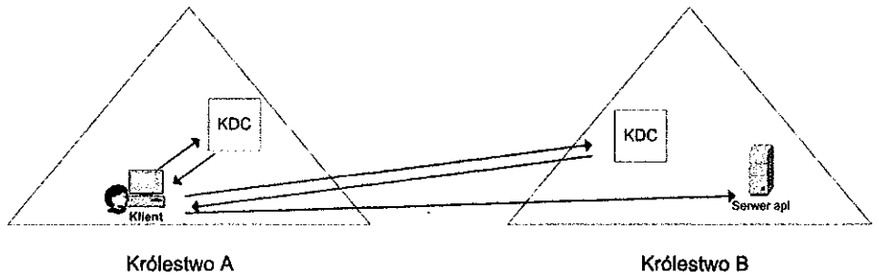
Rys. 68. Przyznawanie dostępu do aplikacji w systemie Kerberos [33]



8.2.2. Rozszerzanie zaufania

Usługi uwierzytelniania AS i przyznawania biletów TGS działają w obrębie serwera Kerberosa niezależnie, a królestwa (domeny) Kerberosa mogą ufać sobie nawzajem. Pozwala to użytkownikowi używać biletu TGT otrzymanego w jednym królestwie do uzyskiwania dostępu do usług w innym królestwie. Używa się w tym celu biletów **międzykrólestwowych** (*referral ticket*) [20] (rys. 69).

Rys. 69. Uzyskiwanie dostępu do aplikacji z innego królestwa [33]



Kiedy użytkownik z królestwa A zwróci się do usługi TGS w swoim królestwie z żądaniem dostępu do usługi dostępnej w królestwie B i jeśli królestwa ufają sobie nawzajem, to

użytkownik otrzyma odpowiedni bilet, który musi przedstawić usłudze przyznawania biletów w królestwie B. Po pozytywnej weryfikacji tego biletu usługa TGS królestwa B wydaje użytkownikowi bilet do aplikacji, który następnie zostaje przedstawiony serwerowi aplikacji.

Relacja zaufania jest przechodnia, tzn. jeśli królestwo A ufa królestwu B i królestwo C ufa królestwu B, to królestwa A i C ufają sobie nawzajem, choć nie ma między nimi bezpośredniej relacji zaufania. Jeśli użytkownik z królestwa A chce skorzystać z usługi udostępnianej w królestwie C, to musi otrzymać najpierw bilet z A do B, a potem dopiero z B do C. W bilecie są zawarte informacje, przez które królestwa przeszedł dany użytkownik.

8.3. Protokoły pośrednie

Wiele protokołów aplikacyjnych opiera się bezpośrednio na protokołach TCP i UDP (które z kolei opierają się na IP). Inne aplikacje korzystają z różnych protokołów pośrednich, wnoszących nową funkcjonalność do rodziny TCP/IP [20]. Bardzo często są to funkcje związane z zapewnianiem bezpieczeństwa przesyłanych danych na różnych poziomach komunikacji sieciowej. Omówione tu zostaną następujące protokoły pośrednie: RPC, CORBA, TLS i SSL, IPsec oraz PPTP i L2TP.

8.3.1. Protokół RPC

Skrót **RPC** (*Remote Procedure Call*) oznacza zdalne wywołanie procedury i odnosi się do pewnego mechanizmu, dzięki któremu program wykonuje coś, co z punktu widzenia programisty wygląda jak wywołanie zwykłej procedury, a w rzeczywistości polega na skontaktowaniu się z innym programem, który może działać na innym komputerze [20]. Pozwala to programistom na takie projektowanie oprogramowania, aby korzystało ono z gotowych serwisów czy innych już działających programów. Używa się w tym celu jednego z wielu dostępnych protokołów, także określanych jako RPC. Najbardziej znane są dwie jego wersje:

- RPC dla systemów uniksowych, opracowany przez firmę Sun, z którego następnie utworzono standard Open Computing Group RPC,
- RPC w systemach firmy Microsoft, standaryzowany przez Open System Foundation w ramach projektu Distributed Computing Environment.

Obie te wersje protokołu RPC nie umieją współpracować ze sobą.

Mechanizmy RPC są używane przez bardzo wiele protokołów aplikacyjnych. W maszynach uniksowych korzysta z nich np. NFS, natomiast w systemach Windows XP – m.in. Microsoft Exchange i DHCP. Z punktu widzenia bezpieczeństwa protokół NFS jest usługą podatną na ataki. Aplikacje Windows XP korzystające z RPC nie są aż tak nieodporne, ale też nie można ich uznać za w pełni bezpieczne.

