

# Rodina protokolů TCP/IP verze 3.0

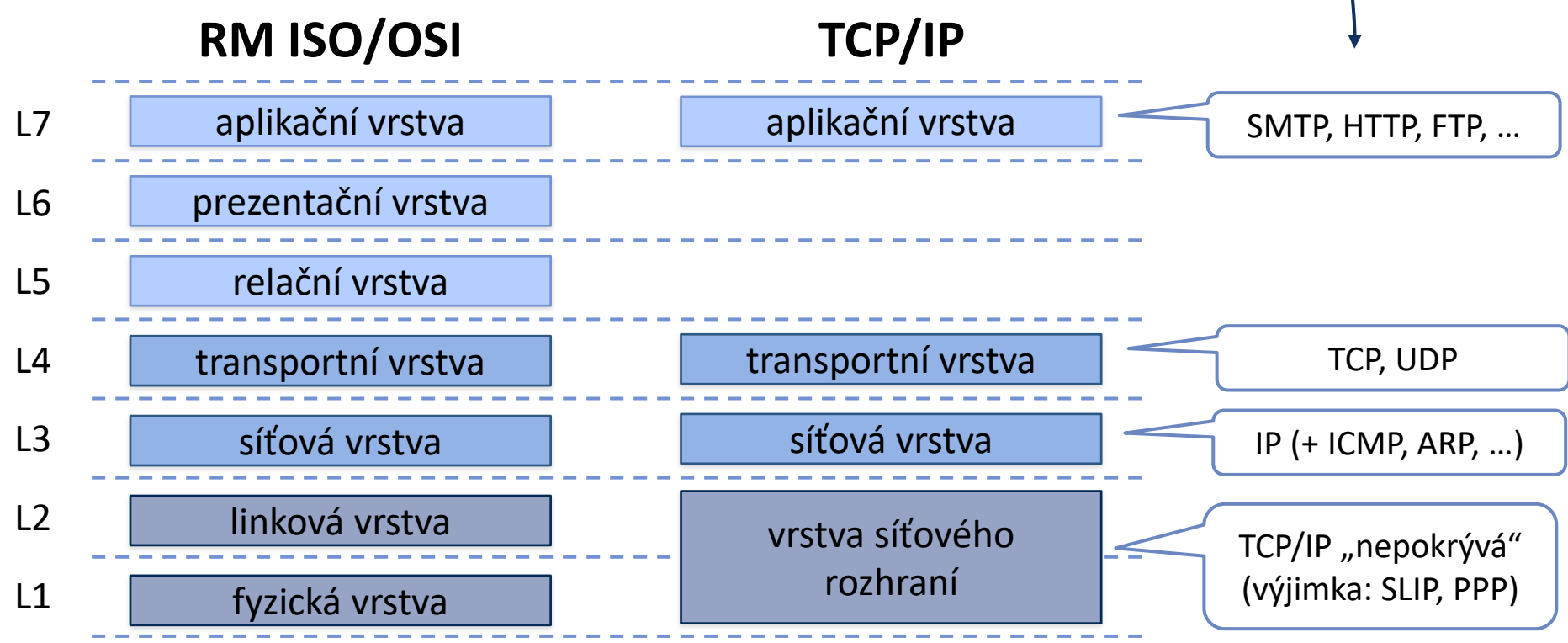
## Téma 3: Architektura TCP/IP

Jiří Peterka

# TCP/IP je síťovou architekturou

- TCP/IP je „rodinou protokolů“ (Protocol Suite)
- ale podle obvyklé terminologie je **síťovou architekturou**
  - **neboť zahrnuje:**
    - představu o počtu vrstev
    - představu o tom, co má mít která vrstva na starosti
    - konkrétní protokoly

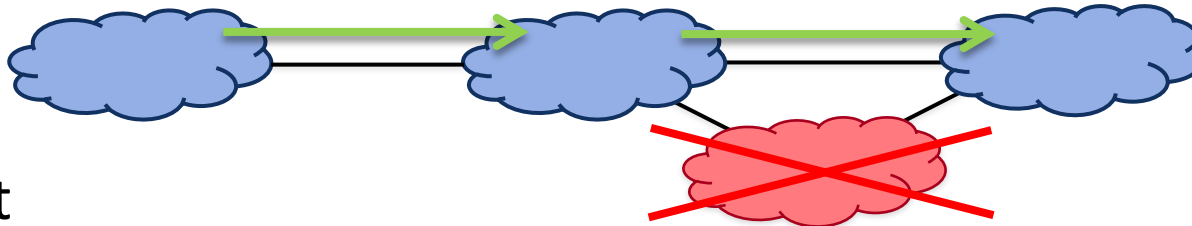
zahrnuje i konkrétní protokoly



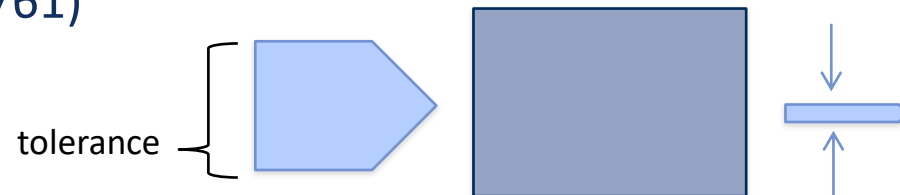
- RM ISO/OSI:
  - přístup autorů odtržený od reality
    - nechtěli přebírat jiná řešení
      - například Ethernet
    - chtěli vymyslet všechno sami
      - a to nestíhali, viz absence ISO/OSI protokolů v referenčním rámci
  - „od složitějšího k jednoduššímu“
    - autoři nejprve vymysleli co nejkompexnější řešení
      - a teprve pak přemýšleli nad tím, zda je realizovatelné
    - následně museli slevovat a hledat implementovatelnou podmnožinu
  - vycházeli z představy, že „potřebují někomu něco prodat“
    - proto: preference „bohatých“ služeb
      - spojované, spolehlivé
      - podpora QoS
- TCP/IP
  - přístup autorů velmi realistický
    - ochota přebírat „cizí“ řešení
      - např. Ethernet od IEEE
    - soustředili se na to, jak „cizí“ řešení využít co nejlépe
      - např. jak vkládat IP pakety do Ethernetových rámců
  - „od jednoduššího ke složitějšímu“
    - nejprve se navrhne jednoduché řešení
      - realizovatelnost je podmínkou standardizace
    - postupně se rozšiřuje a zdokonaluje
      - pokud je o ně zájem
      - pokud je to reálné, použitelné

v TCP/IP vše opačně

- decentralizovaný (distribuovaný) charakter
  - absence centrálních prvků, které by byly „single point of failure“
  - důsledek: i když je část sítě mimo provoz, zbývající části mohou stále fungovat



