

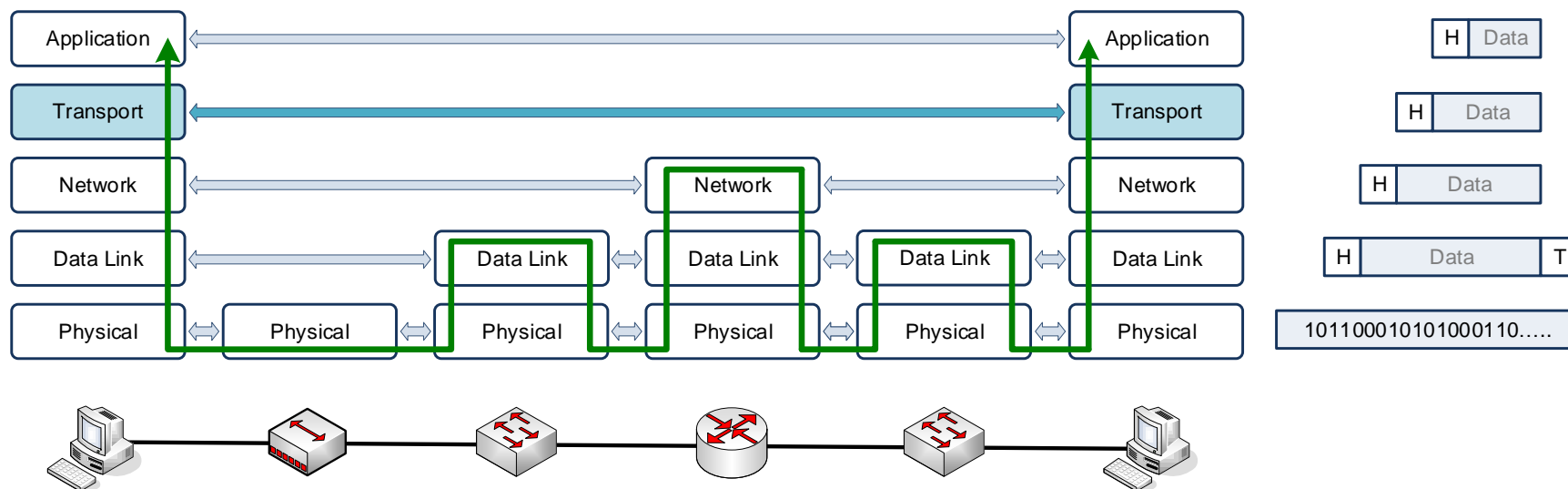
Transportni sloj

Prof. dr Slavko Gajin

2020. god

Transportni sloj

- Transportni sloj (*Layer 4*)
 - Po horizontali – komunikacija *s-kraja-na-kraj* između dva uređaja
 - Po vertikali - veza između aplikacija i mreže
- Zadatak
 - Održavanje višestrukih komunikacija između aplikacija na obe strane



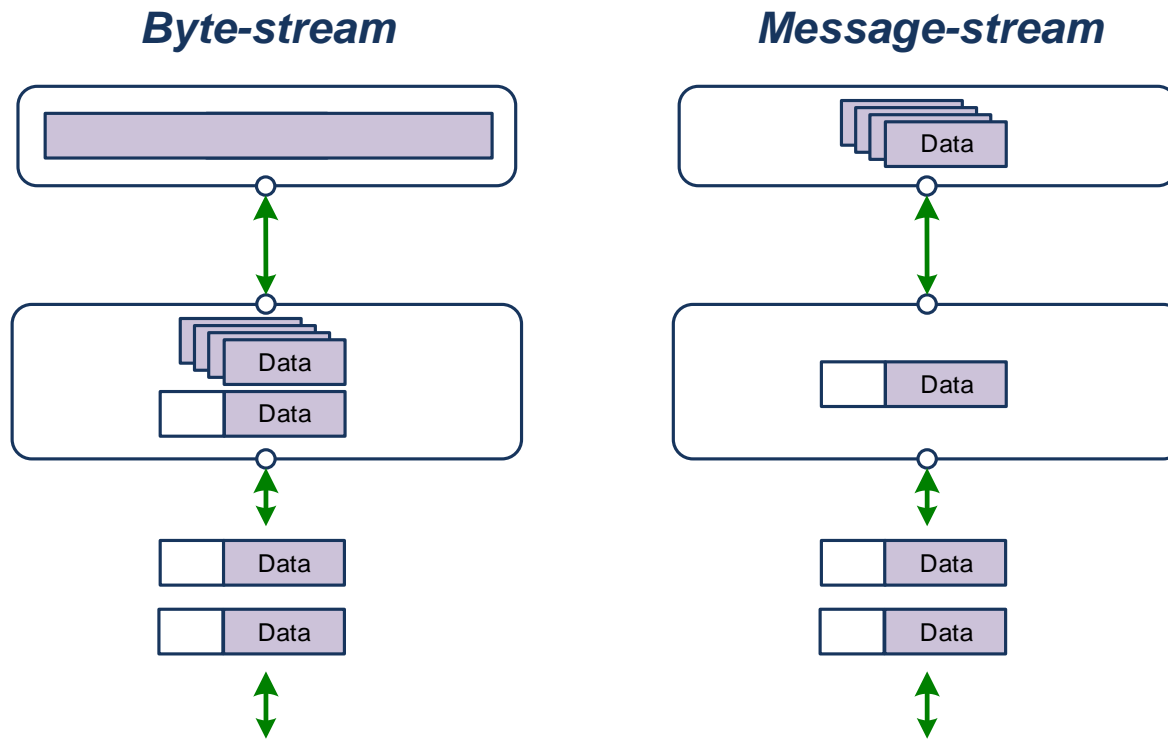
Osnovne funkcije

- **Transport aplikativnih podataka**

- Enkapsulacija i dekapulacija – dodavanje i uklanjanje zaglavlja 4. nivoa

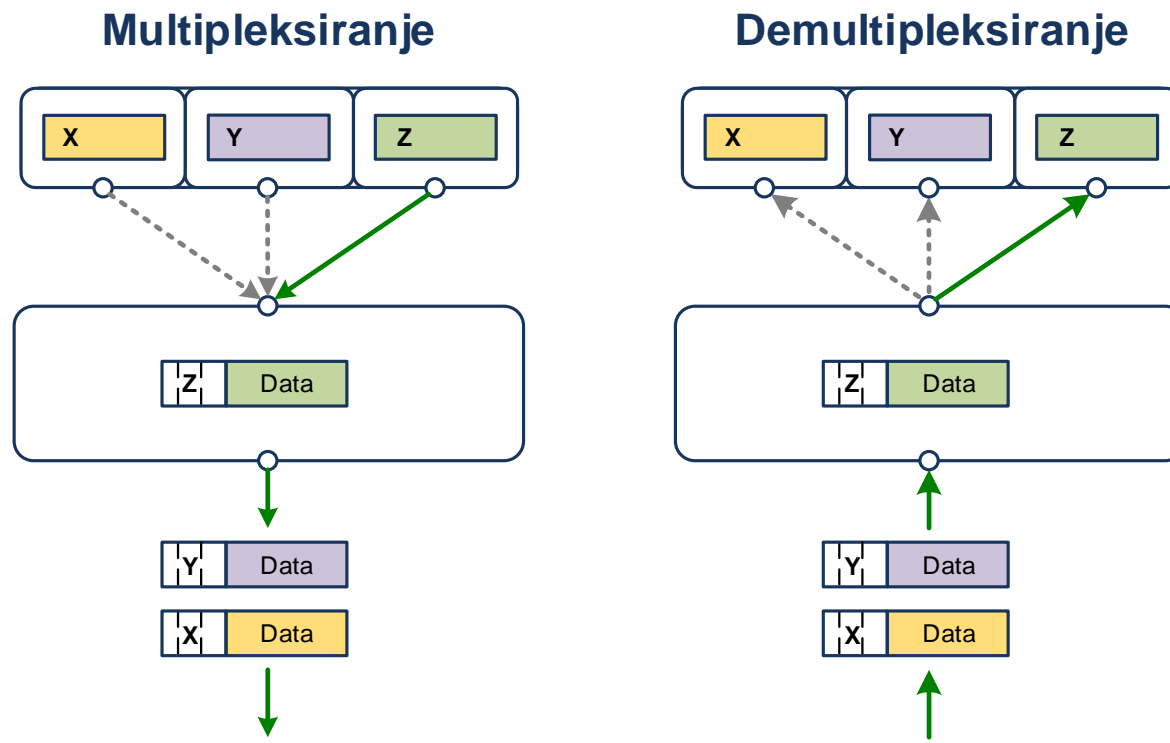
- Dve vrste prenosa

- *Byte-stream* – segmentacija niza bajtova proizvoljne dužine
- *Message-stream* – prenos i enkapsulacija celokupnih poruka



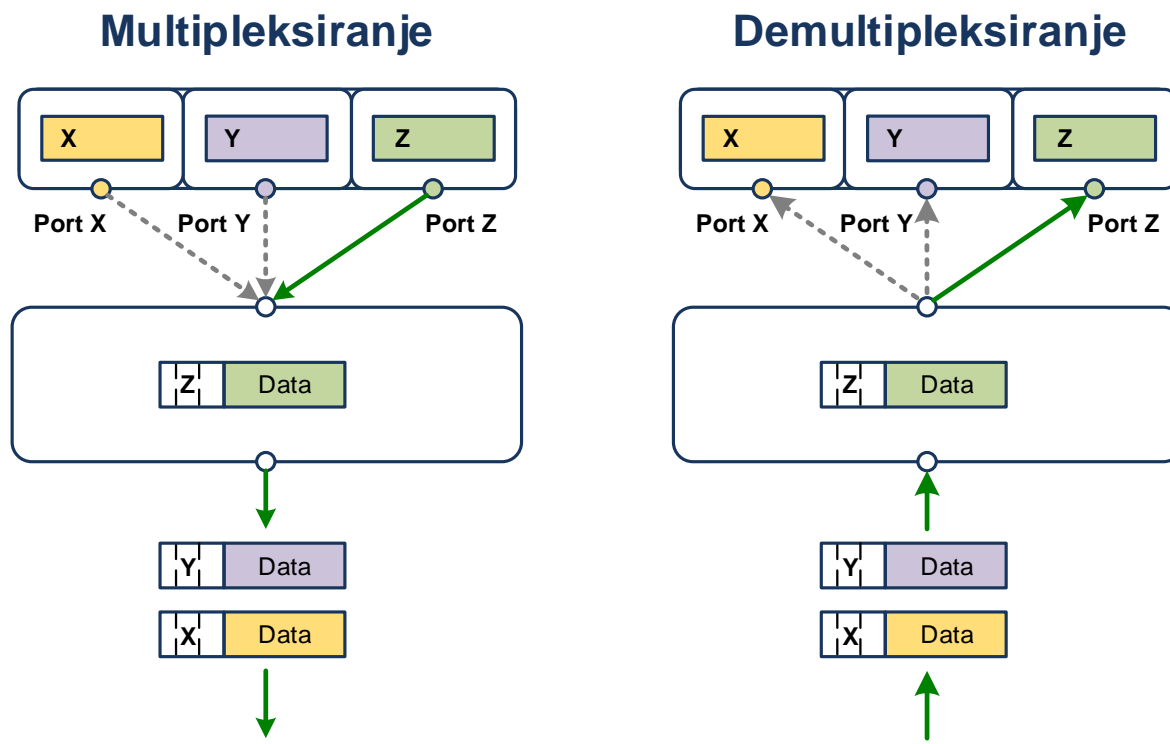
Osnovne funkcije

- **Multipleksiranje i demultipleksiranje aplikativnih podataka**
 - Na strani pošiljaoca
 - Obeležavanje aplikacije prilikom preuzimanja podataka
 - Na strani primaoca
 - Prepoznavanje označene aplikacije i prosleđivanje podataka