W protokołach TCP i UDP aplikacja, do której dociera wiadomość, jest reprezentowana przez numer portu. Pola numerów portów zajmują tylko dwa bajty, co oznacza, że istnieje tylko 65536 możliwych numerów portów dla usług TCP i UDP. Jest to stanowczo za mało, aby przypisać każdej możliwej aplikacji i usłudze niepowtarzalny numer portu. RPC rozwiązuje ten problem, ponieważ każda usługa oparta na RPC ma przypisany niepowtarzalny czterobajtowy **numer usługi RPC**. Daje to ponad 4 miliony dostępnych numerów do przydzielenia. Numery usług RPC są odwzorowywane na porty UDP i TCP, przypisane serwerom RPC na konkretnym komputerze, za pomocą **serwera lokalizującego**. W komputerach uniksowych nazywa się on *portmapper*, natomiast w Windows XP – *RPC Locator*.

Serwer lokalizujący to jedyna usługa RPC, która zawsze działa pod konkretnym numerem portu TCP lub UDP. Kiedy serwer oparty na RPC rozpoczyna pracę, przydziela sobie port TCP i/lub UDP. Następnie kontaktuje się z serwerem lokalizującym, aby zarejestrować swój niepowtarzalny numer usługi RPC i konkretne porty używane w tym momencie. Jeśli nawet serwer wybiera zawsze ten sam numer portu, to nigdy nie będzie miał gwarancji, że ten port będzie zawsze wolny, bo mogła go już zarezerwować inna usługa.

Kiedy program kliencki, oparty na RPC, chce się skontaktować z konkretnym serwerem w danym komputerze, to najpierw musi się połączyć z serwerem lokalizującym, podając mu unikalny numer usługi RPC. Jeśli taka usługa nie została uruchomiona, to klient otrzymuje odpowiedni komunikat. Jeśli usługa działa, to serwer odsyła klientowi numer portu TCP lub UDP, pod którym należy się z daną usługą komunikować. Dalsza komunikacja z usługą odbywa się bez udziału serwera lokalizującego.

8.3.2. Protokół CORBA

CORBA (*Common Object Request Broker Architecture*) jest obiektowym modelem przetwarzania rozproszonego, opracowanym dla systemów uniksowych [20]. Obiekty CORBY porozumiewają się ze sobą za pomocą protokołu Object Request Broker, zwanego **orb**. Obiekty CORBY komunikują się ze sobą przez Internet za pomocą *Internet Inter-Orb Protocol* (IIOP), który opiera się na TCP, ale nie używa stałych numerów portów.

8.3.3. Protokoły SSL i TLS

Protokół **SSL** (*Secure Socket Layer*) został opracowany przez firmę Netscape w 1993 r. [20]. Jego najbardziej rozpowszechnione wersje to SSL 2 i SSL 3, przy czym wersja druga nie jest zalecana ze względu na pewne problemy algorytmów kryptograficznych. Netscape zwrócił się do zespołu IETF z propozycją opracowania standardu internetowego, w wyniku czego powstał protokół **TLS** (*Transport Layer Security*).

Protokoły SSL i TLS umożliwiają:

- uwierzytelnianie serwerów i klientów,
- szyfrowanie danych między węzłami końcowymi,
- ochronę integralności danych.

Ponadto pozwalają klientom na ponowne łączenie się z raz użytymi serwerami bez ponownego uwierzytelniania się czy negocjowania kluczy sesyjnych, jeśli tylko nowe połączenie zostanie nawiązane krótko po zamknięciu poprzedniego.

Oba te protokoły mogą zapewniać bezpieczne, prywatne sesje komunikacyjne, ponieważ:

- klient i serwer negocjują algorytmy szyfrowania i ochrony integralności;
- tożsamość serwera, z którym łączy się klient, jest zawsze weryfikowana, a test tożsamości przeprowadza się przed wysłaniem opcjonalnych informacji uwierzytelniających użytkownika;
- algorytmy wymiany kluczy zapobiegają atakom typu „człowiek w środku";
- pod koniec wymiany kluczy następuje wymiana sum kontrolnych, która wykrywa ingerencję w negocjacje używanego algorytmu;
- serwer może sprawdzić tożsamość klienta na wiele różnych sposobów, może też obsługiwać klientów anonimowych;
- wszystkie wymieniane pakiety danych zawierają test integralności komunikatu, błąd integralności powoduje zamknięcie połączenia;
- dzięki pewnym zbiorom wynegocjowanych algorytmów można używać tymczasowych parametrów uwierzytelniających, które są odrzucane po konfigurowalnym czasie, co zapobiega późniejszemu rozszyfrowaniu zarejestrowanej sesji.

