KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Politechnika Wrocławska

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Technologie Sieciowe

Plan

Literatura

- Andrew S. Tanenbaum:
  "Sieci komputerowe",
  wyd. IV, Helion, 2004.
- Douglas E. Comer:
  "Sieci komputerowe i intersieci",
  wyd. II, WNT, 2001
- Douglas E. Comer:
  "Sieci komputerowe TCP/IP", t. 1, 2, 3,
  WNT, 1997 (wyd. IV - oryg. 2000).
- Elliotte Rusty Harold:
  "Java. programowanie sieciowe",
  wyd. IV, RM, 2001.
- http://docs.oracle.com/javase/tutorial/networking/index.html
- Prezentowane przykłady przygotowano na podstawie dokumentacji języków: C, Perl, PHP, Java, JavaScript.

Sieci komputerowe

Sieci i zadania

Systemy złożone z pewnej liczby komunikujących się komputerów. Zdolność wymiany danych pomiędzy poszczególnymi komputerami.

Składniki

Wymiana danych pomiędzy komputerami sieci komputerowej jest realizowana za pomocą wielu elementów. Dwie grupy: `sprzęt` oraz `oprogramowanie`.

Protokoły sieciowe

Protokół

Zbiór zasad komunikowania się elementów funkcjonalnych sieci.

Zadania protokołów:

- identyfikacja elementów komunikujących
- podanie sposobu określania właściwego adresata,
- podanie sposobu rozpoczynania i kończenia transmisji,
- podanie sposobu przesyłania danych.

Podział

- Ze względu na sposób korzystania z połączeń podczas komunikacji między komputerami: sieci szkieletowe i sieci dostępu lokalnego.
- Ze względu na typ podsieci komunikacyjnej: sieci z transmisją punkt-punkt, charakterystyczną dla sieci rozległych, wielopunktową (ang. broadcast, multipoint), charakterystyczną dla sieci LAN, MAN oraz różnych rodzajów sieci radiowych i satelitarnych.
- Sieci szkieletowe (rozlegle): informacje wymieniane pomiędzy komputerami są przesyłane wewnątrz podsieci komunikacyjnej. Droga sygnału jest zestawem łączy i węzłów danej sieci.

Sieci szkieletowe

- Sieci szkieletowe: wiele linii, każda łączy parę komputerów. Jeśli dwa komputery nie są połączone bezpośrednio wspólną linią, a chcą wymieniać dane, to muszą się komunikować etapami, poprzez inne komputery, z którymi mają połączenie liniami transmisyjnymi.
- Jeśli dane są przesyłane z jednego komputera do innego przy pomocy kilku komputerów pośrednich, to na każdym etapie komunikacji komputery pośredniczące:
  - przesyłają dane do następnego etapu, jeśli odpowiednie linie nie są zajęte,
  - odbierają i zapamiętują dane do chwili, w której linie do następnego etapu zwolnią się.

Sieci LAN

- W sieciach dostępu lokalnego komputery komunikujące się wykorzystują tzw. zasadę rozsiewania informacji w jednym kanale transmisyjnym, do którego wszystkie komputery są przyłączone.
- Cechą takich sieci jest to, że w trakcie komunikacji tylko jeden komputer może nadawać wiadomości przez łącze transmisyjne. Pozostałe komputery mogą w tym czasie jedynie odbierać dane, a nadawać własne wiadomości mogą dopiero wtedy, kiedy kanał komunikacyjny nie jest zajęty. Komputery konkurują o dostęp do kanału komunikacyjnego.

WAN

Rozległa sieć komputerowa WAN - łączy urządzenia znacznie oddalone. Łączy zdalne sieci LAN. Swoim zasięgiem obejmuje rozległe obszary geograficzne (znaczne opóźnienia propagacyjne, większy koszt usług komunikacyjnych). Większa częstość występowania błędów transmisji, większa podatność na uszkodzenia łączy transmisyjnych niż w LAN.

Grafowy model sieci

- Działanie sieci komputerowych można analizować przy użyciu modeli zbudowanych w oparciu o teorię grafów, w których węzły grafu odpowiadają komputerom, a krawędzie grafu połączeniom pomiędzy komputerami.
- Siecią nazywamy parę `S = <G, H>` gdzie, `G` jest grafem, a `H` jest zbiorem funkcji przyporządkowujących każdemu łukowi liczby rzeczywiste zwane wagami łuku.
- Grafowy model sieci odzwierciedla techniczne właściwości sieci komputerowych. W modelu tym węzły grafu odpowiadają komputerom, a łuki pomiędzy węzłami, kanałom komunikacyjnym łączącym komputery.