- robustnost
  - schopnost překonat ne zcela ideální podmínky
  - schopnost vyrovnat se s výpadky, chybami .....
  - projevuje se i v preferenci nespolehlivého a nespojovaného přenosu
  - tolerance vůči nedokonalostem
  - **princip robustnosti** (též: **Postel's Law**):
    - *“be conservative in what you do, be liberal in what you accept from others”*
    - jinými slovy: *toleruj chyby na vstupech, ale sám je na výstupech nedělej*
    - původně formulováno pro TCP (RFC 761)
    - později filosofie celého TCP/IP



# „základní rysy“ TCP/IP

- důraz na efektivnost
  - autoři TCP/IP nemuseli nikomu nic prodávat (myslet na komerční aspekty)
  - proto kladli hlavní důraz na efektivnost přenosových mechanismů sítě

- **důsledky:**

- předpoklad paketového přenosu
  - přenosu dat na principu přepojování paketů
    - které je efektivnější než přepojování okruhů
    - ale které nevychází vstříc multimediálním službám

důsledek ARPANETu a NCP

- paradigma „**hloupá síť, chytré uzly**“

vlastní rozhodnutí autorů

- přenosová síť má být hloupá – ale maximálně efektivní
  - má se maximálně soustředit na svůj hlavní úkol (core business): přenos dat
    - nemá se rozptylovat dalšími úkoly (jako je třeba zajištění spolehlivosti)

- konkrétní důsledky:

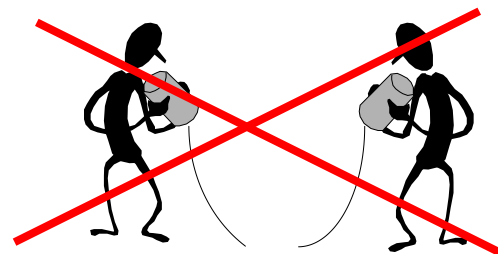
- preference nespojovaných přenosů
- preference nespolehlivých přenosů
- preference principu best effort (před podporou QoS)
- demokracie / možnost volby: **když někdo chce něco jiného, má možnost .....**

protokol IP je:

- nespojovaný
- nespolehlivý
- best effort

# preferenze nespojovaných přenosů

- (hlavní) přenosové služby TCP/IP fungují na nespojovaném principu
  - nenavazují spojení, posílají data v dobré víře, že příjemce existuje a bude ochoten je přijmout
  - přenosový protokol síťové vrstvy (protokol IP) je nespojovaný



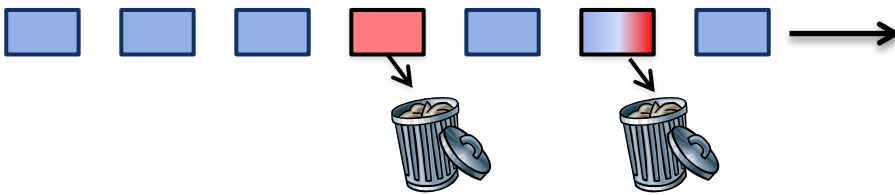
- **výhody:**

- je to bezstavové
  - nemění se stav odesilatele ani příjemce
- není nutné složitě reagovat na změny v přenosové infrastruktuře, rušením a novým navazováním spojení
- vše zajistí adaptivní mechanismy směrování
  - které hledají nejvhodnější trasu pro každý paket v každém „přestupním“ uzlu (směrovači)
- je to výhodné pro "řídké" přenosy
  - přenosy menších objemů dat, hodně rozložené v čase

- **nevýhody:**

- není to výhodné pro "intenzivní" přenosy
  - přenosy větších objemů dat v krátkém časovém intervalu
- různé pakety mohou být přenášeny různými cestami
  - a být doručovány v různém pořadí
- možnost volby:
  - vyšší vrstvy si mohou zvolit spojovaný způsob přenosu
    - skrze volbu protokolu TCP

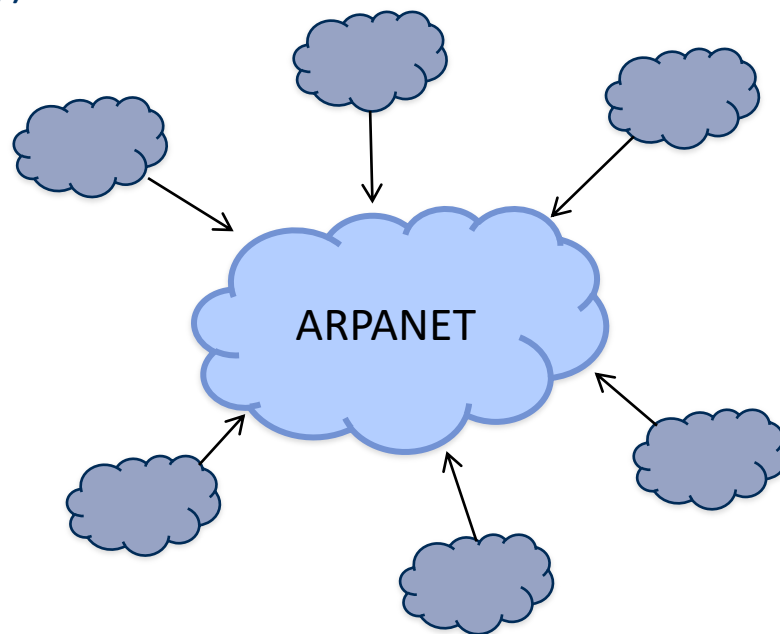
- (hlavní) přenosové služby TCP/IP fungují nespolehlivě
  - když se nějaká data při přenosu poškodí nebo ztratí, nepovažují za svou povinnost postarat se o nápravu
    - přenosový protokol síťové vrstvy (protokol IP) funguje nespolehlivě
- výhody:
  - nenarušuje to plynulost přenosu
  - při nápravě se narušuje opakováním již uskutečněného přenosu
    - plynulost požadují hlavně multimediální aplikace
      - „datovým“ aplikacím je to jedno
  - nenese to režii, spojenou se zajištěním spolehlivosti
    - s potvrzováním, s nutností opakování přenosů, .....
- nevýhoda:
  - data se mohou poškodit/ztratit
- související aspekty:
  - spolehlivost není absolutní
    - ale relativní – má vždy určitou míru
    - různé aplikace mohou požadovat různou míru spolehlivosti
- možnost volby:
  - aplikace si mohou zajistit spolehlivost samy
    - na aplikační vrstvě
  - aplikace mohou využít protokol TCP
    - který funguje spolehlivě



