

# Aplikace

- Aplikace: hlavní smysl existence počítačových sítí
- Aplikace s nízkými nároky na přenosovou síť
- Aplikace s vysokými nároky na přenosovou síť

# Základní parametry sítě pohledem aplikací

- Propustnost: objem přenesených dat za časovou jednotku
- Zpoždění (latence): doba mezi požadavkem a odezvou
- Rozptyl (jitter): kolísání zpoždění

# Ideální síť

- Transparentní
- Pouze end-to-end vlastnosti
- Neomezená propustnost
- Nulové zpoždění
- Nulový rozptyl

# Reálné sítě

- Vnitřní struktura
- Omezená prostupnost
- Zpoždění
  - vlastní přenos
  - aktivní prvky
- Vysoký rozptyl

# Implementace funkcionality

- End-to-end
  - E2E argument: Funkcionalitu požadovanou aplikací je možno zajistit pouze se znalostí a prostřednictvím samotné aplikace
- Hop-by-hop
  - HbH zvýšení výkonu: Je možné dosáhnout výrazného zvýšení výkonu opakováním určité funkcionality na úrovni každého dvoubodového přenosu

# End-to-end rekurzivita

Podle úrovně abstrakce platí, že

- hop by hop v jednom pohledu
- je end-to-end v druhém

End-to-end argument je aplikovatelný rekurzivně jak pro uzly sítě, tak pro jednotlivé vrstvy

# Typy aplikací

- Přístup k informacím
  - pull model
  - push model
- Telepřítomnost
- Distribuované výpočty
- Složené aplikace

# Informace – Pull model

- Webové prohlížeče
- Asymetrický datový tok
- Rozmanité požadavky na propustnost

# Webové prohlížení a propustnost

- Text 1--10 KB/s
- Smíšené stránky 50--500 KB/s
- Statické obrázky 500--1000 KB/s
- Obrazy ve fotokvalitě 10--100 MB/s

Požadavek na odpověď do 100 ms:

- ISDN: 2 KB/s
- Fast Ethernet 1 MB/s
- Gigabit Ethernet 12 MB/s

# Informace – push model

- Asynchronní model
- Přenos dat iniciován automaticky na základě znalosti uživatelského profilu (požadavků)

# Telepresence

- Telekonference
- Televideokonference
- Teleimprese
  - silová zpětná vazda
  - mnohonásobné různorodé proudy dat

# Distribuované výpočty

- Závisí na rozdělení úlohy
- Vysoké nároky na dostupnost
- Adaptabilita

# *Klasické aplikace*

- Přenos souborů (FTP)
- Sdílené systémy souborů (NFS, AFS, DFS)
- Přenos pošty (SMTP)
- chat

# Smíšené aplikace

- Složitost dána nejslabším článkem
- Distanční výuka
- Gridy -- samostatná prezentace

# Základní parametry aplikací

Aplikace	Pásmo	Zpoždění	Ztrátovost	Adaptabilita
Distribované výpočty	vše	1 $\mu$ s-10s	žádná	<i>nízká</i>
Hlas	nízké	30ms	<i>nízká</i>	nízká
Video	vysoké	300ms	nízká	Střední
Haptika	v.nízké	10ms	nízká	nízká
web	střední	100ms-1s	žádná	střední

# Základní parametry aplikací

Aplikace	Pásmo	Zpoždění	Ztrátovost	Adaptabilita
Push model	nízké– střední	1 min– 1 den	střední	vysoká
Telemetrie	nízké– střední	proměnné	žádná	omezená
Streaming Video	střední	1-10s	nízká	vysoká
zálohování	vysoké	1 hodina	nulová	vysoká
e-mail	nízké	1 min– 1 hodina	žádná	vysoká

# Vlastnosti klíčových kategorií

Charakteristika	Informační přístup	Telepřítomnost	Distribuované výpočty
Model	Klient/Server	Peer-to-peer	Různá
Symetričnost přenosu	Asymetrický	Symetrický	Symetrický
Granularita přenosu	Velká	Spojité	Různá
Koncová synchronizace	Žádná	Reálný čas	Různá

# Kvalita služeb

- Ideální versus reálná síť:
  - zajištění propustnosti
  - zajištění doby odezvy
  - zajištění míry rozptylu
- Kvalita služeb (QoS)

# Kvalita služeb

- Spojované sítě
- Nespojované (datagramové sítě)

# Kvalita služeb – spojované sítě

- Signalizace
- Rezervace zdrojů
- Plýtvání
- Problém při výpadku prvku/dráhy

# Kvalita služeb – nespojované sítě

- Signalizace/značkování
- Žádná rezervace zdrojů
- Pouze statistická záruka
- Odolné proti výpadku prvku/dráhy
- Overprovisioning (opět plýtvání)

# Kvalita služeb z pohledu aplikace

- End-to-end garance
- Prioritní třídy
- Adaptace
- Modifikace parametrů během relace