TLS i SSL nie używają określonego algorytmu do operacji generowania kluczy, uwierzytelniania i szyfrowania danych. Mogą korzystać z kilku różnych algorytmów, a właściwie z kilku różnych zestawów algorytmów, co pozwala elastyczniej dopasowywać działanie protokołów do wymagań danej sytuacji, np. zapewniać słabsze, ale szybsze szyfrowanie, albo zaprzestać stosowania algorytmów zbyt często łamanych. Istnieje tylko jeden zestaw algorytmów, który aplikacja musi zawsze obsługiwać, aby można ją było nazwać zgodną z TLS.

Protokoły TLS i SSL są użyteczne tylko wtedy, kiedy są używane w połączeniu z protokołem wyższego rzędu, który rzeczywiście wymienia dane pomiędzy aplikacjami. Jednym sposobem takiego łączenia protokołów jest użycie nowego numeru portu dla połączonych protokołów aplikacyjnego i szyfrującego. Tak działa protokół **HTTPS**, powstały ze zintegrowania HTTP i SSL. Drugim sposobem jest dodanie do protokołu aplikacyjnego nowego polecenia, inicjującego zaszyfrowaną sesję pod tym samym numerem portu, czego przykładem jest protokół ESMTP (*Extended SMTP*). Do protokołu SMTP dodano polecenie *STARTTLS*, inicjujące zaszyfrowaną sesję.

8.3.4. Protokół IPsec

Protokół IPsec jest to protokół bezpieczeństwa, opracowany przez zespół IETF. Działa on bezpośrednio ponad IP i zapewnia kryptograficzne bezpieczeństwo wymiany danych pomiędzy bramami do sieci lokalnych lub pojedynczymi komputerami [20]. Obsługa IPsec jest wymagana w każdej implementacji IPv6 i opcjonalna w IPv4.

Ponieważ IPsec jest realizowany w warstwie IP, może chronić dowolny protokół używający IP, w tym TCP i UDP. Usługi zapewniane przez ten protokół to:

Kontrola dostępu – odmowa wynegocjowania parametrów bezpieczeństwa uniemożliwi nawiązanie komunikacji.

Uwierzytelnienie pochodzenia danych – odbiorca pakietu ma pewność, że pochodzi on od nadawcy, od którego wydaje się pochodzić.

Integralność komunikatów – napastnik nie może zmodyfikować pakietu i liczyć na jego zaakceptowanie.

Ochrona przez odtwarzaniem – napastnik nie może ponownie przesłać raz wysłanego komunikatu i liczyć na jego zaakceptowanie.

Poufność – napastnik nie może odczytać przechwyconych danych.

IPsec składa się z trzech protokołów, zdefiniowanych jako ogólne ramy, które określają układ i rozmiar pól w pakietach i mogą zostać wykorzystane przez różne algorytmy kryptograficzne. Same protokoły nie definiują konkretnych algorytmów kryptograficznych, choć każda implementacja musi obsługiwać określony zestaw algorytmów.

Protokoły tworzące standard IPsec:

AH (*Authentication Header*) – zapewnia integralność komunikatów i uwierzytelnia pochodzenie danych, opcjonalnie może zapewniać usługi przeciwodtworzeniowe. Ochrona integralności w AH obejmuje informacje z nagłówka pakietu (w tym adres źródłowy i docelowy), ale nie uwzględnia takich parametrów często zmienianych przez routery, jak pola *czas życia* w IPv4 lub liczba etapów w IPv6.

ESP (*Encapsulating Security Payload*) – zapewnia poufność (szyfrowanie) oraz ograniczoną ochronę przed analizą przepływu pakietów. ESP obejmuje także niektóre usługi, zwykle realizowane przez AH.

ISAKMP (*Internet Security Association Key Management Protocol*) – zapewnia generowanie współdzielonych kluczy, na których opiera się praca poprzednich dwóch protokołów.

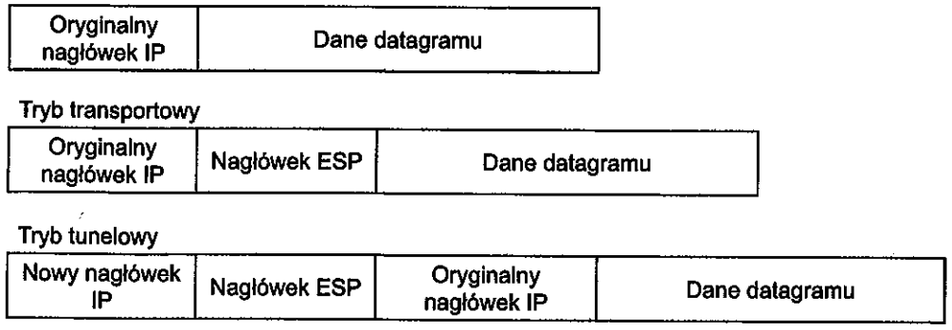
Protokół **IKE** (*Internet Key Exchange*) używa struktury ISAKMP w połączeniu z konkretnymi algorytmami wymiany kluczy do tworzenia kluczy kryptograficznych na potrzeby AH i ESP.