- best effort = všem datům je při přenosu měřeno stejně
  - nerozlišuje se, o jaká data jde – a se všemi se nakládá stejně
  - když se nedostává zdrojů (přenosové kapacity, ....), jsou všechny přenosy kráceny stejně
  - alternativou k best effort je podpora QoS (Quality of Service)
- výhody:
  - nevadí to „počítačovým“ aplikacím
    - jako je přenos souborů, email, ....
  - přenosové mechanismy (protokoly) mohou být jednodušší
    - příklad: protokol IP
  - přenosová infrastruktura (sít) může být levnější a rychlejší
    - důsledky:
      - Internet se mohl rozvíjet snáze, rychleji a levněji
        - než kdyby se snažil podporovat QoS
- nevýhody:
  - vadí to multimediálním aplikacím
    - přenosu zvuku a obrazu
- možné řešení:
  - podporu QoS lze přidat dodatečně
    - technická řešení jsou standardizována
  - ale je to problematické
    - „moc to nefunguje“
    - spíše se řeší předimenzováním kapacit
      - „přístup hrubou silou“



- další ze „základních rysů“ TCP/IP
  - internetworking = vzájemné propojování sítí (do větších celků, internet-ů)
- bylo to důležité již v době, kdy protokoly TCP/IP vznikaly
  - na bázi TCP/IP fungoval zárodečný ARPANET
    - ale na něj se chtěly napojovat další sítě, které již existovaly a využívaly různé technologie (na úrovni fyzické a linkové vrstvy)
- proto:
  - autoři TCP/IP měli v zadání: umožnit snadné napojování takovýchto sítí na „zárodečný“ ARPANET
    - tímto postupem (postupným „nabalováním“ nakonec vznikl dnešní Internet
  - zvolené řešení: vyšší vrstvy (počínaje vrstvou síťovou) budou maximálně nezávislé na nejnižších vrstvách
    - fungování vyšších vrstev nebude využívat žádná specifika nižších vrstev
    - adresování (na síťové vrstvě a vyšších) bude abstraktní, nezávislé na nižších vrstvách



- důraz na internetworking měl konkrétní důsledky pro TCP/IP
- koncepce síťové vrstvy
  - je „neprůhlednou pokličkou“
    - zakrývá specifika nižších vrstev
  - je abstraktní
    - používá abstraktní (IP) adresy
      - které nemají přímé ekvivalenty v linkových adresách
  - od nižších vrstev očekává jen nezbytné minimum
- „nenaplnění“ vrstvy síťového rozhraní
  - TCP/IP sám nepokrývá tuto vrstvu
    - nedefinuje žádné protokoly, které by měly být používány na této vrstvě
      - a předpokládá, že budou použity „takové technologie, jaké existují“
        - například Ethernet od IEEE
    - výjimkou jsou:
      - protokol SLIP (Serial Line IP)
      - protokol PPP (Point-to-Point Protocol)
        - určeny pro dvoubodové spoje
        - kde i nasazení Ethernetu je overkill