# Stručný pohled na Internet

- Co se stane, když v Internetovém prohlížeči zvolíme URL:
- Velmi zjednodušený pohled:  
Pošleme požadavek a dostaneme od příslušného serveru odpověď

# Rozšiřitelnost

- Cache (vyrovnávací paměti) zvyšují rozšiřitelnost
  - dotážeme se (lokální) cache, zda má kopii
  - cache se dotáže serveru, zda neexistuje novější verze
  - Má-li cache nejnovější verzi, splní požadavek, v opačném případě získá novou kopii ze serveru

# Pojmenování (DNS)

- Mapování (symbolických) na síťové adresy (IP adresy v Internetu)
- Síť používá pouze číselné adresy
- Jména musí být přeložena do čísel -- DNS (Domain Name Service)
- Překlady jsou lokálně uchovávány (opět princip cache) pro zrychlení

# Relace (http)

- Jednoduchá webová stránka může být tvořena více „objekty“ (text, obrázky, animace, ...)
- Musí být přenesen každý objekt:
  - sekvenčně
  - paralelně
- Společně tvoří *relaci*

# Spolehlivost

- Zprávy se mohou ztratit nebo mohou být poškozeny
- Příjemce potvrzuje úspěšné přijetí každé zprávy
- Detekci ztracené zprávy, po níž následuje její znovuposlání (retransmission) -- nejčastěji pomocí timeoutů (vypršení časového intervalu)

# Zahlčení

- Potřeba alokovat pásmo mezi vysílající a přijímající stanicí
- Dynamická adaptace na stav sítě
  - vysílající kontroluje stav cesty/trasy
  - přizpůsobuje rychlost vysílání zpráv naměřené dostupné propustnosti

# Pakety

- Paket -- základní jednotka přenášených dat
- Příliš dlouhé zprávy jsou rozloženy
  - Maximum pro Ethernet: 1,5 KB
  - Typická webová stránka (text): 10 KB
- Segmenty jsou číslovány (snazší znovuspojení)

# Směrování (routing)

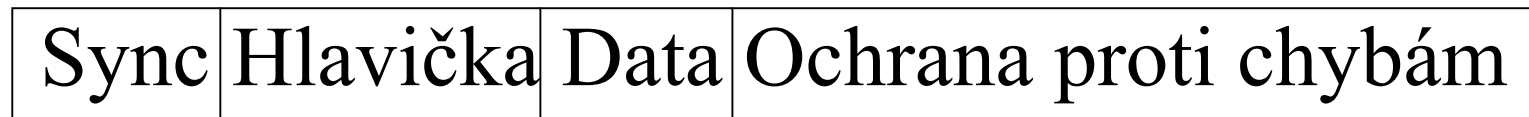
- Pakety prochází řadou směrovačů
- Statické cesty (předem alokované)
  - jednodušší
  - méně flexibilní
- Dynamické cesty
  - složité algoritmy
  - adaptabilní na výpadky
  - nezaručuje pořadí doručení

# Sdílený přístup

- Sdílení linek více relacemi (uživateli)
- Mechanismy dotazování (polling) -- řízený jedním koncem (headend)
  - Headend řídí všechny následné transmise
  - Vyžaduje nižší (lokální) úroveň adresace (např. Ethernetové MAC adresy)

# Rámce a modulace

- Převod dat (payload) do konkrétního signálu



- Např. pro optický přenos se data doplní kódem na detekci (a případně opravu) chyb, přidá se hlavička a rámec (pole Sync výše) a převede se do modulovaného světelného signálu

# Protokol

- Abstrakce -- cesta na zvládnutí složitosti
- **Protokol**: dohoda, definující formu a funkci dat, která si dvě strany vyměňují při vzájemné komunikaci
- Protokol má dvě části:
  - Syntax: „řazení bitů“
  - Sémantiku: jejich význam
- Příklady protokolů: IP, TCP, HTTP, ...

# Model sítě

- Monolitické
- Rozložení do vrstev
- Podle úhlu pohledu

# Rozklad do vrstev

Abstrakce:

- Izoluje funkcionalitu
- Umožňuje asynchronní zpracování
- Zajišťuje nezávislost
  
- Umožňuje separaci funkcí jednotlivých protokolů

# Principy tvorby vrstev

- *Abstrakce* -- způsob uvažování o funkcionalitě
- *Architektura rozdělení protokolů* -- založeno na funkčních požadavcích síťových komponent, topologie, koncových systémů a aplikací
- Mechanismus *implementace*

# Kde definovat vrstvy

- Při změně abstrakce
- Lze-li přesně definovat funkci konkrétní entity
- Při existenci standardních protokolů

Je třeba

- Minimalizovat počet vrstev
- Minimalizovat toky dat přes rozhraní vrstev
- Separovat funkcionalitu