W IPv6 protokoły AH i ESP mogą być używane jednocześnie, dzięki mechanizmowi nazywanemu łańcuchowym połączeniem nagłówków. Pozwala to na zastosowanie trybów uwierzytelniania, którego sam ESP nie zapewnia. W IPv4 nie można używać dwóch protokołów jednocześnie, gdyż pakiet może mieć tylko jeden nagłówek.

IPsec definiuje dwa tryby pracy (rys. 70):

- transportowy,
- tunelowy.

Rys. 70. Tryby pracy protokołu IPsec



W trybie **transportowym** AH lub ESP pojawiają się bezpośrednio za nagłówkiem IP i kapsułkują zaszyfrowaną pozostałą część pierwotnego pakietu. Tryb ten działa tylko między indywidualnymi hostami i służy do ochrony komunikacji. Może on chronić cały ruch pomiędzy hostami lub może stanowić warstwę ochronną dla konkretnych protokołów.

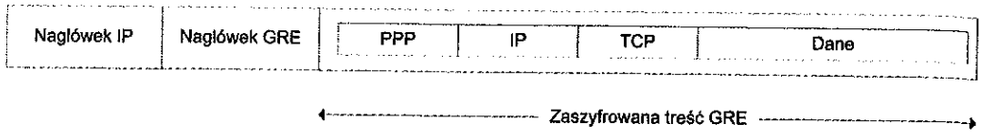
W trybie **tunelowym** cały zaszyfrowany pierwotny pakiet zostaje umieszczony w nowym pakiecie IP i tworzony jest nowy nagłówek IP. **Bramką bezpieczeństwa** nazywamy każde urządzenie, które może pracować w trybie tunelowym, czyli potrafi przyjmować pakiety IP i przekształcać je na pakiety protokołów IPsec. Tryb tunelowy umożliwia zbudowanie wirtualnej sieci prywatnej, opartej na IPsec.

8.3.5. Protokół PPTP

Protokół **PPTP** (*Point-to-Point Tunneling Protocol*) jest protokołem opartym na protokołach **PPP** (*Point-to-Point Protocol*) oraz **GRE** (*Generic Routing Encapsulation*). Protokół PPP został zaprojektowany do pracy m.in. w sieciach telefonicznych. Udostępnia on uniwersalny mechanizm kapsułkowania protokołów na poziomie IP. Może on kapsułkować pakiety protokołów IP, IPX i NetBEUI. PPTP to rozszerzenie protokołu PPP, w którym pakiety PPP są szyfrowane i kapsułkowane w pakietach GRE. Protokół PPTP jest używany do realizowania wirtualnych sieci prywatnych, czyli do tunelowania IP w IP.

Kapsułkowanie pakietu TCP w pakiecie PPTP pokazano na rys. 71.

Rys. 71. Kapsułkowanie pakietu TCP w pakiecie PPTP [20]



Tunelowanie jest procesem przesyłania pakietów do komputerów w prywatnej sieci przez sieć publiczną. Routery sieciowe nie mają dostępu do komputerów znajdujących się w prywatnej sieci. Tunelowanie pozwala na transmisję pakietów do komputera pośredniczącego, np. serwera PPTP, który jest podłączony jednocześnie do ogólnodostępnej sieci przesyłającej pakiety, jak i do sieci prywatnej. Zarówno klient PPTP, jak i serwer PPTP używają tunelowania do bezpiecznego przesyłania pakietów przez sieć, w której routery znają tylko adres odbiorcy.

Protokół PPTP ma jednak pewne wady. Choć jest on protokołem szyfrowanym, to nie wszystkie etapy konwersacji podlegają szyfrowaniu. Zanim serwer PPTP zacznie przyjmować pakiety GRE, negocjacje odbywają się przez TCP. Są one przeprowadzane jawnym tekstem, narażone są więc na podsłuchanie. Dodatkowo negocjacje odbywają się przed uwierzytelnieniem klienta, serwer PPTP jest zatem szczególnie podatny na ataki wrogich klientów.

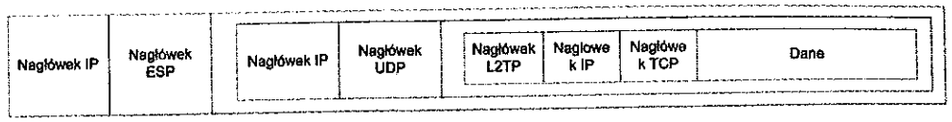
8.3.6. Protokół L2TP

Protokół **L2TP** (*Layer To Transport Protocol*) jest uniwersalnym protokołem kapsułkującym, służącym do tunelowania ruchu IP. Jest on, podobnie jak PPTP, rozszerzeniem protokołu PPP, ale są pomiędzy nimi dwie zasadnicze różnice. PPTP może działać tylko ponad IP, wymaga zatem łączności na poziomie IP. Natomiast L2TP może działać ponad różnymi protokołami, w tym bezpośrednio ponad łączem telefonicznym (jak PPP). PPTP jest protokołem szyfrowanym, natomiast L2TP nie jest protokołem szyfrowanym, ale za to zapewnia wzajemne uwierzytelnienie podczas wstępnych negocjacji i może ukrywać wymieniane wówczas informacje.