aplikační vrstva

transportní vrstva

síťová vrstva

vrstva síťového rozhraní



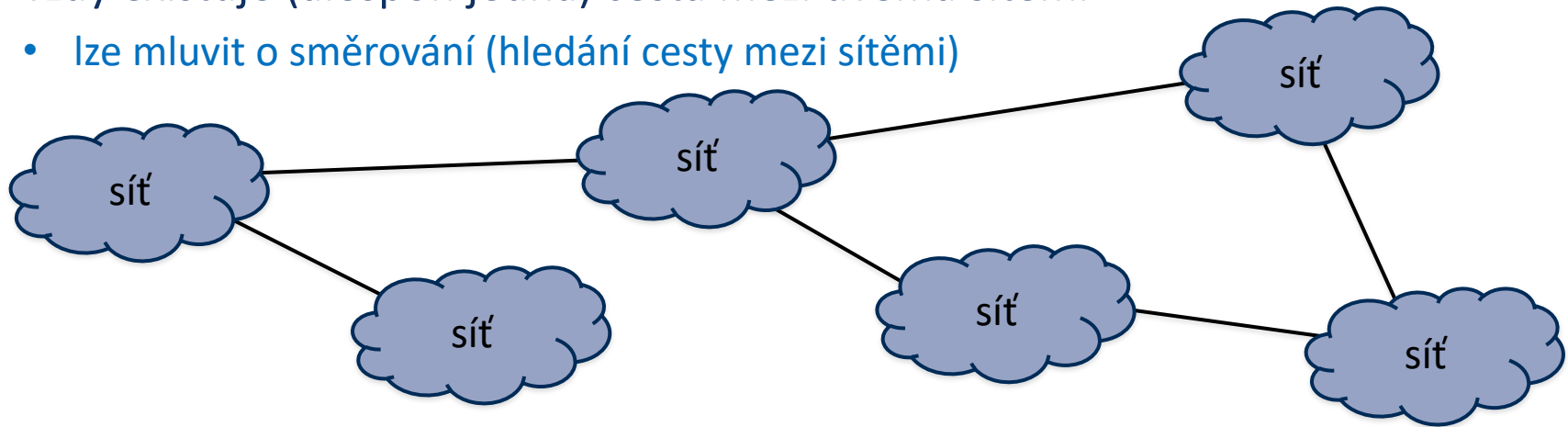
TCP/IP nepokrývá  
výjimka: SLIP, PPP



specifika použité technologie

- rodina protokolů (síťová architektura) TCP/IP je v praxi velmi úspěšná
  - projevuje se to i skrze dva výrazné trendy
- **IP over Everything**
  - slogan, zdůrazňuje že protokol IP dnes může „běžet“ nad (prakticky) jakoukoli linkovou technologií
    - že byl „portován“ na všechny dostupné linkové technologie
      - že IP pakety lze „balit“ do všech linkových rámců, buněk atd.
  - například (IP pakety lze přenášet po):
    - Ethernetu, Token Ringu, ARCnetu
      - linkové technologie sítí LAN
    - ISDN, xDSL, kabelu/CATV, PLC, modemovém spojení
      - „pevné“ technologie
    - GSM, GPRS, EDGE, UMTS, LTE, ....
      - „mobilní“ technologie
- **Everything over IP**
  - slogan, zdůrazňuje že (prakticky) všechny aplikace dokáží fungovat nad protokolem IP
    - že jsou „portovány“ nad protokol IP
      - i když původně předpokládaly jiné prostředí a infrastrukturu
        - a protokol IP pro ně není příliš vhodný
  - (nad IP lze provozovat) například:
    - přenos hlasu
      - technologie VOIP, služby IP telefonie
    - přenos obrazu
      - technologie IPTV, služby OTT (Over the Top)
    - .....

- koncepce protokolů TCP/IP vychází z představy, že svět je tvořen dílčími sítěmi, které jsou mezi sebou (nějak) propojeny
  - ve smyslu:
    - jednotlivé sítě mají svou „identitu“ (síťovou adresu, resp. síťovou část adresy)
      - lze mluvit o příslušnosti uzlů k určité síti
    - vždy existuje (alespoň jedna) cesta mezi dvěma sítěmi
      - lze mluvit o směrování (hledání cesty mezi sítěmi)

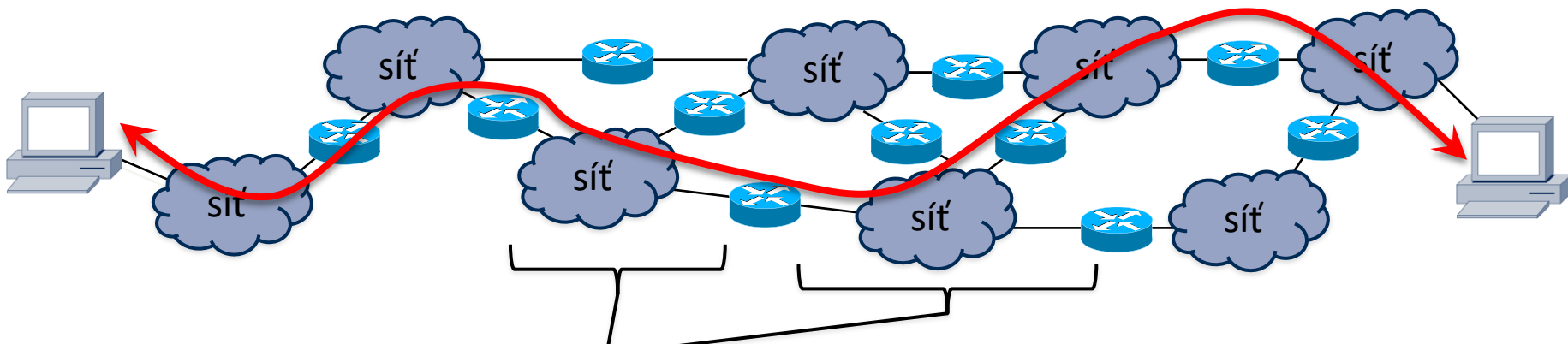


- tato představa odpovídá tzv. **katenetovému modelu**
  - „katenet“ ve smyslu řetězec, posloupnost, .....
  - alternativou by mohla být:
    - jedna jediná (plochá) síť
    - několik samostatných sítí, ale bez vzájemného propojení



## • připomenutí:

- síťová vrstva přenáší **síťové pakety**
- jednotlivé pakety přenáší přes mezilehlé uzly až k jejich koncovým příjemcům
  - mezi jednotlivými sítěmi, přes **směrovače** (router-y)
- k tomu musí zajistit: **směrování** (routing)
  - jako volbu/hledání správného směru dalšího přenosu



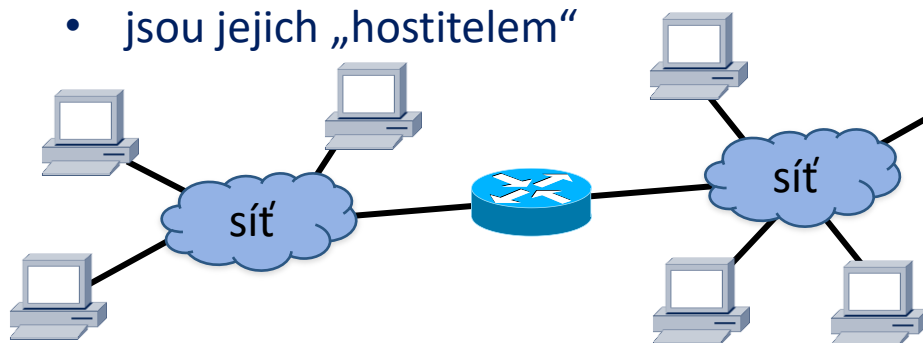
- pro každý jednotlivý „přeskok“ využívá služeb linkové vrstvy (vrstvy síť. rozhraní)
  - a zde použité linkové technologie (např.: Ethernetu, PPP, ATM, ....)
    - různé linkové technologie používají různé (linkové) adresy a mohou se výrazně lišit:
      - například v maximální velikosti linkového rámce
      - mohou fungovat spojovaně/nespojovaně, spolehlivě/nespolehlivě
      - mohou pracovat s prioritami či jinými formami QoS

