
Transportni sloj I: UDP i pouzdani prijenos

Poglavlje 4 — Skripta kolegija *Računalne mreže*

doc. dr. sc. Sandi Baressi Šegota

Fakultet informatike Sveučilišta Jurja Dobrile u Puli

Poglavlje 4

Transportni sloj I: UDP i pouzdani prijenos

Transportni sloj pruža *logičku komunikaciju između procesa* koji se izvode na različitim hostovima. Za razliku od mrežnog sloja, koji povezuje *hostove*, transportni sloj povezuje *proces* i djeluje isključivo na krajnjim uređajima. U ovom poglavlju upoznajemo multipleksiranje i demultipleksiranje, protokol **UDP**, te postupnu izgradnju **pouzdanog prijenosa** nad nepouzdanim kanalom, zaključno s protokolima s cjevovodom **Go-Back-N** i **Selective Repeat**.

Ishodi učenja

Po završetku poglavlja student će moći: objasniti (de)multipleksiranje i razliku između UDP i TCP demultipleksiranja; opisati UDP segment i kontrolni zbroj; izvesti mehanizme pouzdanog prijenosa (ACK, sekvencijski brojevi, timeout); te usporediti Go-Back-N i Selective Repeat.

4.1 Transportne usluge i (de)multipleksiranje

Pošiljatelj transportni sloj poruke aplikacije pretvara u *segmente* dodajući im zaglavlje; primatelj ih sloj rastavlja i predaje odgovarajućem procesu. Taj se posao naziva **multipleksiranje** (skupljanje podataka iz više socketa i njihovo označavanje) i **demultipleksiranje** (predaja pristiglog segmenta ispravnom socketu). Ključ za demultipleksiranje su *brojevi portova* u zaglavlju.

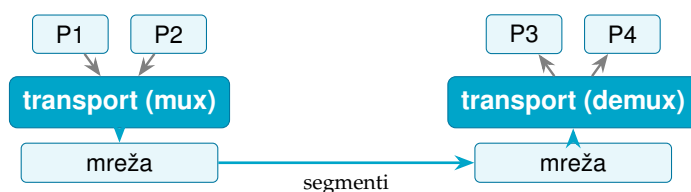


Figure 4.1: Multipleksiranje kod pošiljatelja i demultipleksiranje kod primatelja.

Razlika između protokola je bitna. Kod **UDP-a** socket je određen *dvojema* vrijednostima — odredišnom IP adresom i odredišnim portom — pa svi segmenti s istim odredišnim portom idu istom socketu. Kod **TCP-a** socket je određen *četirima* vrijednostima (izvorišna i odredišna IP adresa te izvorišni i odredišni port), pa poslužitelj može istovremeno razlikovati mnoge veze koje dolaze na isti port.

Primjerice, web-poslužitelj sluša na portu 80. Kada mu se istodobno spoje dva različita klijenta, oba segmenta imaju odredišni port 80, no razlikuju se po izvorišnoj IP adresi i/ili izvorišnom portu; zahvaljujući četvorci, TCP svakom klijentu dodjeljuje zaseban socket i veze se ne miješaju. Kod UDP-a takva razlika ne postoji jer se za demultipleksiranje gleda samo odredište.

4.2 UDP — prijenos bez veze

UDP (*User Datagram Protocol*) je jednostavan transportni protokol *bez veze*: nema rukovanja, nema kontrole toka ni zagušenja. Nudi uslugu „best-effort” — segmenti se mogu izgubiti ili stići izvan redoslijeda. Zauzvrat je brz i ima mali režijski trošak (zaglavlje od svega 8 bajtova: izvorišni i odredišni port, duljina i kontrolni zbroj), pa ga koriste DNS, DHCP, streaming, mrežne igre i QUIC (temelj HTTP/3).

Kontrolni zbroj (checksum) služi za otkrivanje pogrešaka u prijenosu: pošiljalac zbroji 16-bitne riječi segmenta (uz kružni prijenos) i u zaglavlje upiše *komplement* tog zbroja. Primalac zbroji sve riječi zajedno s kontrolnim zbrojem; ako je rezultat „sve jedinice”, poruka je vjerojatno neoštećena. Checksum *otkriva*, ali ne *ispravlja* pogreške.