Grafowy model łączy

- Łuk skierowany jest modelem `kanału simpleksowego` - przepływ informacji tylko w jedną stronę.
- Łuk nieskierowany modeluje `kanał dupleksowy` - możliwy jest równoczesny przepływ informacji w obie strony.
- Schemat rozmieszczenia łuków sieci określony przez graf `G` określa `strukturę topologiczną` sieci `S`. Graf skierowany określa strukturę sieci, której wszystkie kanały są simpleksowe, a graf nieskierowany określa strukturę sieci, której wszystkie kanały są dupleksowe.

Struktura

Funkcje ze zbioru H reprezentują właściwości kanałów komunikacyjnych takie jak niezawodność połączenia, przepustowość lub koszt.
Strukturę sieci opisuje się za pomocą macierzy kwadratowej sąsiedztwa węzłów przyjmując, że wartość elementu macierzy wynosi 1, jeśli graf zawiera dany łuk , albo 0, jeśli graf nie zawiera łuku.

Trasy

- Komunikacja między komputerami - zdolność do wymiany danych - proces, w którym jeden komputer wysyła łączami komunikacyjnymi do innego komputera, pewną liczbę bitów pogrupowanych w jednostki zwane pakietami.
- Węzeł w sieci wysyłający pakiety jest węzłem źródłowym lub źródłem, a węzeł docelowy odbierający adresowane do niego pakiety, węzłem końcowym lub ujściem.
- Komunikacja pomiędzy źródłem i ujściem w sieci - w grafie istnieje co najmniej jedna trasa od źródła do ujścia.

Parametry

Natężenia pakietów

Liczbę pakietów wysyłanych w ciągu sekundy, od węzła źródłowego do węzła końcowego nazywamy `natężeniem` strumienia pakietów pomiędzy tymi węzłami. (opis - macierz natężeń strumienia)

Przepustowość

Maksymalna liczbę bitów, którą można wprowadzić do łącza w ciągu sekundy (wynikającą z ograniczeń technicznych).

Opóźnienie

Czas, mierzony w sekundach, potrzebny na przesłanie pakietu z jednego komputera do innego.

Wskaźniki działania sieci

- Badania zachowania się sieci komputerowych w różnych warunkach prowadziły do opracowywania wskaźników działania sieci, takich jak np.: niezawodność sieci, średnie opóźnienie pakietu w sieci, koszt sieci.
- Wartości wskaźników zależą (m.in.) topologii, tzn. od rozmieszczenia komputerów i połączeń między nimi oraz od parametrów poszczególnych elementów sieci.

Opóźnienie

- Opóźnienie wynikające z czasu transmisji sygnału przewodem elektrycznym lub światłowodem nazywamy opóźnieniem propagacji.
- W sieciach lokalnych, z jednym wspólnym kanałem transmisyjnym, opóźnienie wynikające z konieczności odczekania na dostęp do łącza nazywane jest opóźnieniem dostępu.
- W sieciach szkieletowych, gdzie komunikacja odbywa się etapami, czas związany z wyborem następnego etapu dla pakietu przybywającego do węzła pośredniego, nazywany jest opóźnieniem przełączania, a czas związany z pozostawaniem pakietu w buforze węzła i oczekiwaniem na wysłanie, opóźnieniem kolejkowania.

Opóźnienie

W sieci o dużych wartościach elementów macierzy natężeń (intensywnie wykorzystywanej), opóźnienie w znacznej mierze wynika z czasu przebywania pakietów w buforach węzłów, w których kolejno oczekują one na nadanie do następnego etapu.

Średnie opóźnienie pakietu

średni czas przepływu pojedynczego pakietu od węzła źródłowego do ujściowego w sieci.

Komutacja

- Techniką komutacji - sposób zestawienia połączeń fizycznych (np. elektrycznych), logicznych w danej sieci komputerowej (sieci telekomunikacyjnej).
- Rodzaje: `komutacja kanałów`, `komórek`, `pakietów`.

Komutacja kanałów

`Komutacja kanałów` realizowaną była przez analogowe systemy telefoniczne.
Cechy:

- konieczność zestawiania połączenia przed zainicjowaniem przekazu,
- wyłączność węzłów końcowych na użytkowanie zestawionego połączenia,
- brak opóźnień kolejkowania w przekazie (są opóźnienia propagacji).

Komutacja pakietów i komórek

- Komutacja pakietów i komórek charakteryzuje się możliwością czasowego przechowywania informacji w węzłach podsieci komunikacyjnej, na trasie między stacjami źródłową i docelową.
- Komutacja pakietów jest bardzo rozpowszechnioną techniką, stosowaną w sieciach komputerowych.
- Wiadomości dzielone są na bloki określonej długości do których dodane są części organizacyjne. Tworzona w ten sposób jednostka danych nazywana jest pakietem bądź datagramem.
- Długości komórek są znacznie krótsze od typowych długości pakietów (kilkadziesiąt bajtów).

