

# Aplikační vrstva

PB002: Základy informačních technologií

Eva Hladká

*Slidy připravil: Eva Hladká a Tomáš Rebok*

Fakulta informatiky Masarykovy univerzity

jaro 2018

# Struktura přednášky

- 1 Přehled
- 2 Úvod
- 3 Základní členění aplikací
  - Komunikační modely – Client-Server vs. Peer-to-peer
  - Přístup k informacím – Pull model vs. Push model
  - Nároky na počítačovou síť – nízké vs. vysoké
- 4 Vybrané síťové aplikace
  - Jmenná služba – DNS
  - World Wide Web – HTTP
  - Elektronická pošta – SMTP
  - Přenos souborů – FTP
  - Multimediální přenosy v datových sítích

# Struktura přednášky

## 1 Přehled

## 2 Úvod

## 3 Základní členění aplikací

- Komunikační modely – Client-Server vs. Peer-to-peer
- Přístup k informacím – Pull model vs. Push model
- Nároky na počítačovou síť – nízké vs. vysoké

## 4 Vybrané síťové aplikace

- Jmenná služba – DNS
- World Wide Web – HTTP
- Elektronická pošta – SMTP
- Přenos souborů – FTP
- Multimediální přenosy v datových sítích

# L7. Aplikační vrstva – Přehled

## ISO / OSI



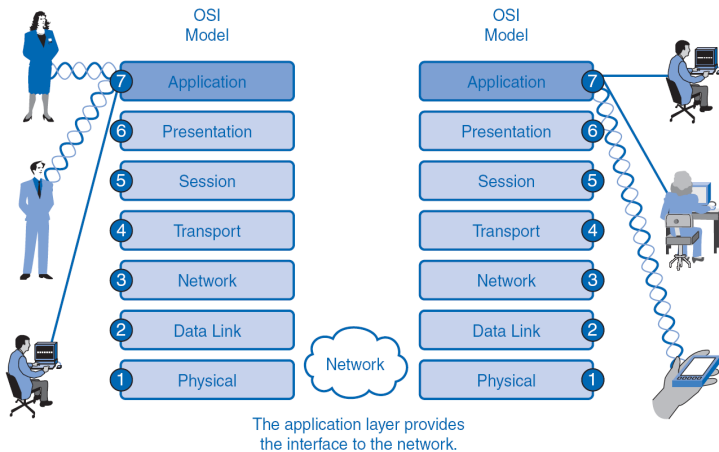
### Proč nestačí L4?

- z pohledu sítě stačí, z pohledu uživatele potřebujeme síťové aplikace

### Co nás nyní čeká. . .

- představení L7
- základní členění aplikací
- vybrané síťové aplikace

# L7 z pohledu sítě – kde se pohybujeme?



- aplikační programy – interface pro uživatele

# Struktura přednášky

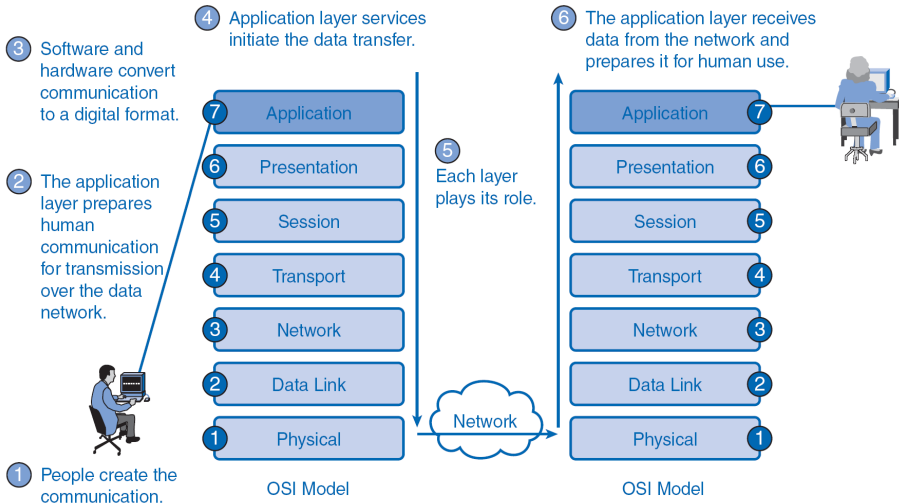
- 1 Přehled
- 2 Úvod
- 3 Základní členění aplikací
  - Komunikační modely – Client-Server vs. Peer-to-peer
  - Přístup k informacím – Pull model vs. Push model
  - Nároky na počítačovou síť – nízké vs. vysoké
- 4 Vybrané síťové aplikace
  - Jmenná služba – DNS
  - World Wide Web – HTTP
  - Elektronická pošta – SMTP
  - Přenos souborů – FTP
  - Multimediální přenosy v datových sítích

