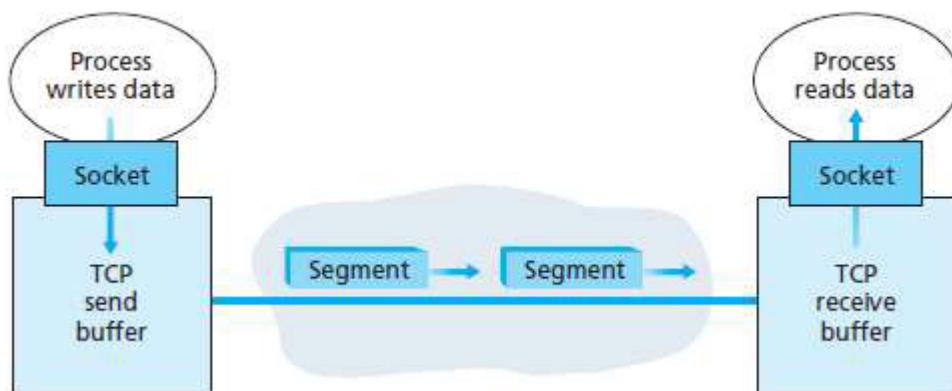


Vežba 5 – TCP

Teorijske osnove

TCP predstavlja protokol transportnog nivoa. Ovaj protokol omogućava istovremeno dvosmernu pouzdanu komunikaciju između klijenta i servera. Komunikacija je realizovana u vidu konekcije koja se uspostavlja pomoću metode rukovanja (eng. *Handshaking*). Iz tog razloga ne podržava multicast.



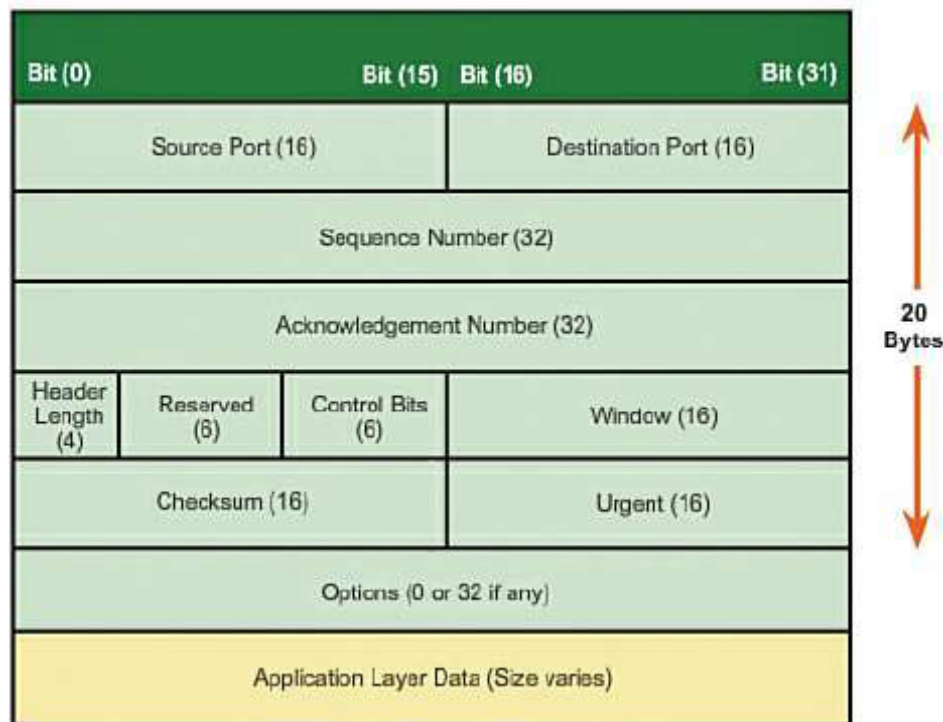
Slika 1. Segmentacija

TCP koristi **segment** za jedinicu prenosa (jedinice podataka transportnog sloja - TPDU). Proces podele originalne poruke aplikativnog nivoa na segmente naziva se **segmentacija**. TCP omogućava praćenje poslatih paketa, kontrolu toka komunikacije, kao i redosleda pristiglih segmenata.

TCP definiše uslugu pouzdane isporuke toka (engl. *stream*) korisničkih podataka. Osobine TCP-a su:

- Obavlja kontrolu toka podataka, TCP obezbeđuje komunikaciju sistema različitih brzina.
- Osnovna jedinica prenosa TCP-a je segment podataka. Segmenti se koriste za prenos upravljačke informacije (npr poruke za uspostavu i raskid veze), ili za prenos podataka.

