

Transportni sloj II: TCP i kontrola zagušenja

Predavanje 5 • kolegij Računalne mreže

doc. dr. sc. Sandi Baressi Šegota

Fakultet informatike Sveučilišta Jurja Dobrile u Puli

Sadržaj predavanja

- 1 TCP: pregled i veza
- 2 Pouzdani prijenos u TCP-u
- 3 Kontrola toka
- 4 Kontrola zagušenja
- 5 Sažetak

TCP: pregled i veza

Ishodi učenja

Nakon ovog predavanja moći ćete:

- opisati TCP segment, sekvencijske i ACK brojeve;
- objasniti uspostavu i zatvaranje veze;
- objasniti procjenu RTT-a i **brzu retransmisiju**;
- razlikovati **kontrolu toka** i **kontrolu zagušenja**;
- opisati **slow start**, **AIMD** i reakciju na gubitak.

TCP: obilježja

- **spojni** (connection-oriented): rukovanje prije prijenosa
- **pouzdan, uređen** tok bajtova
- **dvosmjernan** (full-duplex)
- **kontrola toka** i **kontrola zagušenja**

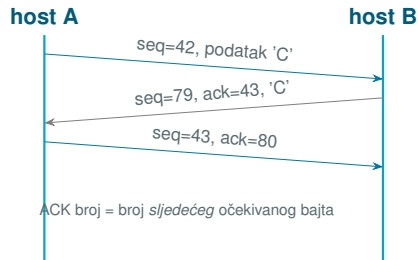
TCP segment

- sekvencijski i ACK broj (bajtovi)
- zastavice: **SYN, ACK, FIN**
- **prozor primatelja** (rwnd)
- checksum



Sekvencijski i ACK brojevi

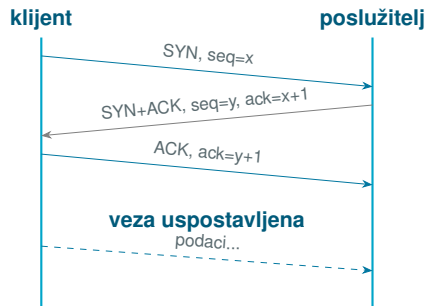
- **seq**: redni broj *prvog bajta* u segmentu
- **ACK**: broj *sljedećeg očekivanog bajta*
- kumulativne potvrde



Uspostava veze: trosmjerno rukovanje

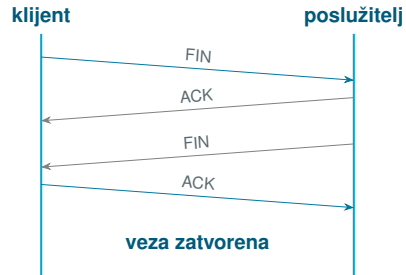
- 1 klijent → **SYN** (seq=x)
- 2 poslužitelj → **SYN+ACK** (seq=y, ack=x+1)
- 3 klijent → **ACK** (ack=y+1)

Tek tada slijede podaci.



Zatvaranje veze

- svaka strana šalje **FIN**
- druga potvrđuje **ACK**-om
- veza se zatvara u oba smjera



TCP vs UDP: sažetak

TCP	UDP
veza + rukovanje	bez veze
pouzdan, ureden tok	best-effort
kontrola toka i zagušenja	bez kontrole
veći troškovi	mali troškovi

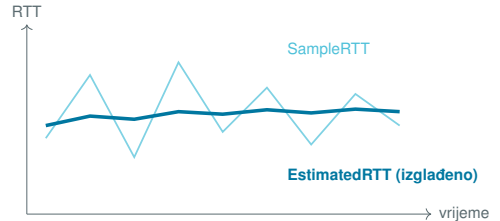
Kumulativne potvrde

- $ACK=n$ znači: „primio sam sve do bajta $n - 1$ ”
- izgubljeni segment $\Rightarrow$ primatelj ponavlja **isti** ACK
- pošiljalac po ACK-u pomiče prozor i šalje nove podatke