Port – identifikacija aplikacije na uređaju

- Multipleksiranje i demultipleksiranje komunikacija
 - polje *Type* u zaglavlju na 2. nivou identifikuje protokole 3. nivoa
 - polje *Protocol* u zaglavlju na 3. nivou identifikuje protokole 4. nivoa
- **Port** - identifikacija aplikacija na jednom uređaju
 - polje *Port* u zaglavlju na 4. nivou identifikuje aplikaciju (ili njen deo)

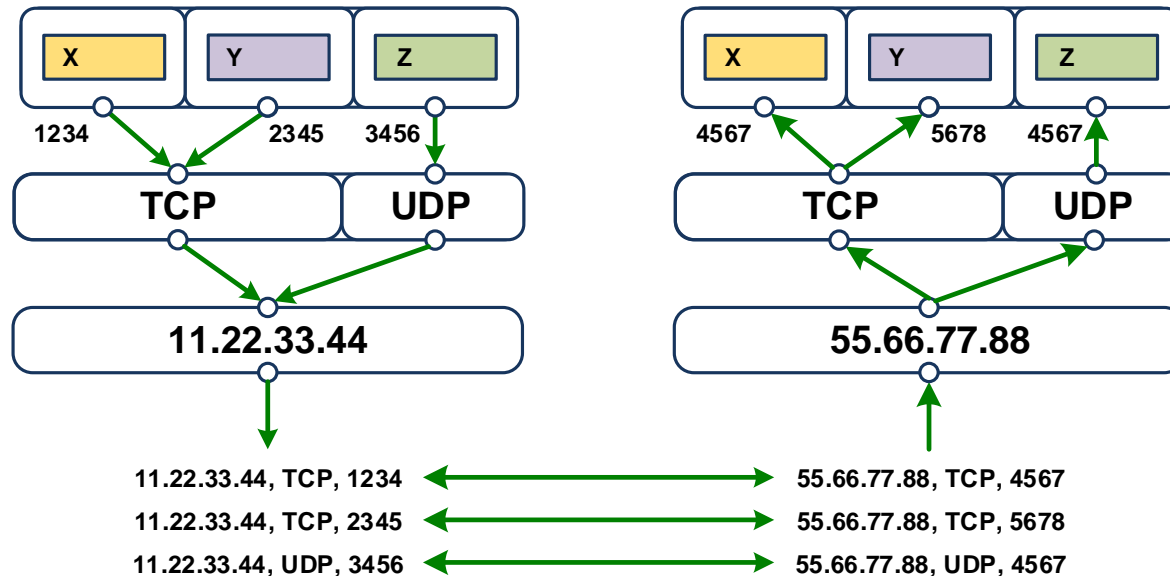


Port – identifikacija aplikacije na uređaju

- IANA – dodeljuje fiksne portove za pojedinačne aplikacije
- Vrste portova (opsezi brojeva za portove različitih namene)
 - **Well-known Ports:** 0-1023 – samo za serverske aplikacije
 - **Registered Ports:** 1024-49151 – za klijentske i serverske aplikacije
 - **Private (Dynamic) Ports:** 49152-65535 – samo za klijentske aplikacije
- Primeri:
 - TCP
 - 21 FTP (*File Transport Protocol*)
 - 22 SSH
 - 23 Telnet
 - 25 SMTP (*Simple Mail Transport Protocol* - slanje email-a)
 - 110 POP3 (prijem email-a)
 - 80 HTTP (*web*)
 - 443 HTTPS (*secure HTTP*) ...
 - UDP
 - 53 DNS (*Domain Name System*)
 - 69 TFTP (*Trivial File Transport Protocol*)
 - 161 SNMP (*Simple Network Management Protocol*)
 - 5004 RTP (*Real Time Protocol*) ...

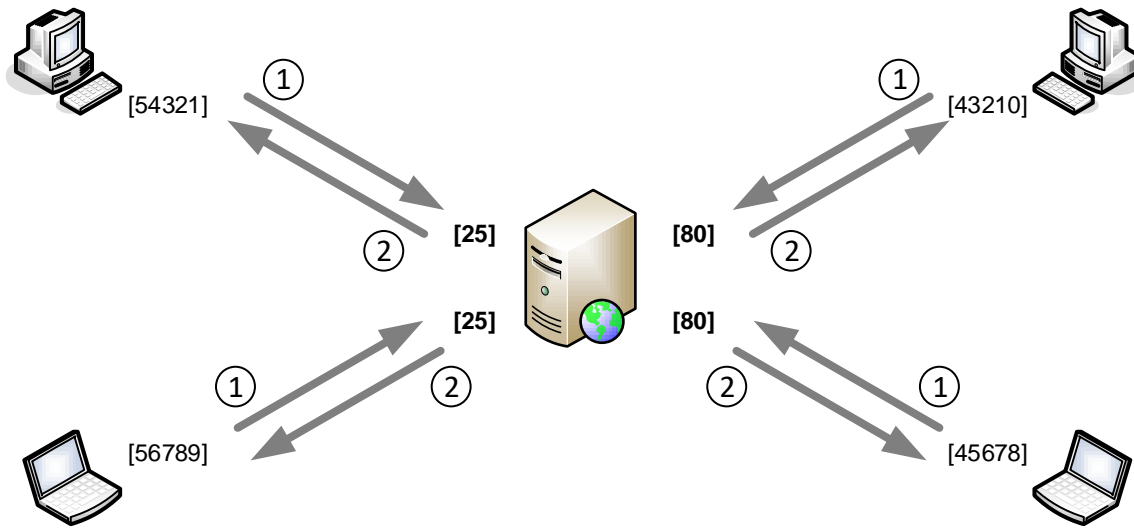
Soket – identifikacija aplikacije na mreži

- *Socket* (soket)
 - Jednoznačno identifikovanje aplikaciju na mreži (ili deo aplikacije koji ostvaruje komunikaciju)
 - Identifikacija „ulaza“ u aplikaciju pri komunikaciji
 - Sadrži:
 - IP adresu
 - Identifikaciju transportnog protokola (TCP ili UDP)
 - Broj porta



Klijent–Server komunikacija

- Serverske aplikacije
 - Aplikacije dostupne (otvorene) za pristup od strane proizvoljnih korisnika
 - Soket: unapred poznata IP adresa i poznat TCP ili UDP port
- Klijentske aplikacije
 - Aplikacije na strani korisnika koje iniciraju komunikaciju sa serverskim aplikacijama
 - Soket: proizvoljna IP adresa i dinamički dodeljen TCP ili UDP port
- Dvosmerna komunikacija između klijentskih i serverskih soketa
 - Zahtev od klijenta prema serveru
 - Odgovor servera prema klijentu



Dodatne funkcije

- **Uspostavljanje veze** – *Connection oriented*
 - Uspostavljanje i održavanje komunikacione sesije
- **Pouzdan prenos** – *Reliable delivery*
 - Garantuje se pouzdan prenos svih aplikativnih podatak u celini
 - Izgubljeni ili oštećeni segmenti se detektuju i ponovo šalju
- **Održavanje redosleda segmenata** – *Ordered data reconstruction*
 - Po različitim putevima segmenti mogu stići u promenjenom redosledu, ali prijemna strana rekonstruiše originalni redosled
- **Kontrola toka** – *Flow control*
 - Dinamičko povećanje i smanjenje protoka podataka
 - Upravljanje prenosom podataka u zavisnosti od mogućnosti i trenutnog opterećenja mreže (smanjenje brzine u slučaju zagušenja)
- Dodatne funkcije su korisne za većinu aplikacija, ali dodatno opterećuju i usporavaju prenos podataka

TCP vs. UDP

Dve osnovne vrste transportnog protokola u TCP/IP modelu

- **UDP – *User Datagram Protocol***

- Obezbeđuje samo osnovne funkcije
- *Connectionless* protokol
 - Jednostavan, nema garancije pouzdanosti prenosa i redosleda
 - Malo posla – malo opterećenja uređaja

- **TCP – *Transmission Control Protocol***

- Obezbeđuje i osnovne i dodatne funkcije
- *Connection oriented* protokol
 - Uspostavljanje veze
 - Pouzdan prenos
 - Održavanje redosleda prijema podataka
 - Kontrola toka
- Više posla - veće opterećenje uređaja

Koji protokol transportnog nivoa koriste aplikacija za video konferencije (Teams, ZOOM, viber...)?

TCP

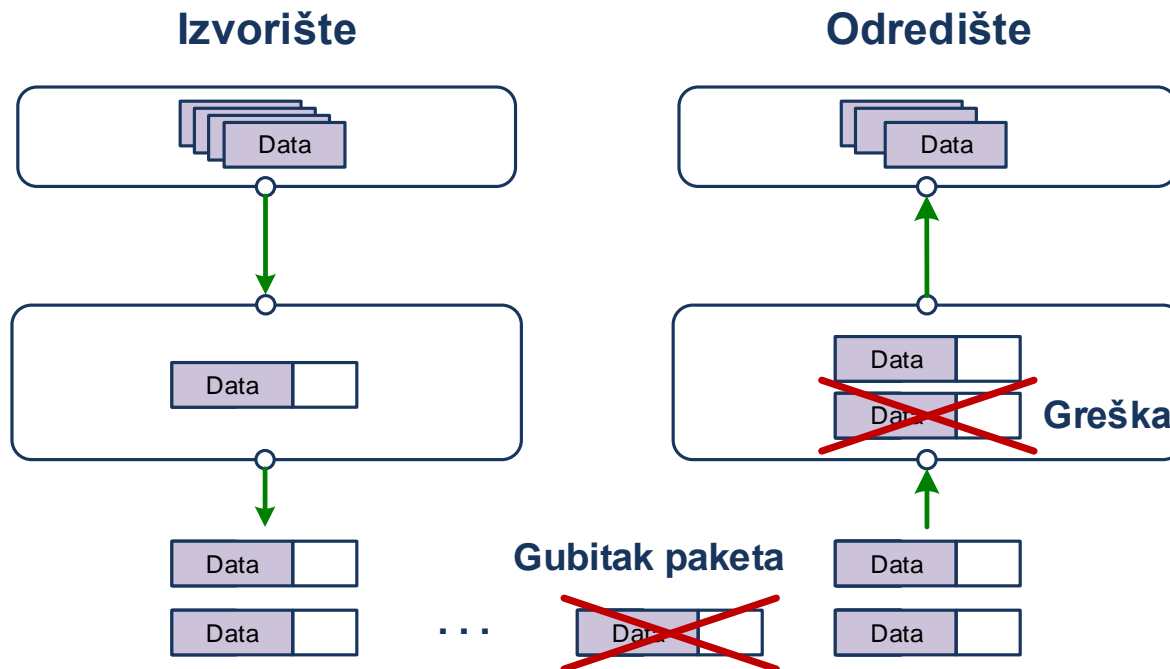
UDP

Nešto treće?

Ne koriste protokol transportnog nivoa.

