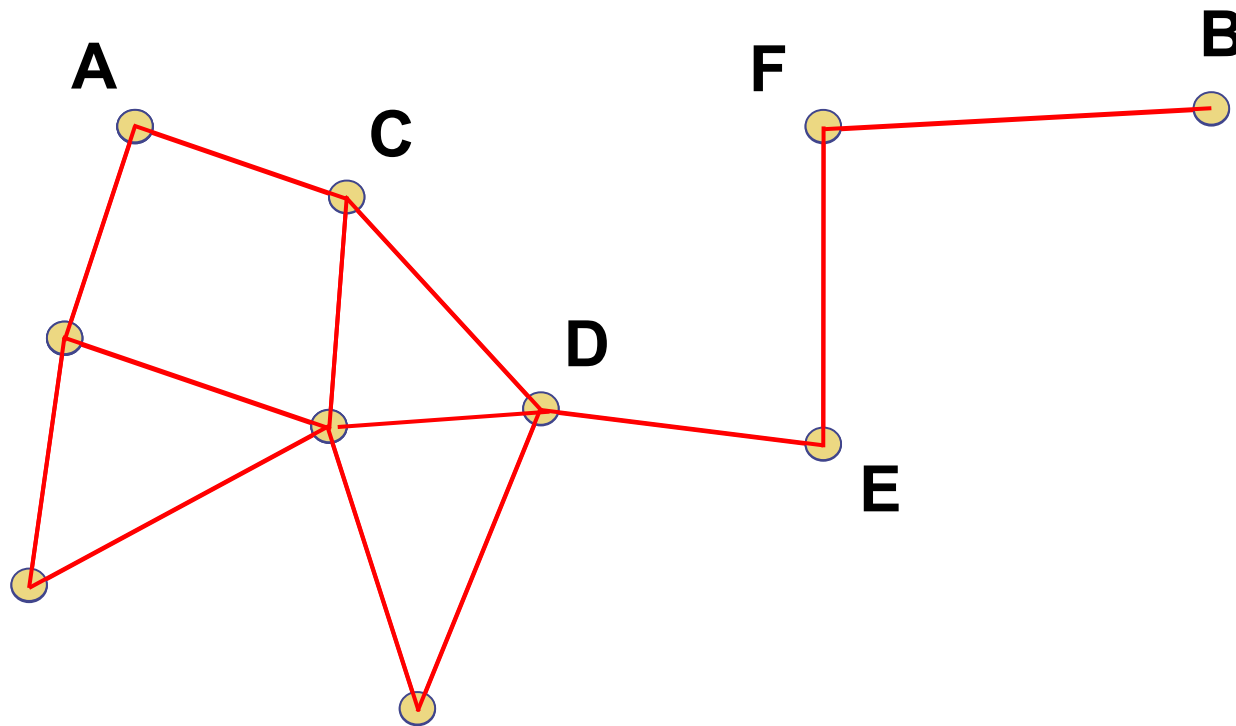


Počítačové siete 2

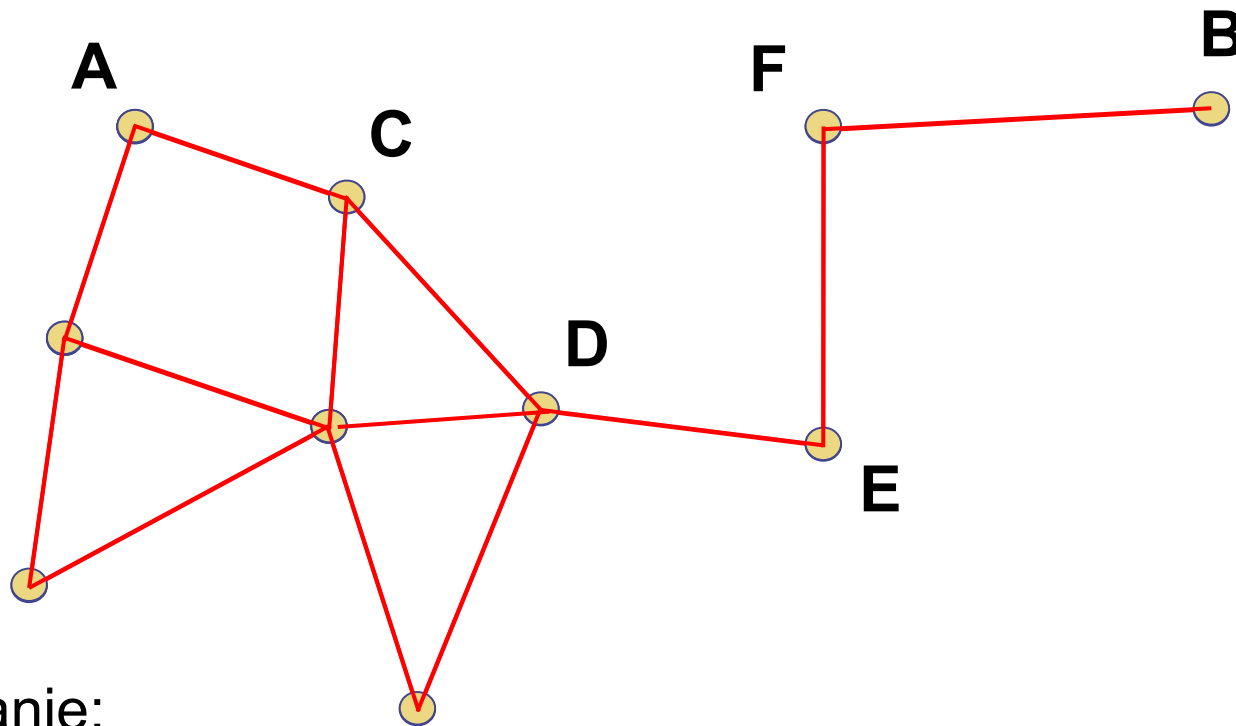
IPv4, IPv6, UDP, TCP, RED

Martin Drozda



MAC:

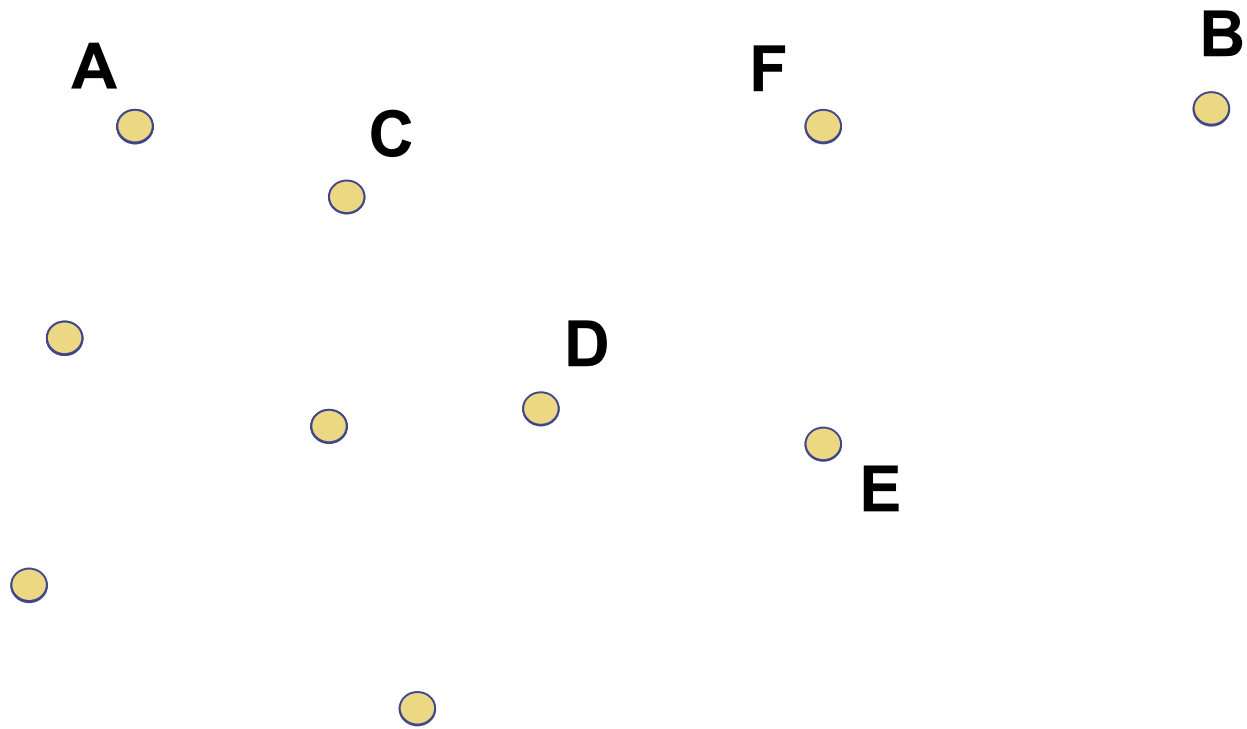
- Rezervuje médium
- Potrebuje informáciu o nasledujúcom uzle od smerovacieho protokolu



Smerovanie:

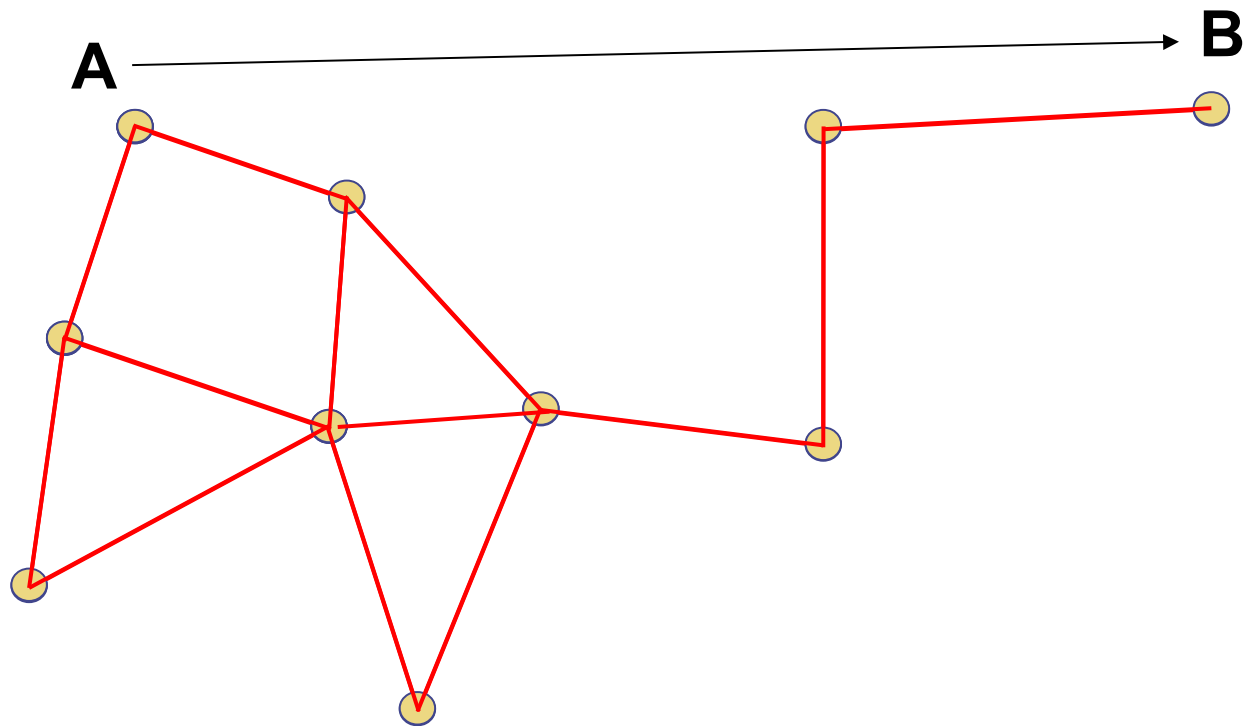
- Výpočet uzlov potrebných na dosiahnutie cieľového uzla
- Štartovací a cieľový uzol sú poskytnuté z transportnej vrstvy

Ad hoc sieť



IP:

- Priradí označenie uzlom (A, B, ..., E) alebo IP adresu



Transportný protokol (TCP, UDP)

- Riadenie spojenia štart-cieľ, ACK paketov na spojení

Životný cyklus dátového paketa

- Dáta sú generované na aplikačnej vrstve (aplikáciou)
- Transportný protokol otvorí spojenie, dáta sú vložené do UDP/TCP paketu s hlavičkami
- Paket je ďalej vložený do IP paketu s hlavičkou
- Cesty do cieľového uzla sú najdené pomocou smerovacieho protokolu
- Paket je vložený do MAC paketa s hlavičkou
- Paket je prenesený fyzickou vrstvou

OSI referenčný model



L7: Aplikačná vrstva (Web, email client, ...)

L6: Prezentačná (reprezentácia dát)

L5: Relačná (spojenie štart/ukončenie)

L4: Transportná (TCP, UDP)

L3: Sieťová (smerovanie) (**IP**; AODV, DSR, TORA, LAR)

L2: Linková (MAC) (802.11, CSMA, MACA, Aloha)

L1: Fyzická (Hardvér, modulácia)

Internet protocol (IP)

IP poskytuje jednoznačné označenie pre každý uzol: IP adresu

IP umožňuje „paketizáciu“, t.j. zabalenie dát do IP paketa s hlavičkou

IP negarantuje kvalitu služieb – spolieha na transportnú vrstvu napr. na TCP

IPv4 používa 32-bit IP adresy

IPv6 používa 128-bit IP adresy

Príklad 128-bitovej adresy:

AAAA:0:F000:0:0:800:800:1A00

Simulačné nástroje ako napr. ns3 podporujú IPv6.