L2TP jest zwykle używany w połączeniu z IPsec, który zapewnia szyfrowanie. W ten sposób powstaje wielowarstwowy stos protokołów. Na rysunku 72 zamieszczony jest przykład przesyłania pakietu TCP za pośrednictwem L2TP w sieci IP. Pakiet IP jest dodatkowo zabezpieczony za pomocą protokołu IPsec, czyli umieszczony ponownie w IP, ale z użyciem protokołu ESP.

Niestety, ceną za dobre zabezpieczenie przesyłanych danych jest tutaj dosyć duży narzut przesyłanych danych, związany z dodatkowymi nagłówkami.

Rys. 72. Kapsułkowanie L2TP pakietu TCP podczas podróży przez sieć IP [20]



8.4. Wirtualna sieć prywatna

Wirtualna sieć prywatna **VPN** (*Virtual Private Network*) pozwala na łączenie lokalnych sieci prywatnych w jedną sieć za pośerdnictwem sieci publicznej, zwykle za pomocą Internetu. Tworzenie prywatnych szybkich i odległych połączeń pomiędzy dwoma ośrodkami jest znacznie bardziej kosztowne niż podłączenie tych samych dwóch ośrodków do sieci publicznej, ale także znacznie bezpieczniejsze. Wirtualne sieci prywatne są próbą połączenia zalet sieci publicznych (niskich kosztów i powszechnej dostępności) z niektórymi zaletami sieci prywatnych (bezpieczeństwo).

„Rozciągnięcie" firmowej sieci polega na zestawieniu bezpiecznego i szyfrowanego połączenia pomiędzy dwoma punktami – tunelu przechodzącego przez publiczną przestrzeń Internetu. VPN umożliwia szyfrowanie i kapsułkowanie w pakietach IP danych przesyłanych w tej sieci, wykorzystując jako środek transportu Internet. Przez tak zabezpieczony transparentny tunel zdalni użytkownicy mogą korzystać z odległych zasobów firmowych i aplikacji sieciowych.

Wirtualne sieci prywatne mogą być dla firmy sposobem na obniżenie kosztów łączności, a dzięki zwiększeniu możliwości korzystania z pracy zdalnej i outsourcingu – również kosztów pracy. Firmy wieloodziałowe lub zatrudniające pracowników mobilnych ponoszą duże koszty łączności pomiędzy rozproszonymi częściami przedsiębiorstwa. Korzystanie z dostępu komutowanego **RAS** (*Remote Access Service*) lub dzierżawa łączy telekomunikacyjnych w postaci rozległej sieci korporacyjnej (WAN) są skutecznymi, ale i drogimi sposobami utrzymywania łączności. Alternatywą jest Internet – jako tańsze medium i wirtualna sieć prywatna – gwarantujące poufność i bezpieczeństwo komunikacji firmowej w obrębie sieci publicznej. Zastosowanie VPN ogranicza do minimum liczbę niezbędnych do obsługi sieci specjalizowanych urządzeń (routerów, serwerów dostępowych, modemów) a tym samym koszty związane z ich amortyzacją i zarządzaniem. Często do zbudowania VPN wystarcza dodatkowe oprogramowanie na routerze czy stacjach klienckich.

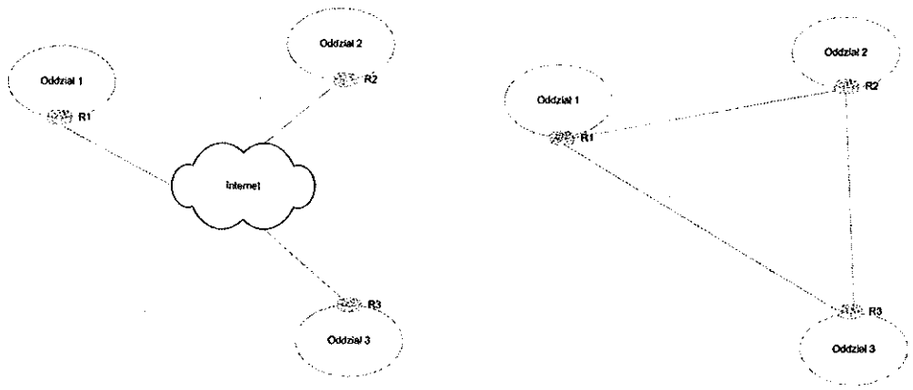
Zalety VPN widoczne są przede wszystkim w firmach działających na rozległych geograficznie obszarach, mających dużą liczbę zamiejscowych czy zagranicznych oddziałów, zatrudniających pracowników mobilnych, zlecających wykonywanie prac biurowych albo o charakterze intelektualnym na zewnątrz firmy. Możliwość pracy zdalnej przekłada się nie tylko na obniżenie kosztów stworzenia miejsca pracy. Stwarza też szansę pozyskania wysoko