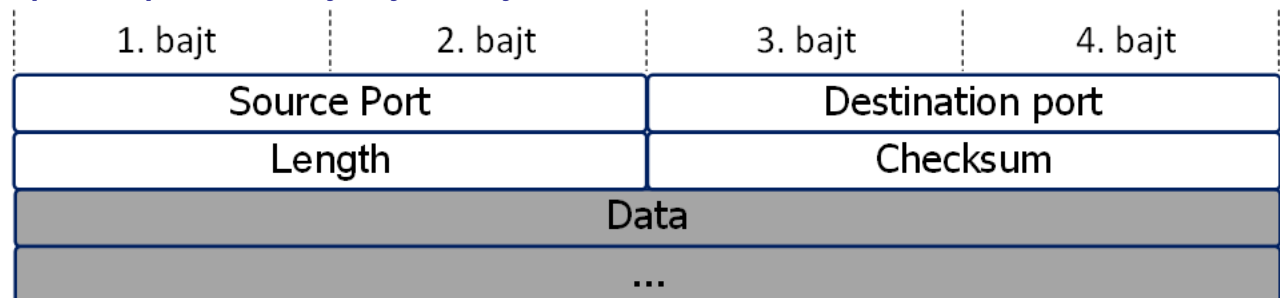
UDP protokol

- **UDP – User Datagram Protocol, RFC 768**
- *Connectionless message-stream* protokol
 - Prenose se celokupne poruke koja se dobijaju od aplikacije – *datagram*
 - Enkapsulacija, obeležavanje porta i prenos bez uspostavljanja veze
 - Nezavistan prenos svakog paketa
 - Nepouzdan, ali jednostavan i brz



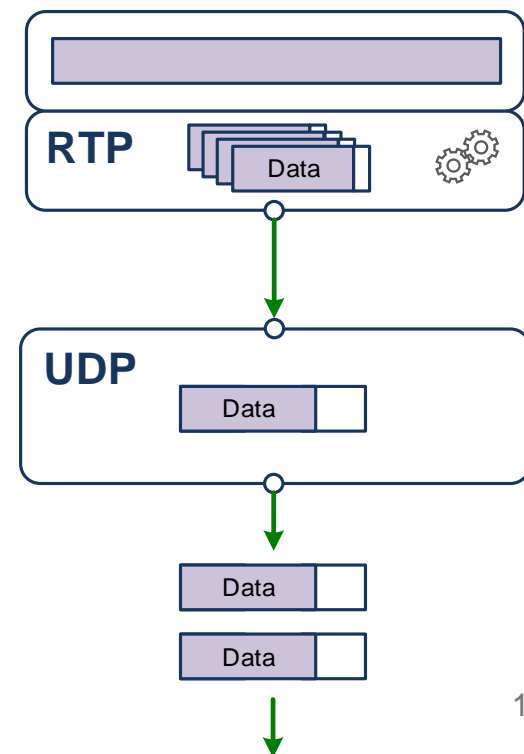
Zaglavlje UDP paketa

- **Source port, Destination port** - izvorišni i odredišni port
 - Po 2 bajta, vrednosti od 0 do 65535
 - IP adresa se prenosi u zaglavlju IP paketa
- **Length** – dužina podataka uključujući i zaglavlje
 - Maksimalna dužina podataka: $65535 - 8 - 20 = 65507$ B
 - Umanjeno za 8B UDP zaglavlja i 20B IP zaglavlja
- **Checksum** – provera greške, 2 bajta
 - Prvi komplement sume UDP zaglavlja, podataka i pseudo-hedera
 - Pseudo-header – izvorišna i odredišna IP adresa, identifikacija UDP protokola (17) i dužina UDP paketa (*Length*)
 - Provera ispravnosti izvorišne IP adrese
 - Donekle se krši princip razdvajanja slojeva



Primena UDP protokola

- Jednostavne aplikacije, periodična komunikacija, kada nije bitna pouzdanost
- *Real-time* aplikacije (IPTV, IP telefonije, video konferencije...)
 - Bitan je kontinuitet pristizanja poruka, čak iako se ne neka poruka izgubi
 - Malo kašnjenje (*delay*) – do 200 ms za interaktivan razgovor
 - Mala varijacija kašnjenja - džiter (*jitter*) +/- 30 ms
- Kada je potreban brodkast ili multikast
- Složenije funkcije se prepuštaju aplikaciji, samostalno ili korišćenjem podslojeva
 - *RTP* – *Real-time Transport Protocol*
 - Dodatna polja: *timestamp*, *sequence number*...
 - Serijalizacija, baferovanje, kontrola džitera...

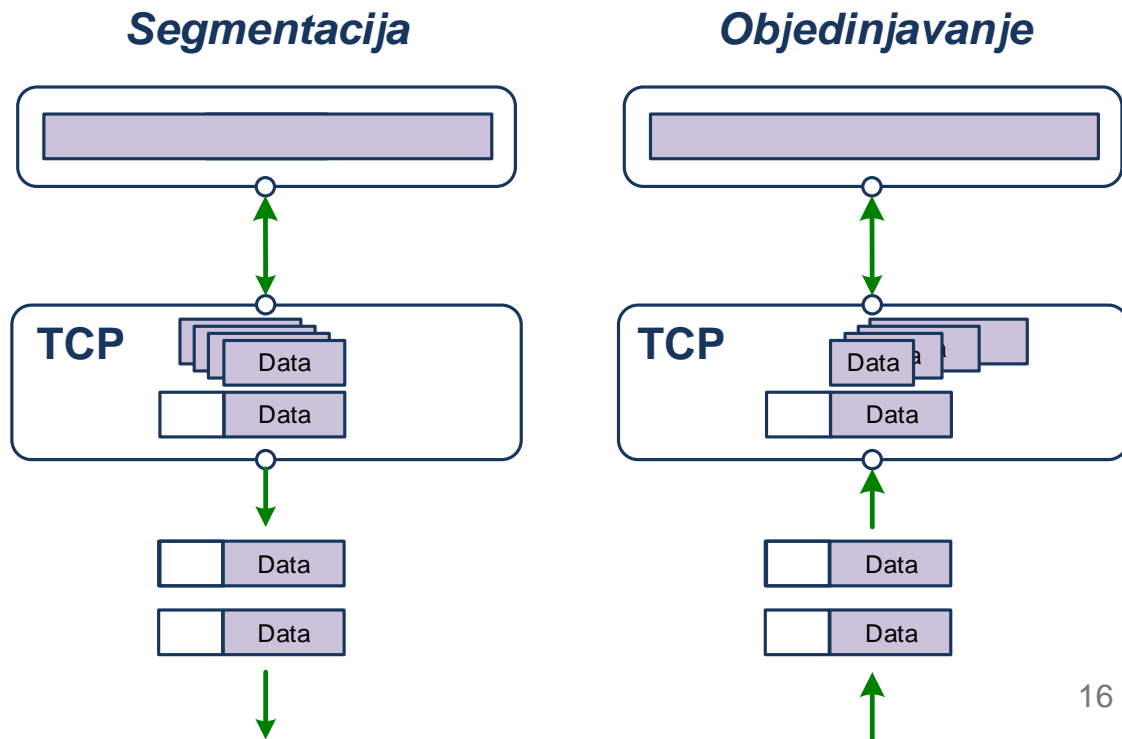


TCP protokol

- **TCP – *Transmission Control Protocol***, RFC 761 (793, 7805)
- *Connection-oriented byte-stream* protokol
 - Inicijalno uspostavljanje veze sa odredištem
 - Segmentacija niza bajtova koji se dobijaju od aplikacije
 - Enkapsulacija, obeležavanje porta
 - Pouzdan prenosa podataka, nezavisno u oba smeru
- Osobine
 - *Point-to-Point*
 - Unikast komunikacija uvek između dva uređaja (ne podržava se brodkast ili multikast)
 - *Full-Duplex*
 - Nezavisna komunikaciju o dva smeru
 - Čak i kada se aplikativni podaci prenose samo u jednom smeru, u drugom smeru se prenose kontrolni podaci

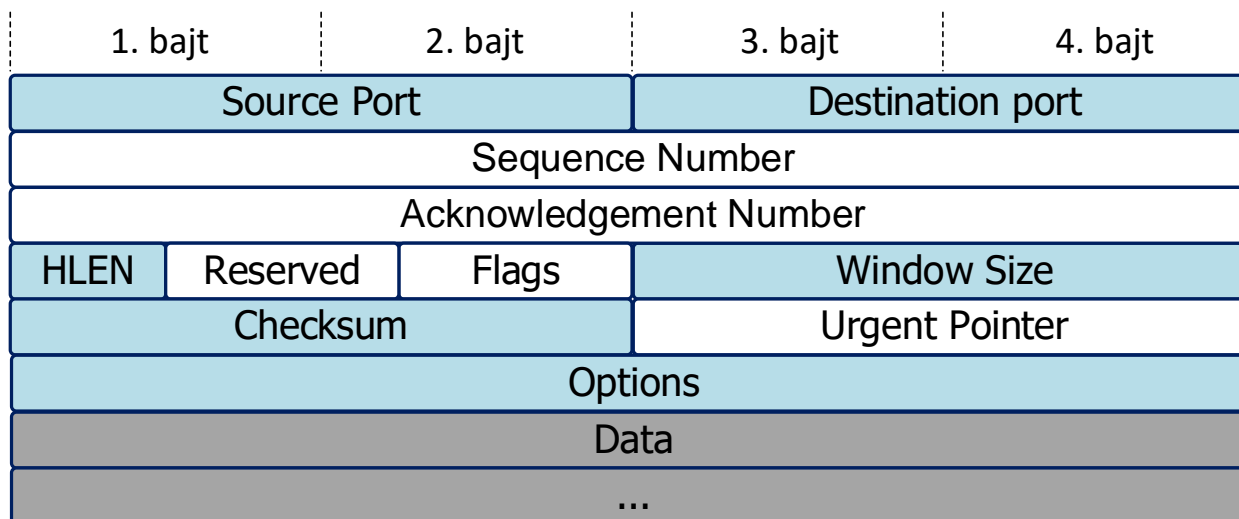
Byte-stream prenos podataka

- **Segmentacija** (*segmentation*) – na strani pošiljaoca
 - Aplikativni sloj predaje niz bajtova, podeljeno u proizvoljne delove
 - Segmentacija – podela većih celina (ili spajanje manjih) u tzv. segmente
 - MMS – *Maximum Segment Size*
 - Difoltna vrednost 536 B, definiše se pri uspostavljanju veze
- **Objedinjavanje** (*reassembling*)
 - Na strani primaoca
 - Rekonstrukcija originalnog niza bajtova aplikativnih podataka



TCP format zaglavlja

- **Source port, Destination port, Checksum**
 - Kao kod UDP protokola
- **HLEN (Header Length)** – 4 bita
 - Dužina zaglavlja u jedinicama od 4 B (32 bita)
- **Window Size** – ukupan broj bajtova koji se mogu poslati pre nego što se čeka na potvrdu
- **Options** – različite opcije, varijabilna dužina



Numeracija bajtova i segmenta

- **Sequence Number (SEQ)**

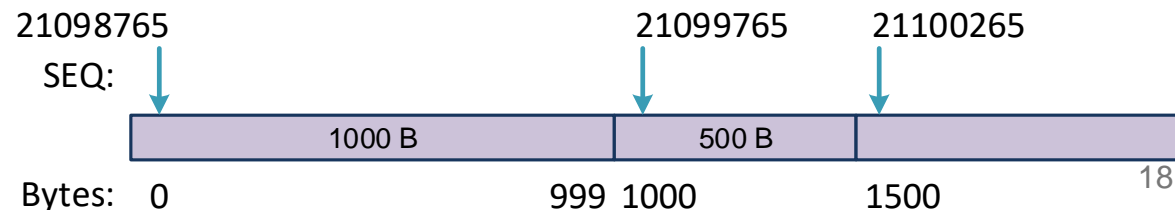
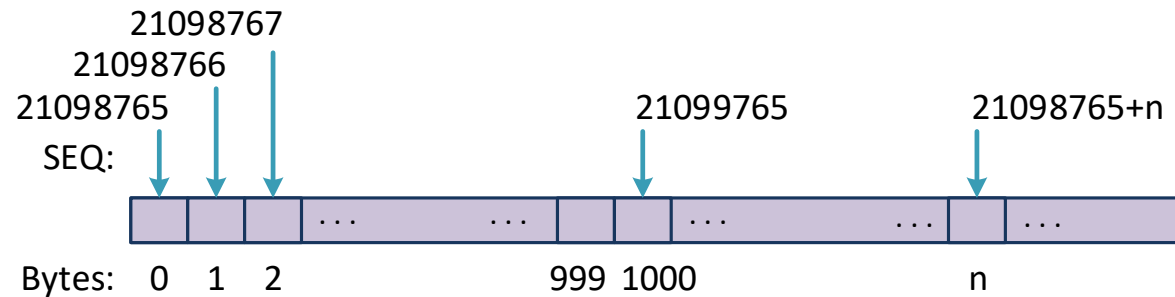
- Inicijalna vrednost - slučajno izabran broj u fazi uspostavljanja veze
- Svaki pojedinačni bajt u nizu aplikativnih podataka ima svoj redni broj relativno u odnosu na inicijalni SEQ

1. bajt	2. bajt	3. bajt	4. bajt
Source Port		Destination port	
Sequence Number			
Acknowledgement Number			
HLEN	Reserved	Flags	Window Size
Checksum		Urgent Pointer	
Options			
Data			
...			