128-bitové adresy sú potrebné pre:

- Internet vecí
- IP telefóniu
- Senzory, roboty, roje robotov...každá logická jednotka potrebuje IP adresu

Základné vlastnosti:

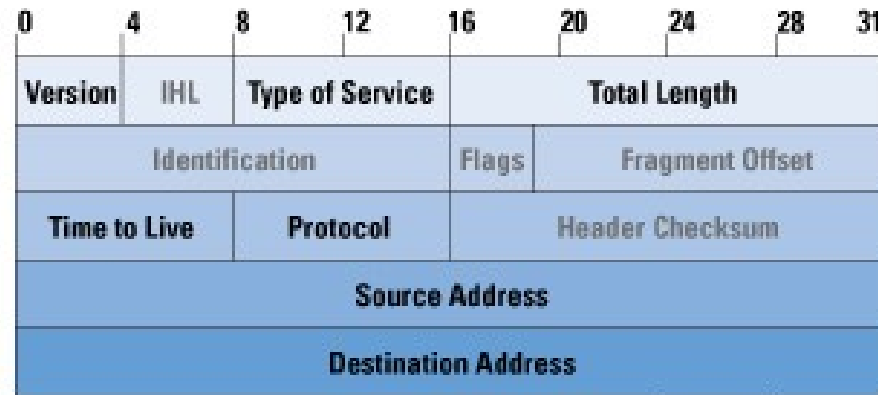
- Žiadne kontrolné sumy (spoliehanie na MAC protokol)
- Flexibilná architektúra hlavičky, podpora pre rozširujúce hlavičky
 - extension headers (optional): fragment header, destination options, hop-by-hop header, routing header, authentication header (MD5 hash, ...)
- Označenie toku: každé spojenie uzlov, ktoré je logicky rozdielne od iného spojenia potrebuje vlastné označenie toku
- Hop limit: maximálny počet preposlaní pre daný paket
- 128-bit zdrojová a cieľová adresa
- Verzia, dĺžka

Základné vlastnosti:

- version, header length, total length, fragment id, time-to-live (TTL), protocol used in data part, source address, destination address, header checksum, and other.
- Protokol v dátovej časti
- Offset k dátovej časti

IPv4 vs IPv6

IPv4 Header



IPv6 Header



IHL = Internet Header Length = dĺžka hlavičky, dĺžka hlavičky je variabilná

Figure source: shivasoft.in

OSI referenčný model



L7: Aplikačná vrstva (Web, email client, ...)

L6: Prezentačná (reprezentácia dát)

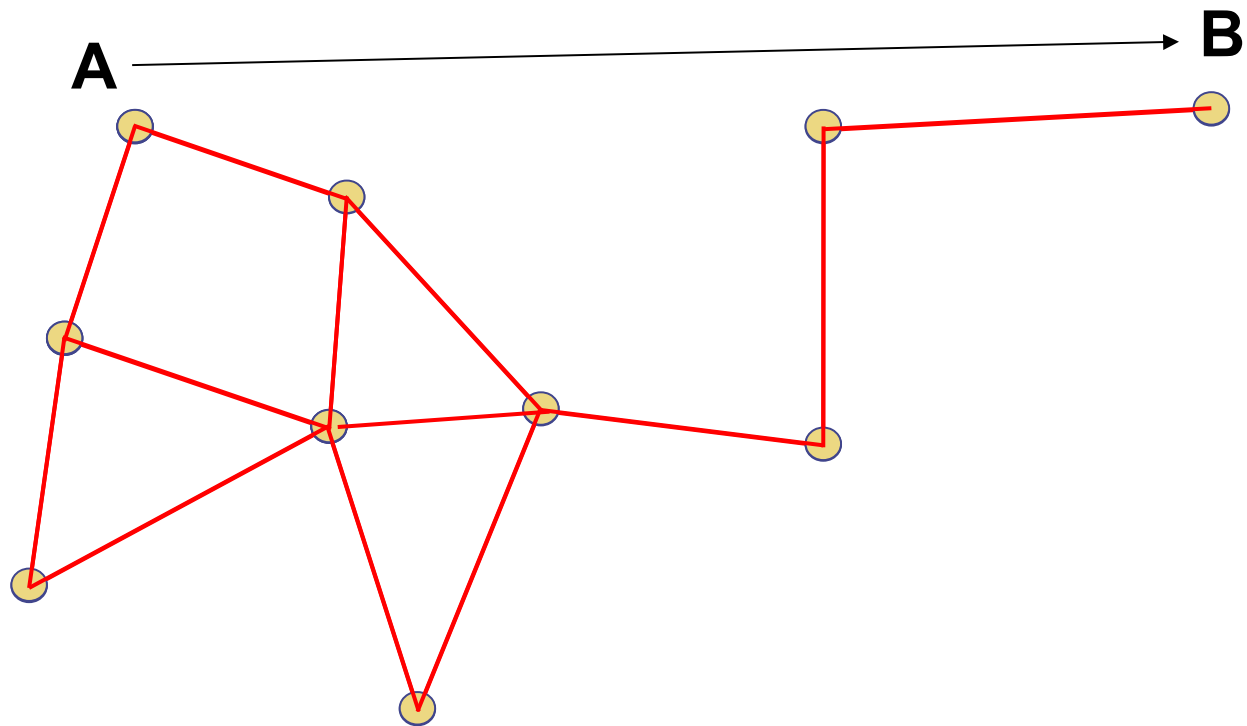
L5: Relačná (spojenie štart/ukončenie)

L4: Transportná (TCP, UDP)

L3: Sieťová (smerovanie) (IP; AODV, DSR, TORA, LAR)

L2: Linková (MAC) (802.11, CSMA, MACA, Aloha)

L1: Fyzická (Hardvér, modulácia)



Transportný protokol umožňuje komunikáciu medzi vzdialenými procesmi na rôznych uzloch.

TCP: Transmission control protokol

TCP je spoľahlivý, využívajúci spojenie, full-duplex (obojsmerný) protokol s riadením toku a zahltenia.

TCP je schopný doručiť pakety v správnom poradí, v prípade, že pakety sú doručené pomocou rôznych ciest alebo v prípade zahltenia.

TCP umožňuje „packet switching“

TCP rozdeľuje pakety do menších častí, tzv. Segmentov.

UDP: User datagram protocol

UDP je nespoľahlivý protokol bez riadenia toku a nedokáže doručiť pakety v správnom poradí.

UDP je „best effort“ protokol, spolieha na iné protokoly, že dokážu ovplyvniť doručenie paketov.

Porovnanie s poštou:

UDP:

- Pohľadnica so známku je poslaná a dôverujeme poštovej službe (Slovenskej pošte), že pohľadnicu doručí.
- Neexistuje logické spojenie zdrojového a cieľového uzla.

TCP:

- List je poslaný doporučené s návratkou. Ak návratka nie je doručená do istého času, list je znova poslaný.
- Existuje logické spojenie zdrojového a cieľového uzla.