- Format segmenta je izabran tako da je moguće potvrđivanje podataka iz jednog smera, njihovim uključivanjem u zaglavlje segmenata koji se šalju u drugom smeru.
- Kontrola toka je realizovana tako što prijemnik oglašava količinu podataka koju je spreman da primi.
- TCP takođe podržava poruke van opsega (engl. “*out of band*”), koje služe za slanje urgentnih podataka i za forsiranje isporuke korišćenjem (engl. “*push*”) podataka.



Slika 2. Izgled TCP segmenta

Na slici 2. je dat prikaz TCP segmenta, koji se sastoji iz zaglavlja i dela u kome se nalaze podaci. Zaglavlje se sastoji iz:

- **Source port** - izvorišni port (port pošiljaoca). **Port ima ulogu u identifikaciji aplikacije.**
- **Destination port** - odredišni port (port primaoca).
- **Sequence number** - broj prvog bajta segmenta u okviru toka podataka. Broj sekvence omogućava praćenje toka podataka. Inicijalni broj sekvence se bira nasumično, kako bi se otklonila mogućnost interferencije između različitih konekcija.
 - $\text{syn}=1$ - Inicijalni broj sekvence.
 - $\text{syn}=0$ - Akumulirani broj sekvence.
- **Acknowledgement number** - broj sekvence narednog segmenta koji se očekuje. Segmenti koji pristignu van redosleda, u zavisnosti od implementacije, mogu se odbaciti ili čuvati.
- **Header Length** - dužina zaglavlja.

- **Reserved** - rezervisano za buduću upotrebu.
- **Control bits** - ukazuje na funkciju i namenu segmenta.
 - **ack** - ukazuje na validnost vrednosti potvrde (acknowledgment-a).
 - **syn, rst i fin** - omogućavaju uspostavljanje i prekid konekcije.
 - **psh** - ukazuje prijemnoj strani da se momentalno pošalju podaci višem sloju.
 - **urg** - ukazuje na postojanje urgentnih podataka u segmentu.
- **Window** - maksimalan broj bajtova koje je moguće poslati, a da prethodno nije potvrđen njihov prijem.
- **Checksum** - koristi se za proveru da li se desila greška poruke nad zaglavljem i podacima prilikom prenosa.
- **Urgent** - lokacija poslednjeg bajta koji je markiran kao urgentan.
- **Options** - dodatne opcije.
- **Application data** - podaci viših slojeva.

Enkapsulacija segmenta

TCP segment se enkapsulira u IP datagram, koji se dalje enkapsulira u okvir nivoa veze, kao što je prikazano na slici 3.



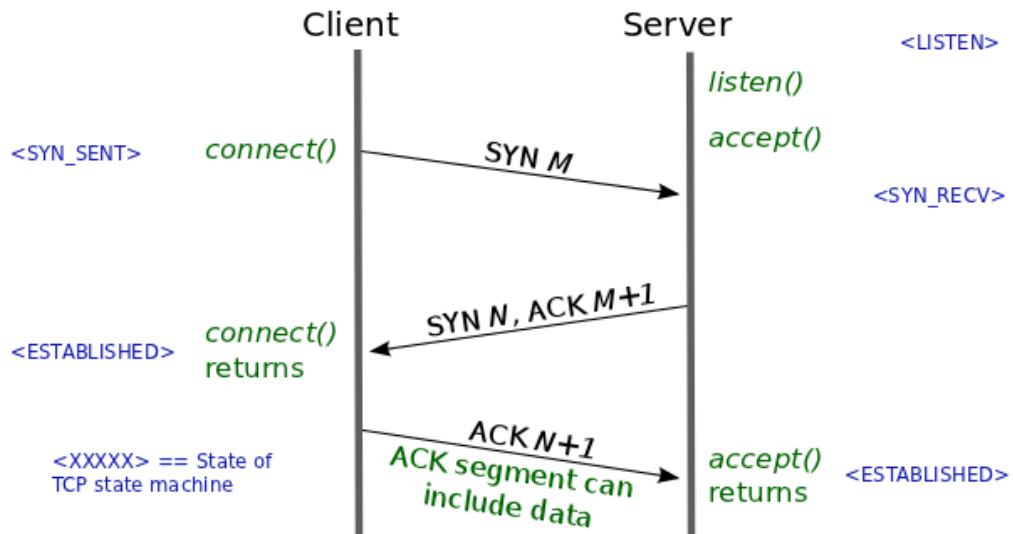
Slika 3. Enkapsulacija TCP segmenta

Uspostavljanje veze

Komunikacija klijenta i servera preko TCP protokola zahteva uspostavu veze koja se ostvaruje tako što se između predajne i prijemne strane iz tri puta razmene poruke sa podešenih odgovarajućim kontrolnim bitima (*TCP three-way handshake*):