- TCP/IP předpokládá, že existují jen 2 typy uzlů:

- **koncové uzly** (host computers, hostitelské počítače)

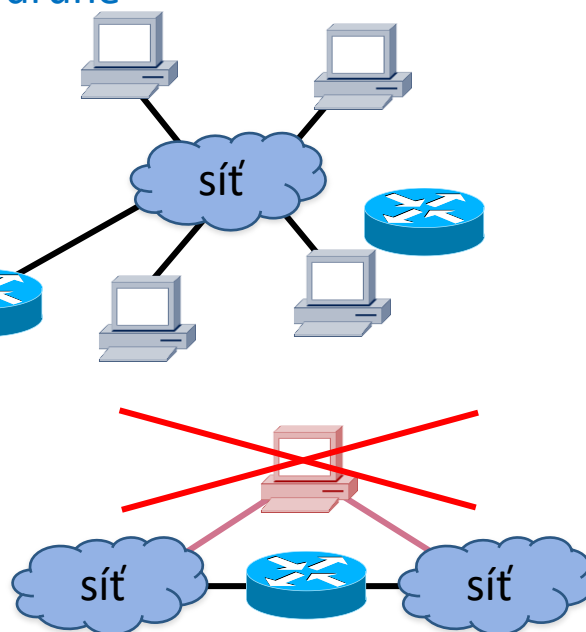


- jsou připojené jen k jedné síti
- slouží (pouze) potřebám uživatelů
  - jako servery, jako pracovní stanice, .....
- „hostitelské“ proto, že „hostí“ zdroje
  - jako jsou aplikace, data, zařízení
    - jsou jejich „hostitelem“



- **směrovače** (routers)

- jsou připojeny do dvou či více sítí
- slouží (pouze) potřebám směrování
  - zajišťují „přestup“ z jedné sítě do druhé



- nedoporučuje: tzv. **multihomed host**

- uzel, který by současně fungoval jako směrovač i jako hostitelský počítač
  - dnes se připouští připojení host. počítačů do více sítí kvůli rozkladu zátěže či zálohování

- při vzniku koncepce TCP/IP připadaly v úvahu 2 možnosti:

a) nezakrývat specifika linkových technologií

- tj. umožnit vyšším vrstvám, aby využívaly specifické schopnosti konkrétních linkových technologií
- výhoda:
  - např. vyšší efektivnost, ...
- nevýhoda:
  - vyšší vrstvy nefungují všude stejně, musí se přizpůsobovat

b) volit síťovou vrstvu jako „jednotnou pokličku“

- tj. zakrýt všechna specifika nižších vrstev, vytvořit jednotné (stejně) prostředí pro vyšší vrstvy
- výhoda:
  - vyšší vrstvy se nemusí přizpůsobovat/měnit
- nevýhoda:
  - nelze využít specifické výhody konkrétních linkových technologií
    - pokud takové existují

aplikační vrstva

transportní vrstva

síťová vrstva

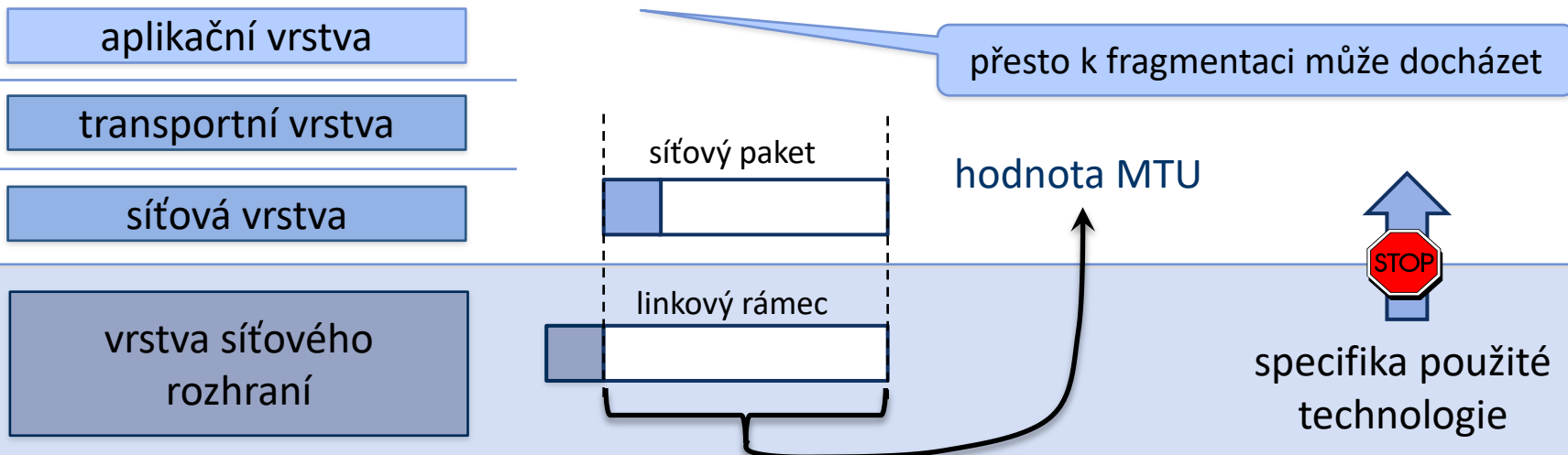
vrstva síťového rozhraní



specifika použité technologie

# koncepce síťové vrstvy TCP/IP

- nakonec se prosadila varianta b: **jednotná poklička**
  - specifika linkových technologií jsou zakryta, využívá se jen minimum přenosových schopností, které by měla podporovat každá technologie
  - přesto:
    - **jednu odlišnost nelze / nemá smysl „zakrývat“: velikost linkového rámce**
  - proto:
    - existuje výjimka (z pravidla, že všechna specifika jsou zakryta):
      - vyšší vrstvy „vidí“ parametr **MTU** (Maximum Transmission Unit)
        - ten udává, kolik bytů se vejde do „nákladové části“ linkového rámce
        - podle něj vyšší vrstvy „porcují“ data k přenosu na menší části – aby se ještě vešly to linkového rámce, a nedocházelo k tzv. **fragmentaci** (nutnosti rozdělení na části)