- Relativna numeracija bajtova

- Obezbeđuje identifikaciju segmenata:

- Održavanje redosleda segmenata na prijemu
- Pouzdan prenos segmenata – potvrda prijema segmenta – *Acknowledgement*

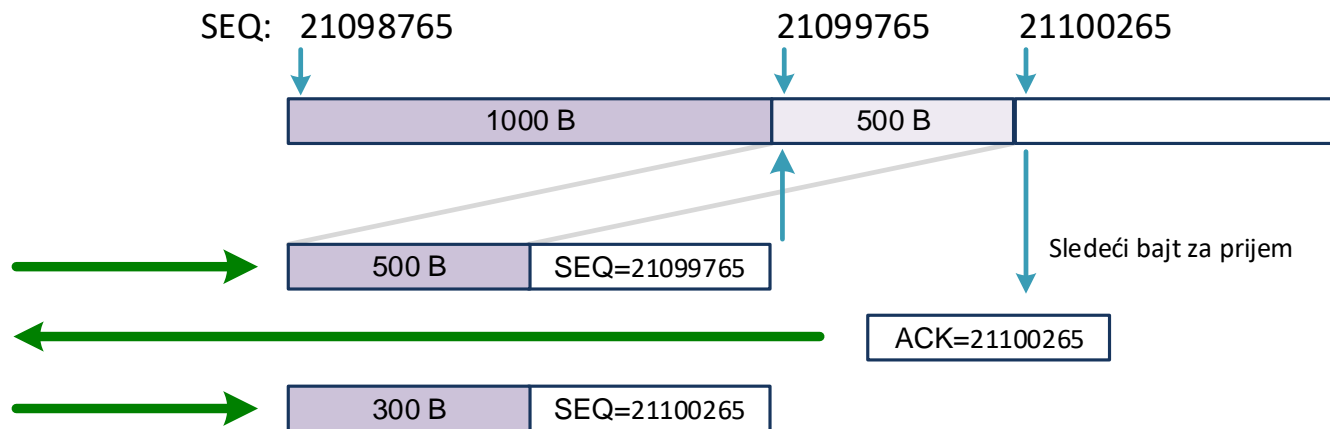


Potvrda uspešnog prijema

- **Acknowledgement Number (ACK)**

- Potvrda prijema kontinualnog niza bajtova
- Predstavlja relativni redni broj (SEQ) sledećeg bajta koji se očekuje za prijem
- Značenje
 - “Ovo je pozicija sledećeg bajta koji se očekuje za prijem, a svi prethodni su uspešno primljeni”

1. bajt		2. bajt		3. bajt		4. bajt	
Source Port				Destination port			
Sequence Number							
Acknowledgement Number							
HLEN	Reserved	Flags		Window Size			
Checksum				Urgent Pointer			
Options							
Data							
...							



Kontrolni flegovi

- **Flags**

- Kontrolni flegovi koji opisuju značenje paketa i drugih polja u zaglavlju

- Vrste flegova:

- **SYN**

- Inicijalizacija *Sequence Number* vrednosti (*Synchronization*)

- **ACK**

- Polje *Acknowledgment Number* je validno (*Acknowledgment*)

- **FIN**

- Poslednji segment, završetak konekcije u jednom smeru (*Finish*)

- **PSH**

- Zahteva se momentalna predaja segmenta aplikaciji na prijemu, bez baferovanja i čekanja ostalih segmenata (*Push*)

- **URG**

- Polje *Urgent Pointer* je validno (*Urgent*)

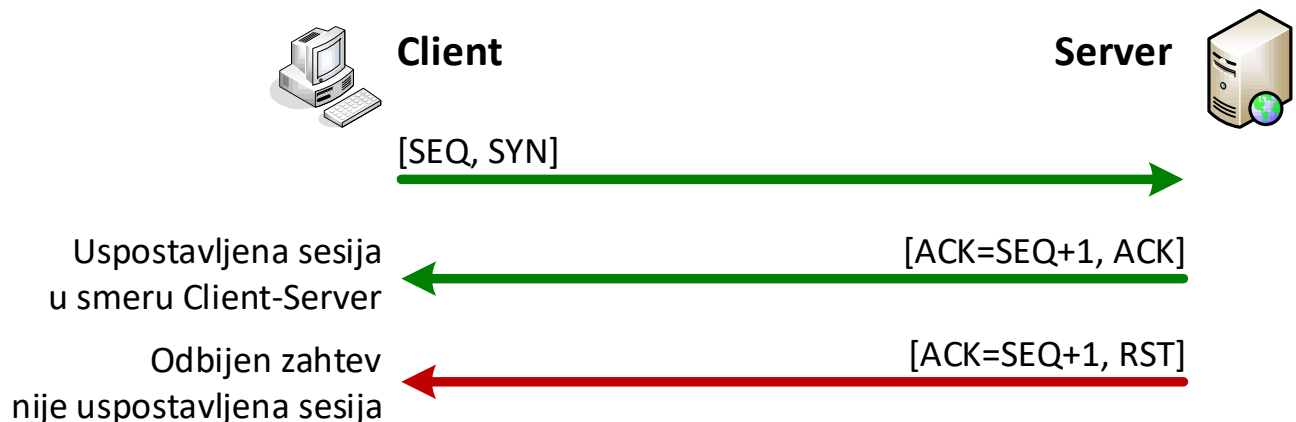
- **RST**

- Resetovanje konekcije (*Reset*)

1. bajt		2. bajt		3. bajt		4. bajt	
Source Port				Destination port			
Sequence Number							
Acknowledgement Number							
HLEN	Reserved	Flags		Window Size			
Checksum				Urgent Pointer			
Options							
Data							
...							

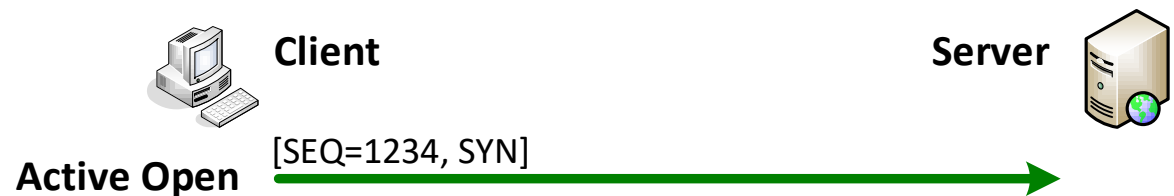
Uspostavljanje sesije

- TCP – dvosmerna komunikacija (*full-duplex*)
 - Dve odvojene komunikacione sesije, u oba smera po jedna
 - Dva nezavisna para *Sequence Number* i *Acknowledgement Number*
 - Uspostavljanje, održavanje i raskidanje obe komunikacione sesije
- Uspostavljanje TCP sesije - nezavisno u oba smera
 - Zahtev – Postavljanje slučajne SEQ vrednosti, postavljanje SYN flega
 - Odgovor
 - Potvrda prijema ($ACK=SEQ+1$), postavljen ACK fleg (bez podataka, 0 bajtova)
 - Odbijanje zahteva, postavljen RST fleg (npr. ne postoji aplikacija sa naznačenim portom)



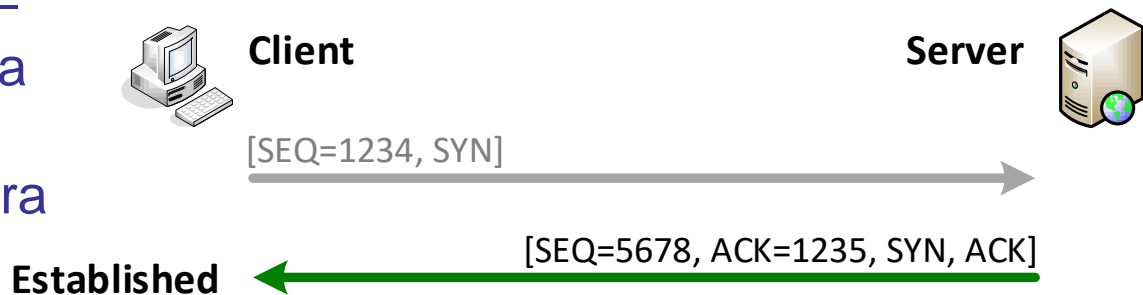
Uspostavljanje sesije

- Uspostavljanje TCP sesije u oba smera – u 3 koraka (*Three-way handshake*)
- **1. korak**
 - Zahtev za otvaranje sesije u jednom smeru, tzv. „sinhronizacija“ (od klijenta do servera)
 - Slanje inicijalne vrednosti SEQ
 - Postavljan SYN fleg
 - *Active Open* stanje



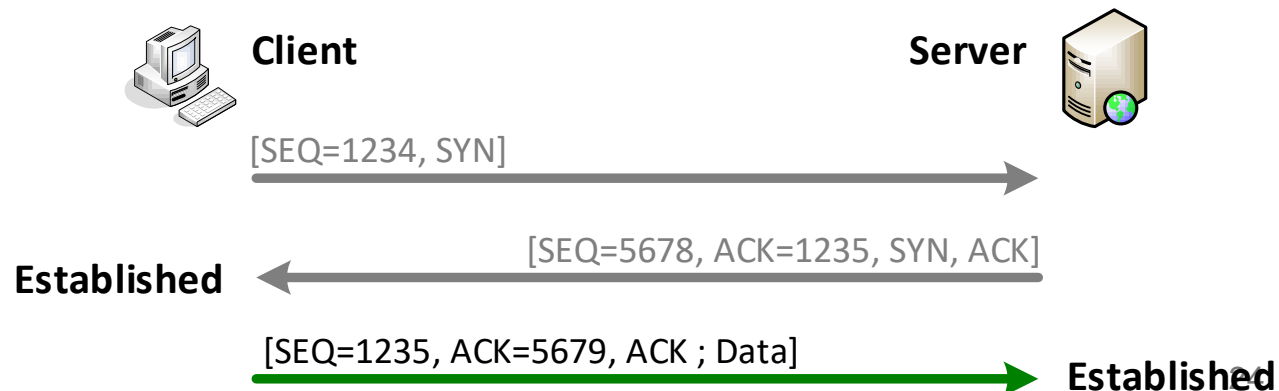
Uspostavljanje sesije

- Uspostavljanje TCP sesije u oba smera – u 3 koraka (*Three-way handshake*)
- **2. korak**
 - Potvrda otvaranja sesije u prvom smeru (od klijenta do servera)
 - ACK vrednost za jedan veća od SEQ vrednosti
 - Postavljan ACK fleg
 - Zahtev za otvaranje sesije u drugom smeru, tzv. „sinhronizacija“ (od servera do klijenta)
 - Slanje inicijalne vrednosti SEQ
 - Postavljan SYN fleg
 - *Established* stanje – Uspostavljena sesija u smeru od klijenta do servera



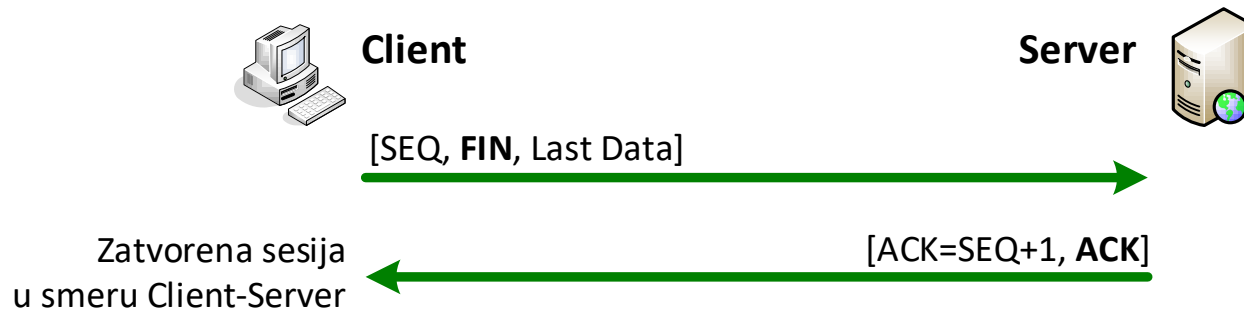
Uspostavljanje sesije

- Uspostavljanje TCP sesije u oba smera – u 3 koraka (*Three-way handshake*)
- **3. korak**
 - Potvrda otvaranja sesije u drugom smeru (od PC-2 do PC-1)
 - ACK vrednost za jedan veća od SEQ vrednosti
 - Postavljan ACK fleg
 - PC-1 šalje podatke koji počinju od SEQ=1235
 - Uspostavljena sesija u oba smera
- TCP soket – ulaz u aplikaciju samo za uparenu stranu
 - Može se posmatrati i kao 5-torka: (srcIP, srcPort, dstIP, dstPort, TCP)