# Úvod I.

## aplikační vrstva:

- poskytuje služby pro *uživatele*:
  - aplikační programy (aplikace) specifické pro požadovaný účel
    - např. elektronická pošta, WWW, DNS, atd. atd.
  - aplikace = hlavní smysl existence počítačových sítí
- zahrnuje *síťové aplikace/programy* a *aplikační protokoly*
  - aplikační protokoly (HTTP, SMTP, atd.) jsou **součástí** síťových aplikací (web, email)
    - nejedná se o aplikace samotné
    - protokoly definují formu komunikace mezi komunikujícími aplikacemi
  - aplikační protokoly definují:
    - typy zpráv, které si aplikace předávají (*request/response*)
    - syntaxi přenášených zpráv
    - sémantiku přenášených zpráv (jednotlivých polí)
    - pravidla, kdy a jak aplikace zprávy vysílají

# Úvod II.





# Struktura přednášky

- 1 Přehled
- 2 Úvod
- 3 Základní členění aplikací
  - Komunikační modely – Client-Server vs. Peer-to-peer
  - Přístup k informacím – Pull model vs. Push model
  - Nároky na počítačovou síť – nízké vs. vysoké
- 4 Vybrané síťové aplikace
  - Jmenná služba – DNS
  - World Wide Web – HTTP
  - Elektronická pošta – SMTP
  - Přenos souborů – FTP
  - Multimediální přenosy v datových sítích

# Základní členění aplikací

*Dle využitého komunikačního modelu:*

- Client-Server model
- Peer-to-peer model

*Dle přístupu k informacím:*

- pull model
- push model

*Dle nároků na počítačovou síť:*

- aplikace s nízkými nároky na přenosovou síť
- aplikace s vysokými nároky na přenosovou síť

# Komunikační modely – *Client-Server* vs. *Peer-to-peer* I.

## Client-Server

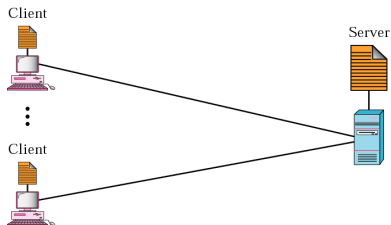
- komunikace iniciována klientem (klient = aplikační program ovládaný uživatelem)
- po ustavení komunikačního kanálu klient zasílá požadavky na server, ten mu odpovídá (mechanismus *request-response*)
- po ukončení komunikace je komunikační kanál uzavřen
- (centralizace zdrojů)
  
- valná většina aplikací v Internetu (WWW, FTP, DNS, SSH, ...)

## Peer-to-peer

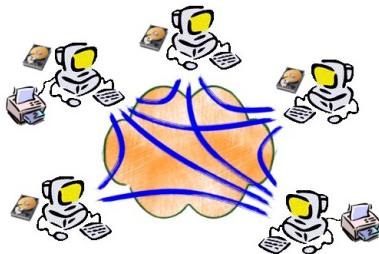
- jednotliví klienti spolu komunikují přímo (uzly jsou si rovnocenné)
- každý uzel poskytuje své zdroje (výpočetní síla, úložná kapacita, atp.) ostatním uzlům
- každý uzel využívá zdrojů poskytovaných ostatními uzly
- (decentralizace zdrojů)
  
- např. sdílení souborů (Gnutella, G2, FastTrack), Skype, VoIP, atp.

