# Multicast

## Adresování na L3

**IPv4:** vrchní 4 bity = 1110
→ třída D → 224.0.0.0 — 239.255.255.255

Lokální: 224.0.0.0 — 224.0.0.255 (256)

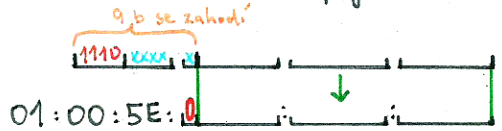
Globální: 224.0.1.0 — 238.255.255.255

Administrat.: 239.0.0.0 — 239.255.255.255 ($2^{24}$)

**IPv6:** prefix FFXX, kde XX určuje dosah
Interface-local (rozhraní): FF01
Link-local (linka): FF02
Global: FF0E

### Mapování na L2:

Prefix MAC 01:00:5E + bit 0 - mapuje se 23b:




32-to-1 overlapping problem

### Mapování na L2:

Prefix MAC 33:33 - mapuje se 32 b

## IGMP (u IPv6 MLD)

All multic. nodes: 224.0.0.1 / FF02::1
All routers: 224.0.0.2 / FF02::2

V1: 1) router periodicky zasílá membership query: Má někdo zájem o jakýkoliv multicast?
2) stanice musí odpovědět membership report: Já mám (stále) zájem o tento multicast.

V2: 1) přidává možnost routeru zeptat se na specifický multicast
2) přidává stanicím možnost odhlásit se - leave group

V3: přidává možnosti vysílat na stejné multic. adrese (skupině) z více zdrojů a určovat, z jakého zdroje konkrétně jej chci odebírat

Hlavička: umístí IP hlavičky - obsahuje typ, max. response time, checksum a adresu skupiny

Multicastové adresy jsou vždy v polích destination address.

## Jak k tomu přistupuje L2 - switche:

- hloupý switch by ve své CAM table nenašel multicastové „MAC" adresy, takže by dělal flooding
→ IGMP snooping - switche se dívají do L3 hlaviček (:) - chce to HW podporu
- stačí se dívat na 01:00:5E / 33:33

# Multicastové směrování

- unicast: hledáme, kam mají být data doručena
- multicast: řídíme se tím, od koho data tečou
- protokoly: PIM (uvnitř autonomního syst.) a další - M-BGP, MSDP (mezi autonomními syst.)
- distribuční stromy - opět hledáme (minimální kostry) - shortest-path trees
  - využíváme zdrojové stromy (S,G) a sdílené stromy (*,G)

(S,G): zvlášť pro každý multic. zdroj → vždy nejkratší cesta = nejmenší zpoždění,
ale routery si musí pamatovat celé ty stromy pro každý zdroj
a nejvíc si jich —ll— ty pravděpodobně nejhorší routery
(u klientů)

(*,G): jako center-based approach - nemusí vždy téct nejkratší cestou

## PIM (Protocol Independent Multicast) - protokol typu IGP - uvnitř autonom. sys.

- kooperuje s unicastovým směrovacím protokolem - dívá se do routovací tabulky (unicastové)
  ↓
  buduje podle nich distr. stromy
- povídají si na 224.0.0.13 (FF02::d)
- zajišťuje techniku RPF (viz broadcast)

• Dense mód: inkluzivní přístup - „multicast chtějí všichni"
pravidelně zaplavuje síť nějakým multicast provozem, stanice říkají ne (pruning)
používá pouze (S,G) - zdrojové stromy
vhodný do topologie s jedním zdrojem multicastu
„bezstavový" - flooding je pravidelný ✗ velmi jednoduchý

• Sparse mód: exkluzivní přístup - příjemci upozorňují kořen, že mají zájem o multicast
pracuje s oběma typy stromů - (S,G) i (*,G).
→ je třeba vhodně zvolit rendezvous point - RP - centrální směrovač
typický topologie s více zdroji multicastu
všechny ostatní routery musí vědět kdo je RP!
zprávy typu Hello, Register, Register-Stop, Join, Prune
↓
zdroj → RP

„data z vysílačů tečou po zdrojových stromech do RP, kde přejdou do sdíleného stromu a jedou ke kl."