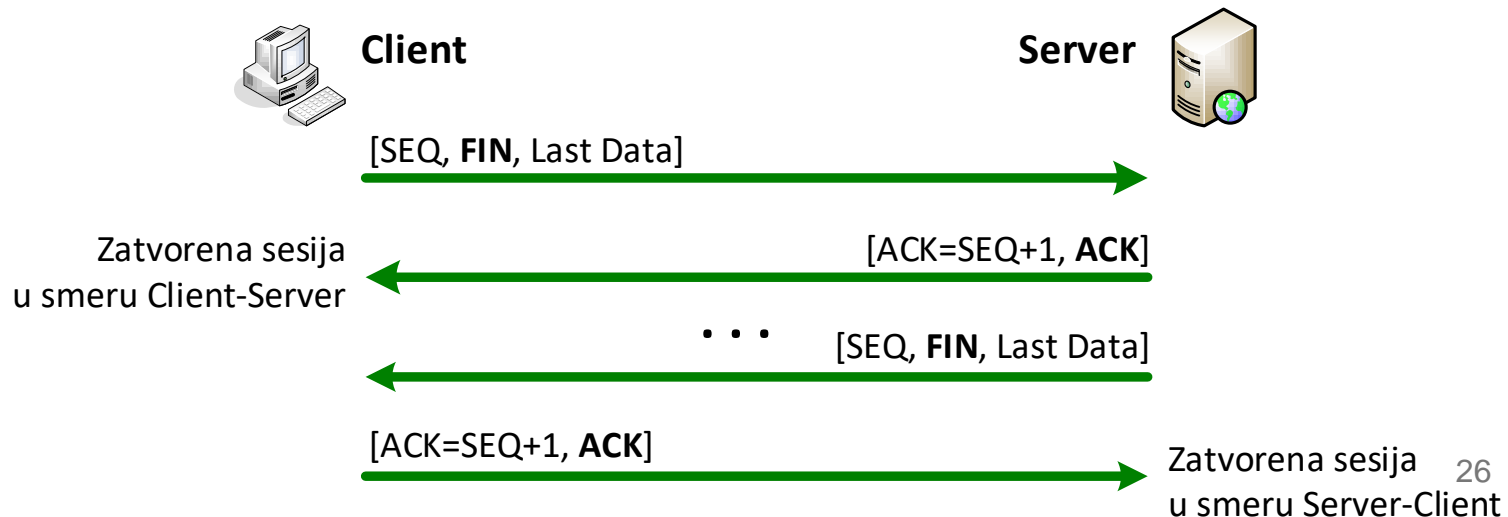
Raskidanje sesije

- Raskidanje TCP sesije u jednom smeru – u 2 koraka (*Two-way handshake*)
 - **1. korak**
 - Kada jedna strana nema više podataka za slanje, šalje se FIN fleg
 - **2. korak**
 - Druga strana potvrđuje prijem poslednjeg paketa slanjem ACK za taj paket
 - Sesija u tom smeru je zatvorena



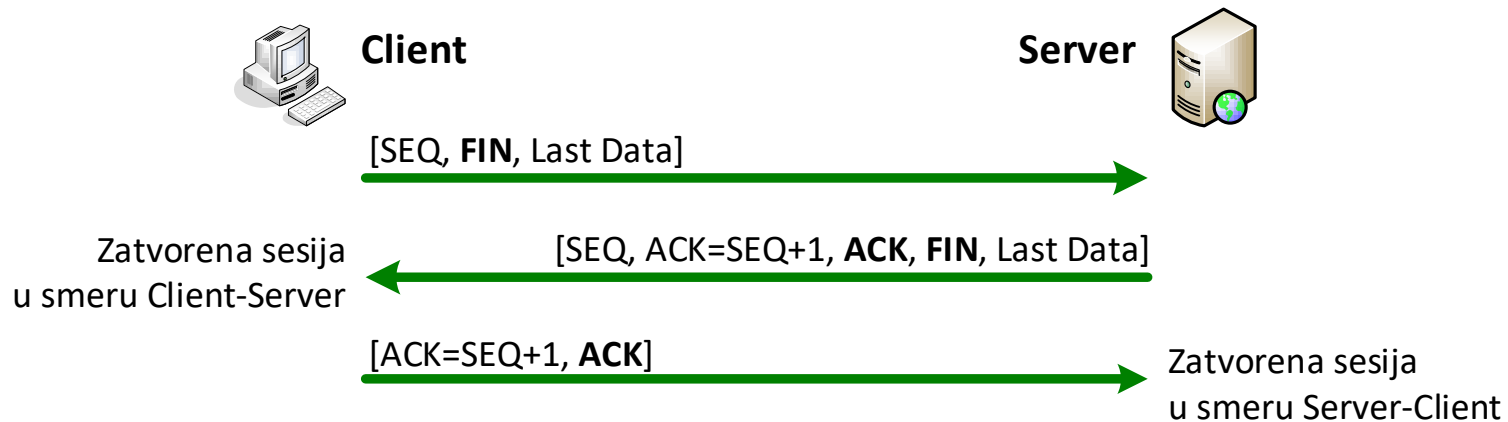
Raskidanje sesije

- Raskidanje TCP sesije u oba smera – u 4 koraka (2x *Two-way handshake*)
 - **1. i 2. korak**
 - Raskidanje sesije u jednom smeru
 - Ako druga strana ima još podataka za slanje, oni se šalju u dugom smeru
 - **3. korak**
 - Kada nema više podataka za slanje, šalje se FIN fleg
 - **4. korak**
 - Potvrđuje se prijem poslednjeg paketa slanjem ACK



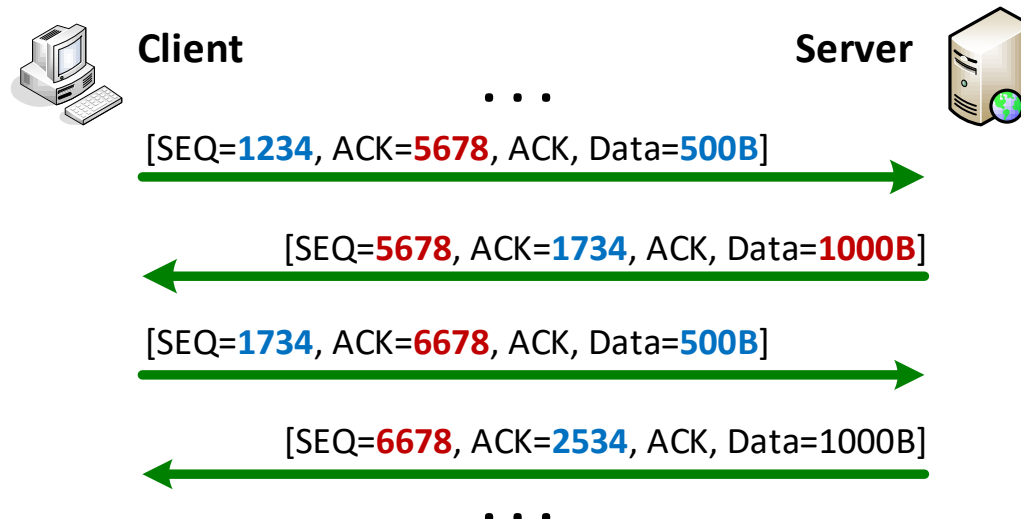
Raskidanje sesije

- Raskidanje TCP sesije u oba smera – u 3 koraka
 - **1. korak**
 - Kada jedna strana nema više podataka za slanje, šalje se FIN fleg
 - **2. korak**
 - Druga strana potvrđuje prijem poslednjeg segmenta slanjem ACK
 - Ako druga strana nema podataka za slanje ili ih može poslati u poslednjem paketu, šalje se FIN fleg
 - **3. korak**
 - Potvrđuje se prijem poslednjeg paketa slanjem ACK



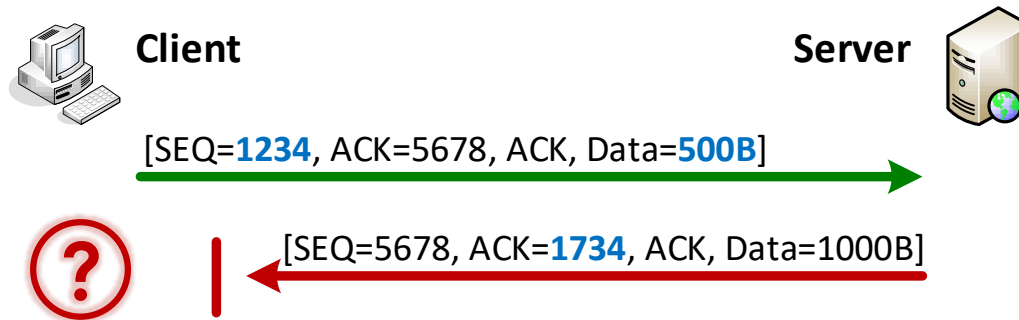
Pouzdan prenos

- Pouzdan prenos svakog segmenta (*Reliability*)
 - Potvrda primljenih podataka (SEQ-ACK)
- Nezavisno u oba smera
 - Različiti SEQ u oba smera
 - Prenos podataka i SEQ u jednom smeru, zajedno sa ACK za potvrdu prenosa u drugom smeru
 - $SEQ_{cs} + DATA_{cs} = ACK_{sc}$
 - $SEQ_{sc} + DATA_{sc} = ACK_{cs}$



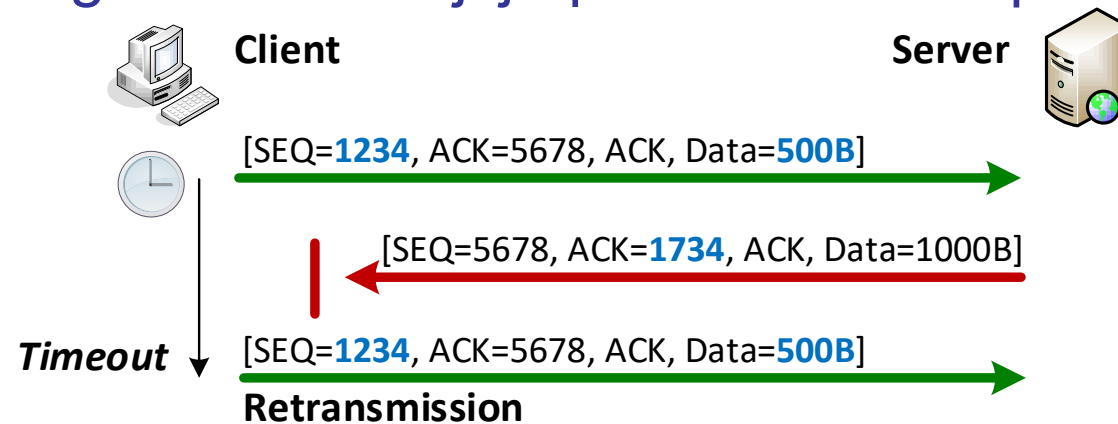
Pouzdan prenos

- Šta ako se izgubi paket koji prenosi potvrdu?
 - Čeka se određeno vreme za svaki poslani segment – *timeout*
- Koliko se čeka?
 - Vreme tajmera treba da bude veće od RTT (*Round Trip Time*)
 - RTT varira tokom vremena
- Kako proceniti RTT za pristizanje potvrda?