Zalety komutacji pakietów

- możliwość zajmowania i zwalniania medium komunikacyjnego zgodnie z potrzebami transmisyjnymi
- możliwość współużytkowania łącza przez pakiety pochodzące z różnych stacji źródłowych i kierowanych do różnych adresatów,
- możliwość obliczania funkcji kanałów (np. naliczania opłat) zgodnie z ilością przesłanych danych, a nie z czasem trwania połączenia oraz zazwyczaj
- możliwość obsługi ruchu interakcyjnego.

Problemy komutacji pakietów

- możliwość wystąpienia nagłego spiętrzenia ruchu na określonych trasach - zablokowanie węzłów i utrata pakietów - duże opóźnienia w ich transmisji,
- możliwość wystąpienia zmiany kolejności w dostarczaniu pakietów do węzła docelowego.

Fragmentacja

- W sieci z komutacją pakietów istnieją zwykle górne ograniczenie wielkości jednostkowego bloku przesyłanych danych. Długość tego bloku może być różna dla różnych sieci.
- W przypadku "przesyłania" wiadomości dłuższej niż maksymalna długość bloku, istnieje potrzeba podziału wiadomości na mniejsze "porcje", które następnie umieszczane są w pakietach i transmitowane przez sieć.

OSI

Organizacja ISO opracowała Model Referencyjny Połączonych Systemów Otwartych (model OSI):

- Ułatwienie realizacji otwartych połączeń systemów komputerowych
- Model jest zbiorem zasad komunikowania się urządzeń sieciowych.
- Siedem warstw, z których każda zbudowana jest na bazie warstwy poprzedniej.
- Model nie określa fizycznej budowy poszczególnych warstw, określa sposoby ich współpracy.
- Poszczególne warstwy sieci stanowią niezależne poziomy.
- Każda warstwa może być implementowana przez producenta niezależnie.

Warstwy OSI

- Komunikacja logiczna na poziomie odpowiadających sobie warstw.
- W rzeczywistej realizacji sieci komputerowej komunikacja odbywa się na poziomie warstwy fizycznej.
- Dane przekazywane są do sąsiedniej niższej warstwy - aż do warstwy fizycznej.
- Pomiędzy warstwami z wyjątkiem fizycznej istnieje komunikacja wirtualna możliwa dzięki istnieniu połączenia fizycznego.

Warstwa fizyczna

- Warstwa najniższa.
- Przesyłanie strumieni bitów.
- Konwersje bitów na sygnały.
- Określa parametry:
  - amplitudowe sygnału
  - czasowe sygnału,
  - wartości napięć,
  - fizyczne aspekty łączy,
  - znaczenie styków.

Warstwa łącza danych

Funkcje:

- Odbierania i nadawania.
- Końcową zgodność przesyłania danych - odbiór i konwersję strumienia bitów bez błędów (sumy kontrolne).
- Ramkowanie,tworzenie, rozpoznawanie granic ramki, przesyłanie bitów w tamce.
- Synchronizacja szybkości przesyłania danych.

Warstwa sieciowa

- Przesyłanie danych pomiędzy węzłami sieci.
- Wyznaczanie trasy przesyłu.
- Określanie drogi transmisji pakietu w oparciu o tablice tras.
- Możliwość dynamicznego określania trasy na bazie funkcji łuków (np. bieżących obciążeń linii).
- Brak wbudowanych mechanizmów korekcji błędów.
- Do łączenia z węzłów znajdującymi się poza lokalnymi segmentami sieci LAN.
- Własna architektura trasowania, niezależna od adresowania fizycznego warstwy 2.

Warstwa transportowa

- Końcową integralność transmisji.
- Niezawodność.
- Reakcja na przeciążenia.
- Pozytywne potwierdzenia.
- Retransmisja.
- Strumień oktetów.
- Identyfikacja oryginalnej sekwencji.
- Ustawienie w oryginalnej kolejności przed przekazaniem do warstwy sesji.

Warstwa sesji

- Zarządzanie przebiegiem komunikacji (sesją) podczas połączenia miedzy dwoma komputerami.
- Określa, czy komunikacja może zachodzić w jednym, czy obu kierunkach.
- Zarządzanie wykonywaniem bieżących żądań (wzorzec żądanie - odpowiedź).

Warstwa prezentacji

- Zarządzanie sposobem kodowania danych.
- Translacja między niezgodnymi schematami i formatami.
- Szyfrowania i rozszyfrowywanie wiadomości.

Warstwa aplikacji

- Warstwa najwyższa.
- Interfejs pomiędzy aplikacjami użytkownika a usługami sieci.
- Inicjuje sesje komunikacyjne.