- Predajna strana A šalje poruku sa podešenim kontrolnim bitom $SYN = 1$, ostali su podešeni na 0, pri čemu nasumično odabere redni broj segmenta (SEQ_a).
- Prijemna strana B odgovara porukom sa kontrolnim bitima SYN i $ACK = 1$, takođe nasumično bira broj segmenta (SEQ_b), a za ACK uzima broj $ACK = (SEQ_a) + 1$. Na ovaj način je uspostavljena veza na liniji od predajne ka prijemnoj strani.
- Slanjem poruke sa podešenim kontrolnim bitom $SYN = 1$ od prijemne strane, ona zahteva da predajna strana potvrdi uspostavljanje veze od prijemne ka predajnoj strani. Predajna strana to čini slanjem poruke sa podešenim kontrolnim bitom $ACK = 1$ i

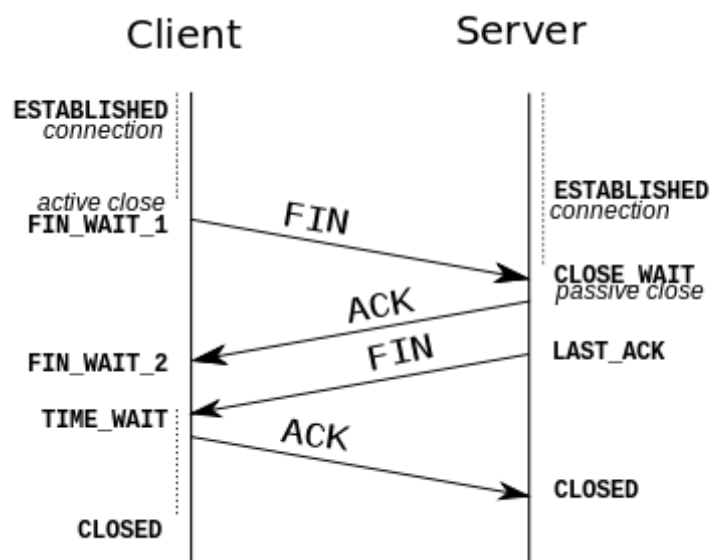
uzima vrednost ACK broja $ACK = (SEQ_b) + 1$. Na ovaj način je uspostavljena konekcija između klijenta i servera.



Slika 4. TCP uspostava veze

Prekid veze

1. Klijent inicira prekid klijent-server konekcije
 - $FIN=1$
2. Server potvrđuje zahtev za prekid klijent-server konekcije
 - $ACK=1$
3. Server šalje zahtev za prekid server-klijent konekcije
 - $FIN=1$
4. Klijent odgovara na zahtev za prekid server-klijent konekcije
 - $ACK=1$