TCP segment (paket)

Základné vlastnosti:

- Port zdrojového uzla
- Port cieľového uzla
- Sekvenčné číslo
- ACK sekvenčné číslo
- Dĺžka hlavičky
- Flags: SEQ, ACK, FIN, ...
- Kontrolná suma
- Dáta

TCP: otvorenie spojenia

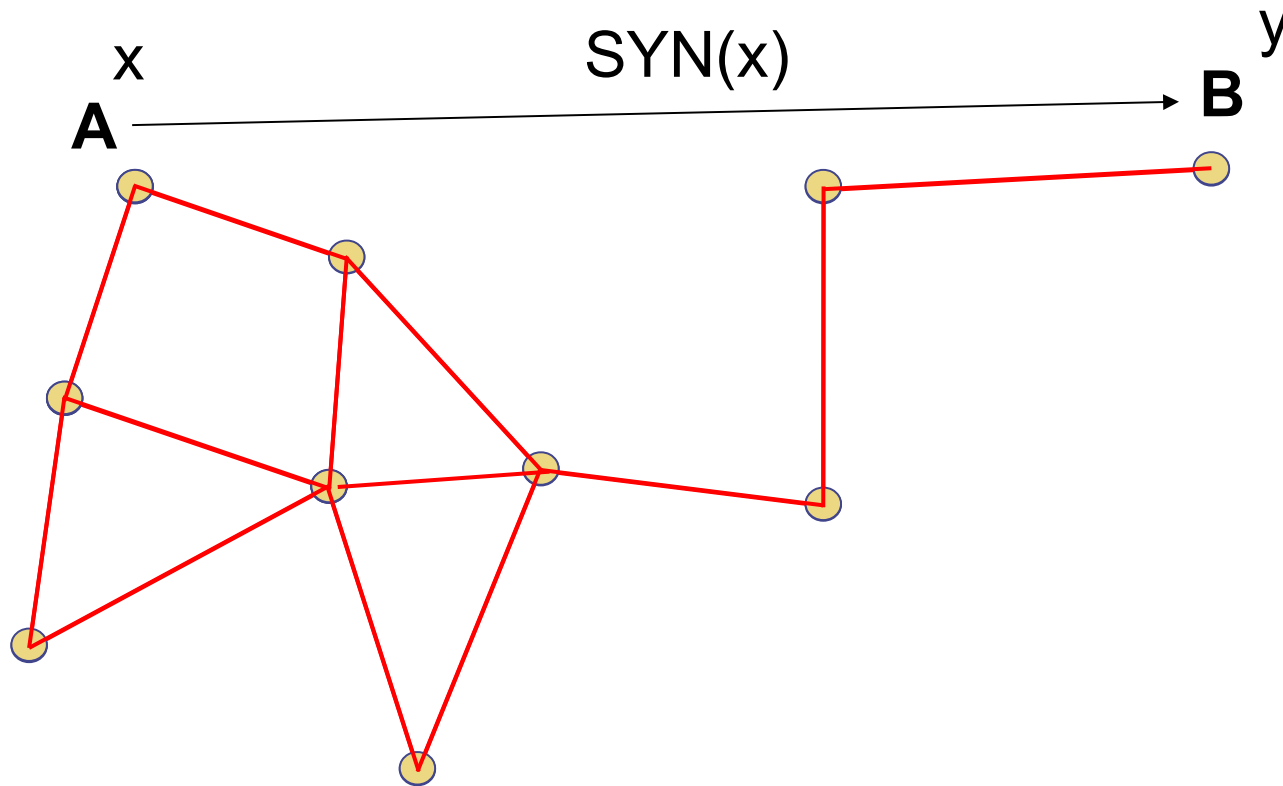
Server otvorí port pre príjem dát:

- Client pošle SYN, teda pošle tzv. “active open call”.
- Server odpovie pomocou SYN-ACK.
- Client odpovie pomocou ACK