4.3 Pouzdani prijenos podataka (rdt)

Cilj je izgraditi *pouzdan* kanal nad *nepouzdanim*. Rješenje gradimo postupno, dodajući mehanizme kako uvodimo nove vrste kvarova.

rdt1.0

pretpostavlja savršen kanal — nisu potrebne potvrde ni ponavljanja.

rdt2.0

uvodi *bit-pogreške*: primalac šalje potvrdu ACK (ispravno) ili NAK (oštećeno), a pošiljalac na NAK ponavlja paket (mehanizam ARQ).

rdt2.1

rješava problem *oštećenog ACK/NAK-a* uvođenjem *sekvencijskih brojeva* (0 i 1), čime primalac prepoznaje i odbacuje duplikate.

rdt2.2

postiže isto *bez NAK-a*: primalac potvrđuje ACK-om s brojem zadnjeg ispravno primljenog paketa; dva ista ACK-a djeluju kao NAK.

rdt3.0

uvodi *gubitak paketa*: pošiljalac koristi *timeout* i po isteku ponavlja paket („protokol izmjeničnog bita”).

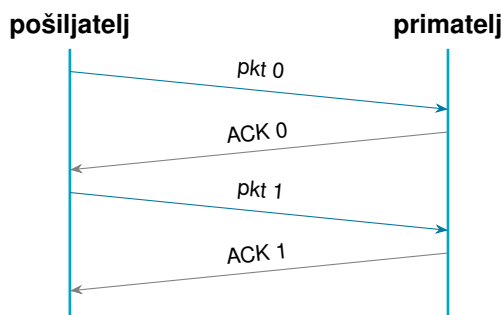


Figure 4.2: Rad protokola „stani-i-čekaj”: jedan paket po vremenu povratnog puta.

Gradivni mehanizmi pouzdanog prijenosa

Kontrolni zbroj otkriva oštećenja; *potvrde* (ACK) javljaju uspjeh; *sekvencijski brojevi* razlikuju nove pakete od duplikata; *timer* otkriva gubitak i pokreće ponavljanje; a kod cjevovodnih protokola *prozor* ograničava broj istovremeno nepotvrđenih paketa. Kombinacijom tih pet elemenata gradi se pouzdana usluga nad nepouzdanim kanalom.

4.3.1 Performanse i motivacija za cjevovod

Protokol „stani-i-čekaj” šalje samo jedan paket po vremenu povratnog puta, pa je iskoristivost linka

$$U = \frac{L/R}{RTT + L/R}. \quad (4.1)$$

Za $L = 8000$ bita, $R = 1$ Gb/s i $RTT = 30$ ms dobiva se $U \approx 0,00027$, dakle svega oko 0,03%. Rješenje je **cjevovod** (*pipelining*): slati više paketa bez čekanja pojedinačnih potvrda, čime se iskoristivost povećava razmjerno veličini prozora (do zasićenja linka). Cjevovod zahtijeva širi raspon sekvencijskih brojeva i međuspremnik.

4.4 Protokoli s cjevovodom: Go-Back-N i Selective Repeat

Postoje dva klasična pristupa. **Go-Back-N** (GBN) dopušta do N nepotvrđenih paketa; primatelj prihvaća samo paket *koji je na redu*, a sve izvan redoslijeda odbacuje i ponavlja zadnji *kumulativni* ACK. Pošiljalac ima jedan timer za najstariji nepotvrđeni paket; po isteku ponovno šalje *sve* pakete od njega nadalje.



Figure 4.3: GBN: klizni prozor. Tamno = potvrđeno, svijetlo = poslano i čeka ACK, prazno = dostupno.

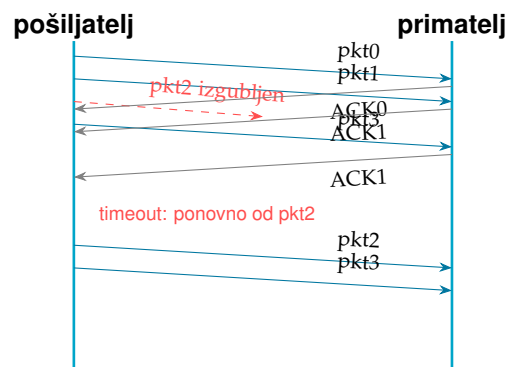


Figure 4.4: Go-Back-N pri gubitku: primatelj odbacuje pakete izvan reda, a pošiljalac po isteku timera ponavlja cijeli niz od izgubljenog paketa.

Selective Repeat (SR) potvrđuje *svaki* ispravno primljeni paket zasebno i *međuspremljuje* pakete izvan redoslijeda; pošiljalac održava zaseban timer za svaki paket i po isteku ponavlja *samo* taj paket. SR je učinkovitiji kad su gubici češći, ali je složeniji. Važno ograničenje jest da veličina prozora kod SR-a mora biti najviše *polovica* raspona sekvencijskih brojeva; inače primatelj ne može razlikovati nov paket od ponovljenoga.