# Komunikační modely – *Client-Server* vs. *Peer-to-peer* II.

## Client-Server



## Peer-to-peer



# Komunikační modely – *Client-Server*

## Tenký (*Thin*) vs. Tlustý (*Fat*) klient

### Tenký (**Thin**) klient

- aplikace, u nichž se na straně klienta vykonává minimum aplikační logiky (většina se vykonává na straně serveru)
  - větší hardwarové nároky na stranu serveru a na komunikaci
- + jednodušší, menší nároky na HW (může tak být levnější)
- – menší škálovatelnost (příliš mnoho práce dělá server), většinou vyšší objemy přenášených dat, existence *Single point of failure* (server)
- *příklad*: vzdálené terminály

### Tlustý (**Fat**) klient

- přesný opak tenkého klienta – většina aplikační logiky se vykonává na straně klienta
  - větší hardwarové i softwarové nároky na klienta
- + menší nároky na server (⇒ dobrá škálovatelnost), většinou nižší objem přenesených dat, možnost práce *offline*
- – komplexní provedení i instalace, značná spotřeba lokálních zdrojů (CPU, paměť, disk)
- *příklad*: Firefox

# Přístup k informacím – *Pull model vs. Push model*

## Pull model

- přenos dat iniciován klientem (forma požadavek-odpověď)
- např. webové prohlížeče
- *vlastnosti:*
  - asymetrický datový tok
  - rozmanité požadavky na propustnost

## Push model

- přenos dat iniciován serverem automaticky na základě znalosti uživatelského profilu (požadavků)
- např. streaming multimedií (IPTV)
- *vlastnosti:*
  - jednosměrný datový tok
  - definované (a stálé) požadavky na propustnost (a zpoždění, jitter, atp.)

## Nároky na počítačovou síť – nízké vs. vysoké

Základní parametry sítě pohledem aplikací:

- *ztrátovost (loss)* – pravděpodobnost ztráty (poškození) přenášených dat
- *propustnost (bandwidth)* – objem přenesených dat za časovou jednotku
- *časová omezení (timing)* – *zpoždění (delay)* (doba nutná pro přenos dat po síti) a *kolísání zpoždění (jitter)*

| Application            | Data Loss     | Bandwidth  | Time sensitive?    |
|------------------------|---------------|--|--------------------|
| file transfer          | no loss       | elastic  | no                 |
| electronic mail        | no loss       | elastic  | no                 |
| Web documents          | no loss       | elastic  | no                 |
| real-time audio/video  | loss-tolerant | audio: few Kbps to 1Mbps<br>video: 10's Kbps to 5 Mbps | yes: 100's of msec |
| stored audio/video     | loss-tolerant | same as interactive audio/video                        | yes: few seconds   |
| interactive games      | loss-tolerant | few Kbps to 10's Kbps                                  | yes: 100's msec    |
| financial applications | required      | elastic  | yes and no         |

# Struktura přednášky

- 1 Přehled
- 2 Úvod
- 3 Základní členění aplikací
  - Komunikační modely – Client-Server vs. Peer-to-peer
  - Přístup k informacím – Pull model vs. Push model
  - Nároky na počítačovou síť – nízké vs. vysoké
- 4 Vybrané síťové aplikace
  - Jmenná služba – DNS
  - World Wide Web – HTTP
  - Elektronická pošta – SMTP
  - Přenos souborů – FTP
  - Multimediální přenosy v datových sítích



# Vybrané síťové aplikace

- *Jmenná služba – DNS*
- *World Wide Web – HTTP*
- *Elektronická pošta – SMTP*
- *Přenos souborů – FTP*
- *Multimediální přenosy – RTP, RTCP*

# Jmenná služba – DNS

**Domain Name System (DNS)** – služba pro překlad doménových jmen na IP adresy a zpět

- např. `aisa.fi.muni.cz` ↔ `147.251.48.1`

V začátcích Internetu řešeno za pomoci tzv. **host** souborů

- soubory s dvojicemi *doménové jméno, IP adresa*
- neškálovatelné řešení
  - s růstem Internetu nemožné mít tyto soubory (obsahující doménová jména celého Internetu) na každém uzlu
  - navíc v nich vyhledávat, aktualizovat, ...

