Chyba spoznamkovane P2P, Ad-Hoc a Multimedia.

Na konci je vela vypracovanych otazok ku vsetkym temam.

GLHF

-Zeleninari

[**Síťové charakteristiky**](#_vzztgju2uw6n) **3**

[**Protokol**](#_6sdodofbyjpw) **3**

[**Standardizace protokolu**](#_pcpsg2q4ab2x) **3**

[**ISO/OSI**](#_p6jrqwnvgrmn) **3**

[**IPv6**](#_4imu5qlyxta8) **4**

[Hlavička a segment](#_xi73gu3q30ap) 4

[Adresy](#_sx8qy0s6gcg0) 5

[Šintoks](#_dn0c8j5dty9b) 5

[Bezpečnost](#_ssclm5x68c2y) 5

[IPsec](#_pxpiuu7k0tp0) 6

[Mobilita](#_9v3rtr6199jt) 7

[QoS](#_3fx856mhrp83) 7

[Porting](#_vsjcvmgkuzkm) 8

[Neighbor Discovery Protocol](#_pbyvl13sbn6l) 8

[DAD - Duplicate address detection](#_2gin5r9vnwnf) 9

[NUD - Neighbor Unreachability Detection](#_utp3wqcmmeo1) 9

[Autokonfiguracia](#_fryvcab6d7ys) 9

[**Routing**](#_74wyfzap6tmn) **11**

[Distance vector DV](#_84ugq3mgtuys) 11

[Link State LS](#_7ihl8hu1xktk) 11

[Path Vector PV](#_kiksow4ke78z) 11

[Autonomní systémy AS](#_59ymwaokt40l) 11

[Interní směrování - IGP - Interior Gateway Protocols](#_pxlau7og89r0) 12

[Vnější směrování - EGP - Exterior Gateway Protocol](#_9sa53hbq07j4) 12

[DV - Protokoly směrování](#_x82eh9nyt1hy) 12

[RIP](#_1ddh8fluos7g) 12

[IGRP](#_f6zkbj9pshuu) 12

[EIGRP](#_h39d4flllvtl) 13

[LS - Protokoly](#_cm13eqnn84m1) 13

[OSPF](#_1t5vsm2bvfjj) 13

[IS-IS](#_6536f37hoeh3) 13

[PV - Protokoly](#_jsrgx7324v0x) 13

[BGP](#_jmf7z3am259i) 13

[Routery - Směrovače](#_jx1d4e2znagc) 14

[Uloha](#_p7so61v4nza1) 14

[Flow](#_qwbl9nwnx6wl) 14

[Elementy](#_a1k6fxspmpw2) 14

[**TCP**](#_f19m1a67l70z) **14**

[Traditional TCP](#_yrovia1rnzvq) 14

[Tahoe](#_n75ygq2xw3qw) 15

[Reno](#_8ocvq4abnyq9) 15

[Vegas](#_uyoge2u70n7g) 15

[Improving traditional TCP](#_jj4yljkbl5rk) 16

[Multi-stream TCP](#_55b6gd8pz24z) 16

[Web100](#_upf1byrufyv1) 16

[Konzervativne rozšírenie TCP](#_ysew7qgzfkef) 16

[GridDT](#_5zub8865t3ik) 16

[Scalable TCP](#_coc2pucom96l) 16

[HSTCP](#_cf023qpx803r) 17

[H-TCP](#_b0x243n3iioq) 17

[BIC-TCP](#_l1vkifjleip7) 17

[CUBIC-TCP](#_1cdqejcexo8g) 17

[Rozsirenia TCP s podporou IP](#_m5q67ontsg75) 17

[Quickstart](#_jzed0lpt5e2r) 17

[E-TCP](#_rv6j2uhhj1tz) 18

[FAST](#_7325yzy8gbsw) 18

[Hipsterciny ine ako TCP](#_hr9v54nkg0jl) 18

[Tsunami](#_83xcjew91iko) 18

[Reliable Blast UDP](#_jvn4oqqsvkkl) 18

[eXplicit Control Protocol](#_9ebkz2morr4e) 18

[SCTP](#_bj5zf2p84iot) 18

[DCTP](#_t1a87xouvqwf) 19

[STP](#_sndlkvgbmwb) 19

[Reliable UDP](#_n3j7i6rhax6l) 19

[Zhrnutie](#_ypl8xlye1nf6) 19

[**Otazky**](#_e4a67tezswpy) **20**

[ISO/OSI & Basics](#_t3vhfdfyzz3p) 20

[IPv6](#_mrwv8lr1hn6f) 22

[Routing](#_3wp3lg68tngd) 25

[EIGRP](#_ei50f7yoijvq) 27

[TCP](#_nbxa27hkizvm) 28

[P2P](#_5gxrf2tv6df5) 29

[AD-HOC](#_f9w4td280cpi) 35

[Media](#_skd88bm9zqfo) 37

[**Otázky - Vyjebaný bonus (30.1.2018)**](#_eg1yj2vzggqm) **39**

[**Otázky - Vyjebaný bonus 2 (12.12.2017)**](#_ntfej8jrzp0q) **39**

# 

# Síťové charakteristiky

**Delivery** - Jsou data doručená správnému cíli?

**Accuracy** - Jsou data doručená všechna?

**Timeliness** - Jsou data doručena včas?

# Protokol

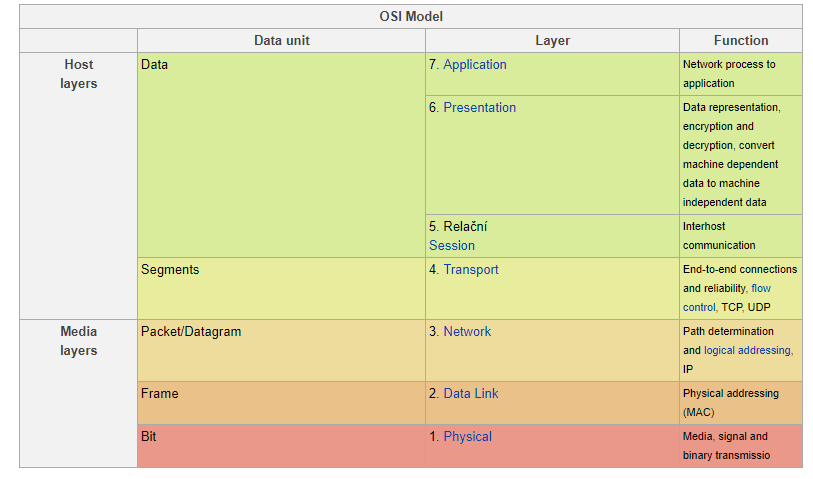
je sada pravidel, které definují formát a pořadí zpráv mezi dvěma komunikujícíma entitama. Protokol také definuje akce, které mezi přenosem vykonávají.

# Standardizace protokolu

de jure - standard vytvořen uznanou organizací

de facto - standard, na kterém se dohodli výrobci

# ISO/OSI



L1 - IEEE 802

L2 - PPP

L3 - IPv4

L4 - TCP / UDP

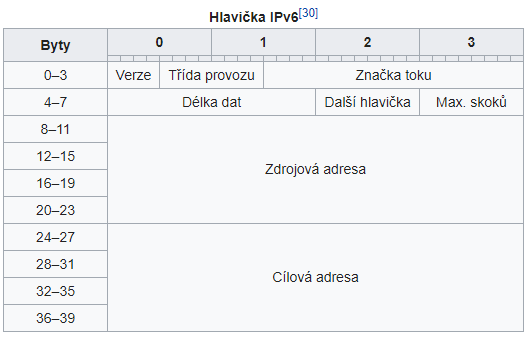
L5 - ASP

L6 - AFP

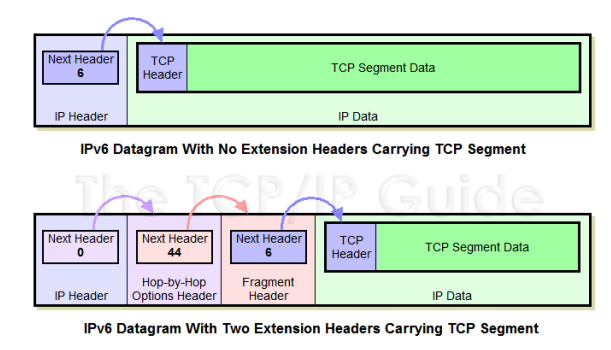
L7 - FTP / HTTP

# IPv6

## Hlavička a segment



* verze — 4 bity, verze 6
* třída provozu — 8 bitů na prioritu paketu. Úroveň priority se dělí na rozsahy: kde zdroj podporuje kontrolu přetížení a bez podpory kontroly přetížení.
* Značka toku — 20 bitů pro správu QoS. Původně určeno pro speciální obsluhu aplikací reálného času, nyní se nepoužívá.
* Délka dat — 16 bitů pro délku těla paketu. Při vynulování se nastaví „jumbo“ tělo (skok za skokem)
* Další hlavička — 8 bitů, určuje další vnořený protokol. Hodnoty se shodují s hodnotami definovanými pro IPv4.
* Zdrojová a cílová adresa — 128 bitů na každou adresu.



## Adresy

Třídy zatím neexistují

Sítě jsou definovány pomocí CIDR

* **Unicast** - posílání dat jednomu příjemci
* **Multicast** - posílání dat více příjemci
* **Anycast** - posílí snídat skupině příjemců, přičemž dat ase pošlou pouze tomu nejbližšímu

## Šintoks

2001:0db8:0000:0000:0000:0000:1428:57ab to samé jako 2001:db8::1428:57ab

Existují ale spešl adresy jako ff00::/8 - multicast

tá 8 znamena 8 zarezervovanych bitov (ff)

## Bezpečnost

* CIA
  + **Confidentiality-** informace nesmí být přečtena ani zmodifikována neautorizovanou entitou
  + **Integrity** - veškeré modifikace informace musí být jasně identifikovatelné
  + **Availability** - veškeré informace musí být přístupné autorizovaným uživatelům
* AAA
  + **Autorizace** - přístup
  + **Autentizace** - práva
  + **Accounting** - Účtování znamená sledování využívání síťových služeb uživateli. Tyto informace mohou být použity pro správu, plánování, účtování, nebo další účely
* Nerozporuplnost - poslání, získání nebo smazání by nemělo být nikdo zakázáno jinou entitou

### IPsec

Bezpečnost je integrována v IPv6 nativně, a jeho použití je povinné.