wykwalifikowanych pracowników niezależnie od ich miejsca zamieszkania oraz zwiększa elastyczność zatrudnienia.

8.4.1. Działanie VPN

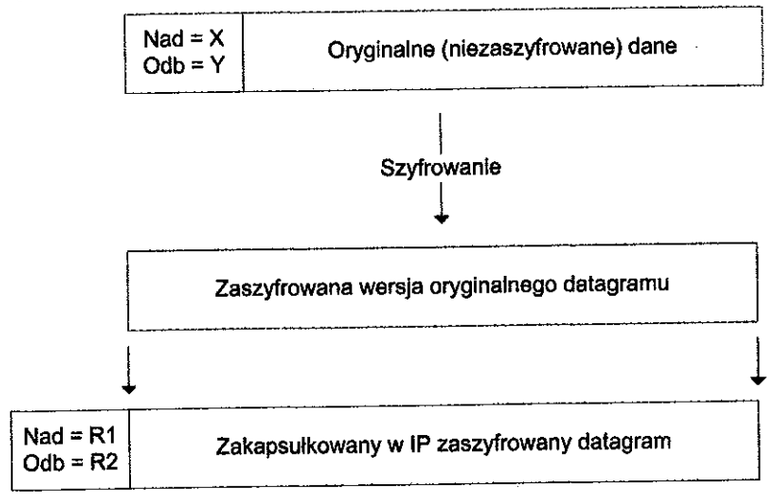
Aby zastosować VPN, firma dla każdego swojego oddziału zakupuje odpowiedni zestaw sprzętu i oprogramowania, oznaczonego na rysunku rys. 73 jako R1, R2 i R3. Zestaw ten jest instalowany między prywatną (wewnętrzną) siecią firmy a siecią publiczną. Każdy z tych zestawów musi zostać tak skonfigurowany, aby znać adresy innych systemów VPN firmy. Na tej podstawie oprogramowanie VPN ogranicza odpowiednio ruch pakietów. Po pierwsze, system filtruje pakiety przychodzące – dany pakiet nie wejdzie do sieci, jeśli nie pochodzi z jednego z oddziałów firmy. Po drugie, w każdym oddziale są filtrowane pakiety wychodzące – pakiet nie wyjdzie z oddziału, jeśli nie jest zaadresowany do innego oddziału. Dzięki temu po skonfigurowaniu VPN oddziały mogą komunikować się wyłącznie ze sobą, odcięte od reszty sieci. Oprogramowanie VPN szyfruje każdy wychodzący datagram, aby zapewnić mu ochronę przed odczytaniem przez osoby postronne podczas wędrówki przez sieć publiczną. Na miejscu przeznaczenia datagram jest odszyfrowywany przez oprogramowanie VPN przed przekazaniem go ostatecznemu odbiorcy. Fizycznie dane są przesyłane za pomocą ogólnodostępnej infrastruktury, ale z punktu widzenia użytkownika korzystanie z VPN przypomina korzystanie z dedykowanego łącza prywatnego.

Rys. 73. Połączenia fizyczne i połączenia logiczne VPN [6]



Bezpieczeństwo przesyłanych danych zapewnia mechanizm **tunelowania**. Jeśli nagłówek datagramu zostanie zaszyfrowany razem z całym datagramem, to routery w Internecie nie będą mogły interpretować pól nagłówka i poprawnie go przekazywać. Jeśli nagłówek nie zostanie zaszyfrowany, to osoby postronne mogą zapoznać się z wewnętrznymi adresami nadawcy i odbiorcy i na tej podstawie wydedukować dodatkowe informacje. Dlatego oprogramowanie VPN korzysta z mechanizmu **tunelowania IP w IP**. Oznacza to, że cały wysyłany datagram jest szyfrowany i wynik jest umieszczany w nowym datagramie, który jest wysyłany do adresata (rys. 74). W nagłówku zamiast adresów konkretnych komputerów są umieszczone adresy routerów, łączących sieć wewnętrzną z siecią publiczną.