SYN obsahuje náhodné sekvenčné číslo

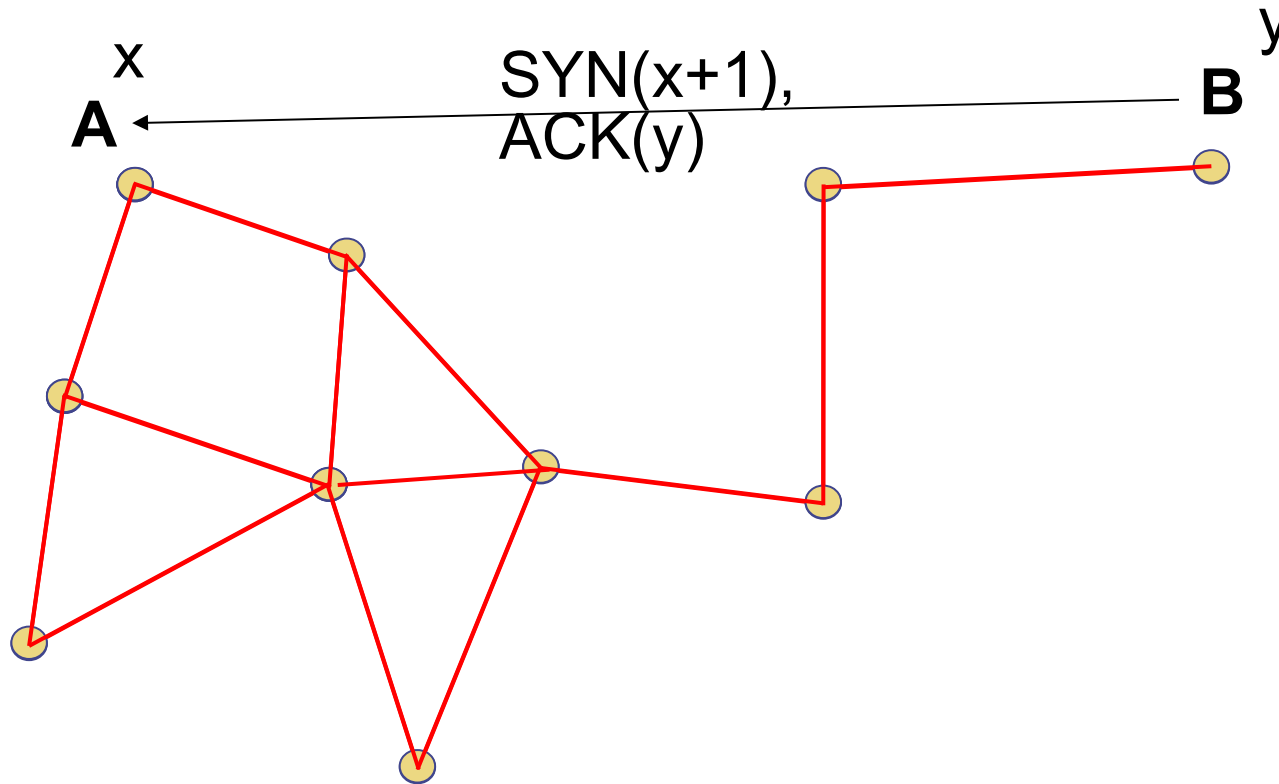
ACK obsahuje náhodné sekvenčné číslo

TCP: príklad



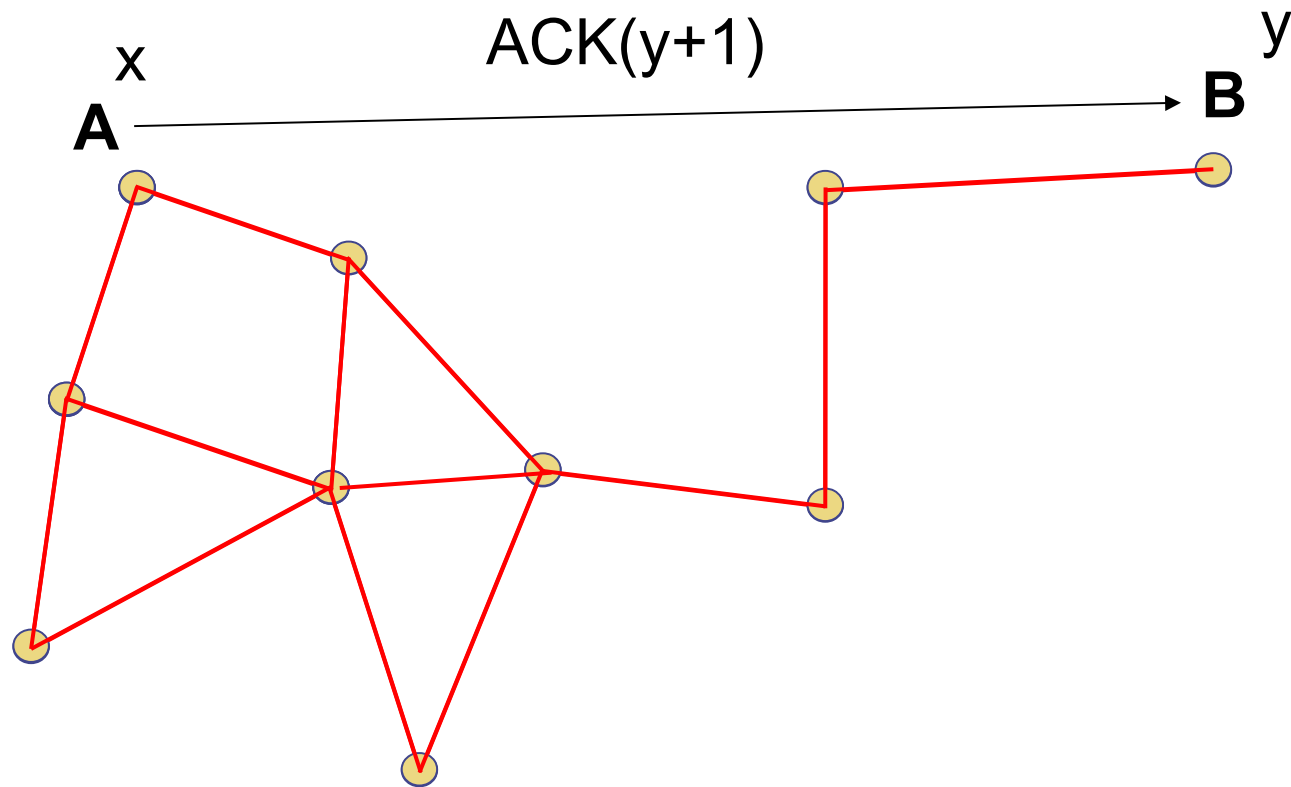
Klient pošle SYN so sekvenčným číslom x

TCP: príklad



Server pošle SYN-ACK.
SYN obsahuje sekvenčné číslo $x+1$.
ACK obsahuje sekvenčné číslo y .

TCP: príklad



Klient pošle ACK so sekvenčným číslom $y+1$

TCP: RTT

Dátový paket je opätovne poslaný, ak ACK nie je prijaté do určitého času, do time-out odvodeného od RTT

RTT = round trip time

ACK = v tomto prípade povrdenie prijatia dátového paketa/segmentu.

Výpočet RTT:

- Zmeraj RTT pre každý pár: segment a ACK
- $\text{EstimatedRTT} = a * \text{EstimatedRTT} + b * \text{SampleRTT}$
kde $a+b = 1$, $a = 0.8-0.9$, $b=0.1-0.2$
- $\text{timeout} = 2 * \text{EstimatedRTT}$

TCP: ukončenie spojenia

Ukončenie:

- Uzol (proces), ktorý už nepotrebuje posielanie dát, pošle segment (paket) s nastaveným FIN príznakom
- Druhý uzol (proces) potvrdí prijatie tohto FIN segmentu (spojenie je „half-open“)
- Druhý uzol (proces) pošle segment (paket) s nastaveným FIN príznakom
- Prvý uzol (proces) potvrdí prijatie tohto FIN segmentu

TCP: sliding window flow control

RFC 793, Jacobson (SIGCOMM'88).

- Každý paket (segment) musí byť potvrdený cieľovým uzlom (procesom)
- „Sliding window“ je max. počet segmentov, ktoré môžu byť poslané bez potvrdenia (bez ACK)
- Viacero segmentov môže byť potvrdených jediným ACK
- Ak ACK nie je prijaté do time-outu, segment je znova poslaný
- Počet pokusov poslať segment po time-oute nie je predpísaný špecifikáciou
- Po každom prijatom ACK je „sliding window“ posunuté o jednu pozíciu

Veľkosť „sliding window“ sa dynamicky mení, podľa zaťaženia siete.

TCP: slow start/congestion avoidance



1 segment je poslaný



1 segment je potvrdený, veľkosť cwnd je inkrementovaná



1 segment je potvrdený, 1 nie je zatiaľ potvrdený



2 segmenty sú potvrdené

Exponenciálny pomalý štart je implementovaný na báze potvrdenia každého segmentu

TCP: tok a riadenie zahltenia

Riadenie toku: riadenie rýchlosti posielania dát

Riadenie zahltenia: preťaženie uzlov z dôvodu neprimeranej rýchlosti posielania dát

TCP prepokladá, že straty paketov sú z dôvodu zahltenia. V ad hoc sieťach sú straty pravdepodobné aj z iných dôvodov:

- Bezdrôtové médium má nízku kapacitu
- Počasie, pohyb...