Pouzdani prijenos u TCP-u

Timeout i procjena RTT-a

- mjeri se **SampleRTT**
- izglađuje u **EstimatedRTT**
- $\text{timeout} = \text{EstimatedRTT} + \text{rezerva}$



Formule za timeout

$$\text{EstimatedRTT} = (1 - \alpha) \text{EstimatedRTT} + \alpha \text{SampleRTT}$$

$$\text{DevRTT} = (1 - \beta) \text{DevRTT} + \beta |\text{SampleRTT} - \text{EstimatedRTT}|$$

$$\text{Timeout} = \text{EstimatedRTT} + 4 \cdot \text{DevRTT}$$

Ideja

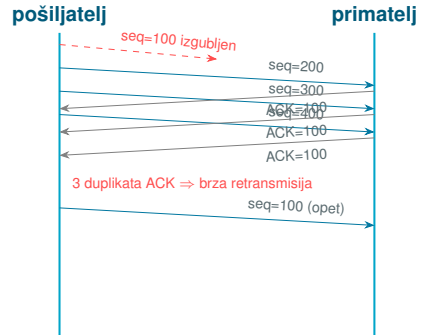
Timeout prati **prosjek** i **promjenjivost** RTT-a (tipično $\alpha = 0,125$, $\beta = 0,25$).

TCP retransmisija

- jedan timer za najstariji nepotvrđeni bajt
- **timeout** $\Rightarrow$ ponovno pošalji taj segment
- po primitku ACK-a pomiče se prozor, timer se resetira

Brza retransmisija (3 duplikata ACK)

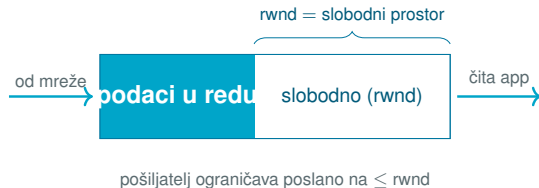
- primatelj na prazninu šalje **isti** ACK
- **3 duplikata** $\Rightarrow$ vjerojatan gubitak
- pošiljatelj šalje *prije* isteka timera



Kontrola toka

Kontrola toka: ne preplavi primatelja

- primatelj javlja **rwnd** (slobodni prostor)
- pošiljatelj drži „u letu” $\leq \text{rwnd}$
- štiti **međuspremnik primatelja**



Razlika

Kontrola toka štiti **primatelja**; kontrola zagušenja štiti **mrežu**.

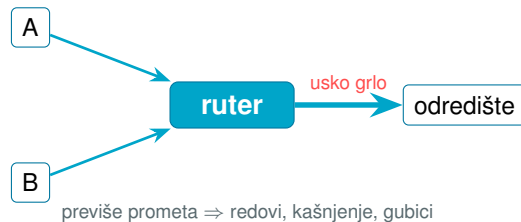
Nulti prozor i persist timer

- ako je $rwnd = 0$, pošiljalatelj mora stati
- ali kako će saznati da se prostor oslobodio?
- **persist timer**: povremeno šalje mali „ispitni” segment

Kontrola zagušenja

Što je zagušenje?

- previše izvora šalje **prebrzo** za mrežu
- posljedice: **redovi**, kašnjenje, **gubici**
- razlikuje se od kontrole toka



Posljedice zagušenja

- veliko **kašnjenje** kako se redovi pune
- **gubici** zbog prelijevanja spremnika
- **uzaludan rad**: retransmisije troše kapacitet

Pristupi kontroli zagušenja

Kraj-do-kraja

Mreža ne pomaže; krajnji uređaji **zaključuju** zagušenje iz gubitaka/kašnjenja (tako radi TCP).

Uz pomoć mreže

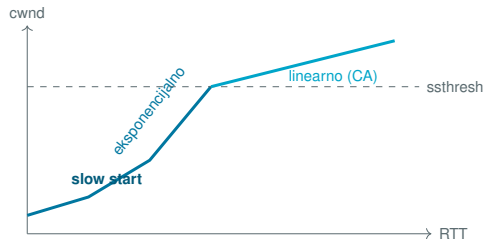
Ruteri **signaliziraju** zagušenje (npr. ECN).