Elementy

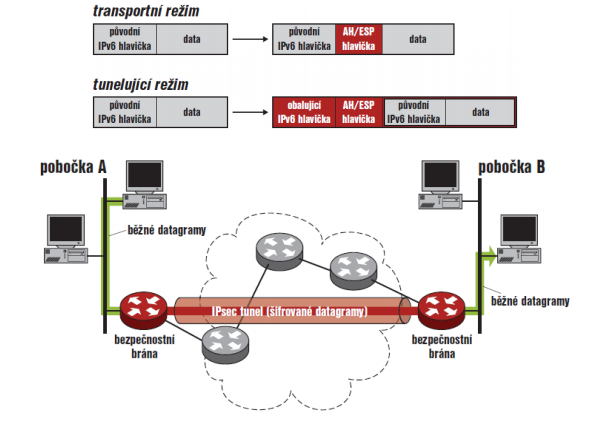
* Protokoly
  + **AH** protokol - autentizační protokol - autentizuje celý datagram + jednodušší a nevyžaduje takovou výpočetní sílu
    - Hlavička
  + **ESP** protokol - Encapsulating Security Payload - lepšejší
    - Hlavička
    - Trailer
    - Auth Data
  + Oba jsou důležité
* Co použít za šifrovací algo
* Co použít za bezpečnostní pravidla mezi dvěma uzly (**SA**)
  + Klíč
  + Šifrovací a autentizační mechanismus
  + Parametry
    - **SPI** - Security Parameter Index - 32bit číslo pro identifikaci SA
    - IP cílová adresa
    - Identifikátor bezpečnostního protokolu (AH nebo ESP)
* Správá klíčů

**IKE verze 2** - Internet Key Exchange version 2

* Automaticky vytvoří SA mezi dvěma uzly
* zařídí i jejich vzájemnou autentizaci

Mody přenosu

* Transport
* Tunnel



## Mobilita

* Princip trvalé adresy ve skutečném světě
  + Home address - globální unicast trvalá adresa, přes kterou je mobilní uzel vždy přístupný.
  + Care-of address - globální unicast adresa, mobilního uzlu, který je v cizí síti (jiná než Home adresa)
* **HA** - home agent
  + router v domácí síti, přes který je mobilní uzel vždy přístupný
  + Podporuje Home Agent Address discovery - dohledání HA jiných uzlů
  + Spravuje uložený seznam ostatních HA v síti
  + Zpracovává Care-of adresy
  + Přeposílá Care-of adresy
* **Return Routability Procedure** - povoluje odpovídajícímu uzlu získat jistotu, že mobilní uzel je adresovatelný CoA a HA
* Kroky RRP
  + Mobilní uzel pošle odpovídajícímu uzlu nějakou **HoTI(Home Test Init)** zprávu přes HA
  + Mobilní uzel posílá **CoTI(Care-of Test Init)** zprávuodpovídajícímu uzlu napřímo
  + Odpovídající uzel odpovídá, že posílá **HoT** zprávu přes HA
  + Odpovídající uzel odpovídá, že posílá **CoT** zprávu
  + Oba uzly nyní mohou si vypočítají 20 bajtový klíč, který bude sloužit jako ochrana k Binding Update zprávy

## QoS

* Dvě hlavní architekury
  + Integrated Services
    - Aplikace oznámí síti, své požadavky
    - Síť čekuje zda dokáže splnit (Admission Control)
    - Pokud ne tak aplikace se pokusí snížit požadavky
    - Až jsou požadavky cajk, tak síť informuje všecky uzly
      * rezervují se zdroje (RSVP, YESSIR)
    - Nevýhody: je třeba spravovat stavy každého uzlu v síti -> škálovatelnost na hovno
  + Differentiated Services
    - Není potřeba informovat síť ohledně požadavků -> rezervace není potřeba
    - Každý paket je před odesláním označen prioritou
    - Výhody
      * netřeba spravovat stavy uzlů v síti -> škálovatelné dobře
      * jednodušší na impl
      * menší delay (netřeba nic rezervovat)
* Hlavička
  + Možno použít dvě místa
  + **Traffic class**
    - 6 bit + poslední 2 bit
  + **Flow label** 
    - 20 bit

## Porting

* Dvojitá implementace - při přechodu jednoduše smažeme implementaci pro 4. Je třeba ale implementovat pro 2 protokoly. Je to nákladné a zabírá zdroje.
* Překládač
* Obalení IPv6, který by se navenek choval jako IPv4 - Ušetří mnoho práce, ale nemůžou se používat featury IPv6.

## Neighbor Discovery Protocol

* Způsob pomocí kterého můžeme získat adresu cílového uzlu.
* Podobné ARP u IPv4
* Oproti ARP ma nove funkcionality:
  + Autokonfigurovanie IPv6 adries
  + Ziskavanie prefixov siete a routerov
  + Duplicate Address Detection
  + …
* Pouziva 5 ICMP (kontrolnych) sprav:
  + **R**outer **S**olicitation (**RS**)
  + **R**outer **A**dvertisement (**RA**)
  + **N**eighbor **S**olicitation (**NS**)
  + **N**eighbor **A**dvertisement (**NA**)
  + **Redirect**
* Dost dobre vysvetlene [v tomto videu](https://www.youtube.com/watch?v=A3LFt7CHpgs).
* **NA** moze byt aj nevyziadana - v pripade zmeny L2 adresy

### DAD - Duplicate address detection

* Pocas autokonfiguracie
* Host posle **NS** multicast so svojou adresou ako cielovou adresou
* Zdrojom je adresa :: (iba 0)
* Ak existuje node s rovnakou adresou, odpovie **NA** -> adresa je zabrana

### NUD - Neighbor Unreachability Detection

* Node periodicky kontroluje dosiahnutelnost svojich susedov, s ktorymi komunikuje
* Mozne cez protokol na vyssej urovni (napr. TCP), alebo vlastnou implementaciou v IPv6
* Adresa suseda moze byt v stavoch:
  + **Incomplete** - Prave sa vykonava adress resolution, caka sa na response/timeout (**NS** sent, **NA** not received yet)
  + **Reachable** - Sused je dosiahnutelny a jeho *ReachableTime* je stale ok
  + **Stale** - Preslo viac ako *ReachableTime* suseda od poslednej spravy od suseda
  + **Delay** - Ako stale, ale spomalene kvoli protokolu na vyssej vrstve (?)

### Autokonfiguracia

* Cielom je obist manualnu konfiguraciu
* Oba druhy autokonfiguracie sa daju kombinovat
* **Stateful**
  + Velmi podobna **DHCP** (Nazyvana **DHCPv6**)
* **Stateless**
  + Pocita s predpokladom routerov, ktori vedia vsetko potrebne
  + Periodicky informuju celu siet o aktualnej konfiguracii (**RA**)
  + V pripade pripojenia noveho nodeu si poziada o konfiguraciu pomocou **RS**, alebo pocka na periodu
  + **RA** obsahuje informacie ako MTU (max. velkost), prefix, …
  + Pri generovani svojich adries postupuje nasledovne:
    1. **Link-Local Address Generation**
       1. Vygeneruje si *tentative address*
       2. Adresa s prefixom FE80, **54 nulami** a **64-b** **identifikatorom**
       3. Identifikator je bud **MAC** adresa, alebo **nahodne ID**
    2. **Link-Local Uniqueness Test**
       1. Skor pri pouziti ID ako MAC adrese, nepravdepodobne
       2. Pomocou **DAD**
    3. **Link-Local Address Assignment**
       1. Assignuje tuto adresu ako svoju IP
    4. **Router Contact & Direction**
       1. Pouzije **RA** a **RS** na informacie co dalej
       2. V pripade specifickych instrukcii z **RA** ich vykona (napr. stateful konfiguracia IP o DHCP server)
    5. **Global Address Configuration**
       1. Zlozena z **prefixu siete**, ktory doda router, a **identifikatoru** z prveho kroku
  + Podporuje pripadne flagy prijate z **RA**
    1. **M** - Pouzitie stateful konfiguracie na adresy (napr. DHCPv6)
    2. **O** - Pouzitie stateful konfiguracie na ine veci ako adresy

# Routing

proces hledání cesty v síti mezi dvěma uzly. Jeho hlavním cílem je co nejefektivněji optimalizovat tyto cesty.

**Hop-By-Hop** - Router jako takový se nestará o celou cestu ale pouze o cestu k dalšímu uzlu

## Distance vector DV

* Bellman-Ford algo
* sousedící směrovače si pravidelně vyměňují a kopírují celou směrovací tabulku
  + nebo pokud se změní topologie sítě
* “informace o celé síti poskytuji pouze sousedům”
* vhodný iba pre menšie siete (max. vzdialenosť 15)
* nezvládá cykly, count-to-infinity problem

## Link State LS

* Dijskrův algo
* směrovače si pravidelně vyměňují informace o jejich spojeních(links) ke kterým jsou připojeny
* algo využívám pro vyhledávání nejkratší
* “informace o mých sousedech všem”
* vhodný pre vacsie siete

## Path Vector PV

* varianta DV
* narozdíl od DV jsou posílány celé cesty
  + snadné nalezení cyklů
  + je možno nadefinovat pravidla, pomocí kterým se můžou směrovače řídit

## Autonomní systémy AS

* Menší sítě (domény)
* Cílem je snížení přehnaného přesměrovávání -> minimalizace výměny dat mezi směrovači
* ASN - 16 bit - identifikátor
* Každý AS je spravován jednou organizací - např. CESNET
* Styly připojení AS k Internetu
  + Stub AS
  + Multihomed AS
  + Transit AS

### Interní směrování - IGP - Interior Gateway Protocols

* Směrování mezi uzly uvnitř AS
* Hlavním cílem je efektivita

### Vnější směrování - EGP - Exterior Gateway Protocol

* Směrování mezi AS
* Hlavním cílem je podpora stanovených pravidel a škálovatelnost

## 

## DV - Protokoly směrování

### RIP

* počet hopu slouží jako metrika
* směrovače posílají informace každých 30s
  + přes UDP
  + timeout 180s
  + podpora aktualizace stavu spojů mezi uzly
* Vhodný pro menší sítě a stabilní spojení
* Nevhodný pro redundantní sítě
* Verze
  + RIPv1 - dnes nepoužívaná
    - !!zprávy jsou broadcastované !!
  + RIPv2
    - maska podsítě
    - multicast (broadcast už ne)
    - 16 hopů = nekonečno
  + RIPng
    - rozšíření verze 2
    - podpora IPv6 adres a sítí (IPv6 native)

### IGRP

* Interior Gateway Routing Protocol
* podobné jako RIPv1
* multicast
* DV aktualizace zahrnují 5 různých metrik pro každou cestu
* povoluje multiple cesty -> load balancing
* pracují přímo s IP
* má checksum
* Výpočet ceny spojů mezi uzly
  + metrika = koeficienty(K1, ..., K5) \* bandwidth (B) \* delay (D) \* reliability (R) \* load (L)

### EIGRP

* Enhanced Interior Gateway Routing Protocol
* IGRP ale “lepší”
  + směrování bez cyklů
  + konvergence
  + rychlé
  + chatty protocol

## LS - Protokoly

### OSPF

* Open Shortest Path First
* nejpoužívanější LS protokol
* získává informace od přístupných směrovačů a sestrojí si topologii sítě
* metrika je cena (1- 65535)
* do verze 2 byla přítomna autentizační zpráva (pak začlo spoléhat na IPv6)
* load balancing
* směrování v prostoru
  + vrstvy hierarchie
  + AS mohou být rozděleny do subdomén - routing areas
* CIDR
* multicast

### IS-IS

* velmi podobné jako OSPF
  + vrstvy hierarchie
  + support variable length subnet masks
* metrika 0-63 (případně 0-16000 000)
* nemá datagramy -> bettah sekjurity
* lépe pracuje s IPv6

## PV - Protokoly

### BGP

* Border Gateway Protokol
* CIDR
* HOP metrika v 2. úrovni
* TCP, port 179
* Internal BGP
  + mechanismus poskytující informace sousedícím AS svým interním směrovačem
  + peer oznámí cesty získané přes EBGP svým IBGP peerům
  + ALE peer neoznamuje cesty získané přes jiné IBGP

## Routery - Směrovače

### Uloha

* Najít cestu v okolí
* Posílat data správným směrem

### Flow

* Přišel paket
* Přečte IP Hlavičku
* Snížit TTL a vyhodnocení TTL
* Checksum rekalkulace
* Vyhodnocení MTU
  + Fragmentace - IPv4
  + ICMP zpráva - “packet too big” -IPv6
* Klasifikace paketů

### Elementy

* Network Interfaces
* Forwarding Engines
* Queue Manager
* Traffic Manager
* Backplane
* Route Control Processor

