
Transportni sloj II: TCP i kontrola zagušenja

Poglavlje 5 — Skripta kolegija *Računalne mreže*

doc. dr. sc. Sandi Baressi Šegota

Fakultet informatike Sveučilišta Jurja Dobrile u Puli

Poglavlje 5

Transportni sloj II: TCP i kontrola zagušenja

TCP (*Transmission Control Protocol*) najvažniji je transportni protokol Interneta: pruža *pouzdan, uređen i dvosmjernan* tok bajtova između dvaju procesa, uz kontrolu toka i kontrolu zagušenja. U ovom poglavlju upoznajemo strukturu TCP veze i segmenta, mehanizme pouzdanog prijenosa (procjena RTT-a, retransmisija, brza retransmisija), kontrolu toka te — najvažnije — kontrolu zagušenja.

Ishodi učenja

Po završetku poglavlja student će moći: opisati TCP segment te sekvencijske i ACK brojeve; objasniti uspostavu i zatvaranje veze; izvesti procjenu RTT-a i brzu retransmisiju; razlikovati kontrolu toka i kontrolu zagušenja; te objasniti slow start, AIMD i reakciju TCP-a na gubitak.

5.1 TCP: veza i segment

TCP je *spojni* protokol: prije prijenosa podataka strane uspostavljaju vezu **trosmjernim rukovanjem**. Klijent šalje segment sa zastavicom SYN i početnim sekvencijskim brojem; poslužitelj odgovara sa SYN+ACK; klijent potvrđuje ACK-om, nakon čega slijede podaci. Veza se zatvara izmjenom FIN i ACK segmenata u oba smjera.

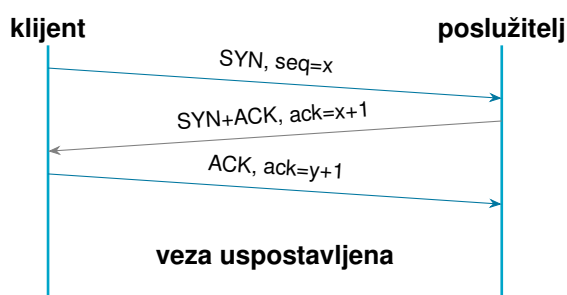


Figure 5.1: Trosmjerno rukovanje pri uspostavi TCP veze.

TCP segment sadrži izvorišni i odredišni port, **sekvencijski broj** (redni broj prvog bajta podataka u segmentu), **ACK broj** (broj *sljedećeg* očekivanog bajta), zastavice (SYN, ACK, FIN), **prozor primatelja** (*rwnd*) i kontrolni zbroj. Potvrde su *kumulativne*: ACK vrijednosti n znači da su primljeni svi bajtovi do $n - 1$.

Budući da su seq i ACK brojevi izraženi u *bajtovima*, a ne u paketima, TCP prirodno podržava tok bajtova proizvoljne veličine. Primjerice, ako host A pošalje segment sa seq = 42 koji nosi jedan bajt podatka, host B potvrđuje s ACK = 43 (očekujem bajt 43); istodobno B može u istom segmentu slati vlastite podatke (dvosmjernost). Izgubi li se segment, primatelj i

dalje šalje ACK s brojem *prvog* bajta koji nedostaje, pa uzastopni identični ACK-ovi otkrivaju prazninu.

5.2 Pouzdani prijenos

TCP pouzdanost temelji na *jednom* timeru za najstariji nepotvrđeni segment. **Procjena RTT-a** ključna je za odabir vremena isteka: iz uzoraka *SampleRTT* (kraće: *S*) računa se izgladena procjena *EstimatedRTT* (*E*) i njezino odstupanje *DevRTT* (*D*):

$$E \leftarrow (1 - \alpha) E + \alpha S, \quad (5.1)$$

$$D \leftarrow (1 - \beta) D + \beta |S - E|, \quad (5.2)$$

$$\text{Timeout} \leftarrow E + 4D, \quad (5.3)$$

uz tipične vrijednosti $\alpha = 0,125$ i $\beta = 0,25$. Timeout tako prati i *prosjeck* i *promjenjivost* kašnjenja.

Osim isteka timera, TCP koristi i **brzu retransmisiju**: kada primatelj primi segment izvan reda, ponavlja *isti* ACK (za prazninu). Primi li pošiljalatelj *tri duplikata* istog ACK-a, zaključuje da je segment izgubljen i ponovno ga šalje *prije* isteka timera, čime skraćuje oporavak.

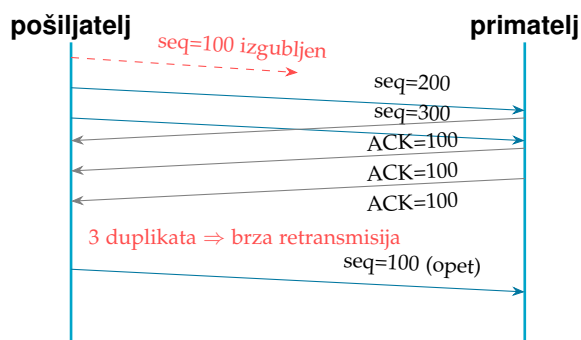


Figure 5.2: Brza retransmisija nakon tri duplikata ACK-a.

5.3 Kontrola toka

Kontrola toka sprječava da pošiljalatelj *preplavi* međuspremnik primatelja. Primatelj u svakom segmentu javlja *rwnd* — količinu slobodnog prostora u svom međuspremniku — a pošiljalatelj drži količinu nepotvrđenih podataka „u letu” manjom ili jednakom *rwnd*. Ako *rwnd* padne na nulu, pošiljalatelj povremeno šalje mali ispitni segment (*persist timer*) kako bi saznao kada se prostor oslobodio. Valja razlikovati: kontrola toka štiti *primatelja*, dok kontrola zagušenja (u nastavku) štiti *mrežu*.