TCP: prozor zagušenja (cwnd)

- pošiljatelj drži „u letu” $\leq \min(\text{cwnd}, \text{rwnd})$
- **cwnd** se prilagođava prema stanju mreže
- propusnost $\approx \frac{\text{cwnd}}{\text{RTT}}$

Slow start

- počni s malim cwnd
- **udvostruči** svaki RTT (eksponencijalno)
- do praga **ssthresh**, pa pređi na CA



Primjer: rast cwnd

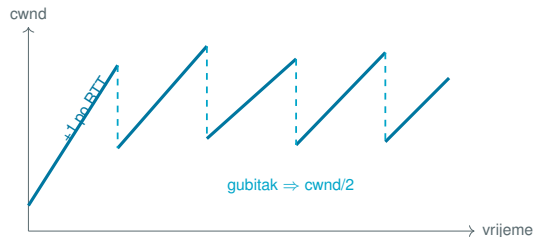
- slow start: $cwnd = 1, 2, 4, 8, 16$ (udvostručenje svaki RTT)
- nakon $ssthresh = 16$: $cwnd = 17, 18, 19 \dots$ (aditivno)
- pri gubitku: $cwnd$ na pola $\Rightarrow$ „zub pile“

Propusnost

Prosječna propusnost TCP-a $\approx \frac{3}{4} \cdot \frac{W_{max}}{RTT}$ (uz periodičan gubitak).

Izbjegavanje zagušenja: AIMD

- **aditivno povećanje**: +1 MSS po RTT
- **multiplikativno smanjenje**: pri gubitku $\text{cwnd} \rightarrow \text{cwnd}/2$
- karakteristični „zubi pile“



Reakcija na gubitak

- **3 duplikata ACK** (blagi znak) $\Rightarrow$ cwnd na pola, **fast recovery**
- **Timeout** (ozbiljan znak) $\Rightarrow$ cwnd na 1, ssthresh na pola, natrag u **slow start**

Stroj stanja kontrole zagušenja



Moderni TCP: CUBIC

- klasični AIMD sporo puni **brze** linkove velikog RTT-a
- **TCP CUBIC** raste kubnom funkcijom vremena od zadnjeg gubitka
- danas zadan u mnogim operacijskim sustavima

Pravednost (fairness)

- više TCP tokova na istom grlu teži **jednakom** udjelu
- AIMD dinamika vodi prema pravednoj raspodjeli
- UDP tokovi **ne** surađuju $\Rightarrow$ mogu „ugušiti” TCP

ECN: eksplicitna obavijest o zagušenju

- ruter **označi** pakete (umjesto da ih odbaci) kad se puni
- primatelj vrati oznaku pošiljatelju
- pošiljatelj smanji brzinu *bez* gubitka paketa

Bufferbloat (napredno)

- preveliki spremnici u ruterima $\Rightarrow$ dugi redovi
- TCP koji reagira *tek* na gubitak puni te redove
- posljedica: veliko **kašnjenje** unatoč propusnosti
- rješenja: pametno upravljanje redovima (AQM), ECN

Sažetak

Ključne poruke

- TCP daje **pouzdan, ureden** tok uz seq/ACK brojeve
- veza se **uspostavlja** rukovanjem i **zatvara** FIN-om
- **timeout** i **brza retransmisija** nose pouzdanost
- **kontrola toka** štiti primatelja, **kontrola zagušenja** mrežu
- **slow start** + **AIMD** prilagođavaju cwnd stanju mreže

Sljedeće predavanje

Mrežni sloj — podatkovna ravnina: IP, adresiranje, NAT.

Pitanja za provjeru

- Što znače seq i ACK brojevi u TCP-u?
- Kako TCP procjenjuje timeout?
- Zašto tri duplikata ACK pokreću retransmisiju?
- Koja je razlika između kontrole toka i zagušenja?
- Kako TCP reagira na timeout, a kako na 3 duplikata ACK?

Hvala na pažnji!

Pitanja i rasprava

doc. dr. sc. Sandi Baressi Šegota • FIPU