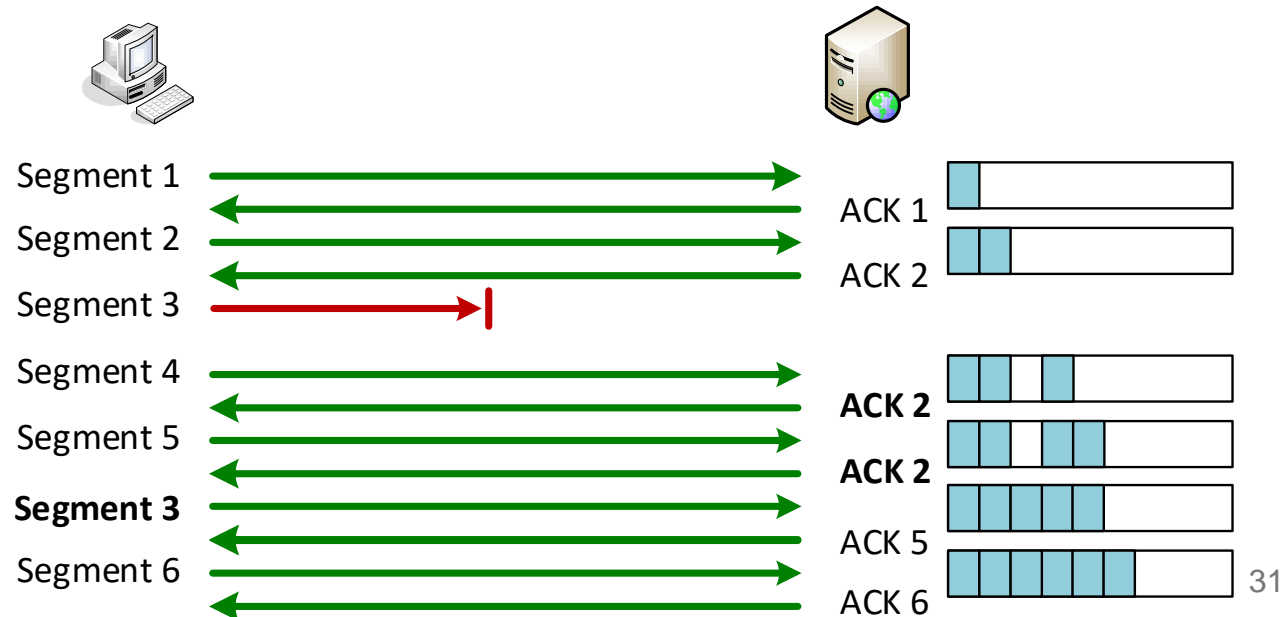
Pouzdan prenos - oporavak od greške

- Dinamičko određivanje vremena čekanja – procena RTT
 - RTT za poslednji poslati i potvrđeni segment - RTT_{new}
 - Pamti se prethodna procena – RTT_{old}
 - Procena RTT za naredni segment
$$RTT = a * RTT_{old} + (1-a) * RTT_{new}$$
 - a – faktor od 0 do 1 (veća vrednost – veći uticaj RTT_{old} , sporije promene)
- $Timeout = b * RTT$
 - b – faktor uvećanja (tipično $b=2$)
- Retransmisija segmenta za koji je prošao *timeout* period



Pouzdan prenos - oporavak od greške

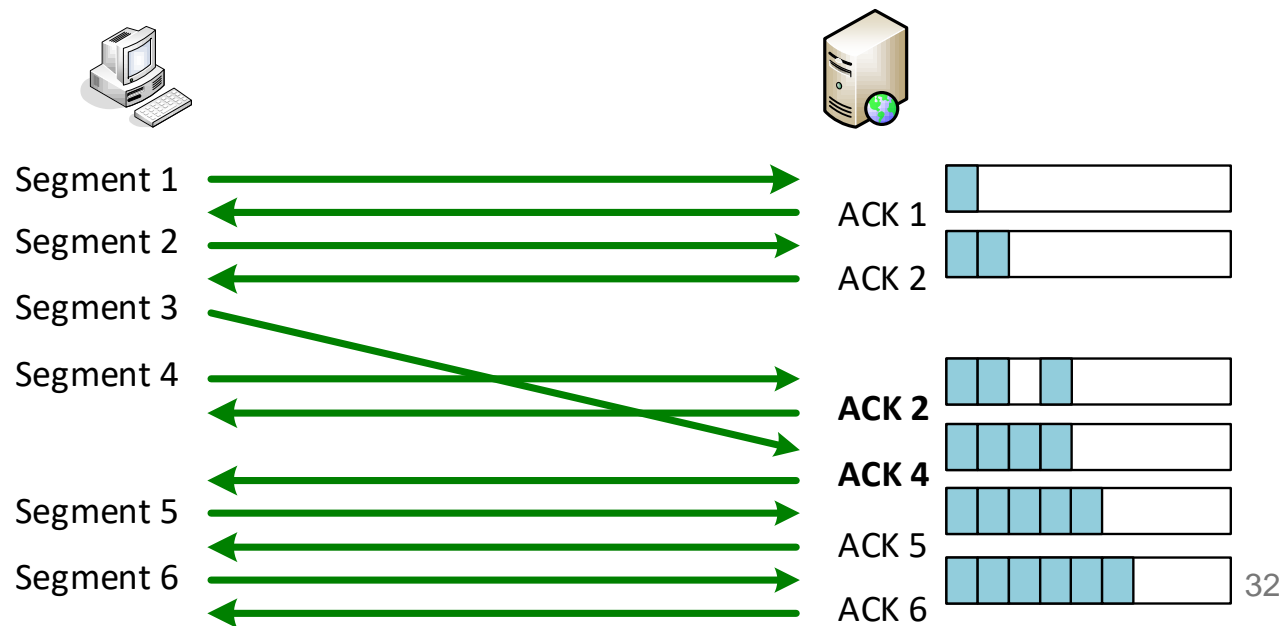
- **Oporavak od greške** (*Error Recovery*)
 - Na osnovu SEQ vrednosti, zna se gde se nalaze podaci
 - ACK se šalje samo kada pristigne neki segment
 - **ACK se šalje samo za poslednji ispravni segment u nizu**
- Višestruki ACK:
 - Označava da je jedan segment izgubljen, ali su drugi segmenti uspešno pristigli



Rekonstrukcija redosleda segmenata

- **Rekonstrukcija redosleda segmenata**
(*segments reordering*)

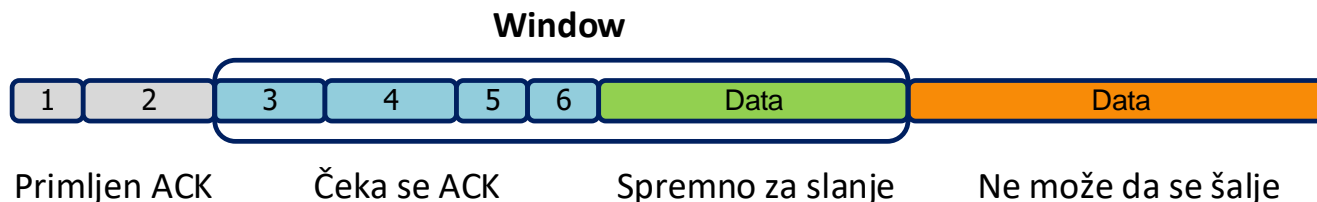
- Različiti segmenti mogu da stignu različitim putanjama
 - Može da dođe do promene redosleda prijema segmenata
 - Prijemna strana će da rekonstruiše originalni redosled na osnovu SEQ vrednosti primljenih segmenata



Kontrola toka

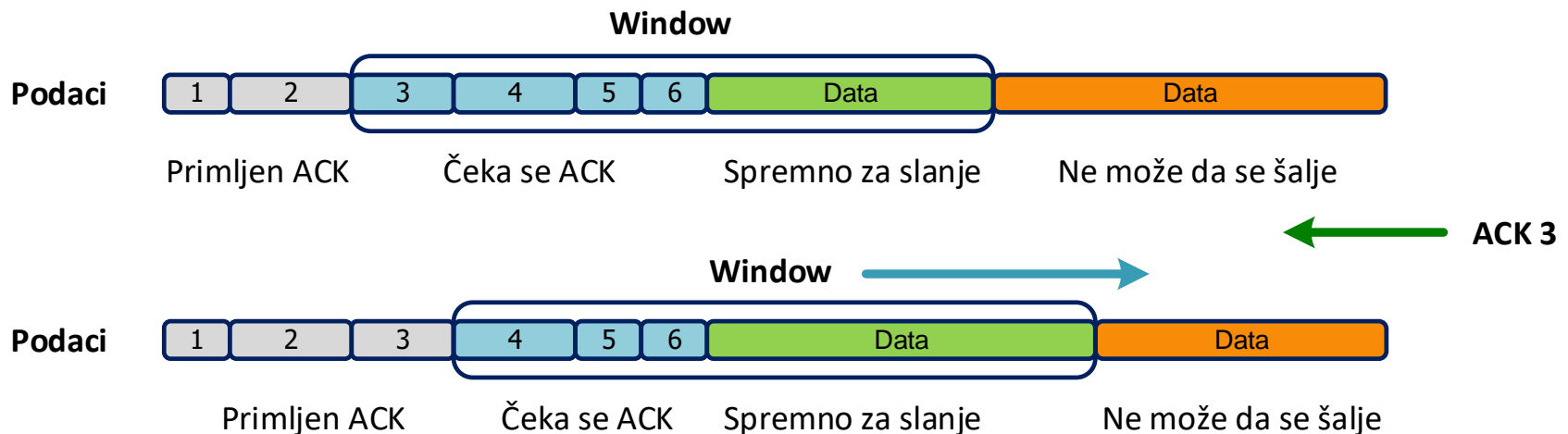
Kontrola toka (*Flow control*) – mehanizam prozora (*Window*)

- Na strani koja šalje podatke
- Označava ukupan broj bajtova koji se mogu poslati pre nego što se čeka na potvrdu
- Sprovodi se nezavisno u oba smera
- Prozor obuhvata deo podataka u baferu podataka koji se šalju
 - Pre prozora
 - Poslati podaci (segmenti) za koje je primljen ACK
 - U prozoru
 - Poslati podaci (segmenti) – čeka se ACK
 - Podaci koji se mogu poslati – spremno za slanje
 - Posle prozora
 - Podaci koji se ne mogu poslati



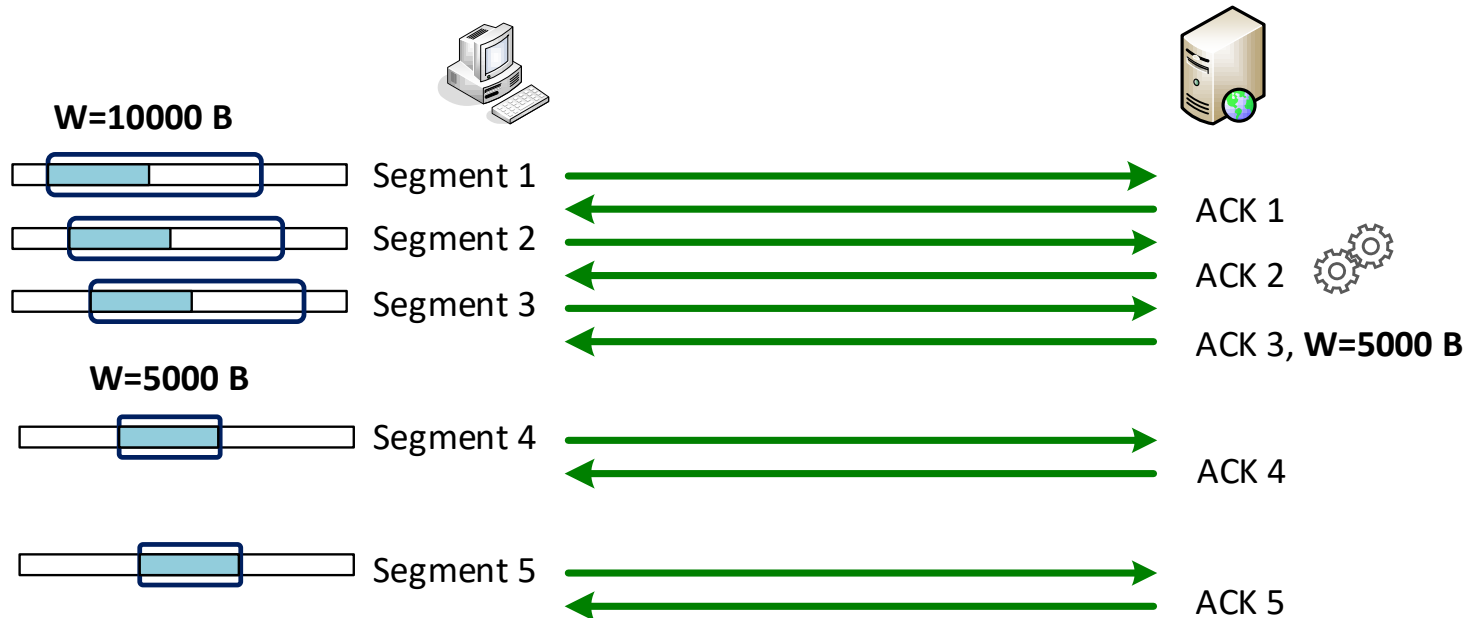
Kontrola toka

- Prozor se pomera kada se dobije ACK na segment na početku prozora – *Sliding Window*
- Kada se popuni prozor
 - Obustavlja se sa slanjem novih segmenata
 - Čeka se potvrda prethodnih segmenata sa početka prozora
 - Potvrđeni segmenti se oslobađaju, prozor se pomera i oslobađa se prostor u prozoru za slanje novih segmenata na kraju prozora
- Kontrola toka se uspostavlja veličinom prozora
 - Manji prozor - sporije slanje, veći prozor - brže slanje