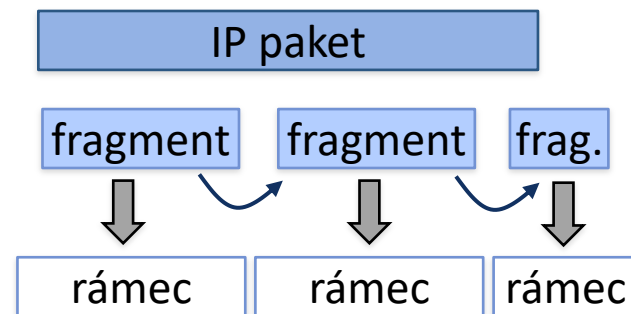
- síťová vrstva má jen 1 přenosový protokol: **IP** (Internet Protocol)
  - přenáší bloky označované jako pakety: **IP pakety**
  - funguje:
    - nespojovaně
      - nenavazuje spojení, díky tomu funguje bezestavově
    - nespolehlivě
      - nestará se o nápravu, pokud dojde k poškození nebo ztrátě dat
    - stylem best effort
      - všem datům měří stejně, nepodporuje QoS, nerozlišuje mezi přenášenými daty
  - je **minimalistický**, nabízí jen „holý přenos“
    - důsledek paradigmatu: „hloupá síť, chytré uzly“
      - a představy, že přenosová část sítě má pouze přenášet data, ale nic jiného
  - negarantuje, že:
    - data vůbec přenesou (důsledek nespolehlivosti)
    - data stihne přenést včas (nejpozději za dobu t)
    - data bude přenášet pravidelně (se stejným přenosovým zpožděním/latencí)
    - data doručí ve správném pořadí (důsledek nespojovaného charakteru)

proto jsou označovány také jako **IP datagramy** !!

protokol IP se nehodí pro multimediální přenosy

# podpora fragmentace

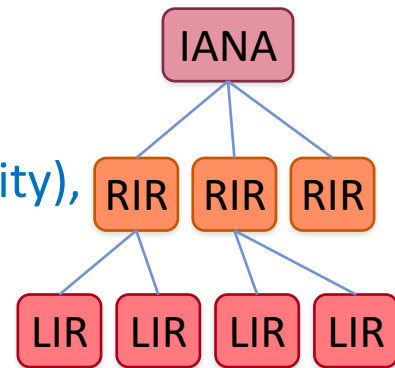
- protokol IP musí mít zabudovanu podporu fragmentace
  - možnost rozdělit „příliš velký paket“ na menší části (fragmenty), tak aby se vešly do linkových rámců
  - zahrnuje:
    - označení jednotlivých fragmentů
      - tak, aby se poznalo, že „patří k sobě“
    - identifikaci pořadí / posloupnosti
      - aby se vědělo, jak jdou fragmenty „za sebou“ a který z nich je poslední
    - minim. velikost paketu, který vždy projde bez fragmentace: IPv4: 576 B, IPv6: 1280 B
  - umožňuje:
    - nefragmentovat
      - odmítnout fragmentaci, pokud by k ní mělo dojít
    - fragmentovat pakety přímo u odesilatele
      - pokud není respektována hodnota parametru MTU
    - fragmentovat pakety „po cestě“
      - ve směrovačích, které pracují s cestami s menším MTU
    - sestavit z přijatých fragmentů původní (větší) paket
      - možnost „poskládat“ fragmenty – pokud došly všechny



umožňuje jen IPv4  
(IPv6 nikoli)

dělá vždy až  
koncový příjemce

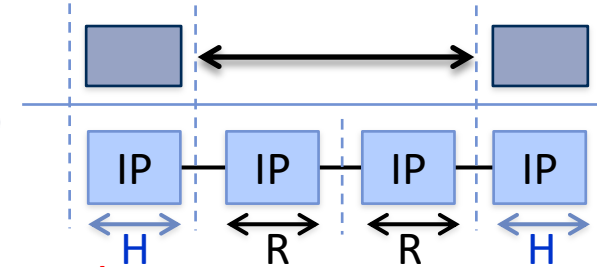
- používá jednotné adresování
  - abstraktní **IPv4 adresy** (32 bitů) – nejsou nijak závislé na linkových (HW) adresách
  - jsou logicky dvousložkové:
    - mají síťovou část (určuje síť jako celek)
    - mají relativní část (určuje relativní adresu uzlu v rámci sítě)
- s tím souvisí i celý systém distribuce IP adres
  - centrální přidělovatel (IANA, Internet Assigned Numbers Authority),
  - regionální přidělovatelé (RIR, Regional Internet Registries)
  - lokální přidělovatelé (LIR, Local Internet Registries)
- dále pravidla pro alokaci (přidělování IP adres „po kvantech“)
  - třídy A, B a C (i D a E)
  - koncept subnettingu a supernettingu
  - koncept privátních IP adres
  - mechanismus CIDR (Classless InterDomain Routing)
- dnes také:
  - **IPv6 adresy** (128 bitů), systém distribuce a přidělování



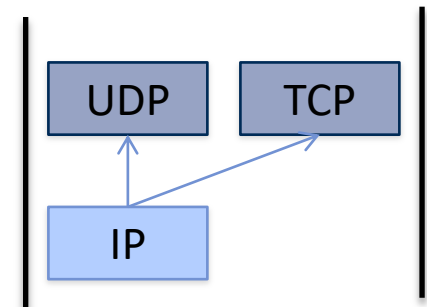
také pravidla přidělování IP adres hostitelským počítačům a směrovačům