Warstwa fizyczne w LAN - Ethernet

- Pierwsze podejście: Magistrala (szyna) składa się z pojedynczego kabla łączącego węzły.
- Kabel nie jest zarządzany przez żadne urządzenia zewnętrzne (brak nadzorcy).
- Kabel obsługuje jeden kanał (niektóre technologie oparte na magistrali korzystają z więcej niż jednego kabla, dzięki czemu obsługiwać mogą więcej niż jeden kanał).
- Na końcach magistrali znajdują się terminatory (oporniki), które powodują, że sygnał nie odbija się.
- Sygnał wychodzący ze stacji rozchodzi się w obydwu kierunkach.
- Wszystkie przyłączone do sieci urządzenia słuchają transmisji przesyłanych magistralą i odbierają pakiety do nich zaadresowane.

Kable

Skrętki

Nośniki dla sieci Ethernet obejmują kable nieekranowanej (UTP) i ekranowanej skrętki (STP) (np. kategorii 5). Uwaga na lab. wizyta w serwerowni.

LAN - Ethernet

- Ethernet jest powszechnie używaną techniką sieciową o topologii szynowej.
- Został on opracowany przez Xerox Corporation's Palo Alto Research Center we wczesnych latach siedemdziesiątych.
- Wtedy: sieć półdupleksowa, urządzenia łączone za pomocą grubego kabla koncentrycznego.
- Prędkość przesyłania sygnału wynosiła 10 Mbps.

LAN - Ethernet

- Ethernet jest najpopularniejszą topologią sieci LAN.
- Możliwość współpracy z szerokim zakresem typów okablowania.
- Niskie ceny sprzętu.
- Klasyczne podejście - algorytm CSMA/CD (Carrier Sense Multiple Access with Collision Detection).

Collision Detection

- Kiedy jednocześnie transmitują dwie stacje, ma miejsce kolizja. Kolizja - zakłócenie fal elektromagnetycznych.
- Wszystkie stacje objęte kolizją sygnalizują, że nastąpiła kolizja.
- Stacje, odczekują jakiś czas i następnie ponownie przesyłają dane.
- Każda stacja odpowiedzialna jest za własny interwał oczekiwania przed ponowną transmisją (każda stacja odczekuje losowy czas, przez co zmniejsza się ryzyko kolejnej kolizji).
- Jeśli nastąpi kolejna kolizja, każda stacja będzie musiała podwoić przedział z którego losuje czas oczekiwania przed ponowną próbą (dwójkowy wykładniczy czas oczekiwania).

Tryb nasłuchu - *promicious mode*

- Każdy węzeł stale monitoruje transmisje w łączu.
- Istnieje możliwość takiej konfiguracji stacji roboczej, która pozwalałaby na odczytywanie wszystkich otrzymywanych informacji.
- Umożliwienie funkcjonowanie specjalnym programom (tzw. analizatorom sieci).
- Stosowany przez administratorów w celu monitorowania sieci z jednej stacji centralnej.
- Nieuczciwe osoby mogą podsłuchiwać przesyłane wiadomości.

Ramkowanie

- Rozpychanie bajtów (bitów).
- Błędy transmisji.
- Bity parzystości (kontrola parzystości).
- Kody korekcyjne i kody detekcyjne.
- CRC.
- Lokalność odwołań (miejscowa i czasowa).

Ramka Ethernet

Ramka Ethernet to ustruktualizowany zbiór bitów przesyłanych w nośniku za pomocą sygnału. Wielkość od 64 do 1518 oktetów i zorganizowana jest w cztery sekcje: preambuła, nagłówek, dane, ciąg kontrolny.

- Preambuła to zdefiniowana seria bitów, która informuje odbiorcę o nadawaniu. Standardowa preambuła składa się z ośmiu znaków. Służy do zsynchronizowania nadajnika i odbiornika. Preambuła ma postać siedmiu oktetów postaci 10101010 i jednego 10101011.
- Nagłówek: nadawca, odbiorca, wielkość danych, itp.
- Przesyłane dane. Ramka Ethernet ma długość od 64 do 1518 oktetów (bez preambuły), długość pola danych: od 46 do 1500 oktetów. Jeżeli dane miałyby być krótsze niż 46 oktetów zostają uzupełnione zerami do 46 oktetów.

Adresy Ethernet

Adresy są fizycznymi kodami adresowymi urządzeń (MAC).

- Sprzętowy adres przeznaczenia. Każda karta ma unikalny 6 oktetowy adres, z których 3 pierwsze oktety oznaczają producenta.
- Sprzętowy adres nadawcy. Każda karta ma unikalny 6 oktetowy adres, z których 3 pierwsze oktety oznaczają producenta.
- Adres: FF-FF-FF-FF-FF-FF - adres typu broadcast - wskazuje, że przekazywany komunikat przeznaczony jest dla wszystkich stacji danej sieci. Wszystkie stacje odbierające ten pakiet powinny odczytać znajdujące się w nim dane.