Rys. 74. Kapsułkowanie IP w IP, używane przez VPN [6]



Wirtualna sieć prywatna stanowi także metodę połączenia się pracownika znajdującego się poza firmą przez Internet z siecią lokalną swojej firmy. Oprócz szyfrowania wykorzystuje się tu techniki uwierzytelniania, aby mieć pewność, że z siecią lokalną połączą się tylko użytkownicy autoryzowani.

Całe połączenie w sieci VPN jest szyfrowane, aby dane przesyłane przez Internet nie zostały przechwycone i wykorzystane przez niepowołane osoby. Wykorzystywane są w tym celu protokoły **PPTP** (rozdz. 8.3.5.) oraz **L2TP** (rozdz. 8.3.6.), który z kolei wykorzystuje technologię **IPSec** (rozdz. 8.3.4.).

Szyfrowanie może być wykonywane dwojako:

- jako część metody transportu, w której to host decyduje o szyfrowaniu generowanego przez siebie ruchu;
- jako tunel, w którym ruch jest szyfrowany i deszyfrowany pomiędzy źródłem pakietów a ich celem przeznaczenia.

Ważne jest położenie miejsca szyfrowania pakietów względem filtru pakietów, przy czym każde ma swoje wady. Jeśli pakiety są szyfrowane i deszyfrowane w obrębie sieci lokalnej, to muszą przechodzić przez filtr pakietów bez żadnej analizy, co wiąże się z wystawieniem się na możliwość ataku. Jeśli szyfrowanie i deszyfrowanie zachodzi poza granicami filtrowania

pakietów, to pakiety podlegają pełnej analizie filtrowania, ale za to łatwiej je podsłuchać napastnikom, bo wtedy takie szyfrowanie odbywa się na zewnątrz większości zabezpieczeń.

8.4.2. Stosowane technologie

Ze względu na odbiorcę i rodzaj zastosowania można wyróżnić dwie technologie IP VPN. Tradycyjny protokół IPSec umożliwia stworzenie sieci VPN przy wykorzystaniu publicznego Internetu, nie gwarantuje jednak jakości pasma. IPSec jest rozwiązaniem dla tych użytkowników, którym nie zależy na bardzo dobrych warunkach transmisji, np. głosu. Bardziej wymagającym i zasobniejszym użytkownikom ciekawe możliwości stwarza technologia **IP MPLS** (*Multiprotocol Label Switching*) [36]. Jest to stosunkowo nowa technologia, niezależna od technologii sieci podkładowej, zlokalizowana w 2 i 3 warstwie modelu OSI. Może ona wykorzystywać zarówno sieci Frame Relay i ATM (główne technologie dotychczasowych sieci korporacyjnych), jak też Packet over SONET/SDH czy Ethernet. Umożliwia to „rozpięcie" sieci wirtualnej ponad mediami transmisyjnymi. Główną ideą MPLS jest oznaczanie pakietów, co pozwala nadawać im odpowiednie priorytety i wydzielać spośród nich klasy jakości QoS (*Quality of Service*), np. klas głosowych, co gwarantuje transmisjom głosowym znacznie lepszą jakość niż w dotychczas stosowanych rozwiązaniach. Etykieta pozwala rozpatrywać każdy pakiet jako przynależny do strumienia danych, który można spójnie obsługiwać na ustalonych wcześniej warunkach. Pozwala to na wbudowanie w sieć pewnej inteligencji, dzięki której każda transmisja będzie obsługiwana stosownie do jej specyfiki i priorytetu.

Ofertę zewnętrznych dostawców VPN można podzielić na trzy kategorie:

- usługa z pasmem transmisji bez żadnych gwarancji,
- usługa z pasmem transmisji z gwarancją części ruchu,
- usługa z podziałem na klasy głosu, aplikacji biznesowych oraz pozostałego ruchu.

Dla transmisji głosu jest zagwarantowane maksymalne pasmo i minimalne opóźnienia, natomiast dla aplikacji biznesowych gwarantowana jest przepustowość niezależnie od spiętrzenia innego ruchu. ■

Bibliografia

Publikacje książkowe i artykuły z czasopism:

[1] *Akademia sieci Cisco 1 & 2 semestr*, wyd. 3, Mikom, Warszawa 2004. Seria: Cisco Press.

[2] *Akademia sieci Cisco 3 & 4 semestr*, wyd. 3, Mikom, Warszawa 2004. Seria: Cisco Press.

[3] Albitz P., Liu C., *DNS i BIND*, Wydawnictwo RM, Warszawa1999.

[4] Bojarski B., *Zmierzch kabli*, „Chip" nr 10/2004, s. 18–23.

[5] Bruno A., Kim J., *CCDA Oficjalne materiały przygotowujące do egzaminu*, Mikom, Gliwice 2004.

[6] Comer D.E., *Sieci komputerowe i intersieci*, WNT, Warszawa 2003.