Kontrola toka

- Dinamičko uspostavljanje veličine prozora - *Dynamic Window*
 - Inicijalno se obe strane dogovore o veličini prozora kod uspostave veze
 - U slučaju opterećenja prijemne strane ili gubitka paketa
 - Od pošiljaoca može da se zahteva smanjenje trenutne veličine prozora
 - Usporavanje slanja novih segmenata
 - Ako nema grešaka, veličina prozor može da se postepeno povećava

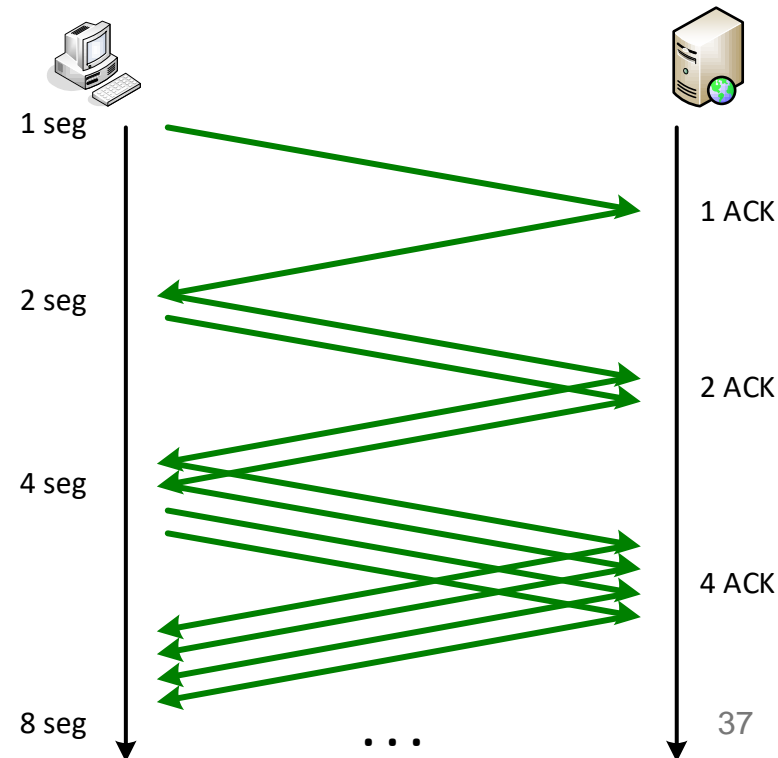


Kontrola zagušenja

- **Kontrola zagušenja** – TCP *Congestion Control*, RFC 5681
- Pošiljalac se dinamički prilagođava trenutnom opterećenju
- Algoritmi kontrole zagušenja:
 - *Slow Start*
 - *Congestion Avoidance*
 - *Fast Retransmit*
 - *Fast Recovery*

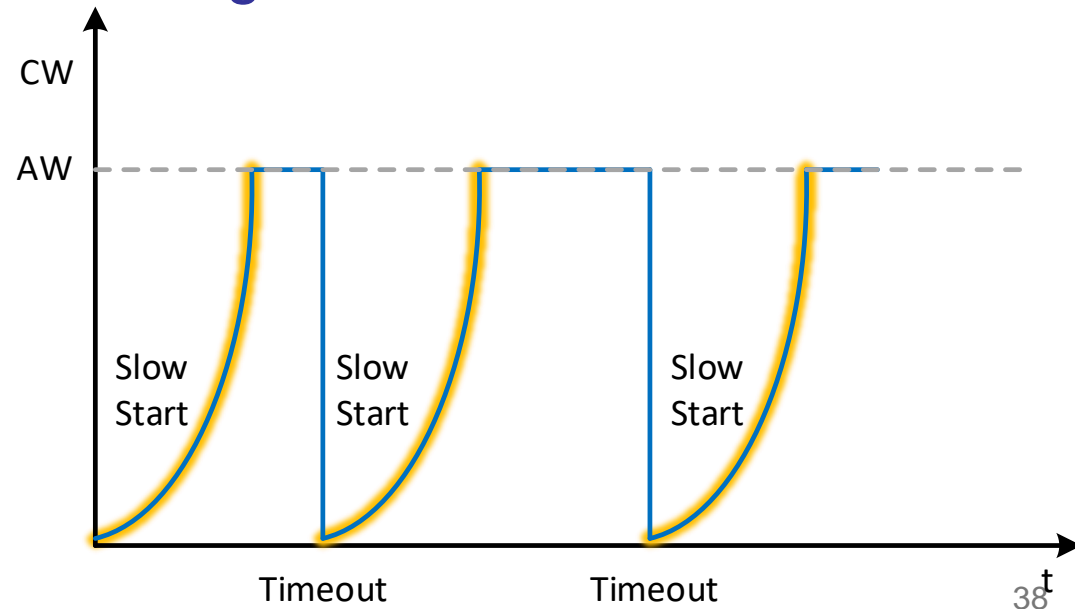
Slow Start

- *Advertised Window (AW)*
 - Inicijalna vrednost prozora postavljeno pri uspostavljanju veze
 - Postavlja primalac – kontroliše brzinu prijema segmenata
- *Congestion Window (CW)*
 - Da bi se izbeglo zagušenje, pošiljalac postepeno povećava stvarnu veličinu prozora (brzinu slanja) do maksimalne vrednosti (AD)
 - Inicijalna vrednost za CW je 1 segment
 - CW se povećava sa svakim ACK
 - Ako se potvrđuje svaki segment, CW se u svakom koraku povećava za 2
 - 1, 2, 4, 8 eksponencijalno
 - Tipično jedan ACK šalje za 2 segmenta, ali je rast i dalje eksponencijalan



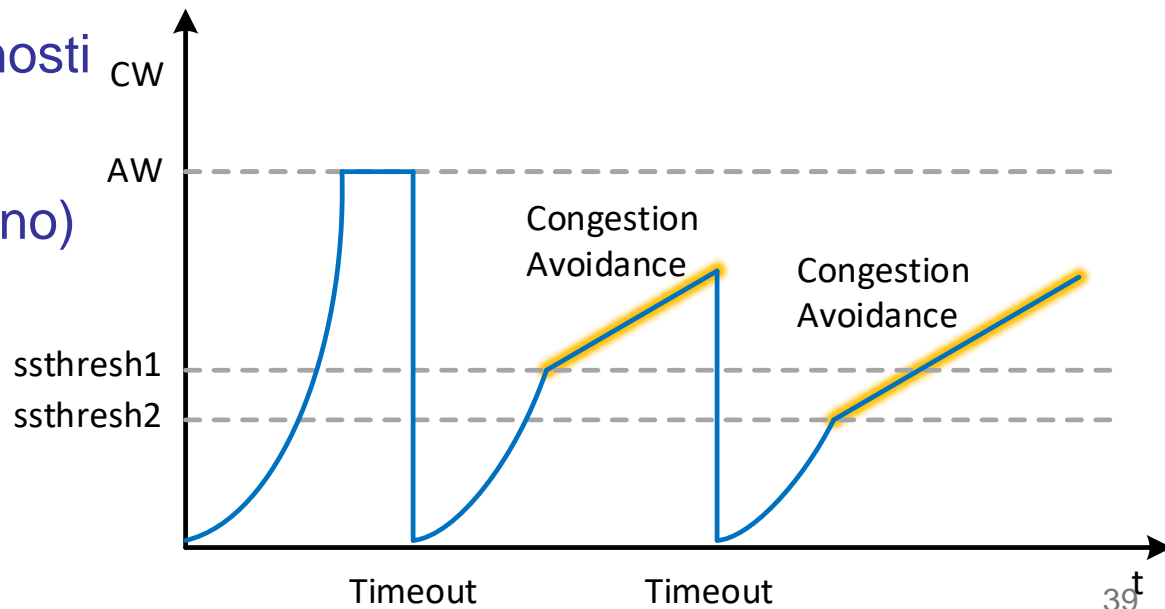
Slow Start

- **Slow Start**
 - Početak slanja sa $CW=1$
 - Eksponencijalno povećanje CW do maksimalne vrednosti
- Cilj
 - Krenuti oprezno sa malom brzinom prenosa
 - Ubrzo dostići maksimalnu brzinu, ako mreža može da podnese
- Ako nastane *timeout* – gubitak segmenta
 - Novi *Slow Start*



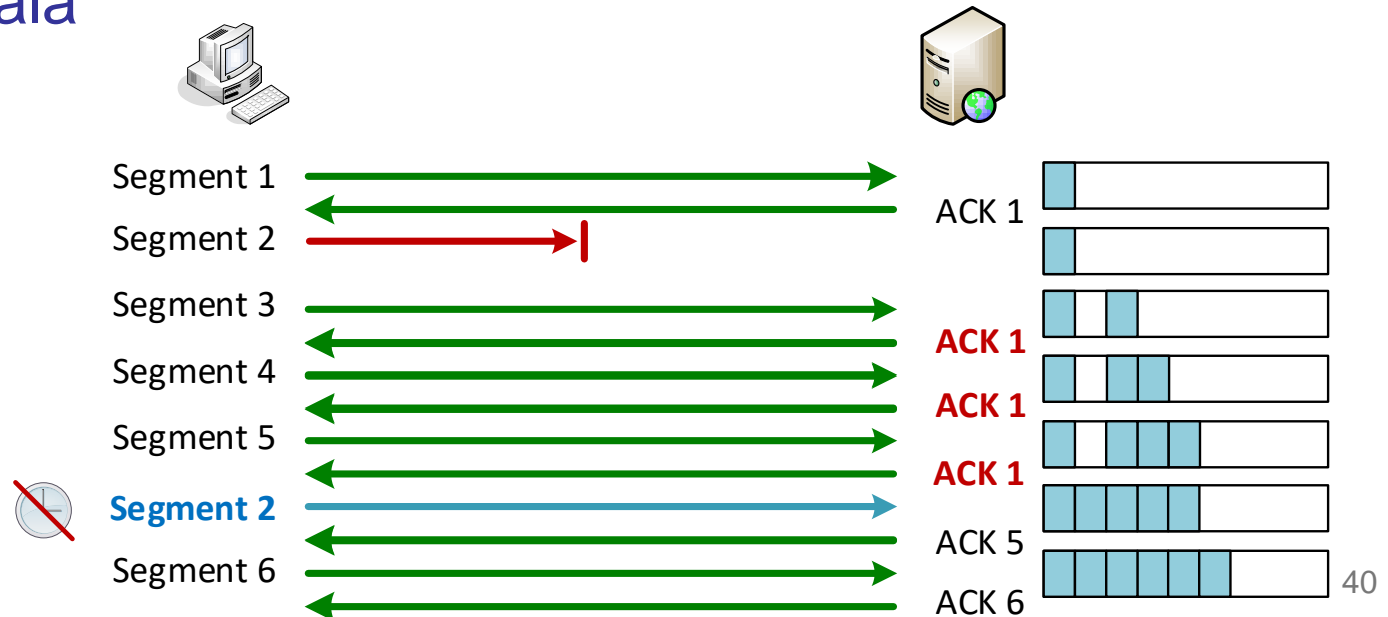
Congestion Avoidance

- *Congestion Avoidance* - Izbegavanje zagušenja
 - Linearno povečanje prozora, umesto eksponencijalnog
- *Slow Start Threshold Size* – *ssthresh*
 - Veličina prozora dokle traje eksponencijalni rast u *slow start* fazi
- Ako ne stigne ACK za neki segment
 - *ssthresh* se smanjuje na polovinu poslednje vrednosti
 - $cwnd < ssthresh$
slow start (eksponencijalno)
 - $cwnd > ssthresh$
congestion avoidance
(linearno)