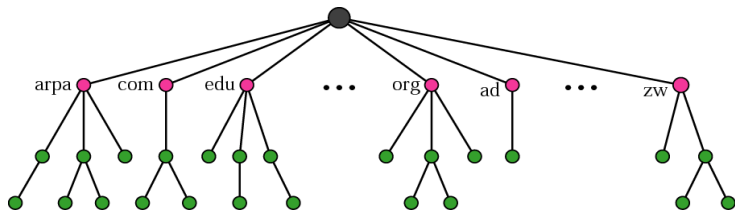
⇒ **Domain Name System (DNS)**

# Jmenný prostor

- *Jmenný prostor*  $\approx$  způsob pojmenování předmětných entit
- 2 základní varianty:
  - *plochý jmenný prostor* – jména bez jakékoliv vnitřní struktury
    - např. `mujRouterDomaVBrne`
    - hlavní nevýhoda: nemožnost využití ve velkém systému (nutnost centrální kontroly pro zamezení duplicit)
  - *hierarchický jmenný prostor* – jména s hierarchickou vnitřní strukturou
    - jména sestávají z několika částí, každá s definovaným významem
    - např. `mujRouter.DomaVBrne.cz`
    - hlavní výhoda: možnost decentralizace správy (přidělování a kontroly) jmen  
(zodpovědnost vždy za určitou (pod)část doménového jména)

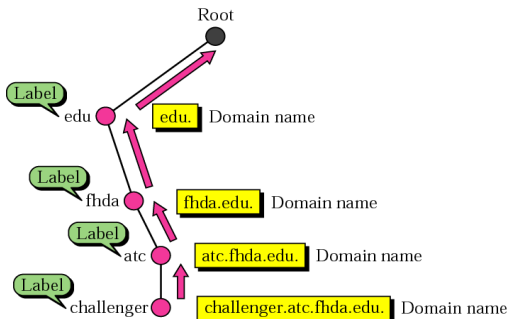
# Jmenný prostor Internetu I.

- *Jmenný prostor Internetu* – **doménový jmenný prostor (Domain Name Space)**
  - varianta hierarchického uspořádání
  - struktura invertovaného stromu
  - maximální počet úrovní = 128



## Jmenný prostor Internetu II.

- každému uzlu přidělena tzv. *jmenovka (label)* a *doménové jméno (domain name)*
  - label* – řetězec (max. 63 znaků) popisující daný uzel
    - jmenovka kořenového uzlu* je prázdný řetězec
  - domain name* – sekvence jmenovek (oddělená znakem .) od daného uzlu ke kořenovému
    - plné doménové jméno vždy končí znakem .



# Domény v Internetu

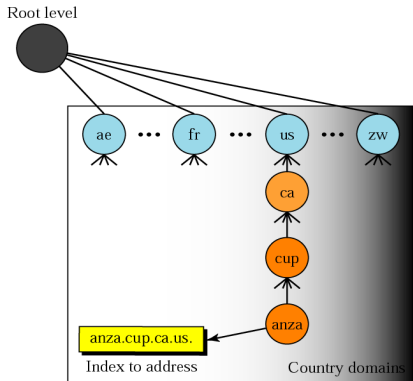
## Základní domény – Tabulka I.

| Label      | Description                 |
|------------|-----------------------------|
| <b>com</b> | Commercial organizations    |
| <b>edu</b> | Educational institutions    |
| <b>gov</b> | Government institutions     |
| <b>int</b> | International organizations |
| <b>mil</b> | Military groups             |
| <b>net</b> | Network support centers     |
| <b>org</b> | Nonprofit organizations     |

# Domény v Internetu

## Národní domény

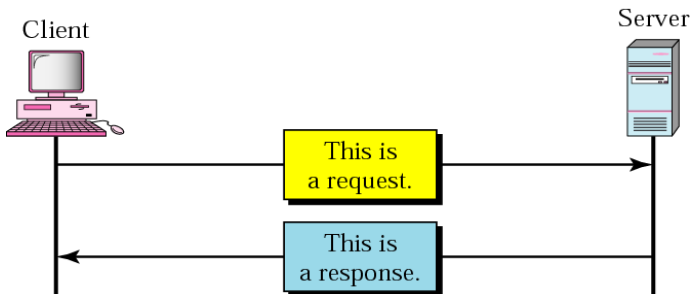
Národní domény (*Country Domains*) definují uzly podle jejich příslušnosti ke státu. Na první úrovni jsou využity dvou písmenné zkratky státu (cz, sk, ca, us, ...)