# TCP

**Aggressiveness** - Schopnost vyuzit cely bandwidth

**Responsiveness** - Schopnost zotavit sa pri strate paketov

**Fairness** - Ferova priepustnosť siete pri vyuzivani viacerymi uzivatelmi

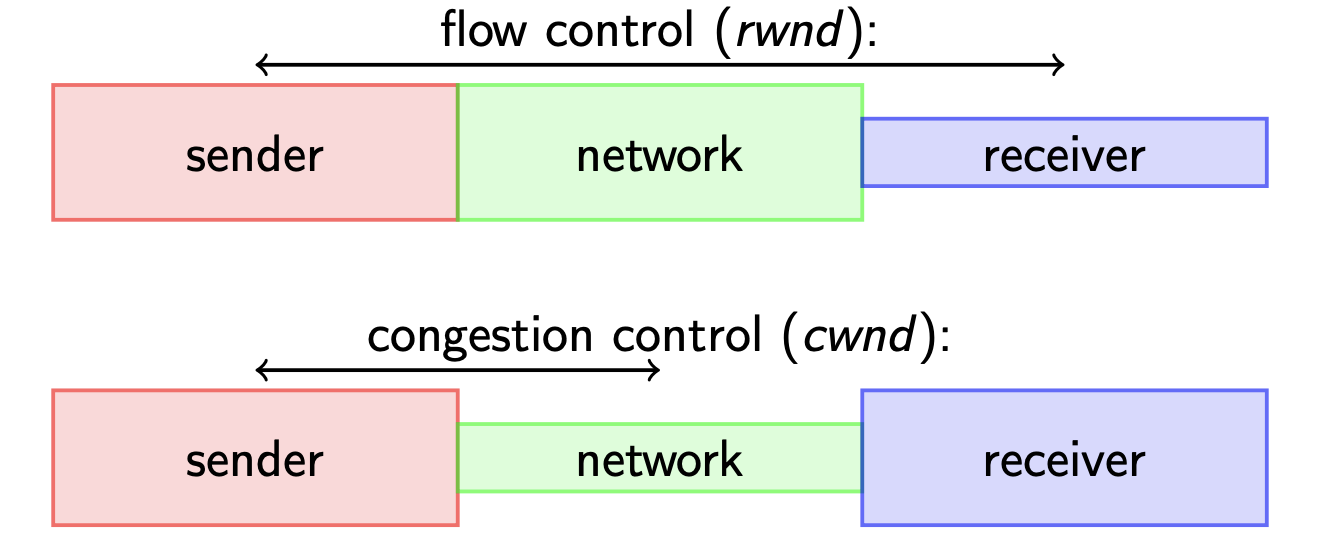
## Traditional TCP

**Flow Control** - Deterministicky feedback od príjemcov (*rwnd*)

**Congestion Control** - Odhad odosielateľa o volnej priepustnosti siete (*cwnd*)

^ O tomto odhade su vsetky tie dalsie algoritmy!

Výsledná velikost okna pri pouziti oboch: ***ownd* = min{*rwnd*, *cwnd*}**

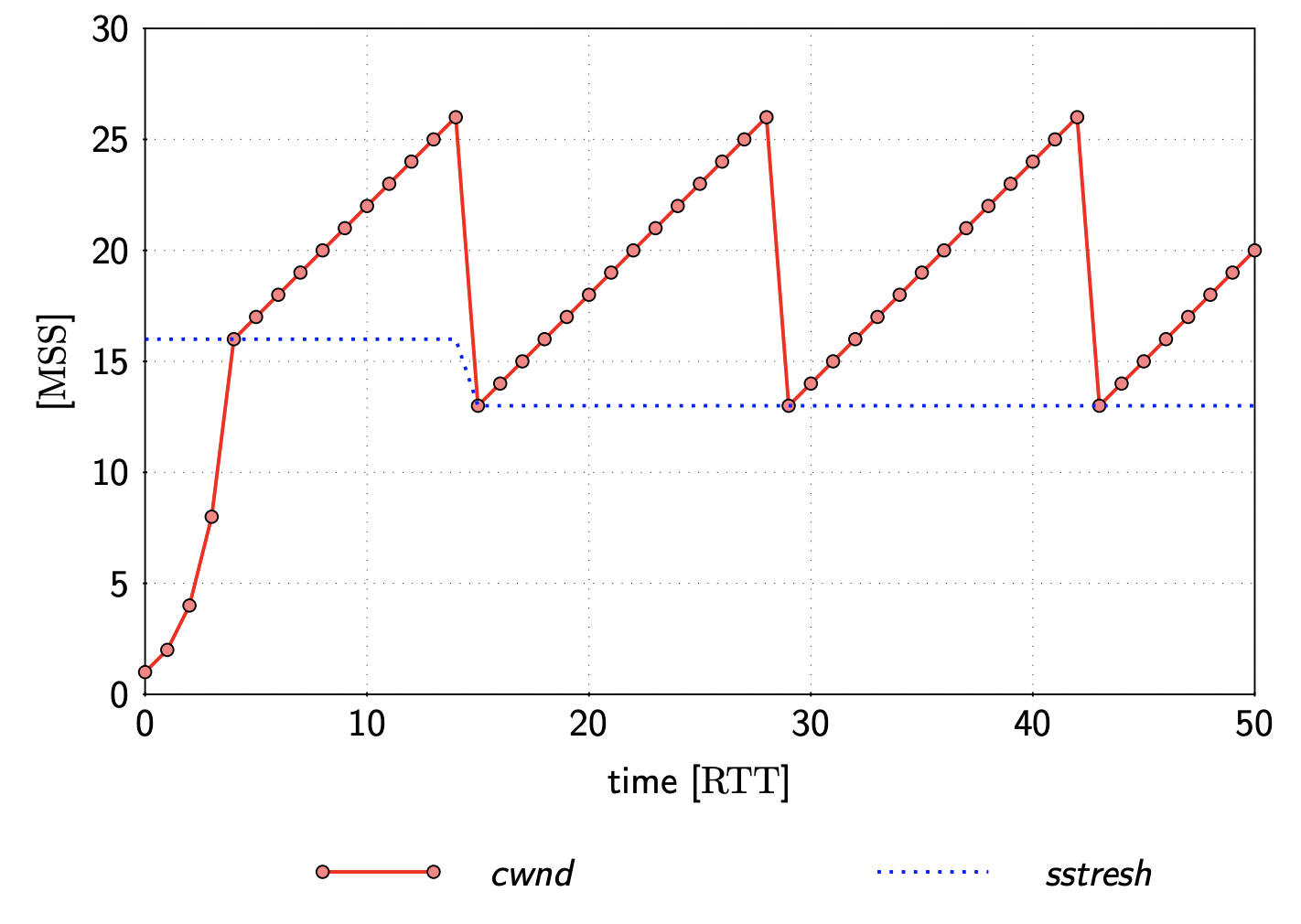
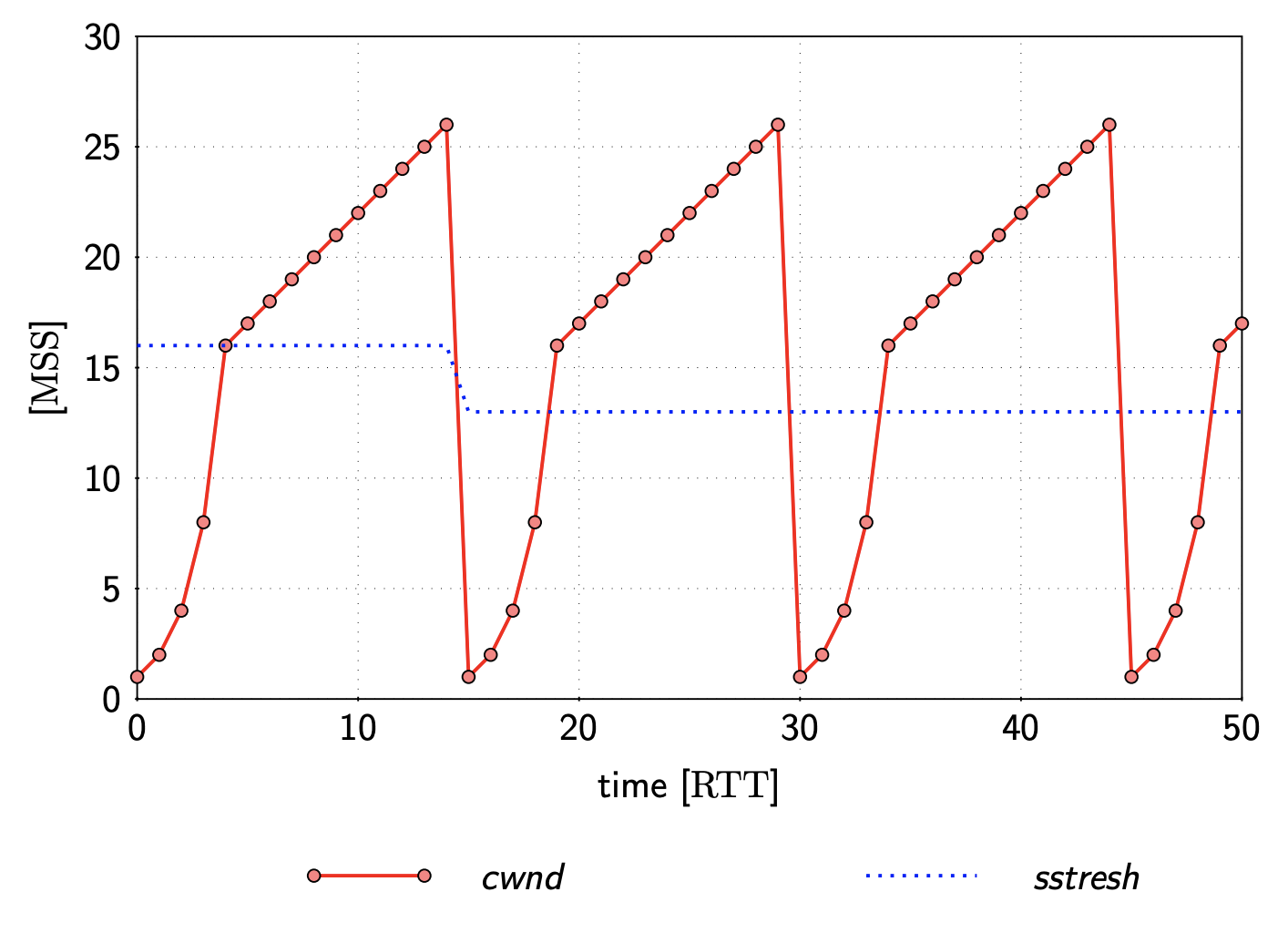
****

### Tahoe

* **On successful RTT**
  + cwnd++above *sstresh* (slow start threshold)
* **On loss**
  + sstresh = 0.5\*cwnd
  + cwnd = 1

### Reno

* Reno adds fast retransmission & fast recovery.
* TCP prijemca hned odosle ACK ked pride out-of-order segment
* 3 duplicate ACKs = loss
  + On receiving 3 ACKs, **TCP performs a fast retransmission** & ignores the retransmission timer
* **On loss**
  + fast recovery - slow start only when retransmission timeout
  + sstresh = 0.5\*cwnd
  + cwnd = 0.5\*cwnd



**Tahoe Reno**

### Vegas

* Iny koncept congestion control
* Monitoruje RTT pocas prenosu
* Ak je siet zahltena, **RTT sa zvysuje -> linearne znizuje cwnd**

## Improving traditional TCP

### Multi-stream TCP

* Je zalozeny na principe viacerých TCP streamov pre jeden data flow
* Lepsia performance ako TCP pri izolovanej strate paketov (tzn. iba z jedného streamu)
* **Drawbacks:**
  + Komplikovanejsie ako bezne TCP (viac threadov)
  + Linearne zrychlujuci startup

### Web100

* Software implementujuci nastroje (e.g. *zero copy, HW cooperation*) do Linuxoveho TCP/IP stacku
* Dve casti
  + **Kernel patch** - Pridava instrumenty do kerneloveho jadra
  + **Libraries** - Pridava nastroje na pristup ku novym instrumentom kerneloveho jadra
* Features:
  + Monitorovanie, statistiky
  + Tuning, auto-tuning, hw tuning

## Konzervativne rozšírenie TCP

### GridDT

* Ad-hoc modifikacie :(
* Rychlejsi slowstart (upraveny sstresh)
* **AIMD** na congestion control (**A**dditive **I**ncrease, **M**ultiplicative **D**ecrease)
  + **On success** - cwnd = cwnd + a
  + **On loss** - cwnd = b \* cwnd
  + a, b - zvolene konstanty
* Staci notifikovat odosielatela

### Scalable TCP

* **MIMD** na congestion control (**M**ultiplicative **I**ncrease, **M**ultiplicative **D**ecrease)
  + **On success** - cwnd = 1.01 \* cwnd
  + **On ACK** - cwnd = cwnd + 0.01
  + **On loss** - cwnd = 0.875 \* cwnd
* Pri malej velkosti okna, alebo vysokej stratovosti prepina na **AIMD**