5.4 Kontrola zagušenja

Zagušenje nastaje kada previše izvora šalje podatke prebrzo za kapacitet mreže; posljedice su rastući redovi u ruterima, veliko kašnjenje i gubici paketa (uz uzaludno trošenje kapaciteta na retransmisije). TCP koristi pristup *kraj-do-kraja*: mreža ne šalje eksplicitne signale (osim kod ECN-a), pa pošiljalatelj zagušenje *zaključuje* iz gubitaka.

Pošiljalatelj održava **prozor zagušenja** *cwnd* i drži podatke „u letu” ograničene na $\min(\text{cwnd}, \text{rwnd})$; propusnost je približno cwnd / RTT . Prozor se prilagođava u dvije faze. U **slow startu** *cwnd* počinje malen i *udvostručuje* se svaki RTT (eksponencijalni rast) dok ne dosegne prag *ssthresh*. Tada se prelazi u **izbjegavanje zagušenja** (*congestion avoidance*), gdje

cwnd raste *aditivno* (za jedan MSS po RTT-u). Pri gubitku prozor se *multiplikativno* smanjuje — to je pravilo **AIMD** (*additive increase, multiplicative decrease*), koje daje karakteristične „zube pile“.

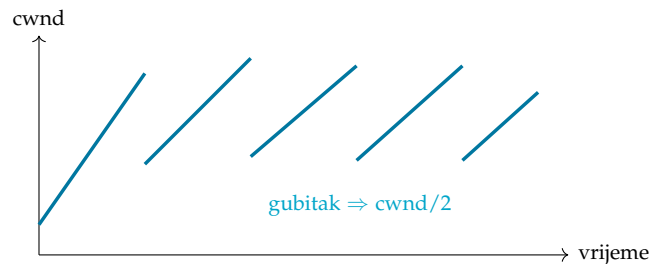


Figure 5.3: AIMD: aditivni rast i multiplikativno smanjenje prozora zagušenja.

Reakcija ovisi o *vrsti* znaka gubitka. Tri duplikata ACK-a (blaži znak, mreža još isporučuje pakete) vode u *fast recovery*: cwnd se prepolovi i nastavlja rasti aditivno. *Timeout* (ozbiljniji znak) vraća cwnd na 1 MSS, postavlja ssthresh na polovicu i vraća protokol u slow start. Više TCP tokova na istom grlu, zahvaljujući AIMD dinamici, teži *pravednoj* (približno jednakoj) raspodjeli propusnosti; UDP tokovi, koji ne surađuju, mogu narušiti tu ravnotežu. Moderne inačice (npr. **TCP CUBIC**) mijenjaju zakon rasta radi boljeg iskorištenja brzih linkova velikog RTT-a, a **ECN** omogućuje ruterima da *označe* pakete umjesto da ih odbace, signalizirajući zagušenje bez gubitka.

Primjer rasta prozora

Konkretno, u slow startu prozor može rasti kao $cwnd = 1, 2, 4, 8, 16$ (udvostručenje svaki RTT); po dosezanju praga (npr. $ssthresh = 16$) prelazi u aditivni rast $17, 18, 19, \dots$ sve do gubitka, kada se prepolovi. Uz periodičan gubitak, prosječna propusnost jednog toka približno je $\frac{3}{4} \cdot W_{max} / RTT$, gdje je W_{max} prozor neposredno prije gubitka. Tako TCP neprestano „pipa“ dostupni kapacitet i prilagođava mu se.

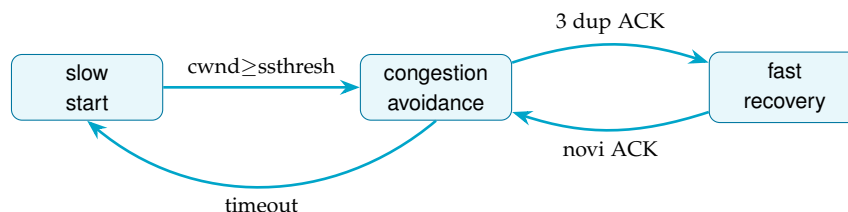


Figure 5.4: Pojednostavljeni stroj stanja TCP kontrole zagušenja.

Kontrola toka vs kontrola zagušenja

Iako oba mehanizma ograničavaju brzinu pošiljatelja, cilj im je različit. *Kontrola toka* (prozor $rwnd$) sprječava da pošiljatelj preplavi *međuspremnik primatelja* i tiče se samo dviju krajnjih točaka. *Kontrola zagušenja* (prozor $cwnd$) štiti *mrežu* u cjelini od preopterećenja. Pošiljatelj u svakom trenutku poštuje oba ograničenja: količina nepotvrđenih podataka „u letu“ ne smije premašiti $\min(cwnd, rwnd)$.

Sažetak poglavlja

TCP pruža pouzdan, uređen tok bajtova uz sekvencijske i kumulativne ACK brojeve. Veza se uspostavlja trosmjernim rukovanjem i zatvara FIN-om. Pouzdanost nose procjena RTT-a, timeout i brza retransmisija (tri duplikata ACK-a). *Kontrola toka* (rwnd) štiti primatelja, a *kontrola zagušenja* (cwnd) mrežu: slow start eksponencijalno, potom AIMD; timeout vraća u slow start, a tri duplikata u fast recovery. AIMD vodi pravednosti, a CUBIC i ECN suvremena su poboljšanja.