# OSI model

7 Aplikační vrstva

6 Prezentací vrstva

5 Relační vrstva

4 Transportní vrstva

3 Síťová vrstva

2 Datové spoje

1 Fyzická vrstva

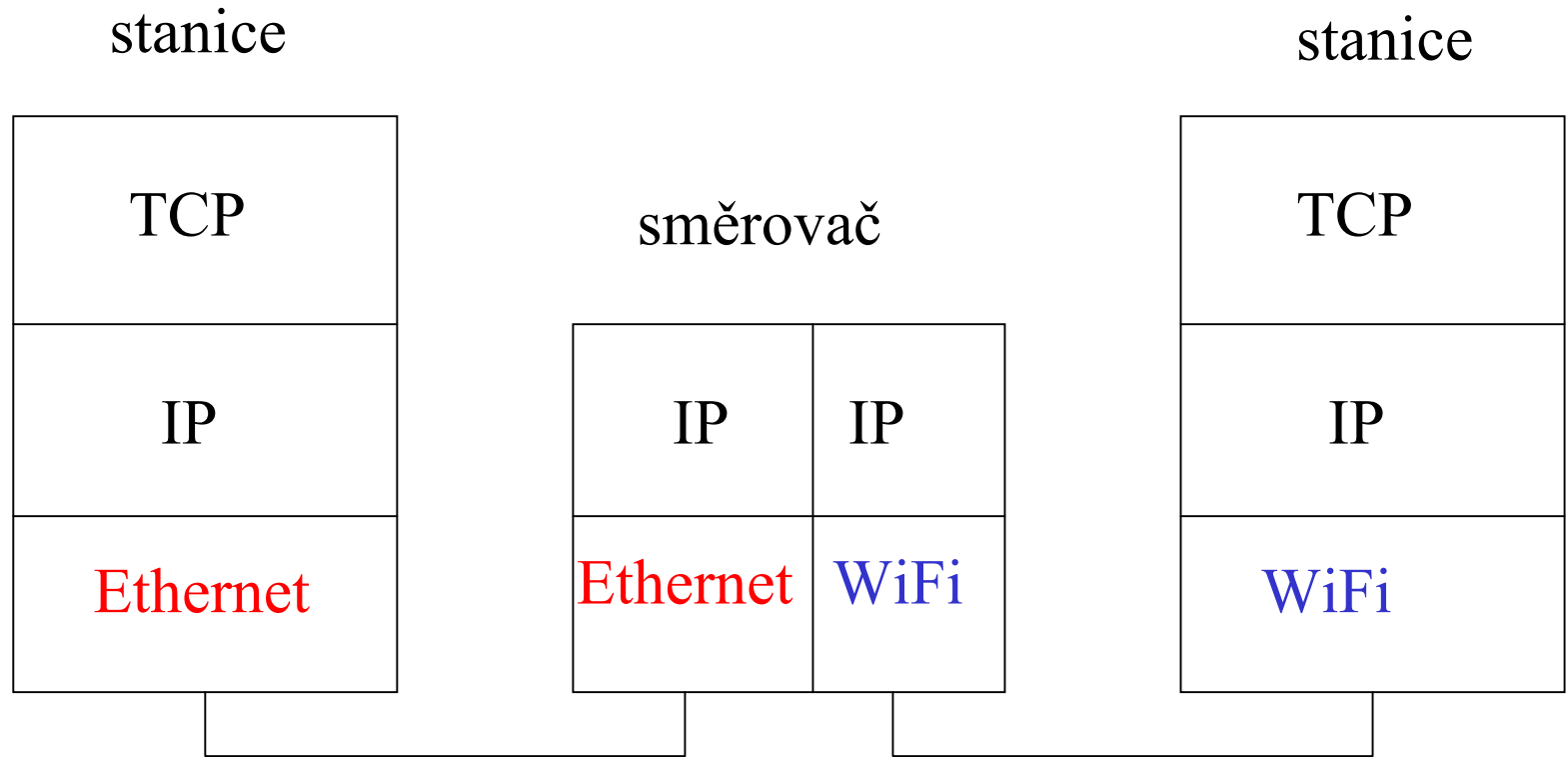
# Tok informací

- Peer-to-peer: odehrává se mezi ekvivalentními vrstvami komunikujících systémů
- Interlayer: odpovídá za předávání informací mezi jednotlivými vrstvami

# Implementační pravidla

- Přímá (naivní) implementace vrstev je vysoce neefektivní
- Princip redundance: Konkrétní funkce se smí vyskytovat pouze v jediné, a to logicky nejvyšší vrstvě (existují ale výjimky)
- Přesýpací hodiny: Musí být zachována jednotná adresace a směrovací a signalizační protokoly musí být vysoce kompatibilní