### HSTCP

* **H**igh**-S**peed **TCP**
* **AIMD** aj **MIMD** na congestion control
  + **On success** - cwnd = cwnd + a(cwnd)
  + **On ACK** - cwnd +
  + **On loss** - cwnd = cwnd \* b(cwnd)
  + a(), b() - zvolene fcie
* Pri malej velkosti okna, alebo vysokej stratovosti emuluje tradicne TCP

### H-TCP

* Pomenovane po Hamilton Institute
* **AIMD** na congestion control
  + Pouziva funkcie zavisle od ∆ poslednej congestion
  + Zvysuje agresivitu cim vacsia je ∆

### BIC-TCP

* Default v Linux jadrach
* Pouziva binarne vyhladavanie
* 4 fazy:
  + Packet-loss reaction
  + Additive Increase
  + Binary Search
  + Maximum probing
* **AIMD** pocas 2 a 4 fazy
* Stabilnejsia velkost okna -> Lepsie vyuzitie siete
* Dost agresivne na pomalych sietach

### CUBIC-TCP

* Velmi podobne BIC-TCP
* Pouziva kubicku funkciu

## Rozsirenia TCP s podporou IP

### Quickstart

* Slow-start faza sa neda zefektivnit bez zmeny v nizsich vrstvach siete
* **Proposal**: Pridanie novej 4-b IP hlavicky s QS TTL a Initial Rate
  1. Odosielatel nadstavi hlavičku QS TTL na vysoku hodnotu a Initial Rate na pozadovany rate a posle SYN packet
  2. Kazdy router po ceste znizi TTL a (ak je to potrebne) znizi Initial Rate
  3. Prijemca posle spat SYN/ACK packet s upravenymi QS TTL a Initial Rate
  4. Odosielatel porovna povodne hodnoty, upravi, nadstavi podla nich cwnd a zacne pouzivat vhodny congestion control mechanizmus (e.g. AIMD)
* **Treba zmeny v IP Vrstve!**

### E-TCP

* Pouziva **E**arly **C**ongestion **N**otification (**ECN**)
  + ^ Komponenta **A**dvanced **Q**ueue **M**anagement (**AQM**)
  + **ECN** je 1-b flag, ktory nadstavuje router, ak prichadza congestion.
  + TCP musi reagovat rovnako, ako pri strate (typicky znizenie cwnd)
  + Nutno dokonfigurovat do routerov
* **E-TCP** zamrazi cwnd ak pride ACK s **ECN** flagom
* Pozaduje drobne synteticke straty aby udrzal **MD**.

### FAST

* **F**ast **A**QM **S**calable **T**CP
* Ak je v queue primalo packetov, zvysi sending rate
* Ak je v queue privela packetov, znizi sending rate
* Podobne TCP Vegas, s rozdielmi:
  + FAST ma variabilnu velkost uprav podla rozdielu medzi idealnym stavom a aktualnym stavom
  + dokaze pouzivat ECN ak je pritomna v sieti a nahradit nim indikatory zahltenia

## Hipsterciny ine ako TCP

### Tsunami

* **TCP** - **kontrolny prenos**
* **UDP** - **prenos dat**
  + **MIMD** na congestion control
  + Vysoko konfigurovatelne

### Reliable Blast UDP

* Ako Tsunami
* Urcene na prenos dat z disku na disk

### eXplicit Control Protocol

* Pouziva feedback z routerov per packet

### SCTP

* Multi-stream
* Spravy spolahlive UDP, congestion control TCP

### DCTP

* Spravy nespolahlive UDP, congestion control TCP

### STP

* Absentuje kvalitna congestion control

### Reliable UDP

* Spolahlive a usporiadane UDP dorucenie

## Zhrnutie

* Vysoka interakcia s L3 (IP Layer)
* Taktiez interakcia s L2
  + Variabilne oneskorenie a priepustnost
* Specificke nadstavenie na flow v routeroch
  + napr. v E-TCP synteticke generovanie packet loss
  + Moze byt napomocne pri velkych naporoch na siet
  + Drahe a tazko skalovatelne

# 

# Otazky

## ISO/OSI & Basics

**Co je to síťový protokol? Napíšte definíciu. Ako sú protokoly štandardizované? Pre jednotlivé vrstvy OSI modelu uveďte príklady protokolov.**

**Sietovy protokol** je sada pravidiel, ktoré definuju format a poradie zprav medzi dvoma komunikujucimi entitami.

Protokoly su standardizovane bud **de facto** - vytvorene výrobcami, alebo **de jure** - vytvorene doveryhodnymi organizaciami

L1 - IEEE 802

L2 - Aloha

L3 - IPv4

L4 - TCP / UDP

L5 - ASP

L6 - AFP

L7 - FTP / HTTP

**Srovnejte TCP/IP a ISO/OSI model, popiste funkce vrstev, protokoly a algoritmy, ktere se zde používají. Zduvodnete existenci obou modelu.**

ISO/OSI - 7 vrstiev

TCP/IP - 4 vrstvy

TCP/IP zjednotuje vrstvy 5-7 ISO/OSI modelu do jednej a vrstvy 1-2 do druhej.

ISO/OSI je konceptovy model, na ktorom sa dobre vysvetľuje, TCP/IP je prakticky používaný.

TCP/IP

* **1. Vrstva - Fyzicka** - ramce na frames, Multiplexing, bit-to-signal transformation, synchronizace (kodovani)
* **2. Vrstva - Datový spoj** - Pakety na ramce, MAC, error control, LAN, Distributed Spanning Tree Alg. (loop prevention), Backward Learning Alg. (find locations)
* **3. Vrstva - Síťová** - Datagramy na pakety, spaja LAN na WAN, unique adresy, routování, control messaging, IP protocol, RIP protocol, …
* **4. Vrstva - Transportni** - Data na datagramy/segmenty, flow & congestion control, QoS, porty ako adresy aplikacii, UDP & TCP, …
* **5. Vrstva - Aplikacni** - sluzby pre uzivatelov, konkretne aplikacie, HTTP, SMTP, FTP, Client-Server, P2P, interface medzi uzivatelom a sietou

**Popiste L2 vrstvu ISO OSI modelu, adresaci na této vrstvě, funkci vrstvy, protokoly a pripadne algoritmy, ktere se zde používají.**

Vrstva **Datoveho Spoja**. Adresacia pomocou MAC adries, ktora urcuje unikátnu adresu fyzickej entity. MAC sa stara o koordinovanie pristupu viacerych zariadeni ku jednomu prenosovemu mediu. Používá protokoly ako ALOHA, CSMA/CD,...

Obaluje pakety z L3 do ramcov, ktore posiela do L1.

Zarucuje flow control, ktorym zabranuje zahlteniu controlleru na L1 (např. stop-and-wait).

Zarucuje error detection, kde posiela bity ako checksum prenesenych dat. Prijemca si tak isto vypocita checksum a tak odhalí chybu. Následne môže požiadať o znovu-zaslanie / pokusit sa o opravu.

Stara sa o vytvaranie LAN sieti, na ktorych vyuziva Spanning Tree Alg (na prevenciu cyklov) a Backward Learning Alg (na najdenie poloh ostatnych nodeov).

**L3 vrstva, účel, funkcie, spôsob adresácie, protokoly, algoritmy protokolov.**

**Síťová** vrstva. Prijima segmenty z L4 a transformuje ich na pakety pre L2.

Spaja LAN na vacsi logický celok, ktorý navonok (vo WAN) pôsobí ako jedna entita.

Dokaze fragmentovat segmenty v pripade nizkeho MTU siete.

Adresovanie - Priraduje unique IP adresy, pakety obsahuju source a destination IP adresy. Používá ARP a RARP protokoly na přiřazení IP ku MAC.

Routovanie - Vyberá cestu po sieti, po ktorej poslat data do ciela. Staticke, dynamicke, centralizovane, distribuovane,... Pouziva DV (Bellman-Ford) alebo LS(Dijkstra) na routovanie. DV -> RIP, RIPv2, LS -> OSPF

Kontrolne spravy - Pomocou ICMP protokolu sa stara o zakladne informacie potrebne ako stav siete, hosta, …

Unicast - identifikacia jednoho konkretneho prijmaca

Broadcast - identifikacia celej LAN

Multicast - see next q

**Co je IP multicast, jaky L4 protokol vyuziva; vytvareni multicastovych skupin (+ protokoly).  
IP Multicast** je metoda posielana IP datagramov zvolenej (konkretnej) skupine prijemcov.

V IPv6 ma prefix ff00::/8

Posiela sa iba jedna kopia dat, **pouziva UDP** a best-effort delivery.

Pouziva Source-based tree (protokoly RIP, OSPF) a Core-based tree (protokol CBT) na routovanie.  
Pouziva **IGMP** a **DVMRP** na tvorenie multicast skupin. <-L3 protokoly

to UDP asi staci to je L4

## 

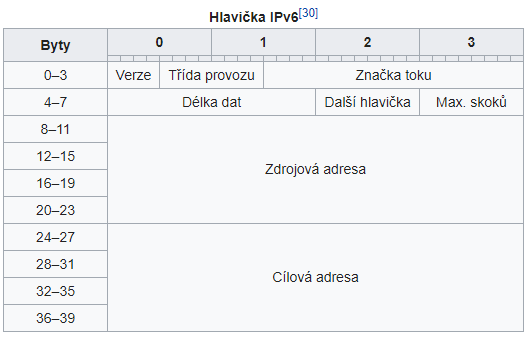
## 

## 

## 

## IPv6

**Hlavicky datagramu v IPv6; typy adres. Uveďte příklad IPv6 adresy a pravidla pro zkracování.**

* verze — 4 bity, verze 6
* třída provozu — 8 bitů na prioritu paketu. Úroveň priority se dělí na rozsahy: kde zdroj podporuje kontrolu přetížení a bez podpory kontroly přetížení.
* Značka toku — 20 bitů pro správu QoS. Původně určeno pro speciální obsluhu aplikací reálného času, nyní se nepoužívá.
* Délka dat — 16 bitů pro délku těla paketu. Při vynulování se nastaví „jumbo“ tělo (skok za skokem)
* Další hlavička — 8 bitů, určuje další vnořený protokol. Hodnoty se shodují s hodnotami definovanými pro IPv4.
* Zdrojová a cílová adresa — 128 bitů na každou adresu.

Typy:

**unicast** - jeden odosielatel/jeden prijimatel

**multicast** - jeden odosielatel/skupina prijimajucich prefix: ff00::/8

**anycast** - novinka (nie je v IPv4) jeden odosielatel/jeden najblizsi zo skupiny prijímateľov

**Príklad adresy**: (128 bitov - 16 Byteov - 32 hexadecimalnych cisel, rozdelenych do 8\*štvorica)  
  
2 typy skracovania:

1. vynechávanie 0 zľava v rámci 4-ice

000d: -> :d:

1. vynechavanie jedneho bloku opakujúcich sa 0

ffee:0:0:0:0:d:0:eeee -> ffee::d:0:eeee

Aplikovaním pravidiel:

2001:0db8:0000:0000:0000:1428:0000:57ab -> 2001:db8::1428:0:57ab

**Aké sú rozdiely medzi tvorbou hlavičiek v IPv4 a IPv6? Ako je v IPv6 riešená v hlavičke fragmentácia datagramu?**

Hlavička IPv6 je oproti IPv4 jednoduchšia, neobsahuje: checksum, options, fragmentation info. Má pevne stanovenú dĺžku (40 B), možnosť rozšírenia funkcionality cez tzv “extension headers” (Ako Higher Order Components). Pomocou toho je aj riešená fragmentácia datagramu.