# World Wide Web – HTTP

**HyperText Transfer Protocol (HTTP)** – protokol pro přístup k datům na World Wide Webu (WWW)

- přenášená data mohou být ve formě textu, hypertextu, audia, videa, atp.
- základní idea: klient vysílá požadavek, WWW server zasílá odpověď
  - komunikace TCP protokolem na portu 80





# World Wide Web – HTTP

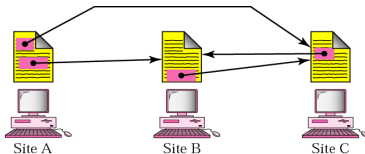
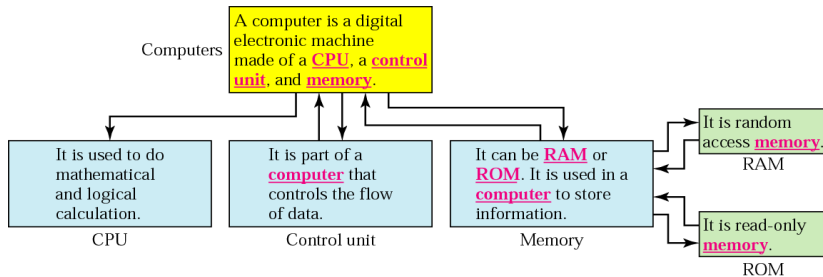
## Hypertext

*Co je to hypertext?*

# World Wide Web – HTTP

## Hypertext

*Co je to hypertext?*



# World Wide Web – HTTP

## Uniform Resource Locator (URL)

Součástí požadavku je tzv. *Uniform Resource Locator (URL)*

- standardní mechanismus pro specifikaci čehokoliv na Internetu
- definuje zdroj, který chce klient získat
- součástí URL je:
  - *method* – metoda (protokol), který má být využit pro přístup k odkazovanému zdroji
  - *host* – uzel, kde se odkazovaná informace nachází (kde má být vyhledána)
  - *port* – volitelná součást, pokud je využit jiný než standardní port
  - *path* – cesta, kde se odkazovaná informace nachází (+ případně další informace (parametry))

URL

Uniform resource locator



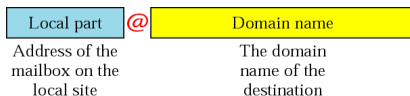
# World Wide Web – Dokumenty

Základní kategorie WWW dokumentů:

- *statické* – na serveru uložené dokumenty s pevným obsahem
  - např. HTML dokumenty
- *dynamické* – neexistují v předem definovaném formátu; jsou tvořeny webovým serverem dle požadavků klienta
  - např. CGI skripty
- *aktivní* – serverem poskytnuté programy spouštěné na straně klienta
  - např. JAVA aplikace