- se síťovou vrstvou úzce souvisí (a patří do ní):
  - „pomocné“ protokoly
    - např. **ICMP** (Internet Control Message Protocol)
      - „posel špatných zpráv“ – slouží pro sdělování nestandardních situací
        - například že konkrétní paket nelze doručit, že neexistuje cesta atd.
      - zajišťuje některé další funkce
        - například „poučuje“ hostitelské počítače o správném směrování
        - slouží utilitám PING a Traceroute
    - **ARP** (Address Resolution Protocol) a **RARP** (Reverse ARP)
      - pro převod mezi síťovými a linkovými adresami, oběma směry
        - ale jen tam, kde je k dispozici broadcast na linkové vrstvě
    - **NAT** (Network Address Translation)
      - pro překlad IP adres (mezi veřejnými a privátními IP adresami)
  - se síťovou vrstvou souvisí i protokoly pro směrování
    - **RIP** (Routing Internet Protocol), **OSPF** (Open Shortest Path First) ....
  - volitelně další mechanismy a protokoly:
    - **IPSec** (IP Security, pro zabezpečené přenosy), **Mobile IP** (pro podporu mobility)

- připomenutí:
  - transportní vrstva je první vrstvou, kterou mají koncové uzly „plně ve své moci“
    - která je implementována až v koncových uzlech (hostitelských počítačích, H)
      - která není implementována ve vnitřních uzlech sítě (směrovačích, R)
  - transportní vrstva zajišťuje end-to-end komunikaci
    - komunikaci mezi koncovými uzly (hostitelskými počítači)
- proto:



- transportní vrstva je první (a současně poslední) možností, kdy (kde) změnit způsob fungování přenosových služeb
  - reprezentovaných fungováním síťového protokolu IP



- zvolené řešení: jen 2 varianty
  - obě „krajní“ (extrémní)

## a) „neměnit nic“

- transportní protokol UDP
  - nespojovaný a nespolehlivý
    - stejně jako protokol IP
  - je to velmi jednoduchý protokol

## b) „změnit všechno“

- transportní protokol TCP
  - spojovaný a spolehlivý
    - na rozdíl od protokolu IP
  - velmi složitý a komplexní protokol

- **UDP (User Datagram Protocol)**

- je pouze jednoduchou („lehkou“) nadstavbou nad protokolem IP
  - funguje stejně jako protokol IP:
    - **nespojovaně**: nenavazuje spojení, data mohou cestovat jinými cestami
    - **nespolehlivě**: nestará se o nápravu poškozených či ztracených dat
    - **na principu best effort**, bez podpory QoS: všem datům měří stejně
- přenáší bloky dat, označované jako **UDP datagramy**
  - „datagramy“ kvůli nespojovanému způsobu fungování
  - velikost UDP datagramů:
    - volitelná – ale musí se vejít do IP datagramu (max.  $2^{16} = 65535$  bytů včetně hlavičky)
      - max. 65507 bytů dat nákladové části
        - +8 bytů UDP hlavička, +20 bytů hlavička IP paketu
    - v praxi typicky mnohem menší (dáno konkrétní implementací TCP/IP)
      - často max. 8196 bytů
- řeší:
  - práci s porty
    - rozlišování různých entit v rámci každého uzlu
  - **neřeší**:
    - spolehlivost
    - řízení toku (aby odesílatel nezahltl příjemce)
    - předcházení zahlcení (aby nedošlo k zahlcení přenosové části sítě)

data „porcuje“ aplikace, UDP je dostává už po blocích

- **TCP (Transmission Control Protocol)**

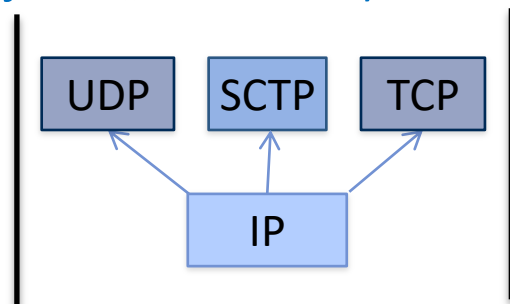
- je velmi složitým protokolem
- je velmi adaptabilní, dokáže se přizpůsobit odlišným podmínkám
  - dokáže fungovat stejně dobře v sítích LAN i WAN
    - i při diametrálně odlišné latenci / přenosovém zpoždění
- mění způsob fungování protokolu IP
  - IP funguje nespojovaně, TCP funguje **spojovaně**
  - IP funguje nespolehlivě, TCP funguje **spolehlivě**
- přenáší **TCP segmenty**
  - jejich velikost si určuje protokol TCP sám
- řeší řadu věcí, které IP ani UDP neřeší (ani řešit nemusí)
  - zajištění spolehlivosti
    - používá kontinuální potvrzování, metoda okénka, volba timeoutů, ....
  - řízení toku
    - skrze metodu okénka
  - předcházení zahlcení
    - dočasný přechod na jednotlivé potvrzování

vytváří iluzi  
bytového proudu

sám si „porcuje“  
data z bytového  
proudu, dle MTU

- řeší:
  - práci s porty
    - rozlišování různých entit  
v rámci každého uzlu

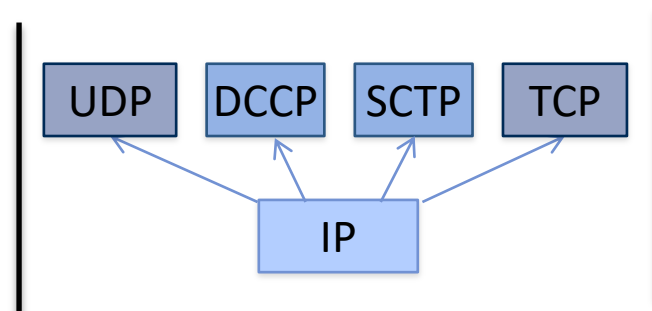
- původně:
  - jen 2 „extrémní“ varianty: TCP („všechno“), UDP („nic“)
- později:
  - různě „vyladěné“ varianty protokolu TCP
    - používají různé postupy/algoritmy/metody při zajištění spolehlivosti, řízení toku, předcházení zahlcení atd.
      - TCP Tahoe, TCP Reno, TCP NewReno, TCP Vegas, .....
- v poslední době:
  - snaha zavést jemnější škálu transportních protokolů (než jen 2 „extrémní“)
  - protokol **SCTP** (Stream Control Transmission Protocol)
    - funguje spolehlivě (stejně jako TCP)
    - funguje spojovaně (ale jinak než TCP)
      - přenáší data po blocích (messages), podobně jako UDP
      - podporuje více proudů (streams) současně (TCP jen 1 proud/stream)
    - podporuje multihoming
      - dokáže využít více síťových rozhraní, pokud je uzel má
    - předchází zahlcení (podobně jako TCP)