**Neighbour discovery v IPv6.**

Časť ICMPv6, slúži na:

* autokonfiguráciu IPv6 adries
* určenie prefixu siete, smerovačov a ďalšie konfiguračne ~~pičoviny~~ informácie.
* detekcia DAD (Duplicate Address Detection)
* určenie L2 adries uzlov na tom istom spojení
* info ktorí susedia sú dosiahnutelní a ktorí nie (NUD)
* detekuje zmeny v L2 adresách

Pozostáva z 5 ICMP (Internet Control Message Protocol) správ:

1. Router Solicitation (RS)
2. Router Advertisement (RA)
3. Neighbor Solicitation (NS)
4. Neighbor Advertisement (NA)
5. ICMP Redirect

**Jakym zpusobem je v IPv6 resena mobilita? Popiste vcetne schématu.**

Funguje na predpoklade, že každé mobilne zariadenie je niekde doma.

Termíny:

Domáca Adresa - globálna trvalá adresa, cez ktorú je mobil stale prístupný.

Care-of Adresa - globálna adresa, na ktorej je mobil kým je v cudzej sieti.

Correspondent Node (CN) - uzol, ktorý sa chce dorozumievať s naším Mobilným uzlom.

Mobilný uzol (MN)

Domáci Agent (HA) - smerovač v domácej sieti, cez ktorý sa dá s MN vždy komunikovať.

optimalizácia cesty - je možné komunikáciu medzi CN a MN optimalizovať tak aby nekomunikovali cez HA.

RRP Return Routability Procedure - umožňuje túto optimalizáciu v 5 krokoch:

1. **Home Test Init (HoTI):** MN -> HA -> CN (obsah správy: Home Init Cookie (HIC))
2. **Care-of Test Init (CoTI):** MN -> CN (obsah správy: Care-of Init Cookie)
3. **Home Test (HoT):** CN -> HA -> MN (obsah správy: Home Nonce Index a HIC)
4. **Care-of Test (CoT):** - CN -> MN (obsah správy: Care-of Nonce Index a CIC)
5. MN a CN vypočítajú 20 B Management Key a môžu komunikovať napriamo.

**Bezpečnosť v IPv6, spôsob zabezpečenia, schéma.**

Security Associations (SA) - množina bezpečnostných informácii, ktoré opisujú bezpečné spojenie medzi 2 zariadeniami.

Obsahuje:

1. kľúč
2. šifrovací/autentikačný mechanizmus
3. prídavné parametre pre algoritmus

Je to jednosmerná dohoda, takže ak chceme spraviť obojstrannú komunikáciu, poskytujúcu autentizáciu a šifrovanie potrebujeme 4 SAs.

Protokol pre Autentizáciu - AH (Authentication Header)

Ah sa sklada z hlavicky

vytvorí sa Hlavička ktorá má v sebe spočítaný Integrity Check Value z datagramu.

Protokol pre zašifrovanie (a autentizáciu) - ESP (Encapsulating Security Payload)

ESP sa sklada z hlavicky - pred datami, trailer - po datach, ESP authentication Data - po trailery (ked je pouzita aj autentizácia tak obsahuje ICV)

Naco pouzit AH ked ESP toho vie viac, niekedy stačí autentizácia, AH je rychlejšie + autentizuje celý datagram, ESP neautentizuje vonkajšie IP Headers

SA je zadefinovaná 3 parametrami:

1. Security Parameter Index (SPI) - 32-bitové číslo unikátne identifikujúce SA
2. IP Adresa destinácie
3. Security Protocol Identifier - špecifikuje či je to AH alebo ESP

IPsec pracuje v dvoch modoch:

1. Transportný - AH/ESP hlavicka sa vlozi za povodnu IP hlavicku
2. Tunelovací - cely datagram sa v podobe dát obalí do novej hlavicky

**Ako je riešená koexistencia IPv4 a IPv6? Vypíšte tri základné spôsoby jej riešenia a podrobne ich popíšte.**

1. Duálne používanie - zariadenie podporuje aj IPv4 aj IPv6,

+ jednoduché na použitie a flexibilné

- dva rozdielne protokoly zaberaju miesto, vsetky aplikacie musia vediet rozoznať

1. Tunelovanie - IPv6 datagram je zabalený do IPv4

+ jednoduche migrovanie na IPv6

- pridana praca pre smerovace (zabalovanie/odbalovanie)

1. Prekladače (NAT-PT) - zariadenie preklada medzi IPv6 <-> IPv4

+funguje to

- najhorsia moznosť (stracaju sa dodatočné informácie, veľa roboty pre NATs)

## Routing

**Co jsou to Distance Vector smerovaci protokoly? Jak pracuji? Jmenujte 3 zastupce, jednoho z nich popiste detailne.**

Směrovací protokoly sloužící k stanovení cesty a předávání paketu.

Bellman-Ford algo

sousedící směrovače si pravidelně(nebo když se změní topologie sítě) vyměňují a kopírují celou směrovací tabulku

“informace o celé síti poskytuji pouze sousedům”

vhodný iba pre menšie siete (max. vzdialenosť 15)

nezvládá cykly, count-to-infinity problem

zástupci: RIP, IGRP, EIGRP

RIP - metrika hopy, směrovače

počet hopu slouží jako metrika, směrovače posílají informace každých 30s, timeout 180s,

Vhodný pro menší sítě a stabilní spojení

Nevhodný pro redundantní sítě

Verze

RIPv1 - dnes nepoužívaná

zprávy jsou broadcastované

RIPv2

maska podsítě

multicast (broadcast už ne)

16 hopů = nekonečno

RIPng - podpora IPv6 adres a sítí (IPv6 native)

**Funkcie smerovača, schéma, účel**

Účel**:**  Najít cestu v okolí. Posílat data správným směrem.

Funkce

* Přišel paket
* Přečte IP Hlavičku
* Snížit TTL a vyhodnocení TTL
* Checksum rekalkulace
* Vyhodnocení MTU (jestli paket není moc velký)
  + Fragmentace - IPv4
  + ICMP zpráva - “packet too big” -IPv6
* Klasifikace paketů

Schéma:

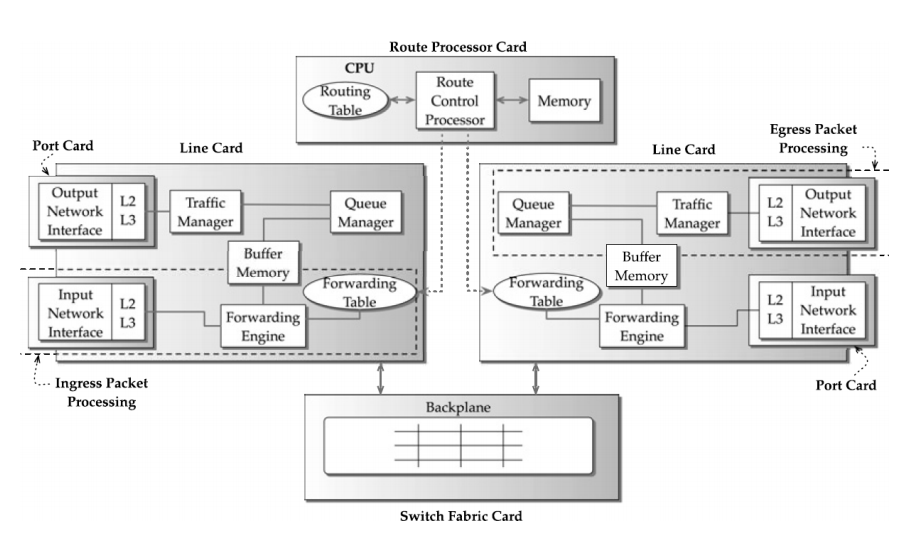
* Network Interfaces
* Forwarding Engines
* Queue Manager
* Traffic Manager
* Backplane
* Route Control Processor

**vplyv CIDR na smerovanie**

Je více komplikovaný protože destinační IP adresa nenese s sebou masku a protože prefixy v přesměrovávací tabulce musí být shodné.

Rychlost - pomalejší

// píchat to tam nebude

**Popíšte fyzické komponenty směrovačů a nakreslite ich do náčrtu.  
**

**RIP vs OSPF - shody, rozdíly.**

RIP funguje na bázi Distance Vector směrovacích protokolů využívající Bellman Forduv vyhledávací algo, OSPF je na Link State směrovacích protokolů fungujících na bazi Dijstruv algo.

Shody:

Oba protokoly patří do skupiny distribuovaných směrování.

Rozdíly:

RIP - zvládá cykly, count-to-infinity problem, vhodný pro menší sítě, “informace po celé síti posílám sousedům”

OSPF - větší sítě, “informace o mých sousedech všem”

**Uveďte, čo je Traffic Engineering, ako a kedy sa používa. Aké metódy TE poznáte?**TE se jedná o hledání alternativních cest pro přenos.

TE vzniklo za myšlenky, kdy by mohla potenciálně jiná cesta, než ta nejkratší být v danou situaci i rychlejší. Obzvlášť aby nedocházelo k zahlcování jednoho uzlu.

Jak: traffic je změřeň, a na jeho základě je vytvořena traffic matice > pomocí této matice se ohodnotí hrany

30% zaplnění je ok jinak by se mělo začít řešit

Metody: Single commodity a Multi commodity

**Protokoly IGRP a EIGRP.**IGRP

* Interior Gateway Routing Protocol
* podobné jako RIPv1
* multicast
* DV aktualizace zahrnují 5 různých metrik pro každou cestu
* povoluje multiple cesty -> load balancing
* pracují přímo s IP
* má checksum
* Výpočet ceny spojů mezi uzly
  + metrika = koeficienty(K1, ..., K5) \* bandwidth (B) \* delay (D) \* reliability (R) \* load (L)

### EIGRP

* Enhanced Interior Gateway Routing Protocol
* IGRP ale “lepší”
  + směrování bez cyklů
  + konvergence
  + rychlé
  + viac ukecané

**Co znamená MPLS (Multiprotocol Label Switching) a jak pracuje; pro site jakého rozsahu je vhodný? Co je to GMPLS?**

Směrovací mechanismus pomocí labelů. Tento label se přidává na začátek peketu. Směrovače potom se pomocí labelu rozhodují, co s paketem udělají. -> přeposlat, otevřít zprávu

Tento mechanismus není náhradou ARP.

Téměř kruciální pro velké sítě.

paket může mít více jak 1 MPLS

Směrovače

Core Label-switched - koncové směrovače

Ingress - přidává label na začátek hlavičky, zhodnocuje cestu na základě hlavičky

Engress - odstraňuje label a přeposílá originální paket cíli

Edge Label-Switched - směrovače “uprostřed” - přepisují label a posílají paket dalšímu směrovači

GMPLS - Generalized MPLS

* MPLS mechanismus pro přesměrovávání i jiných objektů než je paket.

**Co jsou to autonomni systemy a jakym zpusobem se zde řeší směrování? Proc je vyhodne AS zavadet?**

Menší sítě(domény), o které se stará jedna organizace. Zavádí se, aby nenastal routing overhead. Každá doména má svůj identifikátor (**ASN** - 16bit).

Směrování se rozděluje do dvou skupin,

interní(IGP), které se stará o směrování uvnitř AS - **RIP, OSPF, IS-IS**

hlavním cílem efektivita

externí(EGP), které řeší směrování mezi více AS - **BGP-4**

hlavním cílem škálovatelnost a podpora stanovených pravidel

**Jake jsou zakladni typy QoS based smerovacich algoritmu? Alespoň jeden popište podrobněji.**

**Source-based, Hop-by-hop, Hierarchický**

**Source-based**

* každý směrovač má globální stavovou informaci o síti a na jejich základě je vybírána cesta
* jak je cesta nadefinovaná, směrovač informuje všechny ostatní
* nevýhody: škálovatelnost špatná, každý směrovač musí znát dost informací -> hůř se to spravuje

**Klasifikacia paketov a ich filtrovanie.**

**DIFFSERV** v IPv6 hlavicke

// pičovina

## TCP

**TCP, opíšte varianty TCP protokolu pracujúce na L2 vrstve**.

Protokol transportní(L4) vrstvy

aplikace mohou vytvořit mezi sebou spojení, přes které mohou obousměrně přenášet data. Protokol garantuje spolehlivé doručování a doručování ve správném pořadí.