# Příklad využití vrstev



# Datové jednotky

7 Aplikační vrstva	-	ADU
6 Prezentační vrstva	-	PH   APDU
5 Relační vrstva	-	SH   PPDU
4 Transportní vrstva	-	TH   SPDU   TT
3 Síťová vrstva	-	NH   TPDU
2 Datové spoje	-	LH   NPDU   LT
1 Fyzická vrstva	-	coded LPDU

# Transportní protokol

- Požadavky:
  - de-multiplexing
  - spolehlivost
  - řízení toku dat (flow control)
  - reakce na zahlcení (congestion control)

# Řízení toku dat

- Omezení velikosti vysílaných dat
  - ochrana proti zahlcení přijímající stanice
- Koordinace mezi jedním vysílajícím a jedním přijímajícím
  - v podstatě end-to-end, pro každý tok zvlášť

# Reakce na zahlcení

- Omezení toku dat v celé síti
  - „ochrana“ interních bufferů (na směrovačích)
- Problém celé sítě
  - de facto globální optimalizační problém
  - odpovídá systému se zpožděnou zpětnou vazbou

# Zábrana nebo řízené zahlcení

- Stav zahlcení sítě a jeho důsledky
- Zábrana zahlcení
  - k zahlcení nikdy nedojde
- Řízené zahlcení
  - včas detekováno
  - rychlá reakce vedoucí k jeho odstranění

# Požadované vlastnosti

- Účinnost
  - maximální využití kapacity
- Férovost (fairness)
  - stejný přístup ke všem tokům dat
- Decentralizovaná správa
- Rychlá konvergence

# Kvalita transportních služeb

- Platí pro spojované služby
- Dohodne se při navázání spojení
- Parametry:
  - zpoždění při navazování spojení
  - zpoždění při rušení spojení
  - pravděpodobnost neúspěchu
  - chybovost přenosu dat
  - propustnost (minimální, průměrná, špičková)
  - zpoždění (minimální, průměrné, maximální)
  - rozptyl

# Kvalita služeb -- typy sítí

- A
  - spolehlivá síť (minimální ztráty, vysoká spolehlivost)
- B
  - občasné výpadky (zahlcení, technické závady)
- C
  - nespolehlivá síť (vysoká ztrátovost, technické problémy)

# TCP

- Transmission Control Protocol
- Služba
  - poskytuje zaručený proud slabik (žádné ztráty, zachovává pořadí)
- Protokol
  - segmenty (většinou 512 byte)
  - kumulativní potvrzování
  - řízení toku pomocí „okna“ (řízeno vysílajícím)
- Algoritmy
  - korekce ztrát
  - řízené zahlcení

# Uživatelé transportních služeb -- příjemci

- Adresa uzlu + adresa služby
  - aplikace příliš dynamické
  - multiplexing a de-multiplexing
- Adresa služby = **port**

# Porty

- Fronta (s bufferem)
- Číslování portů
  - well-known („dobře známé“) porty
  - dynamické porty
- Well-known porty
  - místo se známou konkrétní službou
  - čísla portů registrována a statická

# Dynamické porty

- Čísla přidělována dynamicky
- Nejsou zveřejňována
  - uživatel číslo získá (pro danou relaci) jako odpověď na požadavek zaslaný konkrétnímu well-known portu

# Obsluha portů

- Proces nad každým portem
  - neefektivní
- Společná obsluha (inetd, Internet démon)
  - spustí (fork) konkrétní službu až na základě explicitního požadavku

# TCP -- 4 základní algoritmy

- Pomalý start (slow start)
- Zábřana zahlcení (congestion aviodance)
- Rychlá retransmise (fast retransmit)
- Rychlé vzpamatování (fast recovery)

# TCP -- potvrzování

- Potvrzují se zasláné zprávy (včetně pozice)
- Používá piggybacking
  - duplexní protokol, potvrzení součástí zprávy posílané opačným směrem
- Jedno potvrzení na dva segmenty
  - pokud je přenos dat dostatečně rychlý
  - čekání max 500 ms
- Duplikované potvrzení
  - přijde neočekávaný segment

# TCP -- pomalý start

- Okno zahlcení (congestion window, cwnd)
  - omezení na straně vysílajícího
  - objem dat, který smí být vyslán, než přišlo potvrzení
- Příjemcem definované okno (receiver advertised window, rwnd)
  - omezení na straně přijímajícího
  - objem dat, který příjemce akceptuje
- $\min(\text{cwnd}, \text{rwnd})$

# TCP -- pomalý start II

- Vysílání zahájeno:  $cwnd=2$  segmenty
- Každé potvrzení zvýší  $cwnd$  o jeden segment
  - efektivně reprezentuje exponenciální růst
- Skončí, jakmile  $cwnd$  dosáhne velikost  $ssthresh$ 
  - $ssthresh$  na počátku nastaven např. na 64K
- Přechází do zábrany zahlcení

# TCP -- zábrana zahlcení

- Zvyšuje cwnd o jeden segment za každý RTT (round trip time, čas který data potřebují na cestu mezi vysílajícím a přijímajícím a zpět)
  - efektivně představuje lineární zvětšování cwnd