Okno zahltenia (TCP), okno exponenciálneho čakania (MAC)

Terminológia:

- Transportný protokol: TCP – okno zahltenia
- MAC protokol: 802.11 – okno exponenciálneho čakania

Rozdielne vrstvy, úplné iná funkcionalita

TCP: slow start/congestion avoidance

$cwnd$ = veľkosť „sliding window“, congestion window

$ssthresh$ = slow start threshold

$cwnd = 1$

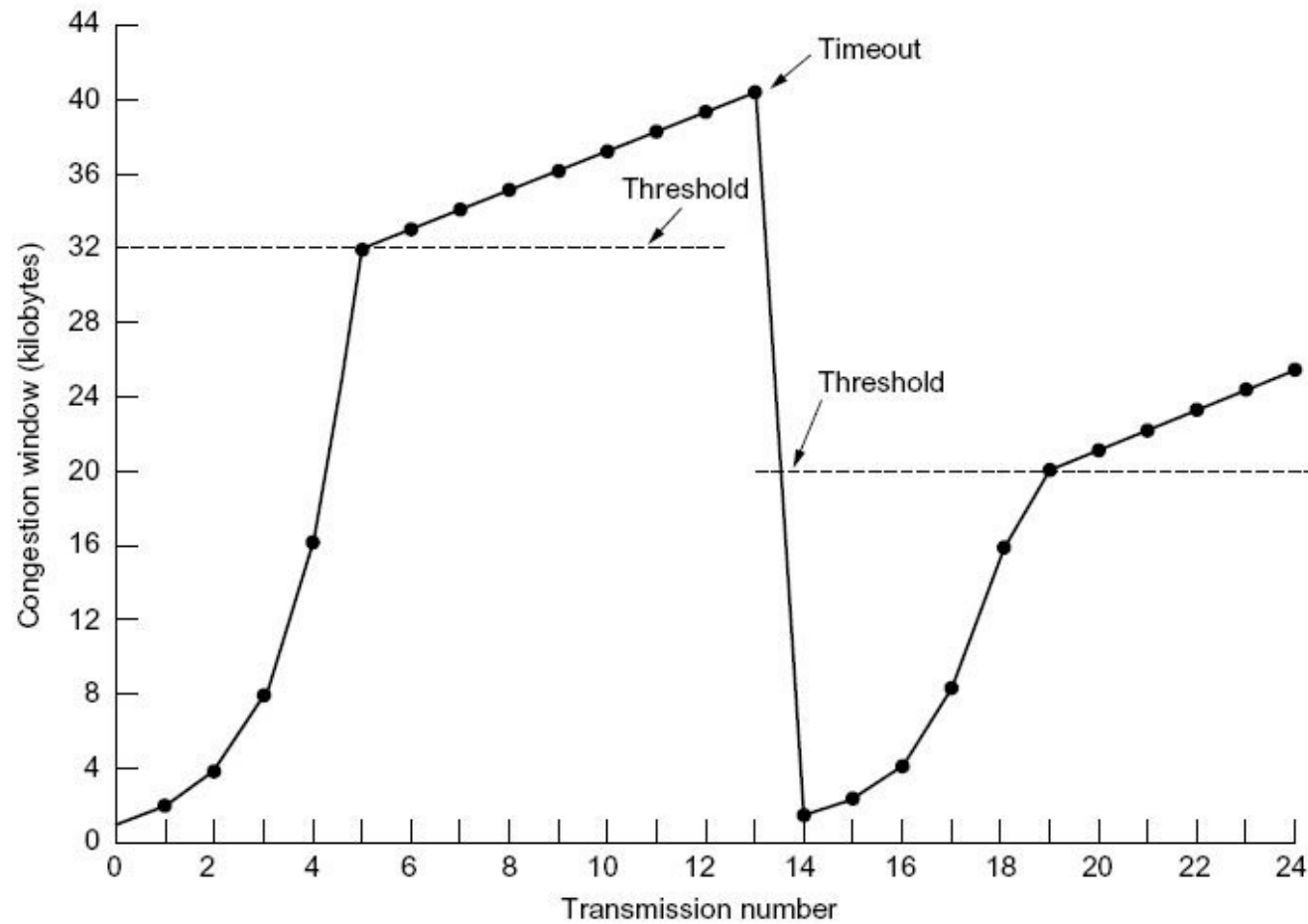
Ak je segment správne prijatý:

- $cwnd = 2 * cwnd$, až po $ssthresh$ (hranica pre pomalý štart)
- $cwnd = cwnd + 1$

Ak nie je segment správne prijatý:

- $ssthresh = cwnd / 2$, t.j. nastal time-out pre ACK alebo boli prijaté duplikáty ne ACK
- $cwnd = 1$, nastane nový pomalý štart

TCP: slow start/congestion avoidance

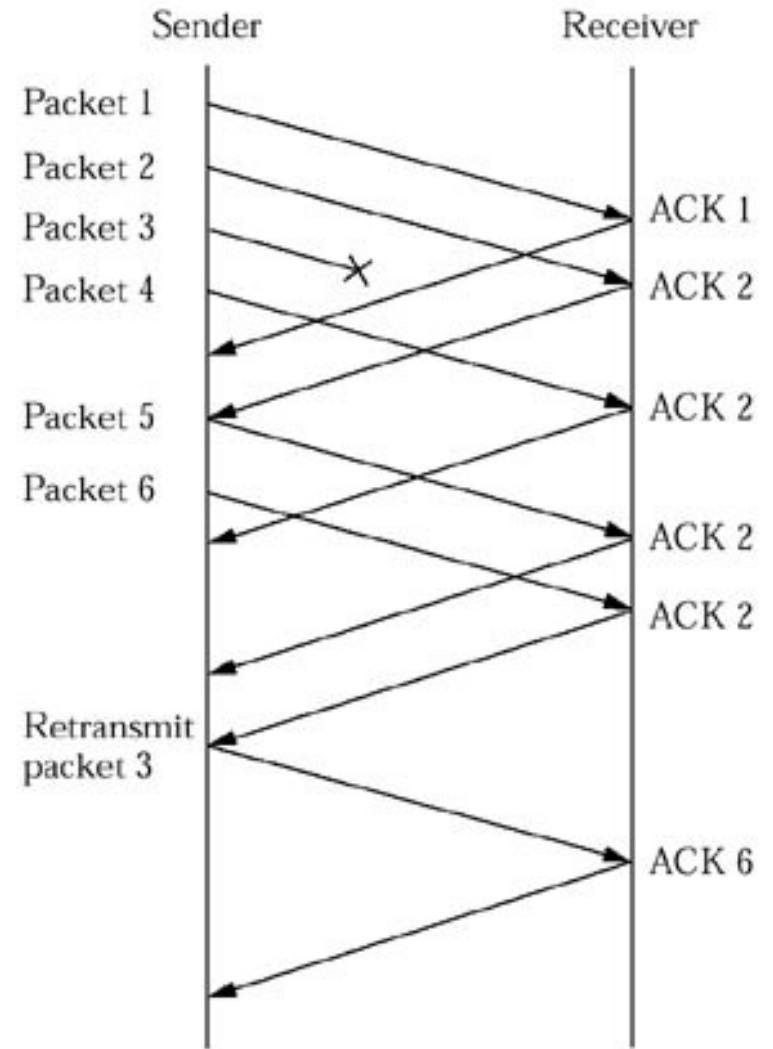


TCP: fast retransmit

Duplicate ACK: ak je prijatý segment mimo správnu sekvenciu

Ak sú prijaté 3 alebo viaceré ACK, potom je segment opätovne poslaný

Fast retransmit = nečaká sa na time-out



TCP: fast recovery

Fast recovery nastane po fast retransmit

- Po 3. duplikátnom ACK, $ssthresh = \max\{cwnd/2, 2\}$
- Nepotvrdený segment je opätovne poslaný
- $cwnd = ssthresh + 3 * seg_size$ (3 fragmenty boli poslané po nepotvrdenom fragmente)
- $cwnd = cwnd + 1$