**opíšte varianty TCP protokolu pracujúce na L2 vrstve**. <- idk to je nějaká blbost

**Rizeni toku a zahlceni v TCP.**

**Flow Control** - Deterministicky feedback od príjemcov (*rwnd*)

**Congestion Control** - Odhad odosielateľa o volnej priepustnosti siete (*cwnd*)

tradiční TCP - Tahoe, Reno

konzervativnější rozšíření - GridDT, STCP(Scalable TCP)

rozšíření s podporou IP - Quickstart, FAST

**Jak se pocita velikost vysilaciho okna v tradicnich TCP** **protokolech?** **Jak to ovlivnuje mnozstvi prenesenych dat na vysokorychlostnich linkach?**

cwnd - congestion control - odhad odesílatele ohledne propustnosti cíle

sstresh - slow start streshold -

Tahoe

* za ok RTT -> cwnd++
* loss -> sstresh = cwnd / 2 a nataví cwnd na 1

Reno

* loss -> sstresh = cwnd / 2 a potom cwnd = sstresh

Vegas

* při zahlcení sítě -> RTT se zvýší -> linearně se snižuje cwnd

Směrovač potom ví, zda dané množství paketů může projít přes spoj a jestli cílový uzel dokáže takové množství zpracovat.

**Funkce TCP, faze prenosu, rozepsat Tahoe/Reno/Vegas.**

**Aggressiveness** - Schopnost vyuzit cely bandwidth

**Responsiveness** - Schopnost zotavit sa pri strate paketov

**Fairness** - Ferova priepustnosť siete pri vyuzivani viacerymi uzivatelmi

**Popiste protokol RBUDP (Reliable Blast UDP). Do jake skupiny protokolu patri? Kdy ho lze s vyhodou pouzit?**  
podobné jako Tsunami

* **TCP** - **kontrolny prenos**
* **UDP** - **prenos dat**
  + **MIMD** na congestion control
  + Vysoko konfigurovatelne

určené pro přenos dat z disku na disk

## P2P

**Porovnanie P2P s Client-Server.**

Client-server ma 2 moduly:

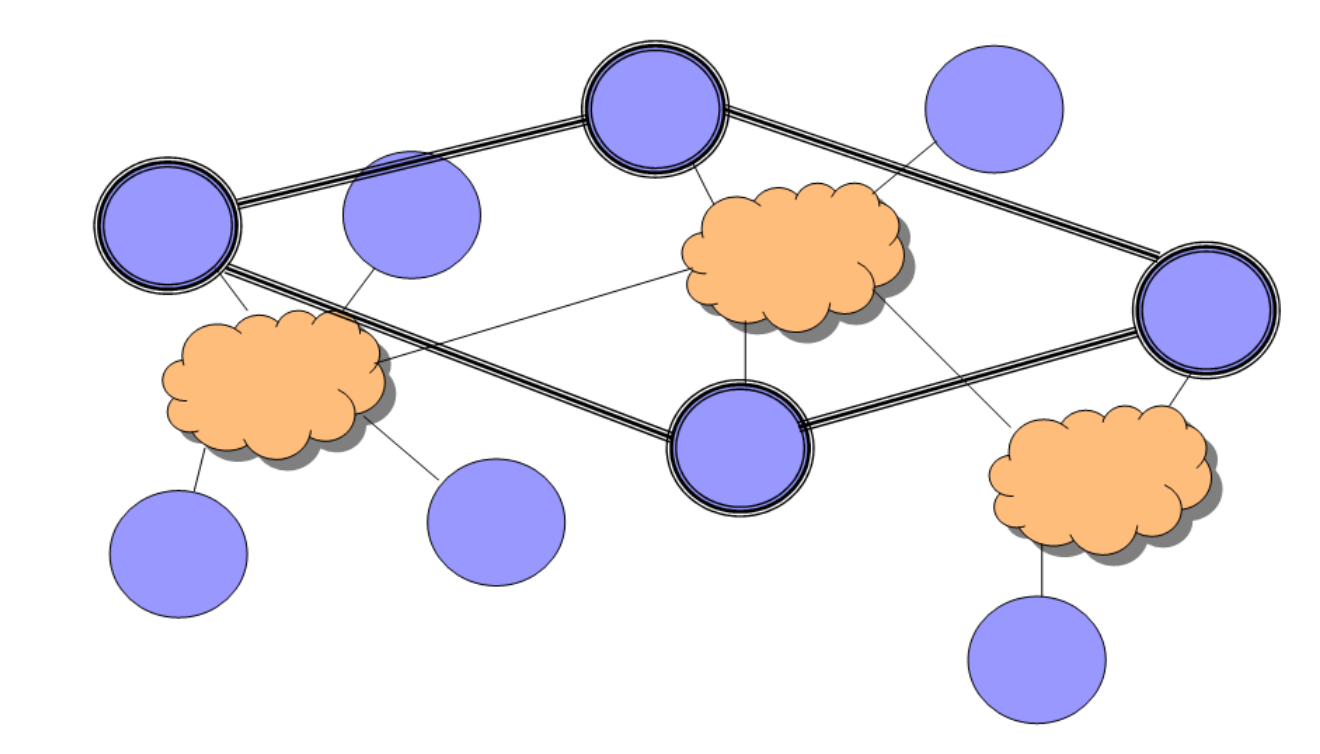
1. Server:
   1. Jedna centralizovaná instance (vnitrne replikovana pro lepsi škálovatelnost)
   2. Poslouchá jestli se připojit klient
   3. Dokaze zpracovavat vice pozadavku od klientu (sekvencne, zaroven, ...)
   4. Vyhrazeny (virtualizovaný) hardware, ktery je spolehlivý
   5. Single point of failure
2. Klient:
   1. Vice distribuovanych instanci
   2. Aktivne iniciuje připojení k serveru
   3. Klienti spolu primo nekomunikuji (jen skrze server)
   4. Vypadek klienta neohrozi system

P2P:

1. Mnoho identickych softwarovych modulu (kazdy na jinem pocitaci)
2. Peerove spolu komunikují primo
3. Kazdy peer je zaroven server a zaroven klient
4. Peerove byvaji nespolehlivi
5. P2P síť prirozene skaluje
6. Pracuje bez vyhrazeného serveru\
7. Poznamka: mohou existovat hybridi, napr. P2P sit s centralizovanym adresarem.

**Co jsou overlay networks a kde se nacházejí?**

Overlay network je virtuální síť nad sítí “fyzickou” (napr. TCP/IP). Spoje v overlay siti jsou implementovany za použití zakladni site (napr. TCP spojeni, UDP zpravy). Peerove, kteri jsou od sebe daleko ve fyzicke siti, mohou byt v overlay siti sousede -> vykon zavisi na strukture overlay site.



**Taxonomie P2P systémů, stručně charakterizovat každou kategorii.**

1. Centralizovane - jeden nebo vice serveru poskytujicich ruzne sluzby - typicky pomahaji peerum najit pozadovane zdroje.
2. Decentralizovane - zadne servery - peerove maji stejna prava a zodpovednost, imunní proti single point of failure.
   1. Struktura
      1. Plocha (single-tier) - funkcionalita a zatez rozdeleny na vsechny uzly
      2. Hierarchická (multi-tier) - vice urovni, napr. mezinarodni uroven, statni uroven, univerzitni uroven, ...
   2. Overlay topologie
      1. Strukturovana - umístění dat ma na starosti předdefinovaná strategie (DHT -> distributed hash table)
         1. **Rooting table**
         2. **zaruka nalezu itemu ak je v siti**
         3. **tazsie vytvorit**
      2. Nestrukturovana - kazdy peer zodpovedny za vlastni data, udržuje přehled o svých sousedech
         1. **Flooding table**
         2. **neni zaruka nalezu itemu v siti**
         3. **lahke vytvoit**
3. Hybridni - bere si vyhody a nevyhody z obou, napr. misto jednoho serveru mame super peery, což jsou výkonnější uzly.

**Popiste heuristicke smerovaci strategie používané v P2P sítích a alespoň jednu z nich podrobneji. Zakladni problem heuristik?**

Iterative Deepening - BFS po urovnich. System poskytne politiku P, ktera udava sekvenci hloubek, ve kterych se má hledat. Zdrojový uzel odešle dotaz do site skrze BFS hloubky D1. Pokud je zde vysledek -> konec dotazu. Jinak se prejde do hloubky D2, atd...

Directed BFS and Intelligent Search - V Directed BFS neposila dany uzel dotaz vsem svym sousedum, ale pouze podmnozine sousedu. Trik je v tom, jak zvolit tyto “dobre” sousedy. Muze byt zalozeno na ruznych heuristikach:

1. Uzly, které minule vrátily nejvetsi pocet vysledku
2. Uzly, které minule forwardovaly nejvetsi pocet zprav
3. ...

Local Indices Search - Kazdy uzel si spravuje indexy pro lokalni data a data sousedu, kteri jsou ve vzdálenosti *k* hopu (*k* = 0 znamená BFS). Existuje globální politika *P*, která specifikuje seznam hloubek ve vyhledavacim strome, kde bude dotaz zpracovavan.

Random Walk - Nahodny vyber souseda (nebo podmnozina sousedu) pro zpracování dotazu

Adaptive Probabilistic Search - Kombinuje k-walker random search a probabilistic search. Dva pristupy pro update pravdepodobnost:

1. Optimisticky - zvysit vsem uzlum na ceste. Jen pokud dotaz skonci neuspechem, tak snizim vsem uzlum na dane ceste.
2. Pesimisticky - snizim vsem uzlum na ceste. Jen pokud dotaz skonci uspechem, tak zvysit vsem uzlum na dane ceste.

Interest-Based Shortcuts - kazdy peer si udržuje zkratky (omezeny pocet). Po kazdom uspesnem dotaz vytvoří dotazujici se zkratku k uzlu, který obsahoval odpověď pro daný dotaz.

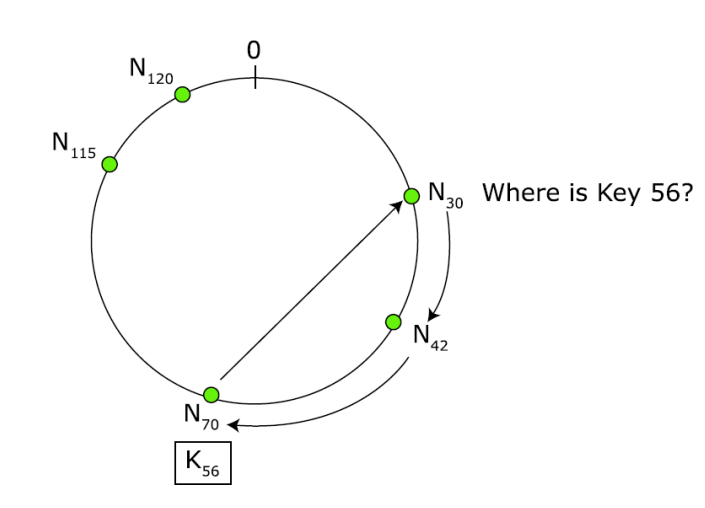
Heuristiky nezarucuji optimalni vyhodnocovani.

**Smerovanie CHORD v P2P, popísať ako a kde sa používa. Ako v nej prebieha smerovanie?**Zalozeno na DHT - distribuovane hashovaci tabulce.

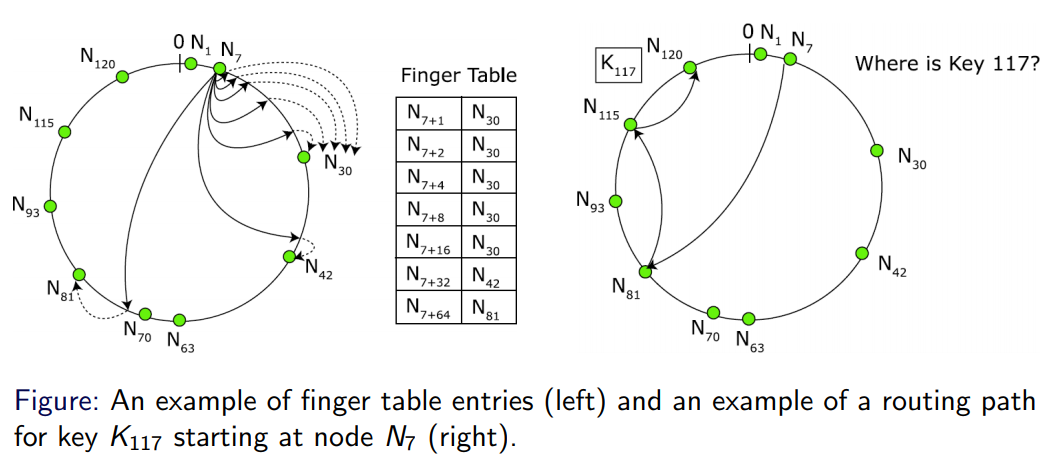
Simple lookup algorithm: Mejme rozsah klicu 64 a mejme uzly (peery) N12, N25 a N50.