# TCP -- reakce na ztrátu

- Detekce ztraceného segmentu
  - timeout
  - duplikované potvrzení
- $ssthresh = \max(\text{FlightSize}/2, 2 * \text{segment})$ 
  - FlightSize -- objem dat „na cestě“ (cca odpovídá cwnd)
- $cwnd = \text{segment}$
- Návrat k pomalému startu

# TCP -- rychlá retransmise

- Reakce na duplikované potvrzení
  - síť je propustná, jde spíše o náhodnou chybu
- Příjem 3 duplikovaných potvrzení detekuje ztrátu segmentu
- Následuje zaslání ztraceného segmentu
- Nedojde k návratu na pomalý start, namísto toho se použije rychlé vzpamatování

# TCP -- rychlé vzpamatování

- $ssthresh = \max(\text{FlightSize}/2, 2 * \text{segment})$
- $cwnd = ssthresh + 3 * \text{segment}$
- pošle nová data
- dostane-li neduplikované potvrzení, nastaví  $cwnd = ssthresh$

# UDP -- User Datagram Protocol

- Nespojovaná služba
- Prostý přenos paketů
- Nezajištěná
  - odpovědnost na aplikaci
- Hlavička:
  - port příjemce a port odesilatele
  - délka
  - kontrolní součet

# IP -- Internet Protocol

- Síťová vrstva
- Doprava dat mezi uzly
  - TCP (UDP) dopravuje data mezi relacemi/aplikacemi
- Určeno pro velké sítě (internetworks)
- paket = payload data unit (PDU)