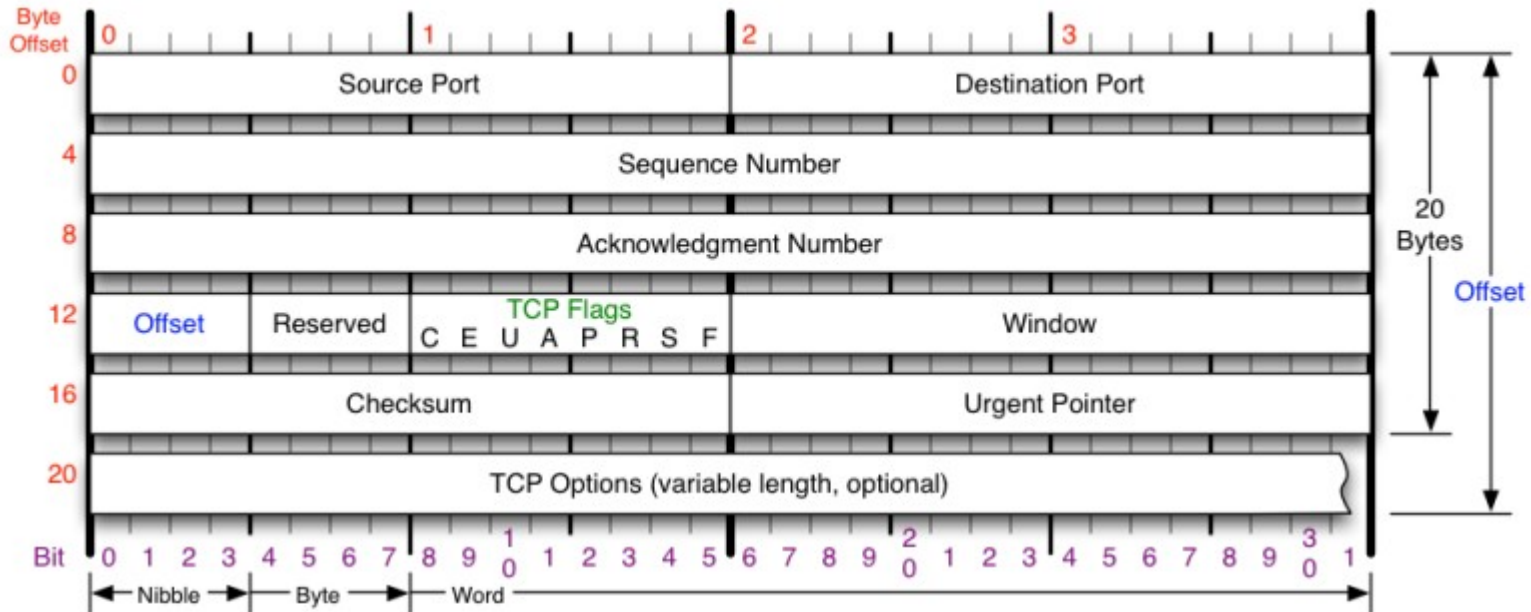
TCP: verzie

TCP Tahoe: Slow start, Congestion avoidance, Fast retransmit

TCP Reno: Tahoe + Fast recovery

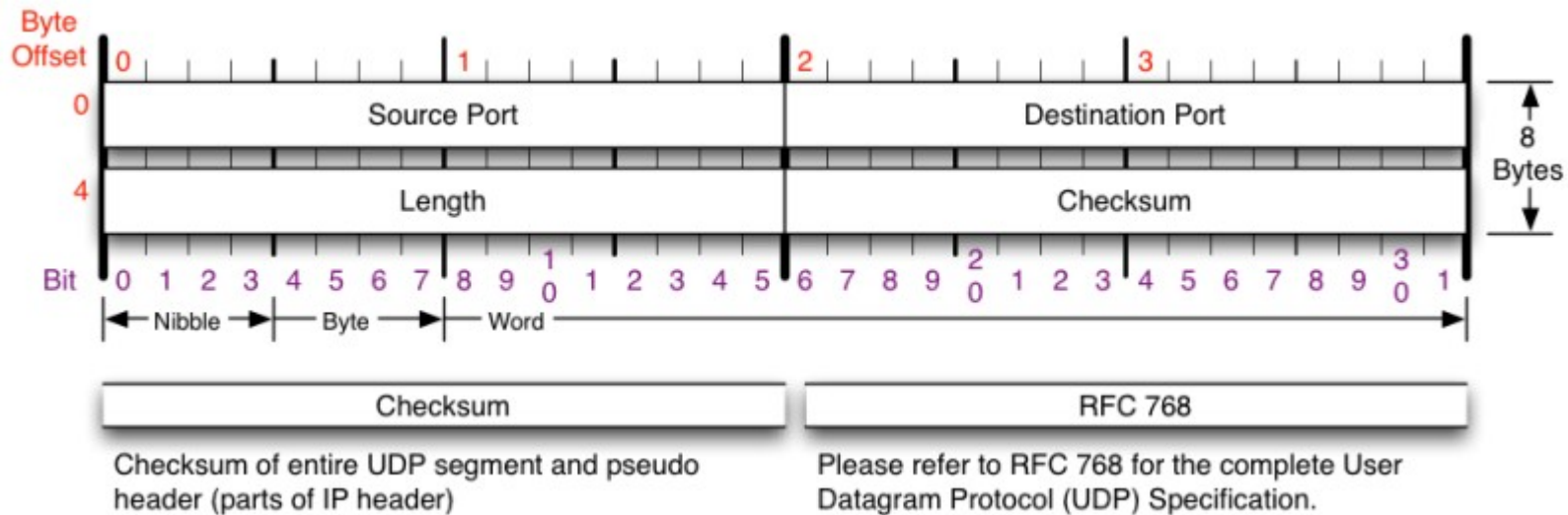
TCP SACK (selective ACK): segmenty sú explicitne potvrdené