Potom:

1. uzel N12 zodpovida za klice K51 az K12.
2. uzel N25 zodpovida za klice K13 az K25.
3. uzel N50 zodpovida za klice K26 az K50.



Scalable lookup algorithm: Obsahuje navic tzv. finger table, viz obrazek.



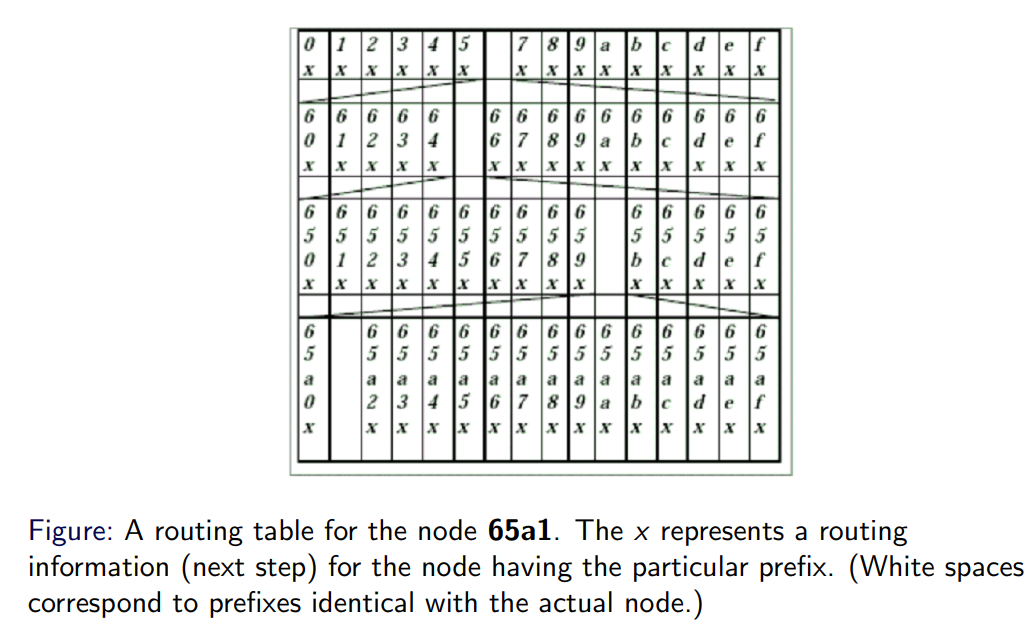
**Popiste P2P sítě využívající PRR stromy a vyhledávání v nich.**

Identifikátor uzlu je m-bitové cislo, ktere je rozdeleno do sekvenci cislic se základem 2b.

Příklad: 128-bitový identifikátor rozdeleny do 32 4-bitovych cislic. Tj. 24 = 16, sekvence hex. cislic.

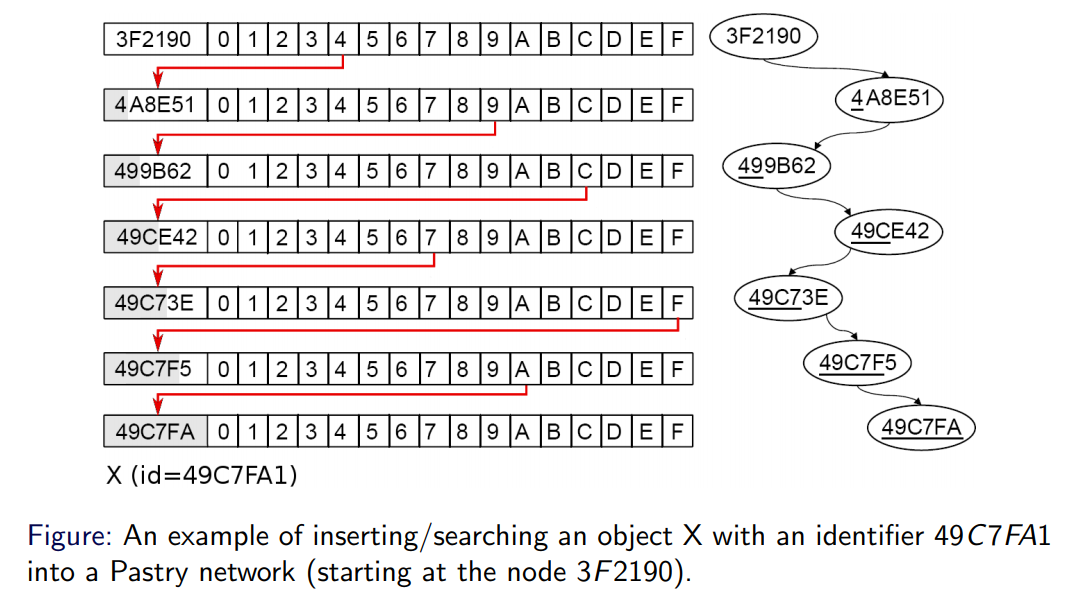
Data jsou v uzlu, který s identifikátorem sdílí nejdelsi prefix. V kazdem kroku je vybran sousedni uzel, ktery ma nejdelsi spolecny prefix s hledaným uzlem.

Priklady: Pastry (prefix), Tapestry (suffix)



**Popisat Pastry. Rozdiel medzi Tapestry a Pastry u P2P sieti.**

Pastry - viz predchozi otazku.



Rozdíl mezi Tapestry s Pastry je ten, ze Pastry pouziva nejvetsi spolecny prefix a Tapestry nejdelsi spolecny suffix.

**Popiste P2P systemy BATON a P-Grid. Jake grafové struktury vyuzivaji a cim se lisi?**

P-Grid - virtualni binární strom, ve kterem se kazdy peer stara o list stromu. System udeli kazdemu peerovi identifikátor - binarni retezec reprezentujici cestu z korene do listu (vice peeru muze mit stejny identifikátor).

Kazdy peer je pote odpovedny za data, jejichz prefix je roven identifikatoru peera. Kazdy peer si udržuje směrovací tabulku.

Strom muze byt nevyvazeny.

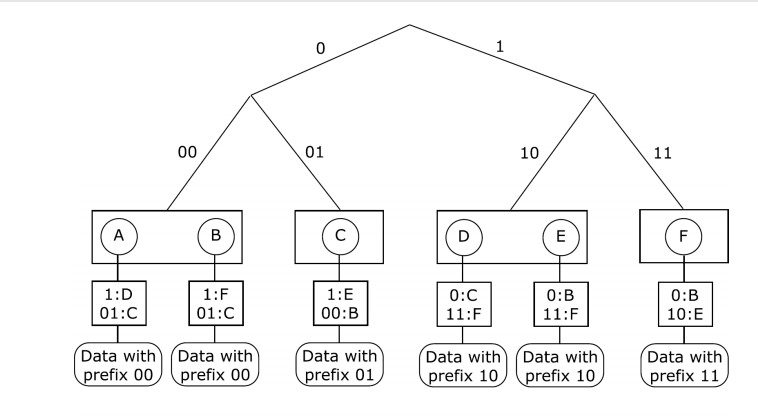
Priklad k obrazku: Jsem uzel B a hledam data s prefixem 11.

Podivam se k sobe, ale bohuzel mam jen data s prefixem 00.

Nahlednu do smerovaci tabulky a vidim, ze uzel F ma data s prefixem 1 a uzel C data s prefixem 01.

Vyberu tedy uzel F, protoze ma nejdelsi shodny prefix s tim, ktery hledam.

Uzel F se se podiva, jaky je prefix jeho dat. ANO jsou to data s prefixem 11, vyborne.

****

BATON - stromová struktura, data uchovaná ve vnitřních uzlech i v listech. Krome linku mezi rodičem a potomkem obsahuje dalsi linky - adjacent links a neighbor links.

Adjacent link spojuje uzel s uzlem, ktery se stara o prilehle rozsahy hodnot.

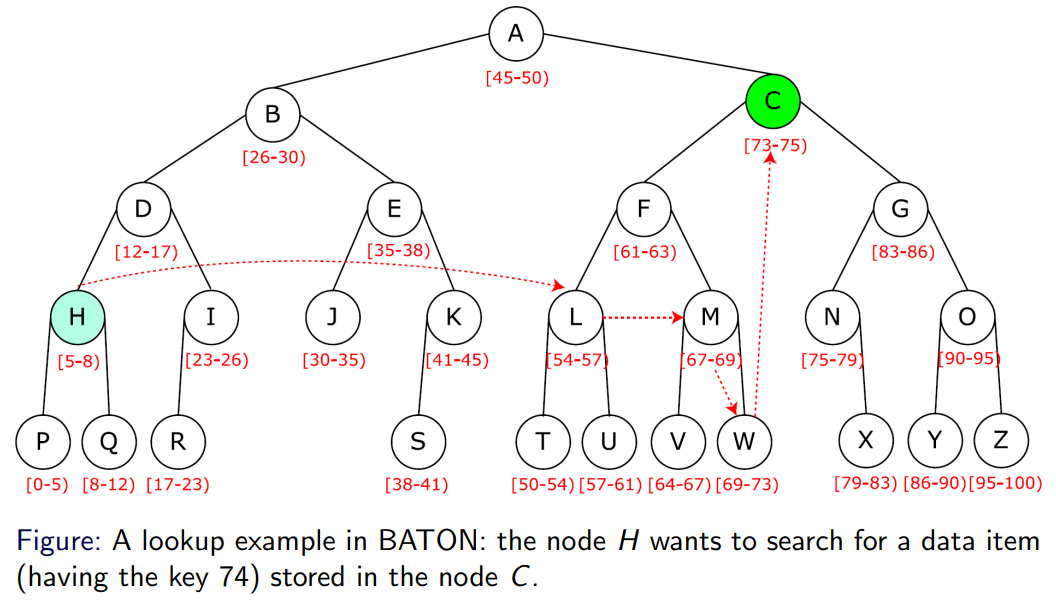
Neighbor link spojuje uzel se sousednimi uzly na stejne urovni, kteri maji vzdalenost 2i, pro kazde i >= 0 (tj. vzdalenost 1, 2, 4, 8, 16, ...).

Postup hledani: Peer x dostane dotaz.

Kdyz hledany klic spadá do rozmezí hodnot, o ktere se stara x, odpovi na dotaz.

Jinak přepošle dotaz na souseda (skrze Neighbor link), ktery je co nejdál, ale zaroven neprestrelni hledany klic.

Pokud takovy soused neexistuje, x přepošle dotaz potomkovi (pokud existuje) nebo uzlu s naslednym rozsahem hodnot (skrze adjacent link).



**Vymenuj smerovacie protokoly používané v neštruktúrovaných P2P sieťach, opíš ako pracuje Interests based shortcuts**

Gnutella, FreeNet

IBS - Kazdy peer prida dalsie linky do vyhľadávacej siete, ktore priamo odkazuju na peerov s podobnymi “zaujmami” (pre neho najcastejsie relevantnymi dátami).