CRC

- 4-bajtowa suma kontrolna ramki.
- Kod wielomianowy.
- Algorytm: cykliczna kontrola nadmiarowa (CRC).
- Podczas wysyłania tworzony jest czterobajtowy.
- Numer zależny od danych.
- System odbiorczy, weryfikuje ciąg kontrolny ramki w oparciu o otrzymane dane.
- Jeśli jest zgodność, system zakłada, że ramka nie zawiera błędów i przystępuje do przetwarzania informacji.

Adresacja IP

- Adres IP jest 32-bitowym identyfikatorem, (adresacja abstrakcyjna - adres przyznawany administracyjnie).
- Zapis w notacji dziesiętnej z kropkami, (ang. dot notation) - cztery liczby dziesiętne rozdzielone kropkami.
- Każda z liczb odwzorowuje 8 bitów.

10	.	5	.	254	.	1
00001010		00000101		11111110		00000001

- Adres IP dzieli się na dwie części: numer sieci, numer węzła

Klasy adresów IP

Początkowo wykorzystywano klasy adresów:

- `Klasa A` jednobajtowy numer sieci i trzy bajty przeznaczone na numer węzła. Najstarszym bitem adresu należącego do klasy A jest 0.
- `Klasa B` dwa bajty na numer węzła i sieci. Dwa pierwsze bity adresu z klasy B to '10'.
- `Klasa C` 24 bity na numer sieci i osiem na numer węzła. Najwyższe bity adresu klasy C to '110'.
- `Klasa D` używana w komunikacji grupowej (ang. multicasting). Każdy adres identyfikuje inną grupę i rozpoczyna się bitami '1110'. Adres z klasy D jest dynamicznie przypisywany za pomocą IGMP (ang. Internet Group Management Protocol) węzłom zgłaszającym akces do danej grupy.
- `Klasa E` jest przeznaczona do badań i eksperymentów. Pakiety zawierające adresy, które rozpoczynają się bitami '1111' nie powinny być przetwarzane przez zwykłe węzły.

Podsieci

- Podział - wyodrębnienie w adresach IP części sieciowej i części hosta.
- Podsieć jest rozszerzeniem pojęcia podziału pól bitowych, jak w klasach A, B i C.
- Część sieciowa jest rozszerzana tak, by zawierała niektóre bity z części hosta, liczba bitów, interpretowana jako numer podsieci, jest określona przez tzw. maskę podsieci lub maskę sieci
- Podsieci są generowane przez administratora, określają wewnętrzny podział sieci (administracyjny, lokalizacyjny, itp.).

Maska sieci (podsieci)

32-bitowa liczba określająca maskę bitową dla części sieciowej adresu IP, Jeśli bit w masce jest jedynką, to odpowiadający mu bit w adresie IP jest interpretowany jako bit adresu sieci Jeśli bit w masce jest zerem to to odpowiadający mu bit w adresie IP należy do części adresu określającej komputer,

```
11000000.10101000.01111011.10000100  Adres IP (192.168.123.132)
11111111.11111111.11111111.00000000  Maska podsieci (255.255.255.0)

11000000.10101000.01111011.00000000  Adres sieci (192.168.123.0)
00000000.00000000.00000000.10000100  Adres hosta (000.000.000.132)
```

Maska sieci (podsieci)

Zwykle dziesiętne maski podsieci po przekształceniu na liczby dwójkowe mają z lewej strony same jedynki, a z prawej same zera. Np.:

Dziesiętnie	Dwójkowo
255.255.255.192	1111111.11111111.1111111.11000000
255.255.255.224	1111111.11111111.1111111.11100000

Zakresy adresów IP zarezerwowanych dla sieci prywatnych, wewnętrznych: A: 10. 0.0.0 - 10.255.255.255 B: 172.16.0.0 - 172.31.0.0 C: 192.168.0.0 - 192.168.255.0

Pakiet - Datagram

Jednostka przesyłanych danych w warstwie IP. Nagłówek datagramu: nadawca, odbiorca, pole typu (identyfikacja zawartości datagramu), dane fragmentacji, opcje. Format pakietu nie jest uwarunkowane sprzętowo.

Adresy IP - adresy MAC

Problem w sieci LAN:

- Nadawca zna tylko adres IP.
- Nadawca musi wysłać ramkę na adres sprzętowy.

