# Sieci komputerowe

* Aplikacji
* Warstwa najwyższa w modelach OSI/ISO i TCP/IP
* Komunikuje się z użytkownikiem
* Obejmuje programy użytkowe
* Telnet
* FTP
* SMTP
* DNS
* HTTP
* Prezentacji
* Odpowiada za:
* Kodowanie danych z warstwy aplikacji
* Kompresję
* Szyfrowanie
* Przykładowe protokoły
* ASCII, UTF
* JPEG, GIF, PNG
* MPEG, AVI
* WAV, MP3
* Sesji
* Kontroluje dialog między komunikującymi się aplikacjami
* Pozwala na nawiązanie sesji i jej zakończenie
* Pilnuje, czy sesja nie wygasła, lub czy ktoś nie próbuje się pod nią podszyć
* Transportu
* Sieci
* Danych
* Fizyczna

# Modele komunikacji w sieciach komputerowych

## Model klient serwer

* Najstarszy i najpopularniejszy model komunikacji
* Jeden komputer, zawierający zasoby, pełni rolę serwera
* Pozostałe komputery to klienci, którzy korzystają z zasobów serwera
* Najwygodniejszy z modeli
* Cetralnie zarządzanie zasobami
* Cetralne zarządzanie dostępem do zasobów
* AAA (Authentication, Authorization, Accounting), polityki bezpieczeństwa

Operacje wykonywane przez klienta

* Pobieranie danych
* Przesyłanie danych
* Operacje na danych

## Model peer to peer

* Sieci peer-to-peer, to sieci w których mamy do czynienia z decentralizacją usług i zasobów
* Informacje mogą się znajdować w dowolnym miejscu sieci (na dowolnym urządzeniu)
* Problem z uwierzytelnianiem – przeważnie brak centralnego uwierzytelniania
  + Konieczność konfiguracji dostępu (np. konta) na każdym z urządzeń

Aplikacje p2p

* Aplikacje (głównie działające w sieciach p2p), w których każde urządzenie może działać jako klient jak i serwer
* Czasem mamy do czynienia z modelem hybrydowym, np. centralny katalog (usług, plikó)
* Przykłady:
  + Współdzielenie plików i drukarek w otoczeniu sieciowym Windows
  + Aplikacje P2P typu Emule, BitTorrent, Gnutella

# Podstawowe protokoły warstwy aplikacji

## Numery portów

* Porty odnoszą się do warstwy transportowej
* Służą do identyfikacji aplikacji, które obsługują połączenia
* „Dobrze znane porty” – ściśle określone, standardowe numery portów, które mówią nam, jaka aplikacja obsługuje połączenia przychodzące na ten port (np. WWW = 80)

## Domain Name System (DNS)

* Łatwiej zapamiętać napis, niż ciąg cyfr (adres IP)
* Odwzorowuje nazwę DNS na adres sieciowy
* Przykład: [www.pkp.pl](http://www.pkp.pl)
  + Napisy oddzielone kropkami
  + Hierarchiczność nazw
* Domena pierwszego poziomu: .pl
* Domena drugiego poziomu: .pkp.pl

## Strefy i ich obsługa

* Strefa (zone) jest częścią drzewa DNS, która jest oddzielnie administrowana
  + Może być podzielona na mniejsze obszary – następuje wtedy delegacja odpowiedzialności

## Zarządzanie

Domenami najwyższego poziomu zarządza ICANN – The Internet Corporation for Assigned Names and Numbers

Domeny najwyższego poziomu (Top-Level Domains) podzielone na dwie podstawowe kategorie

* Domeny ogólne, czyli rodzajowe (.com, .org, .net, .edu, .gov, .mil, .int)
* Domeny krajowe/regionalne

## Zapytania rekurencyjne

Jaki adres IP ma komputer www.up.krakow.pl ?

Serwery, które nie znają tej domeny przesyłają zapytanie do następnych, odpowiedzi wracają tą samą drogą.

## Zapytanie iteracyjne

Jaki adres IP ma komputer [www.up.krakow.pl](http://www.up.krakow.pl)?

Serwery, kolejno odpytywane są o znajomość domeny i bezpośrednio „podpowiadają” propozycję odpytania następnego serwera DNS.