## AD-HOC

**Čo je to VANET a aké sú rozdiely oproti mobilným ad-hoc sieťam?**

**V**ehicular **A**d-Hoc **Net**work, pouziva pohybujuce sa auta ako nody na vytvorenie mobilnej ad-hoc siete.

Oproti MANET je najvacsi rozdiel ze sa nody nepohybuju nahodne, ale organizovaně (typicky po silnici a rychlost je take omezena). Taktiez mozu casto komunikovat s vysielacmi popri ceste. Kvôli tomuto je mozne pouzit specializovanejsie routovacie protokoly

**MANET, čo to je, porovnanie s P2P**

**M**obile **A**d-Hoc **Net**work, je to mobilna, bezdratova ad-hoc siet.

V MANET sa nody dokážu pohybovat v čase a priestore, co moze vyrazne ovplyvnovat topologiu.

Maju vela **podobnosti** **s P2P**: samo-organizujúca siet, dynamicka topologia, rovnaka (p2p) paradigma, absentujuca centralna jednotka

Maju ale aj **odlisnosti**:

* MANET je fyzicka infrastruktura, ktorej logicka korešponduje s fyzickou, P2P je logicka infrastruktura, ktorej fyzicka je odlisna od logickej
* MANET je wireless a s nepriamym spojenim medzi nodami, P2P je typicky wired a s priamym spojenim medzi nodami
* MANET ma peerov v obmedzenom priestore, P2P po celom svete
* MANET dokaze routovat reaktivne aj proaktivne, P2P typicky reaktivne
* MANET broadcastuje fyzicky do vsetkych nodov v okoli, P2P broadcastuje “logicky”; tzn. vysiela viacero unicastov na konkretne nody

**Porovnanie reaktivneho a proaktivneho ad-hoc smerovania, priklady protokolov.**

Reactive:

* Najde routu, az ked ju potrebuje
* Nizsi overhead na routovanie, vyssia latencia
* Protocol napr. **D**ynamic **S**ource **R**outing (**DSR**), alebo **A**d Hoc **o**n Demand **D**istance **V**ector (**AODV**)

Proactive:

* Najde routu skor, ako ju potrebuje
* Vyssi overhead na routovanie, nizsia latencia
* Protocol napr. **D**estination **S**equence **D**istance **V**ector (**DSDV**), alebo **O**ptimized **L**ink **S**tate **R**outing (**OLSR**)

**Jak pracuje reaktivní ad-hoc směrování (Reactive On demand Ad-hoc Routing)? Uvedte priklady konkretnich protokolu se stručným popisem.**

Vysvetlene vyssie.

**D**ynamic **S**ource **R**outing (**DSR**):

* Pouziva RREQ a RREP pakety aby objavil cestu do ciela
* Najprv floodne siet RREQ paketmi
* Nody forwarduju RREQ po tom, ako pripoja svoje ID
* Cielovy node dostane RREQ a unicastne RREP spat s cestou z prijateho RREQ
* Data poslane az ked je overena cesta do ciela

**A**d Hoc **o**n Demand **D**istance **V**ector (**AODV**):

* Podobny proces ako DSR
* Rozdiel je v tom, si nody pamataju, odkial packety prichadzaju vo svojej routing table.
* Ak teda uz nejaky node po ceste vie cestu do ciela, posle spat RREP s touto cestou.

**Kde se používá a jak pracuje DSR protokol.**

Vysvetlene vyssie.

**Pristup k mediu pro WSN a Ad-hoc sítě. Identifikujte hlavní kritérium pouzitelnosti pro tyto protokoly; uveďte jejich klasifikaci a pro kazdou skupinu zastupce.**

**W**ireless **S**ensor **N**etworks.

Hlavne kriterium je **plytvanie energiou.** Casto sposobene koliziami, prepocutim a necinnym pocuvanim.

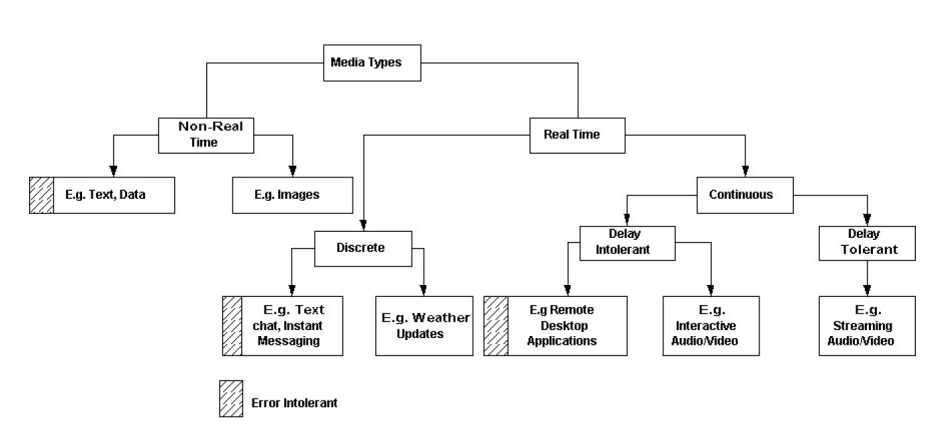
**Klasifikacia:**

* Contention-based protocols - e.g. **BTMA**, **MACA**
* Contention-based protocols with reservation mechanisms - e.g. **MACA/PR**
* Contention-based protocols with scheduling mechanisms, e.g. **LEACH**, **SMACS**

## Media

**Čo sú to multimediálne data a aké typy týchto dat rozlišujeme? Uveďte protokoly, ktoré multimediálne data spracovávajú a v skratke ich popíšte.**

* Multimediální data jsou informace složené z více typů forem (text, obraz, zvuk)
* Data jsou datová reprezentace multimedii
* Rozdělení



* + Real-time vs non real-time
  + discrete media vs continuous media
  + dealy tolerant vs delay intolerant
* SMTP(email), HTTP(web), FTP(soubory), RTP (Real-Time Transport Protocol), RTSP (Real-Time Streaming Transport Protocol)

**Čo sú multimédiá, hlavné delenie na dve skupiny, komunikačné protokoly pre ich prenos s ohľadom na Realtime prenos.**

definice nahoře

**Ako prebieha digitalizácia zvuku, 3 formáty zvukového záznamu.**

pomocí

* sampling - vzorkování - rozkouskování audia na menší části
* quantization - kvantování - mapping signálu do číselných hodnot

wav, mp3, webm

**Popiste prenosove a funkční požadavky multimedialnich sitovych aplikaci + protokoly.**

přenosové - delay, jitter, bandwidth, reliability

funkční - bezpečnost, mobilita, multicasting(IGMP), správa relací(SAP, SDP, RTP, RTCP, RTSP)

**Popsat základní funkce RTP/RTCP/RTSP protokolů. Proč byly zavedeny?**

Kvůli zavedení správy relací(session management).

RTP - Real-Time Transport Protocol - pakety mají sekvenční číslo, to pomáhá s detekcí ztrát. Frame indication. Source indication. Payload identification. Intramedie synchronization.

RTCP - Real-Time Control Protocol

* získávání statistik během přenosu dat v relacích
* přenos dat relačnímu zdroji a jiným účastníkům relace

RTSP - Real-Time Streaming Protocol

* povoluje media přehrávači spravovat přenos během streamu
* NEDĚLÁ : streamování dat jako takových (na to pouzit RTP)

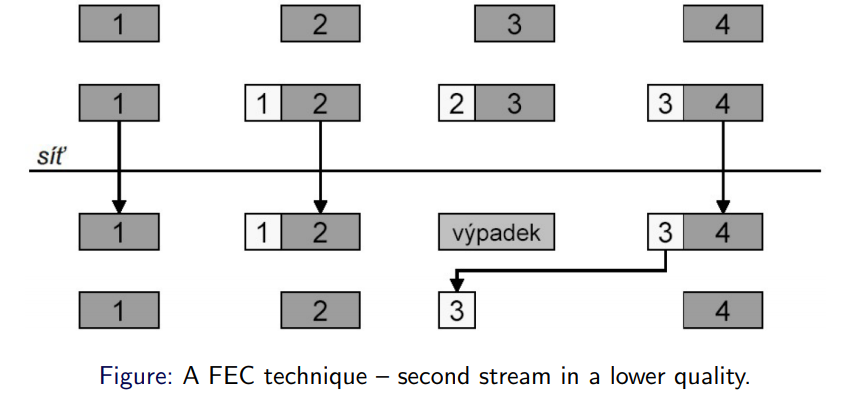
**Jak odstranit nebo alespoň omezit chyby pri prenosu videa. Co to je a jakou ma stim souvislost Interleaving?**

Forward error correction - druhy stream v nizsi kvalite (pocet paketu ale zustava stejny, jen jsou vetsi, to ale typicky nevadi - nejvetsi problem je mnozstvi, ne velikost paketu). Pokud se nejaky paket ztrati, je na strane prijemce nahrazen paketem z druheho streamu v nizsi kvalite.

Interleaving - mozne pouze pokud velikost jednotky media je mensi nez velikost paketu. Zalozeno na “rozsekani” jednotek media pred odeslanim a naslednem slozeni na druhe strane. Efekt: pokud doslo k vypadku paketu, pak to tolik nevadi, protoze ztrata je ve vysledku rozdelena na vetsi cast. Nevhodne pro interaktivni aplikace. Vyhodou je, ze se nezvysi vyuziti bandwidth.

**Co to je Backward Error Correction a Forward Error Correction. Vcem se lisi, co z toho pouziva TCP. Uvedte priklad protokolu, ktery pouziva to druhe.**

Oba řeší problém, kdy nastává error(např. výpadek).



(FEC - pasivní oprava - na základě odesílatele -

BEC neviem - ten tam ani neni nikde zmineny

TCP by poslalo rovnou celý paket znovu ze zdroje aniž by čekl zda náhodou není někde blíž)

**Jake druhy potvrzovani paketu znate a jake pouziva TCP.**

# Otázky - Vyjebaný bonus (30.1.2018)

1 dátová vrstva, ako tam funguje adresovanie, aké sú topológie, čo ešte sa tam deje, aké grafové algoritmy sa tam využívajú?  
2 ako funguje mobilita v ipv6 + načrtni schému  
3 security v ipv4 vs ipv6 (tuším že za najmenej bodov, asi 6 alebo 7)  
4 autonómne systémy, niečo o nich, aký majú identifikátor  
5 ip packet filtering a classification, na čo slúži, aké používa algoritmy + opísať ich  
6 MPLS, načo to je, ako funguje a čo je GMPLS  
7 aké (tuším) systémy využívajú viac kanálov naraz, opíš aspoň jeden z nich (asi tie approaches different from tcp)  
8 taxonómia P2P systémov, ku všetkému uveď minimálne výhody a nevýhody  
9 systém založený na PRR stromoch, opíš   
10 prístup k médiu vo wsn, aké je delenie, porovnaj BTMA a PCM  
11 proaktívny vs reaktívny ad-hoc routing, aký je v nich rozdiel, uveď príklady protokolov a stručne ich opíš  
12 traffic requirement delay, opíš všetky druhy/vplyvy a urči ktoré môžeme ovplyvniť

# Otázky - Vyjebaný bonus 2 (12.12.2017)

1) Popsat L2/L3 datové struktury, adresaci a rozsah sítě. Proč potřebujeme L4?  
  
2) Popsat hlavičky IPv4 a IPv6 a jejich rozdíly.  
  
3) Co je to QoS, jak je implementováno v IPv6; (vyloženě chtěla vědět, že Traffic class využívá v IPv6 pole PRI).  
  
4) Co je to stavová a bezstavová autokonfigurace; co z toho je používáno u IPv4.  
  
5) Popsat jak funguje Path Vector, trochu popsat jeho nejpoužívanější protokol.  
  
6) Co je to Traffic Engineering, jaké metody používá.  
  
7) Rozdíl mezi High Speed TCP a Scalable TCP.  
  
8) Popsat, jak funguje MPLS a jakým způsobem se distribuují značky (labels).  
  
9) Popsat systém BATON a jak se v něm vyhledává.  
  
10) Rozdíl mezi Destination Sequence Distance Vector a Optimised Link State Routing, kde se používají a co je nezbytné pro jejich nasazení.  
  
11) Srovnat autokonfiguraci v P2P a Ad-hoc systémech.  
  
12) Jaké požadavky mají multimediální aplikace, zaměřit se na zpoždění. Vyjmenovat všechny případy, kdy zpoždění vzniká.