# Broadcast

- nejjednodušší: N-way-unicast
  - není vůbec efektivní
  - musíme znát všechny příjemce v síti - ale jak na ně přijít?
- uncontrolled flooding: uzel pošle broadcastový paket všem sousedům
  ti jej zduplikují a pošlou všem sousedům (kromě zdroje)
  - → problém s cykly
  - → broadcast storm - přehlcení sítě (exponenciálně) vznikajícím počtem broadcast paketů
  - ⇒ je třeba podle něčeho usoudit, jestli paket posílat dál
- controlled flooding:
  - sequence-number-controlled: do paketu je vloženo číslo (unikátní pro daný zdroj)
    všichni ostatní mají seznam přijatých dvojic (zdroj, číslo)
    pokud přijde paket, který tam už je, je zahozen i jinak je tam přidán
  - Reverse Path Forwarding (někdy RP Broadcast): routery pošlou broadcast paket dál pouze tehdy, pokud přišel na lince, která je na nejkratší cestě routem ke zdroji
    → router musí vědět, který jeho soused tvoří nejkratší cestu ke zdroji
  - ⇒ zabraňuje vzniku bouře, ale sítí stále putují redundantní pakety
- Spanning tree broadcast: naprosto ideální by bylo vytvořit minimální kostru grafu reprezentujícího síť a posílat broadcast pakety jen po ní
  → ale jak ji vytvořit a udržovat?
  - center-based approach: jednoduchý alg. pro konstrukci kostry - je vybrán jeden „centrální uzel", všechny ostatní mu unicastovým routovacím posílají „tree-join" zprávu. Na strom se tak postupně nalepí všechny uzly (buď zpráva dojde až do centra, nebo dojde k nějakému už přilepenému bodu)

# TCP

- striktně P2P – 1 odesílatel, 1 příjemce
  → ne pro multicast (→ UDP)
- zachovává pořadí bajtů
- pomocí klouzavého okna umožňuje zřetězený přenos více balíků – segmentů najednou
- vyrovnávací paměti
- spojově orientovaný – před komunikací se vyjednává stav
  - obě fáze života: explicitní vytvoření i ukončení spojení

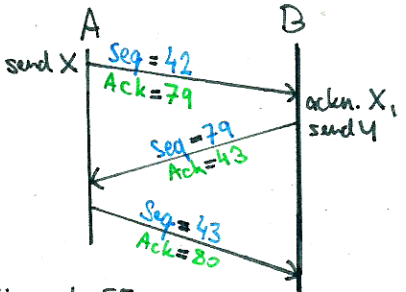
Hlavička:
- source port, dst port
- sekvenční čísla } „stavové prostory" – sekvence pro potvrzování přijatých paketů, jsou rozdílné
- potvrzující čísla } pro komunikaci A→B a B→A
- flagy: URG ACK PSH RST SYN FIN
  - URG → urgentní data
  - ACK → indikuje platné číslo potvrzení
  - SYN → požadavek na vytvoření spojení
  - FIN → pož. na ukončení spojení
- data offset (= velikost hlavičky včetně offsetu a paddingu) – počet 32b slov!
- window: velikost klouzavého okna

TCP je in-order byte stream.

Head-of-line blocking: dokud nemám „začátek" sekvence bajtů, nemůžu je postoupit aplikaci
→ není to data-agnostic

V každém segmentu je kromě dat i potvrzení o tom, co už jsem dostal od protistrany:
(ale typicky se ACK posílají zvlášť...)