## Zapytania – mieszane

Jaki adres IP ma komputer [www.up.krakow.pl](http://www.up.krakow.pl)?

* Decyzję podejmuje oprogramowanie klienta
* Serwer jest konfigurowalny
  + Serwer główny nigdy nie jest rekurencyjny

## Zapytanie proste

Najczęściej występujące zapytanie: o adres IP konkretnego urządzenia

## Przeznaczenie DNS

* DNS nie jest usługą wyszukiwania
  + Nie ma żadnych znaków specjalnych, które reprezentowałyby w zapytaniu dowolny ciąg znaków
  + Nie ma możliwości odnalezienia „najbliższego dopasowania”

## Pamięć podręczna

Wszystkie serwery DNS stosują pamięć podręczną aby zredukować wymianę komunikatów DNS i zwiększyć efektywność działania.

Serwer udziela odpowiedzi **autorytatywne**, które pochodzą od serwerów z odpowiedniej domeny

* Na zewnątrz nie widać, który serwer jest pierwszo-, a który drugoplanowy
* Serwer autorytatywny posiada plik strefy (zone file)

Serwer udziela odpowiedzi nieautorytatywne pochodzące z pamięci podręcznej zawierające informacje na temat serwera

* Klient ma możliwość „osobistej” kontroli nad uzyskaną odpowiedzią
* Wraz z odpowiedzią autorytatywną serwer otrzymuje czas ważności odpowiedzi TTL określający, jak długo należy przechowywać daną informację w pamięci podręcznej

## Rodzaje zapytań

Inne typy

* MX – serwer poczty dla określonej poczty
  + maciek@poczta.pl -> mail.poczta.pl
* NS – autorytatywny serwer DNS dla danej domeny
* CNAME – nazwa kanoniczna (alias)
* HINFO – informacje na temat komputera
* Inne

## Rodzaje zasobów DNS

* Name – definiowana nazwa domenowa
* TTL – (time to live) czas w sekundach, który dany rekord jest ważny
* Class – identyfikuje rodzinę protokołów DNS. W internecie wartość IN
* Type – identyfikuje typ zasobu w rekordzie; ka
* Rdata – wartość zależna od typu

# Hypertext Transfer Protocol

* Przesyłanie zróżnicowanych rodzajów danych – zasobów
  + Strony HTML
  + Pliki graficzne, dane multimedialne
  + Aplikacje
  + Inne
* Zasoby identyfikowane przez URL (Uniform Resource Locator)

## HTTP – cechy

* Protokół klient-serwer
  + Serwer
    - Serwer www
    - httpd
  + Klient
    - Najczęściej przeglądarka HTML
    - Specjalizowane aplikacje wykorzystujące http do transferu danych
* Protokół bezstanowy i bezpołączeniowy
  + Działa w oparciu o model żądanie/odpowiedź
  + Po dostarczeniu danych połączenie najczęściej jest zamykane

## Pakiet HTTP

* Nie ma ścisłego podziału na pola
* Komendy oddzielone są kończem linii
* Postać
  + Typ operacji, jedna linia
  + Zero lub więcej linii z parametrami postaci: nazwa:wartość
  + …

## Żądania i odpowiedzi HTTP

* Żądania HTTP
  + GET
  + POST
  + HEAD
  + Inne – PUT, DELETE
* Odpowiedzi
  + Kod odpowiedzi + tekst
* Po uzyskaniu odpowiedzi, połączenie TCP między klientem a serwerem najczęściej jest zamykane

## Żadania HTTP – GET

* Używane najczęściej
* Ciąg znaków identyikujący zasób na sewerze
  + Najprościej: statyczny zasób serwera
  + Parametry do skryptu lub bazy danych
* Stosowane do przesyłania małych ilości informacji

## Żadania HTTP – POST

* Żądanie nie jest zawarte w URL lecz w samym ciele informacji
* Cżąsto …

## Odpowiedzi HTTP

* 1xx – informacja
* 2xx - Powodzenie, żądanie zrozumiane i zaakceptowane
* 3xx – muisi zostać podjęta dalsza akcja
* 4xx- błąd po stronie klienta
* 5xx – błąd po stronie serwera

## Protokoły przesyłania poczty