definuje RFC 2960 (říjen 2000), nově RFC 4960



- škála transportních protokolů se dále rozšiřuje
- protokol **DCCP (Datagram Congestion Control Protocol)**
  - přenáší datagramy (jako UDP)
    - data členěná na bloky
  - funguje spojovaně (jako TCP)
    - čísluje přenášené datagramy
  - funguje nespolehlivě (jako UDP)
    - ale poskytuje informaci o doručení datagramu (doručen, zahozen, zpožděn ....)
  - předchází zahlcení (jako TCP)
    - nabízí více algoritmů pro předcházení zahlcení
  - podporuje multihoming (jako SCTP)
    - podporuje mobilitu
  - neřídí tok (jako UDP)
- do budoucna: **SCPS-TP**
  - Space Communications Protocol Standard Transport Protocol**
    - pro meziplanetární komunikaci, kde je extrémně velké přenosové zpoždění



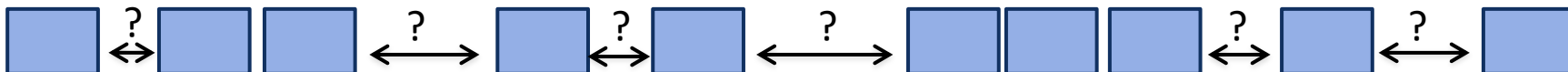
definuje RFC 4340  
(březen 2006)

# aplikace v TCP/IP

- původně malý rozsah aplikací:
  - elektronická pošta (SMTP, RFC 822)
  - přenos souborů (FTP)
  - vzdálené přihlašování (TELNET, rlogin)
    - těmto aplikacím dobře vyhovovalo fungování sítě "na principu maximální snahy, ale nezaručeného výsledku"
- později se prosadily další aplikace:
  - síťové noviny (news, netnews, USENET)
  - sdílení souborů (NFS)
  - Gopher, WWW (HTML, HTTP, ....)
  - on-line komunikace (chat, IRC, ICQ, messengery, ...)
- časem dochází k "platformizaci" aplikací
  - přežívají jen některé aplikace
    - hlavně WWW a el. pošta
  - ostatní ztrácí svou identitu a „skrývají se“ za jiné aplikace
    - jakoby se stávají nadstavbou nad těmi aplikacemi, které zůstaly
      - původně samostatné aplikace se přesouvají do role nadstavby na platformě jiné aplikace
        - platformou je WWW, případně elektronická pošta

pro všechny tyto „počítačové“ aplikace je způsob fungování TCP/IP (hlavně princip best effort, bez podpory QoS) stále ještě akceptovatelný - byť ne ideální

- v době vzniku protokolů TCP/IP se nepočítalo s multimediálními aplikacemi a jejich potřebami
  - zajistit/garantovat nízkou latenci a nízký jitter (pravidelnost doručování)
- bylo zvoleno řešení na bázi **síťové neutrality**
  - přenosová síť je „neutrální“ k přenášeným datům, nerozlišuje je a ke všem se chová stejně
    - na principu best effort: **nedokáže zajistit rychlost ani pravidelnost doručování dat !!**



- **výhoda:**
  - značně to zjednodušilo návrh přenosových protokolů (IP, UDP, TCP)
  - zlevnilo to implementaci TCP/IP infrastruktury
    - celého Internetu
- **nevýhoda:**
  - vadí to multimediálním aplikacím, které časem také přešly na TCP/IP
    - „Everything over IP“
    - i telefonování (VOIP), videokonference, distribuce R a TV signálu (IPTV), .....

- jsou připravena řešení pro dodatečné zavedení podpory QoS
  - **DiffServ** (Differentiated Services): na principu priorit
    - jednotlivé bloky dat (pakety) se mohou „hlásit“ k různým úrovním priority
  - **IntServ** (Integrated Services): na principu rezervace a garance
    - konkrétní přenosy (spojení) si vyžádají vyhrazení určité přenosové kapacity
      - v zásadě: návrat k přepojování okruhů)
- problém:
  - tato řešení se v praxi (moc) neosvědčila
    - jsou velmi komplikovaná a nákladná, dají se nasadit jen v privátních sítích
      - v rámci celého Internetu nelze jejich nasazení očekávat
- jiná řešení:
  - techniky jako client buffering
    - kompenzuje nepravidelnost doručování, vhodné pro neinteraktivní aplikace
- (v praxi) nejjednodušší řešení:
  - **předimenzování** (navyšování přenosové i další kapacity)
    - ale jinak nechat způsob fungování přenosových služeb beze změn !!!!

# princip OTT (Over the Top)

- pro poskytování multimediálních služeb (nad TCP/IP) dnes existují dva různé přístupy
- „privátní“:
  - aplikace, poskytující multimediální služby, je provozována pouze v privátní části TCP/IP sítě, kde je pro ni vyhrazena potřebná kapacita a vytvořeny další podmínky
  - takto fungují např. služby IPTV (O2 TV) či UPC Telefon
  - dokáží fungovat „garantovaným“ způsobem
- OTT (Over the Top)
  - aplikace je provozována „nad“ veřejným Internetem
  - v souběhu s dalším provozem ve veřejném Internetu
    - bez jakýchkoli opatření (QoS)
  - takto funguje např. YouTube, Hulu, Netflix, GoogleTV, Skype, .....
  - mohou fungovat jen „negarantovaným“ způsobem