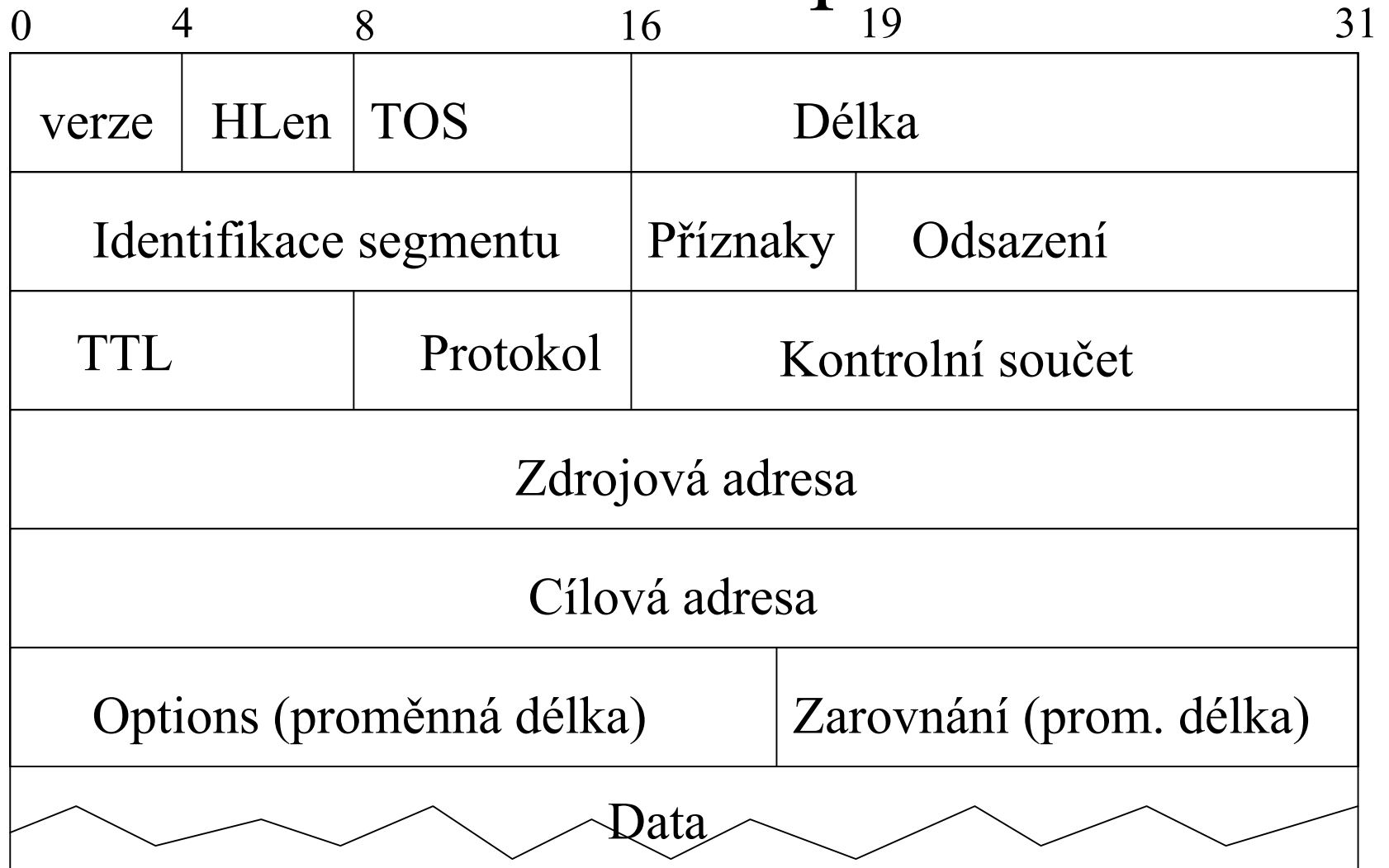
# Modely síťových služeb

- Doručení datagramů
  - connectionless, best-effort, nespolehlivé
  - není zaručeno doručení paketu
  - pakety se sítí pohybují nezávisle
- Virtuální kanály (virtual circuits)
  - connection-oriented
  - signalizace
  - pakety jedné relace prochází stejnou cestou

# IP

- Definován RFC791
  - IPv4
  - IPv6
- Globální hierarchické adresy
  - 32 bit v IPv4, 128 bit v IPv6
  - mapování na adresy nižší úrovně (ARP)
- Přenos sítí -- přepínání a směrování

# Formát IPv4 paketu



# IPv4 paket

- Verze je 4
- Hlen je délka = počet 32bitových slov
- TOS = Type of Service (nepoužíván)
- Délka = délka paketu (ve slabikách), 20--62K
- „slovo“ fragmentů -- při rozdělení paketu
- TTL = time to live
  - každý směrovač sníží o jedničku
  - je-li TTL=0m, paket je „zahozen“
  - zabraňuje nekonečným cyklům

# IPv4 paket

- Protokol = identifikuje protokol vyšší vrstvy (např. TCP, UDP)
- Kontrolní součet
  - přepočítáván směrovači (změna TTL)
- Zdrojová/cílová adresa
  - neměněna směrovači (ne zcela pravda)
  - neautentikováno

# Fragmentace

- Problém nižších vrstev
  - IPv4 rozloží a znovu složí (i rekurzivně)
  - IPv6 vrátí chybu (zdroj se dozví délku)
- Zpomalení přenosu
- Ztráta fragmentu = ztráta paketu
- Nalezení nejmenšího fragmentu na cestě
  - problém dynamických cest

# ICMP

- Internet Control Message Protocol
  - (RFC792)
  - doprovází IP
- Použití
  - odhalení chyb při přenosu paketů
  - zjišťování stavu sítě
- Obsahují část IP paketu který způsobil chybu

# ICMP zprávy

- Destination unreachable
  - „Destination“ může být protokol, port, uzel nebo celá síť
- Redirect
  - zkrácení cest
- TTL expired
  - používán programem traceroute
- Echo request/reply
  - používán programem ping

# ICMP omezení

- Ochrana proti rekurzivnímu generování
- Chybový ICMP paket *není generován* jako reakce na:
  - ICMP chybu
  - broadcast nebo multicast zprávu
  - poškozenou IP hlavičku (špatná cílová adresa)
  - chybu fragmentu (kromě prvního)
- Generování ICMP zpráv často výkonnostně omezeno

# Směrování

- Definice problému:

Nalezení cesty mezi dvěma uzly

- Cesta musí splňovat dodatečné podmínky

- Ovlivňující faktory:

- statické: topologie

- dynamické: zátěž

# Zasílání vs. směrování

- Forwarding vs. Routing
- Zasílání
  - každý směrovač samostatně (lokální proces)
  - představuje proces průchodu paketů směrovačem
- Směrování
  - společná činnost směrovačů (globální)
  - proces nalezení (a údržby) směrovacích tabulek

# Směrovací schemata

- Distribuované nebo centralizované
- „Krok za krokem“ nebo zdrojové
- Deterministické nebo stochastické
- Jedno nebo více cestné
- Dynamický nebo statický výběr cest
  
- Internet je modrý

# Směrovací algoritmus - Funkce

- výběr komunikační cesty
  - Propustnost
  - Zpoždění
- vlastní doručení dat
  - protokoly
  - směrovací tabulky

# Směrovací algoritmus - Vlastnosti

- Správnost
- Jednoduchost
- Robustnost
- Stabilita
- Spravedlivost (fairness)
- Efektivnost
- Optimálnost

# Směrovací algoritmus - Otázky

- Definice optimálnosti
  - Co je *nejlepší* cesta?
- Škálovatelnost (miliony uzlů)
  - minimalizace (objemu) řídicích dat
  - minimalizace směrovacích tabulek
- Robustnost
  - chyby uzlů, přenosových tras, ztráty
  - nezbytný distribuovaný algoritmus

# Globální pohled

- Použití globální znalosti
  - je složité ji získat
  - není aktuální
  - musí být lokálně relevantní
- Rozpor mezi lokální a globální znalostí může způsobit
  - cykly (černé díry)
  - oscilace (adaptace na zátěž)