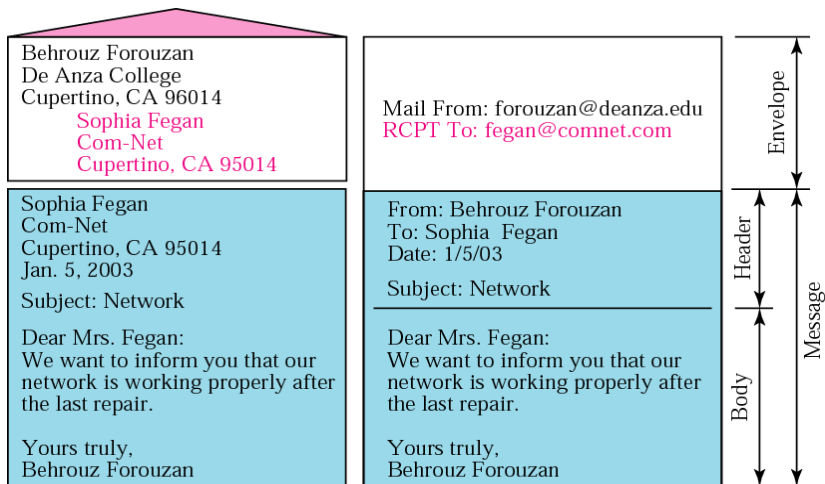
# Elektronická pošta – SMTP

- **Simple Mail Transfer Protocol (SMTP)** – standardní mechanismus pro posílání elektronické pošty (*electronic mail, email*) v Internetu
- struktura SMTP emailové zprávy:
  - *obálka (envelope)* – obsahuje adresu odesílatele, adresu příjemce a další případné informace
  - *vlastní zpráva (message)* – dělí se na hlavičky a tělo zprávy
    - hlavičky – definují odesílatele, příjemce, předmět zprávy, ...
    - tělo zprávy – vlastní přenášená zpráva
- *emailové adresy*:
  - skládají se z tzv. *lokální části a doménového jména*
    - lokální část definuje jméno souboru, kam je doručována pošta předmětného uživatele (tzv. *mailbox*)
    - doménové jméno dané organizace
  - doručení emailu probíhá na základě emailových adres uvedených v **obálce zprávy**



# Elektronická pošta – SMTP

## Příklad emailové zprávy



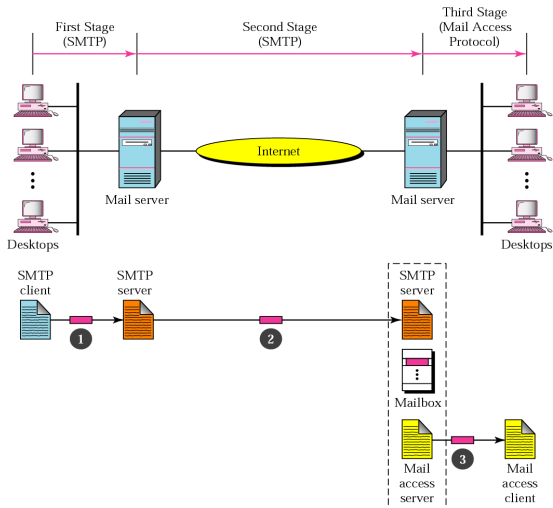
# Elektronická pošta – Doručení elektronických zpráv

Doručení elektronické zprávy se skládá ze 3 fází:

- 1 *doručení emailu lokálnímu poštovnímu serveru (mailserveru)*
  - poštovní klient (*Mail Transfer Agent, MTA*) ustaví TCP spojení (port 25) s poštovním serverem
  - po předání zprávy (s využitím SMTP protokolu) spojení uzavře
- 2 *předání emailu cílovému poštovnímu serveru*
  - lokální mailserver předá emailovou zprávu cílovému mailserveru (SMTP protokolem)
- 3 *předání/čtení emailu cílovým poštovním klientem*
  - iniciováno cílovým uživatelem (poštovním klientem) s využitím protokolu *POP3* či *IMAP4*

# Elektronická pošta – Doručení elektronických zpráv

## Ilustrace doručení emailové zprávy

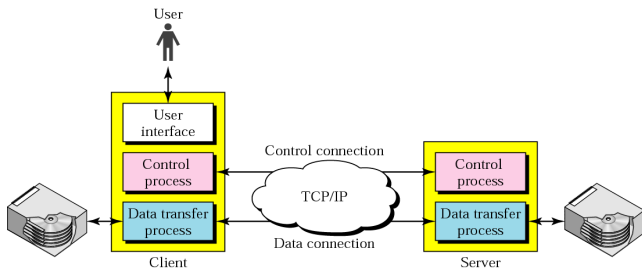




# Přenos souborů – FTP

**File Transfer Protocol (FTP)** – standardní mechanismus Internetu určený pro přenos souborů mezi uzly

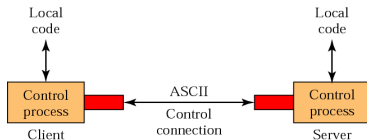
- oproti jiným *klient-server* aplikacím FTP klient s FTP serverem ustavuje **dvě samostatná TCP spojení**
  - řídicí zprávy zasílány tzv. *out-of-band*
- ① *řídicí spojení* (TCP, port 21)
  - udržováno po celou dobu ustavené relace
- ② *datové spojení* (TCP, port 20)
  - otevíráno/zavíráno pro každý přenášený soubor