[7] Derfler F., Freed L., *Okablowanie sieciowe w praktyce. Księga eksperta*, Helion, Gliwice 2000.

[8] *Introduction to TCP/IP*, MSDN Library, July 1999.

[9] Janikowski A., *Nowe oblicze Ethernetu*, „PCkurier" 5/2003.

[10] Łukawiecki R., *IPv6, Materiały konferencyjne*,Tech-Ed 2003.

[11] —, *A-to-Z of Cryptography and Security, Materiały konferencyjne*, Tech-Ed 2003.

[12] —, *A-to-Z of Public Key Infrastructure (PKI), Materiały konferencyjne*, Tech-Ed 2003.

[13] Pierścionek W., *Router Cisco w sieci Frame Relay*, „PCkurier" 12/2001.

[14] Sportack M., *Sieci komputerowe. Księga eksperta*, Wydawnictwo Helion, Gliwice 2004.

[15] Tanenbaum A.S., *Sieci komputerowe*, Wydawnictwo Helion, Gliwice 2004.

[16] Urbanek A., *Ilustrowany leksykon teleinformatyka*, IDG Poland S. A. 2001.

[17] *Vademecum teleinformatyka cz. I*, IDG Poland S. A. 1999.

[18] *Vademecum teleinformatyka cz. II*, IDG Poland S. A. 2002.

[19] *Vademecum teleinformatyka cz. III*, IDG Poland S. A. 2004.

[20] Zwicky E.D., Cooper S., Chapman D.B., *Internet Firewalls. Tworzenie zapór ogniowych*, Wydawnictwo RM, Warszawa 2001.

Publikacje w Internecie:

[21] *RFC 1730 Internet Message Access Protocol – Version 4*, dostępny w http://www.ietf.org/rfc/rfc1730.txt, dostęp 05.09.2005 r.

[22] *RFC 1939 Post Office Protocol – Version 3*, dostępny w http://www.ietf.org/rfc/rfc1939.txt, dostęp 05.09.2005 r.

[23] *RFC 2026 The Internet Standards Process – Revision 3*, dostępny w http://www.ietf.org/rfc/rfc2026.txt, dostęp 05.09.2005 r.

[24] *RFC 2821 Simple Mail Transfer Protocol*, dostępny w http://www.ietf.org/rfc/rfc2821.txt, dostęp 05.09.2005 r.

[25] *RFC 2373 IP Version 6 Addressing Architecture*, dostępny w http://www.ietf.org/rfc/rfc2373.txt?number=2373, dostęp 05.09.2005 r.

[26] *RFC 3501 Internet Message Access Protocol – Version 4 rev.1*, dostępny w http://www.ietf.org/rfc/rfc3501.txt, dostęp 05.09.2005 r.

[27] *RFC 3194 The Host-Density Ratio for Address Assignment Efficiency: An update on the H ratio*, dostępny w http://www.ietf.org/rfc/rfc3194.txt, dostęp 05.09.2005 r.

[28] *Protokół SSH*, dostępny w http://www.amg.gda.pl/siec/ssh.php, dostęp 05.09.2005 r.

[29] *Top-Level Domains (gTLDs)*, dostępny w http://www.icann.org/tlds/, dostęp 05.09.2005 r.

[30] *IPv6*, dostępny w http://www.ipv6.org, dostęp 05.09.2005 r.

[31] *HTTP Made Really Easy*, dostępny w http://www.jmarshall.com/easy/http/, dostęp 05.09.2005 r.

[32] *Windows 2000 Professional*, dostępny w http://www.microsoft.com/poland/windows2000/win2000prof, dostęp 05.09.2005 r.

[33] *Kerberos explained*, dostępny w http://www.microsoft.com/technet/prodtechnol/windows2000serv/maintain/security/kerberos.asp, dostęp 05.09.2005 r.

[34] *Mały leksykon kryptografii*, dostępny w http://www.minrank.org/krypto/kryptleks.html #bezpieczenstwo, dostęp 05.09.2005 r.

[35] „PCkurier", dostępny w http://www.pckurier.pl/webmaster/leksykon, dostęp 05.09.2005 r.

[36] *Multi Protocol Label Switching*, dostępny w http://technologie.pl/content/section/3/30/, dostęp 05.09.2005 r.

[38] Zieliński J., *Krótka historia początków Internetu*, dostępny w http://www.winter.pl/internet/krotka.html, dostęp 05.09.2005 r.

[37] *Wikipedia. Kategoria: kryptografia*, dostępny w http://pl.wikipedia.org/wiki/kategoria:kryptografia, dostęp 05.09.2005 r.

[39] Rafa J., *Telnet*, dostępny w http://www.wsp.krakow.pl/papers/telnet.html, dostęp 05.09.2005 r.

Spis treści