Protokół ARP

Zasada działania:

- Nadawca wysyła ramkę rozgłoszeniową z zapytaniem ARP: "Kto ma zadany IP".
- Odbierają wszyscy.
- Odpowiada węzeł o zadanym IP odpowiedzią ARP: "Ja o takim MAC posiadam zadany IP"
- Dalsza komunikacja za pomoca ramek do konkretnych adresatów.
- Lokalne przechowywanie odwzorowań.

Uwaga! Pokazać przechwycone ramki w programie WireShark.

Protokół kontroli błędów ICMP

- Protokół komunikatów kontrolnych internetu ICMP (ang. Internet Control Message Protocol) umożliwia routerom oznajmianie o błędach oraz udostępnianie informacji o niespodziewanych sytuacjach.
- Wymagana część IP, realizowany przez każdą implementację IP.
- Komunikaty ICMP podróżują w intersieci w części w częściach datagramów IP przeznaczonych na dane.
- Odbiorcą końcowym komunikatu ICMP jest oprogramowanie IP. Komunikat błędu ICMP jest obsługiwany przez moduł oprogramowania ICMP (gdy problem spowodował protokół wyższego rzędu lub program użytkowy, przekaże informacje do odpowiedniego modułu).
- ICMP umożliwia ruterom wysyłanie komunikatów - ale - każda maszyna może wysyłać komunikaty ICMP do dowolnej innej.

ICMP

- W specyfikacji określono sposoby korzystania z ICMP, sugestie na temat działań w odpowiedzi na komunikaty o błędach.
- ICMP nie dla każdego możliwego błędu wyszczególnia działania, jakie mają być zapoczątkowane.
- Gdy datagram powoduje błąd, oprogramowanie ICMP może powiadomić pierwotnego nadawcę o przyczynie.
- Nadawca musi otrzymaną informację przekazać programowi użytkownika, albo podjąć inne działanie rozwiązujące problem.

TCP - Transmission Control Protocol

- Komponent warstwy transportowej - Protokół sterujący transmisją TCP jest komponentem warstwy transportowej i w hierarchii protokołów jest ponad protokołem IP.
- Niezawodna wymiana danych - realizuje usługę niezawodnej wymiany danych w łączach zorientowanych na połączenie (udostępnienie strumienia). Aplikacji sieciowe wykorzystują strumień (tak jak w przypadku pracy w lokalnym systemie plików).
- Wykorzystanie protokołu IP - wykorzystuje protokół IP do przesyłania informacji poprzez sieć. IP oferuje efektywny mechanizm trasowania. Wady IP związane z niezawodnością eliminuje nadrzędny protokół TCP. Dane TCP (oficjalnie nazwane segmentami) są umieszczane w datagramach IP.

Cechy TCP

- Połączenie TCP realizowane jest w trybie full-duplex - poprzez dwukierunkowe kanały wirtualne - sprawia wrażenie realizowanego poprzez dwa niezależne kanały nadawania i odbioru.
- Odbiorca może potwierdzić odebrane datagramy (ACK - acknowledge), aby nadawca mógł być pewny, że dotarły one na miejsce. Potwierdza się najdłuższy ciągły prefiks. Retransmisja
- Idea przesuwającego się okna.
- Kontrola przepływu (flow control) - możliwość aktywnej współpracy węzłów w czasie transmisji danych - pozwala zapobiec nadmiarem i utracie datagramów. Dostosowanie do poziomu ruchu w sieci. Sterowanie rozmiarem okna i czasem retransmisji.
- Numeracja oktetów. Strumień.

Nawiązywanie i przerywanie połączeń

- TCP realizuje sesję, w której następuje bezbłędna transmisja danych.
- Sesja rozpoczyna się nawiązaniem połączenia, następnie przesyłane są dane, po czym połączenie zostaje zamknięte.
- Nawiązanie połączenia polega na przesłaniu żądania połączenia do węzła przeznaczenia.
- Jeśli realizacja połączenia jest możliwa, węzeł odpowiada komunikatem potwierdzającym.
- Oba komputery mogą ustalać parametry połączenia (przesłane w polu opcji segmentu TCP).

Problemy i algorytmy

- Minimalizowanie opóźnienia.
- Retransmisja z adaptacją.
- Algorytm Karna.
- Algorytm Nagle'a.
- Syndrom "głupiego okna".
- Algorytm Clarka.
- Okno przeciążeniowe.
- Wolny start.