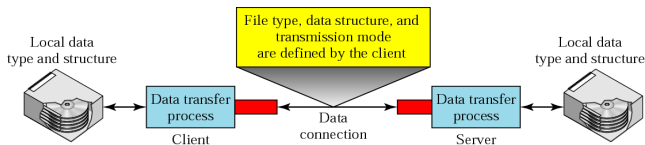
# Přenos souborů – FTP

## Řídící vs. datová komunikace

- *řídící komunikace* – přenos požadavků klienta a odpovědí serveru
  - domluva na parametrech spojení
    - typ souboru (textový vs. binární), vnitřní struktura souboru (obvykle bez struktury) a přenosový mód (proudový, blokový, komprimovaný)
    - nezbytné pro překonání heterogenity komunikujících stran



- *datová komunikace*



# Multimediální přenosy – Úvod

## Multimédia a datové sítě

- posun od využití sítě pro přenos statických dat (emaily, dokumenty, obrázky, ...) k přenosu dynamických dat (přenosy audia&videa)
- vyžadují relativně velké objemy přenášených dat
- specifické nároky na přenos (chybovost, latence, jitter, atp.)
  - požadavky na přenos zásadně ovlivňují možnosti zpracování

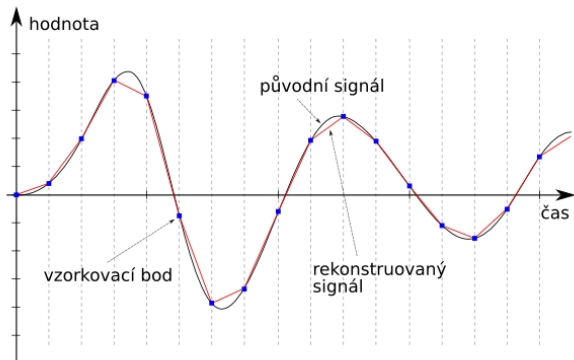
### *Aplikace multimediálních přenosů:*

- Streaming uloženého audia/videoa
- *Streaming live audia/videoa*
  - doručování multimediálního obsahu, který vzniká živě během streamování
- *Videokonference, Internetová telefonie*
  - aplikace požadující zcela konkrétní vlastnosti přenosu (např. minimální end-to-end zpoždění)
  - jednoznačný požadavek na interaktivitu

# Zpracování zvuku

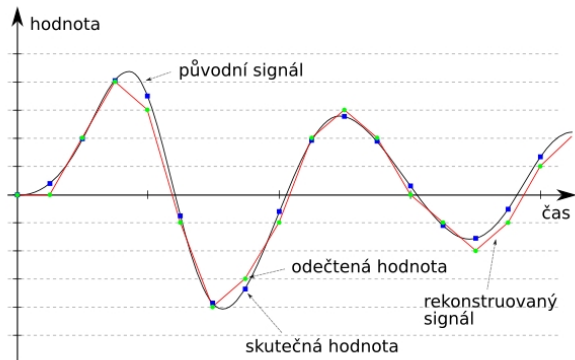
- *zvuk* – podélné mechanické vlnění v látkovém prostředí (vzduch), které je schopno vyvolat v lidském uchu sluchový vjem
  - akvizice mikrofonom  $\Rightarrow$  analogový signál spojitý v čase
- zpracování zvuku:
  - vzorkování a kvantování – převod analogového signálu do digitálního
  - zpracování digitálních dat – použití filtrů (ekvalizace, odstranění šumu/echa, atp.)
  - komprese – snížení datového objemu
    - pro audio data není nezbytná (objem audio dat je relativně malý)
    - MP3 (MPEG audio layer 3), OGG, WMA (Windows Media Audio), RA (Real Audio), ...