Potvrzuje se číslem dalšího očekávaného segmentu

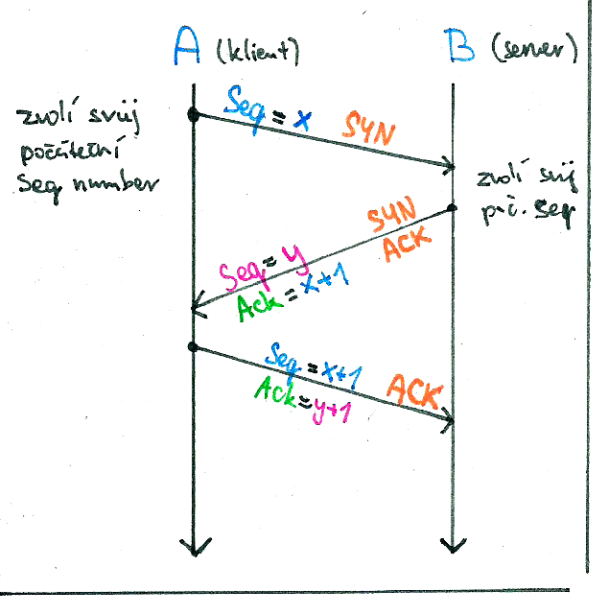



Parametry spojení:
- ISN – Initial Seq Number – přetrká...
- MSL – Maximum Segment Lifetime – jak dlouho hnije v buffem přijatý
- MSS – Max. Segment Size – musí se určit v koordinaci s L3 a L2, podle MTU
  - ideálně nechceme, aby L3 fragmentovala

Spojení – datový tok – Flow je zcela identifikováno 5 parametry:
- zdroj/cíl IP/port
- protokol

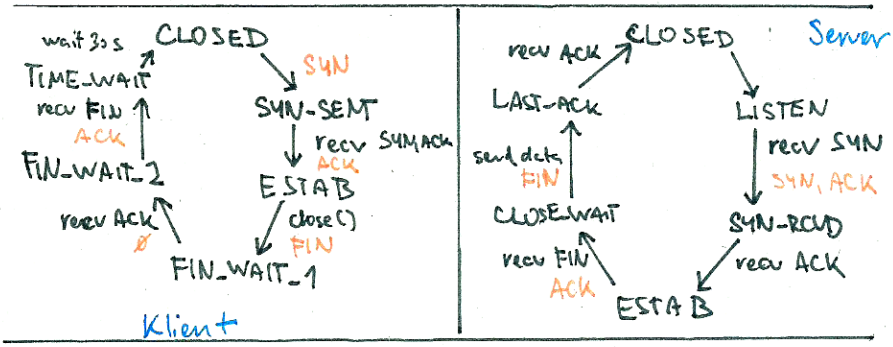
Zahájení spojení: typicky „active-passive" – aktivní klient iniciuje spojení vůči pasivnímu serveru
(ale umožňuje i „active-active")

- 2W handshake je problematický – mohou vznikat napůl otevřená spojení
- užívá se 3WH




## Uzavírání spojení:

- pomocí FIN flagu
- rozlišuje uzavření z jedné strany (half-closed)
  → umožňuje druhé straně doposlat zbytek dat




**Kumulativní potvrzování:** nevadí, když se ztratí ACK, pokud přijde ACK nějakého dalšího balíku

**Co se musí řešit?** Ztráta / zpoždění dat / ACK

**Nastavení timeoutu:**

- timeout < RTT: předčasný timeout → zbytečně se znovu posílá
- timeout > RTT: pomalá reakce na ztrátu → degradace rychlosti

→ Cíl: nastavovat timeout na RTT + něco málo navrch
(to se ale dělá blbě zvlášť kvůli různým peaks)

**Fast retransmit:** pokud je obdrženo více (3+) ACK pro stejný segment, asi se něco ztratilo – pošli hned znovu

**Selektivní potvrzování:** rozšíření TCP (TLV v options) – indikuje se left edge a right edge toho, co jsem přijal

**Zahlcení (network congestion)** – vyšší množství dat než je přenosová kapacita

- **end-end přístup:** L3 neposkytuje informace pro řízení zahlcení
  (x network-assisted) → koncové stanice sledují síť a odhadují stav
  - základ TCP

- jednoduché pravidlo: pakety se neztrácejí = síť nezahlcená → zvýšit rychlost
  pakety se ztrácejí = síť zahlcená → snížit rychlost

→ **Klouzavé okno** – kolik bajtů maximálně odesílám v rámci jednoho segmentu
**cwnd** ≥ last byte sent − last byte **acked**