# Reprezentace

Sít' reprezentována jako graf

- Uzly:
  - adresy/jména
- Hrany
  - ohodnocení = cena komunikace

# Směrovací techniky

- Výkonnostní kritéria
  - minimalizace počtu skoků
  - minimalizace „ceny“ cesty:
    - Zpoždění:  $\text{cena} = \text{délka fronty}$
    - Propustnost:  $\text{cena} = 1 / (\text{přenosová kapacita})$

# Směrovací techniky - Rozhodování

- Okamžik
  - při uzavírání spojení
    - spojované služby, virtuální kanály
  - při příchodu dat (paketu)
    - nespojované služby, datagramy
- Místo
  - jediný uzel: centralizované algoritmy
  - každý uzel: distribuované algoritmy

# Kategorizace

- Statické algoritmy
  - jednorázové (často ruční) tabulky
  - neflexibilní -- vhodné pro statickou topologii
- Dynamické algoritmy
  - aktualizace směrovacích tabulek
  - flexibilní/robustní
  - „potřebují“ protokol pro aktualizaci tabulek

# Dynamické algoritmy

- Centralizované
  - stav se posílá do centra
  - centrum posílá tabulky uzlům
- Izolované
  - každý uzel „sám za sebe“
- Distribuované
  - vzájemná kooperace uzlů

# Statické směrování

- Známa topologie = pevné cesty
- Centrální směrovací tabulka
  - zpracována off-line
  - může být optimální vzhledem k zadaným kritériím
- Next-hop v každém uzlu

Pozitiva: jednoduché

Negativa: citlivé na výpadky nebo zátěž

# Izolované směrování

- Náhodná procházka
  - vysoká robustnost
- Výběr informací z procházejících paketů
- Záplava (broadcast)
  - kopie všem kromě zdroje
  - mimořádná robustnost
  - optimální
  - enormní zátěž sítě
- Vyžaduje zpětnou vazbu

# Dynamické směrování

- Periodická výměna směrovacích informací
- Dynamické výměny tabulek
  - dočasná nekonzistence
- Hierarchie směrování
  - sítě sítí
  - implicitní (default) cesta k neznámým cílům

# Hierarchie směrování

- Hierarchie směrovačů
- Směrování k sítím (autonomní systémy)
- Směrování uvnitř sítí
- Identifikace sítí a uzlů
  - adresa sítě
  - adresa uzlu

# Metrika

- Definice optimality
- Výměna informace o vzdálenosti
- Teorie grafů: Minimální kostra grafu (Minimal Spanning Tree, MST)
- Distance Vector
  - počet přechodů do cíle
- Link State
  - dostupnost sousedů

# Směrování Distance Vector

- Předpoklad:
  - každý směrovač zná pouze cestu a cenu k sousedům
- Cíl:
  - směrovací tabulka pro každý cíl v každém směrovači
- Idea:
  - řekni sousedům svou představu tabulky

# DV Algoritmus

- Distance Vector = dvojice  $\langle \text{Cíl}, \text{Cena} \rangle$
- Každý směrovač udržuje tyto vektory ke všem cílům
  - inicializace: sousedé se známou cenou, zbytek nastaven na nekonečno

# DV Algoritmus II

- Periodicky zašle kopii svých DV sousedům
  - Pokud je cesta v získaném DV zvětšená o cenu cesty k sousedovi, který DV zaslal lepší než současná znalost, aktualizuj vlastní DV a směrovací tabulku
- Konverguje pro statickou topologii
  - reakce na změny topologie

# Změny topologie

- Jednoduchý příklad
  - výpadek hrany -- cena vzroste na nekonečno  
postupně se dozví jinou cestu
- Nebezpečí zacyklení

# Zacyklení

$A/2 \longrightarrow B/1 \xrightarrow{\text{red}} \text{Internet}$

Předpokládejme, že červený spoj selže

# Zacyklení II

- B nastaví cestu do Internetu na nekonečno
- B se dozví, že A zná cestu do Internetu s cenou dvě, tak si nastaví cenu 3
- A se dozví, že B zná cestu do Internetu s cenou tři a zvýší svou cenu na 4 (jinou cestu do Internetu nezná)
- B se dozví, že A zná cestu s cenou 4 ...