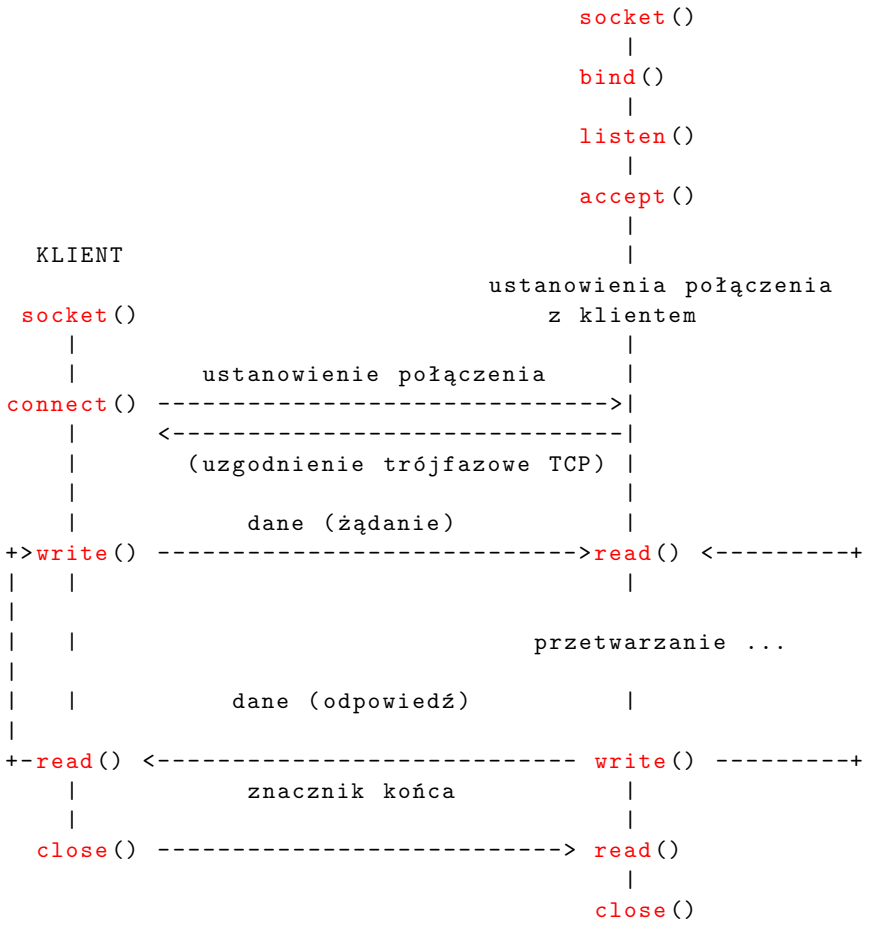
Programowanie sieciowe - interfejs gniazd. Serwer w C - przygotowanie

```
main() {

 int s, z, dl, c;
 struct sockaddr_in adr_serw;
 struct sockaddr_in adr_klnt;

 s = socket(PF_INET, SOCK_STREAM, 0);
 adr_serw.sin_family = PF_INET;
 adr_serw.sin_port = htons(9000);
 adr_serw.sin_addr.s_addr = inet_addr("127.0.0.23");
 dl = sizeof adr_serw;

 z = bind(s, (struct sockaddr *)&adr_serw, dl);

 z = listen(s, 10);              /* nasluchiwanie */

 ....
}
```

Programowanie sieciowe - Serwer w C - pętla akceptująca.

```
main() {

 ...
 z = listen(s, 10);              /* nasluchiwanie */


 for (;;) {                      /* petla serwera */
  dl = sizeof adr_klnt;
  c = accept(s, (struct sockaddr *)&adr_klnt, &dl);
                                 /* akceptowanie polaczenie */
    ....

  close(c);                      /* zamkniecie polaczenia */
 }

 return 0;
}
```

Programowanie sieciowe - schemat



Programowanie sieciowe - serwer w Perl'u

Podobnie jak w języku C

```
use Socket;

socket(SERVER, PF_INET, SOCK_STREAM, getprotobyname('tcp'));
setsockopt(SERVER, SOL_SOCKET, SO_REUSEADDR, 1);
$my_addr = sockaddr_in($server_port, INADDR_ANY);

bind(SERVER, $my_addr) || die "Blad ...";

listen(SERVER, SOMAXCONN) || die "Blad ...";

while (accept(CLIENT, SERVER)) {
# ...
}

close(SERVER);
```

Programowanie sieciowe - klient w Perl'u

```
use IO::Socket;
$socket = IO::Socket::INET->new(PeerAddr => $remote_host,
                                PeerPort => $remote_port,
                                   Proto => "tcp",
                                    Type => SOCK_STREAM)
      or die "Brak polaczenia : $@\n";

# ...
print $socket "Komunikat\n";
$answer = <$socket>;

close($socket);
```