→ additive increase, multiplicative decrease – odesílatel zvedá rychlost lineárně ($cwnd_{t+1} = cwnd_t + 1$)
– v případě ztráty ji sníží na polovinu ($cwnd_{t+1} = cwnd_t / 2$)

slow start with congestion avoidance: – na začátku, do první ztráty, zvyšuje exponenciálně
– po ní roste po 1

# IPv6

ICMPv6 - řídicí protokol pro všechno - zastává roli ICMP, IGMP, ARP
- chyby
- ping
- MLD (~IGMP)
- objevování sousedů (~ARP) = NDP
- informace o uzlu - pěkné, ale nebezpečné
- ...

Objevování sousedů: Neighbour Discovery Protocol
- funkce pro komunikaci uzel - směrovač
- -||- uzel - uzel

- Router Advertisement: periodické zprávy směrovače oznamující jeho existenci
  nese informace o prefixu, MTU apod.
  obrovský problém: má to obsahovat DNS? Není tam... ale je o tom válka
- Router Solicitation: požadavek hosta na zaslání RA

- Neighbour Solicitation: jaká je IP souseda?
- Neighbour Advertisement: odpověď

NS je zaslána na multicastovou SOLICITED-NODE adresu cíle (hledaného)
→ mapuje spodních 24 bitů FF02::1:FFxx:xxxx

Záznamy o mapování jsou uloženy v Neighbour Cache (~ARP table)

Co se děje po připojení?

1) Zařízení si samo vygeneruje link-local adresu (fe80::/10)
   → adresa je v tentative stavu
2) Duplicate Addr. Detection → je zaslán NS na vygenerovanou adresu - resp. na odpovídající solicited-node.
3) Pokud nikdo neodpoví, pomocí Multic. Listener Report se na danou soliciked-node zaregistruje
4) Adresa se přepne do valid režimu → můžeme komunikovat v LANce. Chceme globální adresu.
5) Zašleme Router Solicitation, router odpoví s RA
   → kdo odesílá, jaký je prefix sítě (+ délka prefixu)
6) Zařízení si nageneruje podle toho prefixu svou novou global adresu (resp. interface ID)
   → je v tentative stavu
7) Opakujeme kolečko NS (NA) MLR.
8) NEMÁME DNS :c ⟶ 6b) Podle RA si popovídáme s DHCPv6.

# Protokoly pro zřetězené zasílání dat

- go-back-N: jednoduchý na impl., stačí logika u odesílatele
- selective repeat: složitější, vyžaduje koordinaci na obou stranách

## Go-back-N:

- odesílatel:
  - drží v paměti N nepotvrzených – N = šířka okna – vždy chci mít na trati N balíků
  - nejstarší nepotvrzený má časovač
  - když vyprší, pošlou se všechny nepotvrzené
- příjemce:
  - odesílá kumulativní potvrzení – o naposledy přijatém
  - neodesílá potvrzení, pokud detekuje ztrátu

- k-bitové sekvenční číslo pro každý paket
- když se něco pokazí, musím vždy řešit jen to okno
- v základu opravdu odesílatel sleduje pouze svůj časovač
  - příjemce duplikuje ACK

## Selective repeat

- odesílatel:
  - drží v paměti N nepotvrzených
  - každý nepotvrzený má časovač – okno se zleva smrskne, když je potvrzen nejlevější
  - po vypršení časovače se pošlé pouze ten jeden
- příjemce:
  - také má buffer pro N balíků
  - odesílá individuální potvrzení

- ta okna mohou být jinak velká!

# Směrování

Distance-vector protokoly - pro výměnu informací používají decentralizovaný, distribuovaný přístup

- uzel si iterativně díky komunikaci se sousedy vytváří a udržuje distance vector
  → odhad vzdáleností všech ostatních uzlů
- postupně posílám sousedům informace o sobě a o tom, jaké sítě znám
- očekávám od sousedů totéž
- asynchronní
- Bellman-Fordův alg. a obměny
- typický zástupce: RIP (also BGP)