TCP: hlavička



TCP Flags	Congestion Notification	TCP Options	Offset																											
C E U A P R S F	ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.	0 End of Options List 1 No Operation (NOP, Pad) 2 Maximum segment size 3 Window Scale 4 Selective ACK ok 8 Timestamp	Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.																											
<p>Congestion Window</p> <p>C 0x80 Reduced (CWR)</p> <p>E 0x40 ECN Echo (ECE)</p> <p>U 0x20 Urgent</p> <p>A 0x10 Ack</p> <p>P 0x08 Push</p> <p>R 0x04 Reset</p> <p>S 0x02 Syn</p> <p>F 0x01 Fin</p>	<table border="1"> <thead> <tr> <th>Packet State</th> <th>DSB</th> <th>ECN bits</th> </tr> </thead> <tbody> <tr> <td>Syn</td> <td>00</td> <td>11</td> </tr> <tr> <td>Syn-Ack</td> <td>00</td> <td>01</td> </tr> <tr> <td>Ack</td> <td>01</td> <td>00</td> </tr> <tr> <td>No Congestion</td> <td>01</td> <td>00</td> </tr> <tr> <td>No Congestion</td> <td>10</td> <td>00</td> </tr> <tr> <td>Congestion</td> <td>11</td> <td>00</td> </tr> <tr> <td>Receiver Response</td> <td>11</td> <td>01</td> </tr> <tr> <td>Sender Response</td> <td>11</td> <td>11</td> </tr> </tbody> </table>	Packet State	DSB	ECN bits	Syn	00	11	Syn-Ack	00	01	Ack	01	00	No Congestion	01	00	No Congestion	10	00	Congestion	11	00	Receiver Response	11	01	Sender Response	11	11	<p>Checksum</p> <p>Checksum of entire TCP segment and pseudo header (parts of IP header)</p>	<p>RFC 793</p> <p>Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.</p>
Packet State	DSB	ECN bits																												
Syn	00	11																												
Syn-Ack	00	01																												
Ack	01	00																												
No Congestion	01	00																												
No Congestion	10	00																												
Congestion	11	00																												
Receiver Response	11	01																												
Sender Response	11	11																												

UDP: hlavička



TCP:

- Pôvodne určené pre drôtové siete
- Každá strata segmentu je interpretovaná ako zahltenie

Bezdrôtové siete:

- **Potrebné rozlišovať zahltenie a zníženie prenosovej schopnosti bezdrôtového média**

Monitorovanie paketovej fronty

Každý uzol má paketovú frontu, t.j. buffer pre prijaté pakety
Veľkosť paketovej fronty je obmedzená a môže sa teda naplniť

Možnosti uvoľnenia paketovej fronty:

- tail drop; zmazanie posledných n prijatých paketov
- random drop; náhodné zmazanie prijatých paketov
- Random early detection (RED)

Tail drop / random drop môžu spôsobiť resynchronizáciu TCP, t.j. veľkosť okna zahltenia môže byť zmenená pre mnoho uzlov

Random early detection (RED)

for each packet arrival:
 calculate the average queue size avg
 if $min \leq avg < max$
 with probability p_a :
 mark the arriving packet
 else if $max \leq avg$
 mark the arriving packet

mark = zmaž z dátovej fronty

avg = veľkosť fronty počítaná ako exponenciálne vážený
klzavý priemer

Random early detection (RED)

- Ak je veľkosť fronty menej ako min, žiadne pakety nie sú zmazané
- Ak je veľkosť fronty viacej ako max, všetky nové prijaté pakety sú zmazané
- Ak je $\text{min} \leq \text{veľkosť} < \text{max}$, nový prijatý paket je zmazaný s pravdepodobnosťou p_a

$$p_b = \max_p (\text{avg} - \text{min}) / (\text{max} - \text{min}); \quad p_b = \langle 0, \max_p \rangle$$

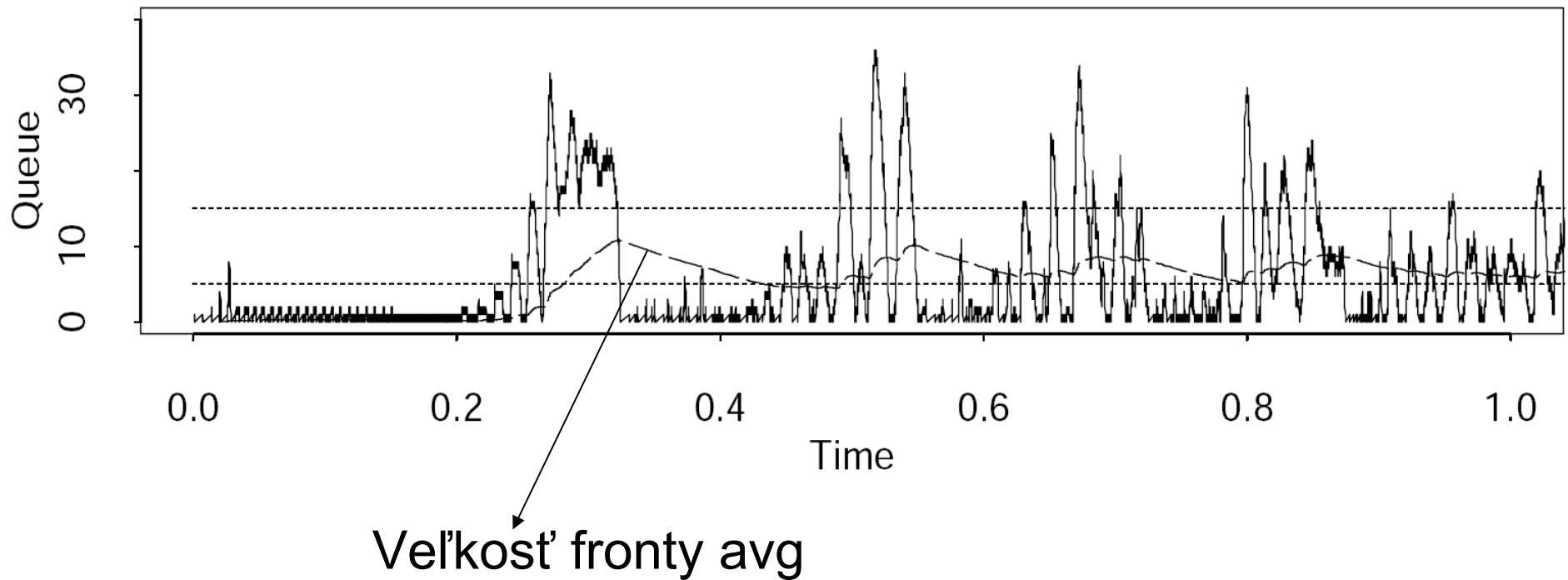
$$p_a = p_b / (1 - \text{count} * p_b);$$

count je počet paketov od posledného zmazaného paketa
 \max_p je napr. 0.02

Vlastnosti:

- Garantuje max. veľkosť fronty
- Menej re-synchronizácie

Random early detection (RED)



Source: S. Floyd, V. Jacobson. „Random early detection gateways for congestion avoidance“, IEEE/ACM Transaction on Networking, 1993.

Sami Iren and Paul D. Amer and Phillip T. Conrad, The transport layer: tutorial and survey, ACM Computing Surveys, vol. 31, no. 4, pp. 360-404, 1999.

Live Q&A

Live otázky a odpovede sú plánované cez Discord.