Serwer HTTP w Perl'u

```
use HTTP::Daemon;
use HTTP::Status;
#use IO::File;

my $d = HTTP::Daemon->new(
         LocalAddr => 'lukim',
         LocalPort => 4321,
     )|| die;

print "Adres serwera: <URL:", $d->url, ">\n";


while (my $c = $d->accept) {
    while (my $r = $c->get_request) {
        if ($r->method eq 'GET') {

            $file_s= "./index.html";     # index.html - jakis istniejacy plik
            $c->send_file_response($file_s);

        }
        else {
            $c->send_error(RC_FORBIDDEN)
        }

    }
    $c->close;
    undef($c);
}
```

Prosty klient HTTP w Perl'u

```
use LWP::Simple;

$content = get("http://www.chip.pl/");
die "Blad pobierania" unless defined $content;

print $content;
```

Klient HTTP w Perl'u

```
use LWP::UserAgent;
$ua = LWP::UserAgent->new;

$ua->agent("FajnaPrzegladarka");

$req = HTTP::Request->new(GET => 'http://www.chip.pl');
$req->header('Accept' => 'text/html');

# wysylanie zadania
$res = $ua->request($req);

# sprawdzenie
if ($res->is_success) {
    print $res->content;
}
else {
    print "Blad: " . $res->status_line . "\n";
}
```

Obsługa sescji w PHP

```
<?php
session_start();
if (empty($_SESSION['count'])) {
        $_SESSION['count'] = 1;
} else {
        $_SESSION['count']++;
}
?>
<p>
Witaj na stronie. Byłeś tu tyle razy ile wskazuje licznik:
  <?php echo $_SESSION['count']; ?>.
</p>

<p>
Aby kontynuowac,
  <a href="ex2.php?<?php␣echo␣htmlspecialchars(SID);␣?>">
  kliknij dowiazanie</a>.
</p>
```

Obsługa sescji w PHP

```
<html>
 <head>
  <title>PHP Test</title>
 </head>
 <body>
<?php
setcookie("cookie[jeden]", "ciastko_raz");
setcookie("cookie[dwa]", "ciastko_dwa");

if (isset($_COOKIE['cookie'])) {
    foreach ($_COOKIE['cookie'] as $klucz => $wartosc) {
        echo "$klucz : $wartosc <br />\n";
    }
}
?>
 </body>
</html>
```

Obsługa przesyłania w PHP - formularz

```
<html>
 <head>
  <title>Test w PHP</title>
 </head>
 <body>
 <form enctype="multipart/form-data"
       action="http://127.0.0.1:8080/t.php"
       method="post">
   <input type="hidden"
          name="MAX_FILE_SIZE"
          value="300000" />
   Wyślij plik:<br />
   <input name="userfile" type="file" /><br />
   <input type="submit" value="Send␣File" />
 </form>
 ...
```

Obsługa przesyłania - PHP

```
...
<?php
$uploaddir = 'c:\\!!\\';
$uploadfile = $uploaddir . $_FILES['userfile']['name'];
print "<pre>";
if (move_uploaded_file($_FILES['userfile']['tmp_name'],
    $uploadfile)) {
      print "Informacje:\n";
      print_r($_FILES);
} else {
    print "Informacje:\n";
    print_r($_FILES);
}
print "</pre>";
?>
</body>
</html>
```

AJAX - obiekt XMLHttpRequest

```
<script type="text/javascript">
function ajaxFunction()
{
var xmlHttp;
try      // Firefox, itp...
  {
  xmlHttp=new XMLHttpRequest();
  }
catch (e)
  {
  try    // IE
    {
    xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
    }
  catch (e)
    {
    try
      {
      xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
      }
    catch (e)
      {
      alert("Your browser does not support AJAX!");
      return false;
      }
    }
  }
  ...
</script>
```

AJAX - funkcja onreadystatechange

```
<script type="text/javascript">
  ...
  xmlHttp.onreadystatechange=function()
    {
    if(xmlHttp.readyState==4)
      {
          document.getElementById("txtHint").innerHTML=xmlH
      }
    }
  xmlHttp.open("GET","c.txt",true);
  xmlHttp.send(null);
  }
</script>
```

Kod strony

```
<body>
<script type="text/javascript">
  ....
</script>

<form name="myForm">
Name: <input type="text"
onkeyup="ajaxFunction();" name="username" />
</form>

<p>Wyniki:
  <pre>
    <div id="txtHint"> </div>
  </pre>
</p>
</body>
```

Usługa WebService - Perl

```
#!perl -w
use SOAP::Transport::HTTP;
use Demo;

my $daemon = SOAP::Transport::HTTP::Daemon
  -> new (LocalPort => 5555)
  -> dispatch_to('Demo')
;

print "WebService działa na: ", $daemon->url, "\n";
$daemon->handle;
```

Klient usługi WebService - Perl

```
use SOAP::Lite +autodispatch =>

uri    => 'http://localhost:5555/Demo',
proxy  => 'http://localhost:5555/';

print witaj();
print "\n".dc2f(20);
```