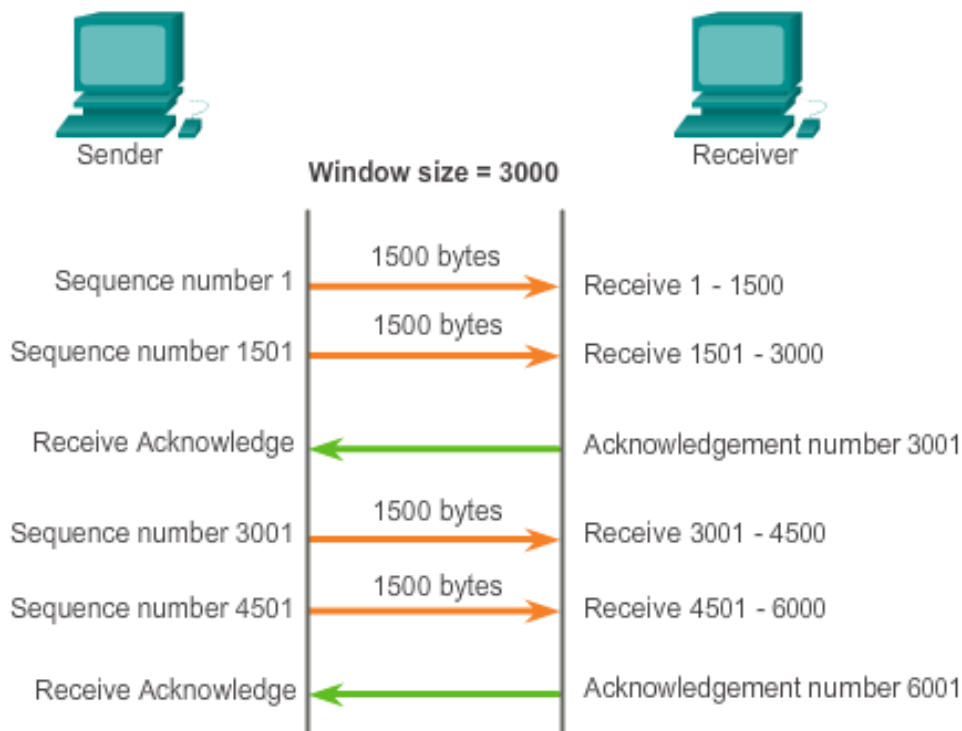
Slika 5. TCP prekid veze

Praćenje isporuke segmenata

TCP protokol omogućava mehanizam za praćenje isporuke paketa, tako što primaoc potvrđuje svaki pristigli paket. Kako bi se zaobišlo čekanje prilikom slanja svakog novog pojedinačnog segmenta, omogućeno je slanje više segmenata odjedanput. U poruci potvrde za se navodi broj primljenih bajtova po redosledu uvećan za 1. Na primer, ukoliko su segmenti sa sekvencama 1500-3000 i 3400-3500 primljeni, vrednost ACK će biti 3001.

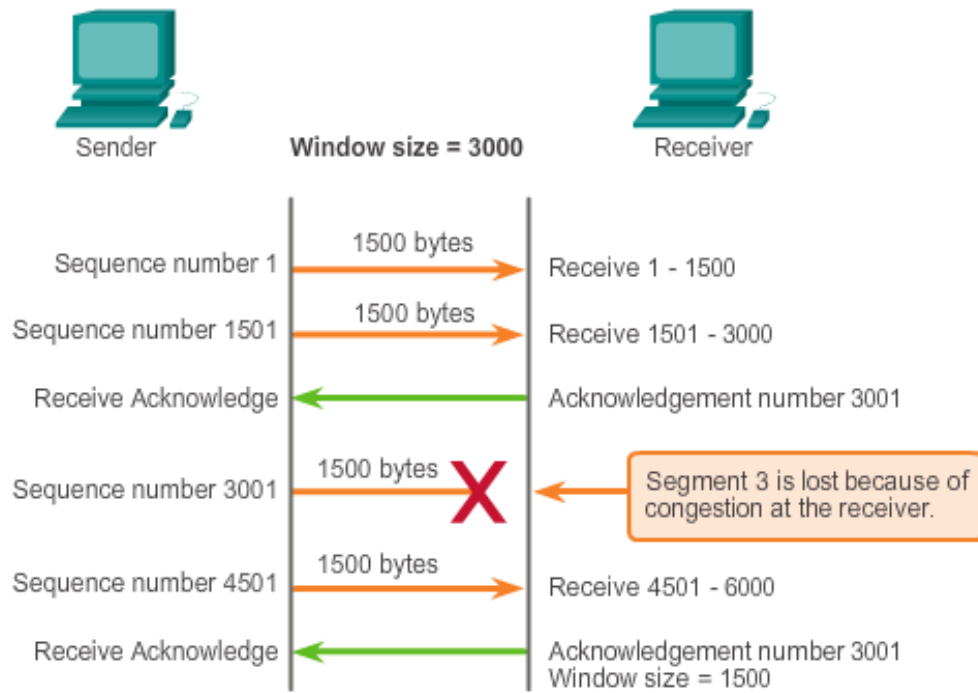
Praćenje toka podataka

Praćenje toka realizovano je definisanjem veličine prozora slanja (eng. *window size*), tj maksimalnim brojem bajtova koje je moguće poslati, a da prethodno nije potvrđen njihov prijem. Segmenti pristigli van redosleda se baferuju.



Slika 6. TCP praćenje toka podataka

Primaoc može zatražiti promenu veličine prozora. To je korisno u slučajevima kada se u baferu nalaze segmenti sa višim brojem sekvence, a da je segment sa nižim izostao. U tom slučaju pošiljaoc će bez čekanja moći poslati samo segmente koji nedostaju.



Slika 7. TCP veličina prozora