# Dělení horizontu

- Triviální řešení uvedeného problému
- Směrovač nikdy nesděljuje cestu zpět uzlu, od nějž se ji dověděl
  - Je možno přímo oznamovat nekonečnou vzdálenost
- Problém zůstává ve složitějších topologiích
  - Navržena řada rozšíření

# RIP (Routing Information Protocol)

- DV protokol používající počet uzlů
  - nekonečno = 16
  - rozdělení horizontu
- Směrovače zasílají informaci každých 30 sekund
  - triggered update při změně stavu hrany
  - časový limit 180s (detekce chyb spojení)

# RIP

- RIPv1 definován v RFC1058
- RIPv2 (přidal např. autentizaci) v RFC1388

# Link State směrování

- Stejné předpoklady a cíle jako DV
- Jiná idea:
  - šíří se topologie, cesty si směrovače počítají samy
  - Dvě fáze
    - šíření topologie (záplava)
    - výpočet nejkratší cesty (Dijkstra)

# Motivace

- V DV je výpočet cesty implicitní, nelze jednoduše rozhodnout co použít, pokud došlo ke změně topologie
- LS rychleji konverguje a mělo by být stabilnější
- Zápor: složitější algoritmus

# Záplava

- Každý směrovač udržuje databázi LS a periodicky zasílá LS pakety (LSP) sousedům
- Každý směrovač posílá dále LSPs které neměl ve vlastní databázi na všechny sousedy kromě toho, od něž tuto informaci dostal
- Spolehlivost: zajištěna potvrzením

# Link State Paket

- Identifikátor uzlu
- Cena spojů k sousedům
- Pořadové číslo
- Doba platnosti (TTL)

# Potenciální problémy

- Výpadek spoje nebo směrovače vynutí odstranění starých údajů
  - SEQNO definuje nová data
  - Pošle nové LSP s cenou rovnou nekonečnu
- Restart směrovače
  - SEQNO=0 a pošle nové LSP s TTL=0
- Obnova spojení (po rozdělení sítě)
  - Synchronizace LS databází

# Dijkstrův algoritmus nejkratší cesty

- Necht'

$N$  je množina všech uzlů v grafu (síti)

$l(i,j)$  označuje nezápornou cenu hrany (spoje)  $(i,j)$

$s$  je aktuální (zdrojový) uzel

$M$  množina uzlů

$C(n)$  cena cesty z  $s$  do  $n$ ;  $\infty$  pokud cesta neexistuje

# Dijkstrův algoritmus

$$M = \{s\}$$

for each  $n$  in  $N \setminus M$

$$C(n) = l(s, n)$$

while ( $N \neq M$ )

$$M = M \cup \{w\}$$

tak, že  $C(w)$  je minimální pro všechna  $w$  z  $(N \setminus M)$

for each  $n$  in  $N \setminus M$

$$C(n) = \min(C(n), C(w) + l(w, n))$$

# OSPF

- Open Shortest Path First
- Nejpoužívanější LS protokol v současnosti
- Prezentovaný protokol s řadou rozšíření
  - autentizace směrovacích zpráv
  - další úroveň hierarchie: směrovací oblasti
  - Load balancing: vícenásobné cesty se stejnou cenou

# Metriky oceňování spojů

- Způsob výběru ceny
  - Propustnost, zpoždění, ztráty, ...?
  - Je funkcí zátěže?
- Statické metriky
  - Počet skoků (hop): snadné, nerozlišuje kvalitu
  - Optimalizace možná manuální úpravou cen
- Dynamické
  - Závisí na zátěži, zabraňují přetížení
  - Mohou oscilovat (nezbytné tlumení)

# Příklad: ARPANET

- Původní metrika:
  - počet paketů ve frontě
- Modifikovaná metrika:
  - cena spoje = průměrné zpoždění
    - časová razítka: doba příchodu (AT) a odeslání (DT)
    - DT-AT charakterizuje „spolehlivost“ spoje
  - zatížení zvyšuje cenu („přidává“ skoky)

# Rozšiřitelnost směrování

- Růst režie
  - velikost směrovacích tabulek
  - objem vyměňované informace
  - objem nezbytných výpočtů
- DV i LS fungují dobře na omezeném počtu uzlů

# Rozšiřitelnost II

- Nové techniky
  - Hierarchická adresace
  - Strukturovaná hierarchie
  - Agregace cest

# IP adresy

- 32 bitů, členění AAA.BBB.CCC.DDD
- Hierarchie -- Sít':uzel
  - třídy (podle nejvyšších bitů)
    - A  $2^{24}$  uzlů (nejvyšší bit 0: např. 10.0.0.0)
    - B  $2^{16}$  uzlů (nejvyšší bity 10, např. 147.251.0.0)
    - C  $2^8$  uzlů (nejvyšší bity 110, např. 195.162.13.0)
- Neefektivní
- Příliš mnoho sítí

# Subnetting

- Řeší problém nedostatku adres
- Namísto dvojice Síť:Uzel je vytvořena trojice Síť:Podsíť:Uzel
- Používá masku: např. 147.251.10.128

# Doručování

D je cílová IP adresa

Směrovací tabulka tvořena trojicemi

(číslo podsítě, maska podsítě, příští uzel)

$D1 = D \& \text{maska podsítě}$

if  $D1 == \text{číslo podsítě}$  then

    if příští uzel je „lokální síť“ then doprav do D

    else doprav do příští uzel