# Zpracování zvuku – vzorkování



- **vzorkování** = **odebírání vzorku signálu** v definovaných časových intervalech (*vzorkovací frekvence*)
  - převádí spojité časový průběh signálu na diskrétní reprezentaci

# Zpracování zvuku – kvantování



- *kvantování* = diskrétní reprezentace **hodnoty intenzity zvuku** v okamžiku odebrání vzorků
  - rozdělení svislé osy zvukové křivky na diskrétní hodnoty

# Zpracování obrazu

- *obraz* – elektromagnetické vlnění s velmi úzkou šířkou spektra (viditelné světlo) odražené od objektů v okolí a dopadající na světlocitlivě buňky sítnice oka
  - akvizice videokamerou  $\Rightarrow$  sekvence diskrétních obrázků
- zpracování obrazu:
  - vzorkování a kvantování – převod analogového elektromagnetického vlnění do digitálního signálu
    - při akvizici obraz rozdělén na diskétní vzorky (typicky  $768 \times 576$  bodů,  $1920 \times 1080$  bodů, atp.)
    - úkolem kvantování je ohodnotit barvu/jas/intenzitu jednotlivých bodů
    - *framerate* = počet obrazových snímků za sekundu pro zachování iluze pohybu (typicky 25 fps)
  - zpracování digitálních dat – úpravy jasu, vyvážení bílé, atp.
  - komprese – snížení datového objemu
    - u video dat nezbytná (velké objemy oproti audio datům)
    - pro účely minimalizace end-to-end latence však může být výhodnější využít nekomprimované video (např. uncompressed HD)
    - MJPEG, MPEG, DV, HD, ...

# Přenosy multimediálních dat – transportní protokoly

## TCP

- zajištění bezchybnosti přenosu je na úkor zvýšení end-to-end latence
- zajištění férovosti nedovoluje dostatečnou šířku pásma na vytížených linkách

## UDP

- nemá režii spojenou s ověřováním (a zajišťováním) bezchybnosti přenosu
- minimalistický, efektivnější, rychlejší
  - minimálně navyšuje latenci přenosu
- ⇒ vhodný pro přenos multimediálních dat
  - výhodný zejména pro interaktivní přenosy
  - využíván v drtivé většině případů (až na speciální výjimky)



# Přenosy multimediálních dat – přenosová síť

- Internet poskytuje best-effort službu
- vzhledem k využití UDP protokolu je zapotřebí vyrovnat se s chybovostí přenosu
  - avšak i s dalšími neduhy – latence, jitter

# Struktura přednášky

- 1 Přehled
- 2 Úvod
- 3 Základní členění aplikací
  - Komunikační modely – Client-Server vs. Peer-to-peer
  - Přístup k informacím – Pull model vs. Push model
  - Nároky na počítačovou síť – nízké vs. vysoké
- 4 Vybrané síťové aplikace
  - Jmenná služba – DNS
  - World Wide Web – HTTP
  - Elektronická pošta – SMTP
  - Přenos souborů – FTP
  - Multimediální přenosy v datových sítích

# Rekapitulace – Aplikační vrstva

- poskytuje služby pro uživatele
  - rozhraní mezi uživatelem a počítačovou sítí
- aplikace lze členit dle nejrůznějších hledisek
  - klient/server vs. peer-to-peer, pull vs. push model, nároky na počítačovou síť, atp.
- příklady stěžejních aplikací a aplikačních protokolů Internetu:
  - jmenná služba (DNS)
  - World-Wide-Web (HTTP)
  - elektronická pošta (SMTP)
  - přenos souborů (FTP)
  - multimediální přenosy (RTP/RTCP)
- *další informace:*
  - PA159: Počítačové sítě a jejich aplikace I. (doc. Hladká)
  - PA160: Počítačové sítě a jejich aplikace II. (prof. Matyska)
  - PV188: Principy zpracování a přenosu multimédií (doc. Hladká, dr. Liška, Ing. Šiler)
  - atd.