Fast Retransmit

- Dupli (višestruki) ACK
 - Određeni segment je izgubljen (ili je došlo do promene redosleda)
 - Naredni segmenti su uspešno primljeni
 - 1 ili 2 dupla ACK – možda segment nije izgubljen, možda je došlo do promene redosleda
 - 3 dupla ACK – segment je ipak izgubljen, treba ga što pre poslati ponovo
- **Fast Retransmit** – ponovno slanje segmenta, pre isteka *timeout* intervala



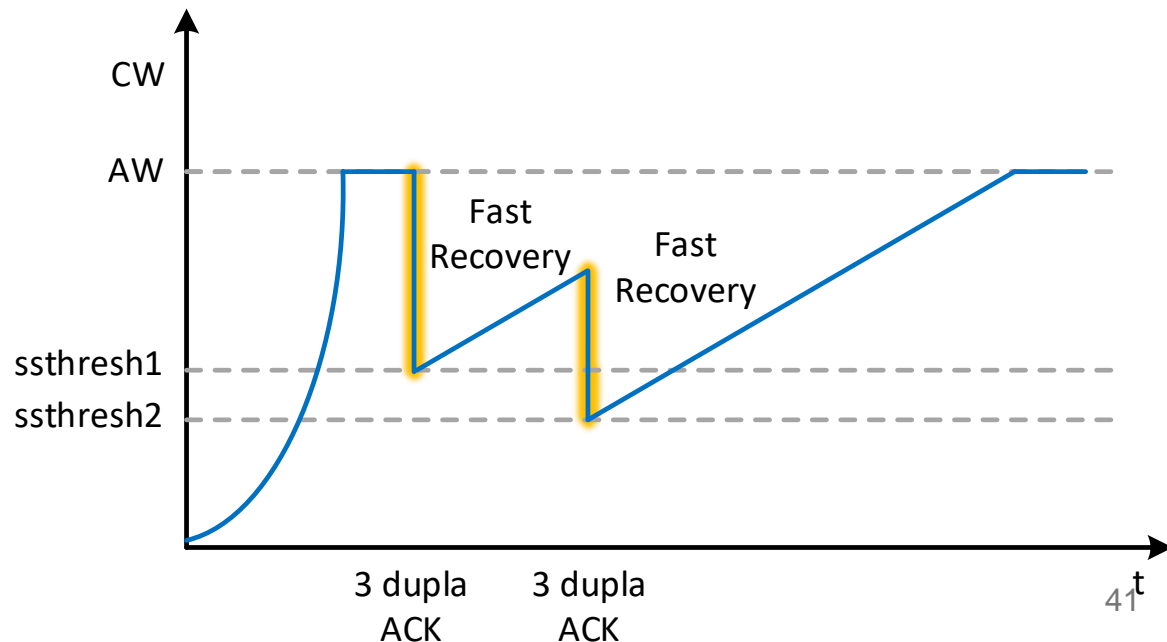
Fast recovery

- *Fast Retransmit*

- Jedan segment je uništen, ali najmanje 3 segmenta su uspešno stigla
- Nema potrebe za drastičnim usporenje slanja podataka

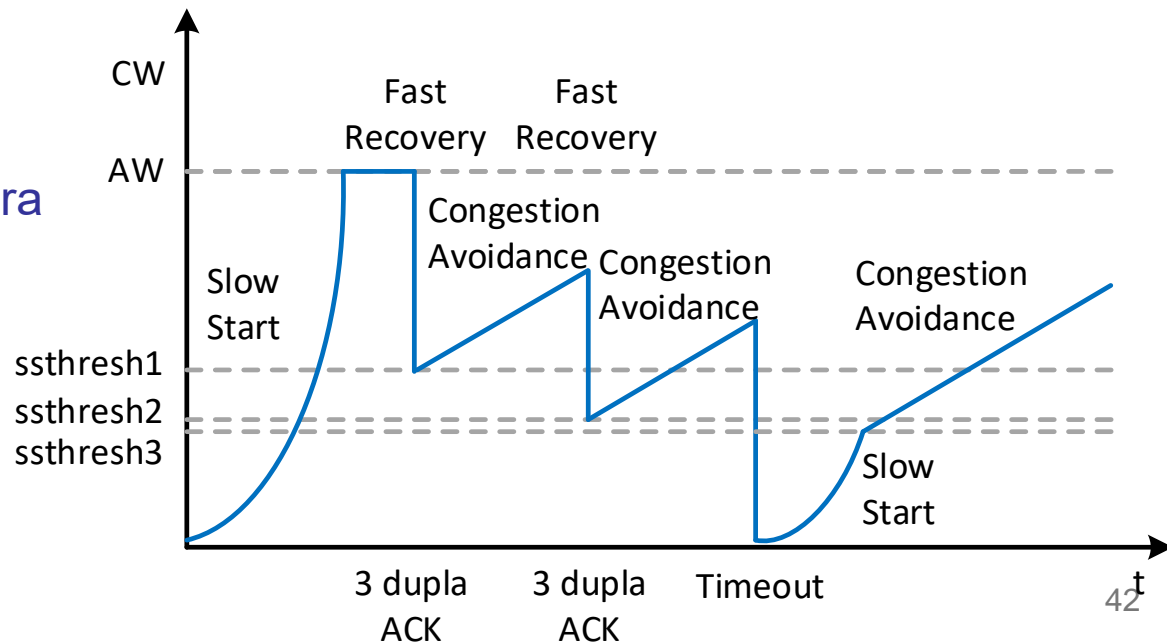
- *Fast Recovery*

- Nakon *Fast Retransmit*
- Ide su direktno u *Congestion Avoidance*, bez *Slow Start* faze



Kontrola zagušanja

- Algoritmi kontrole zagušenja:
 - **Slow Start**
 - Početak od najmanja veličine prozora, uz eksponencijalni rast
 - Na početku ili u slučaju gubitka segmenta uz *timeout*
 - **Congestion Avoidance**
 - Linearni rast, kada se dostigne polovina prethodne veličine prozora
 - **Fast Retransmit**
 - Slanje segmenta nakon 3 dupla ACK
 - **Fast Recovery**
 - Smanjenje veličine prozora na polovinu
 - Ulazak u direkno *Congestion Avoidance* bez *Slow Start*



Alati za proveru TCP konekcija

- **netstat**

- prikaz otvorenih TCP konekcija na lokalnom uređaju

```
C:\Users\user>netstat -n
```

```
Active Connections
```

Proto	Local Address	Foreign Address	State
TCP	147.91.15.32:49509	74.125.133.188:5228	ESTABLISHED
TCP	147.91.15.32:49709	13.92.229.58:443	ESTABLISHED
TCP	147.91.15.32:53532	23.6.112.112:80	TIME_WAIT
TCP	147.91.15.32:53544	92.123.16.209:80	TIME_WAIT
TCP	147.91.15.32:53695	62.67.193.75:443	TIME_WAIT
TCP	147.91.15.32:53701	104.17.58.239:443	ESTABLISHED
TCP	147.91.15.32:53712	64.68.120.41:443	ESTABLISHED
TCP	147.91.15.32:53715	162.247.242.20:443	ESTABLISHED
TCP	147.91.15.32:53722	13.57.4.17:443	ESTABLISHED
TCP	147.91.15.32:53723	172.217.23.206:443	ESTABLISHED
TCP	147.91.15.32:53725	23.7.203.117:80	TIME_WAIT
...			

- **nmap**

- Prikaz otvorenih TCP portova na udaljenom uređaju

```
[root@user~]# nmap 147.91.111.222
```

```
Interesting ports on xxx.bg.ac.rs (147.91.111.222):
```

PORT	STATE	SERVICE
21/tcp	open	ftp
22/tcp	open	ssh
80/tcp	open	http
443/tcp	open	https
514/tcp	open	shell
3306/tcp	open	mysql
8009/tcp	open	ajp13

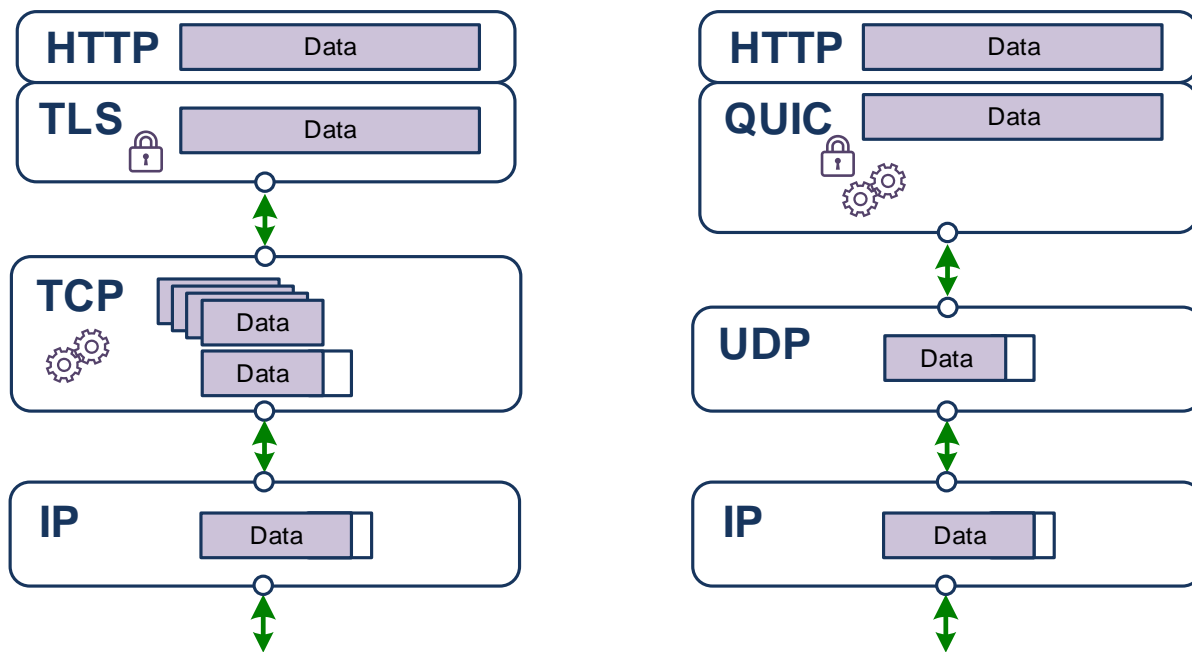
```
...
```

UDP vs. TCP

TCP	UDP
Connction oriented	Connectionless
Prenos kontinualnog niza bajtova	Prenos nezavisnih poruka
Pouzdan prenos (oporavak od grešaka, gubitka paketa, promene redosleda...)	Nema garancije prenosa
Kontrola toka – prilagođava se uslovima u mreži	Nema kontrole toka
Zaglavlja 20 bajtova	Zaglavlje 8 bajtova
Sporiji, složeniji	Brz, jednostavan
Primena: Kada je bitna pouzdanost: <ul style="list-style-type: none">- veb, mejl, prenosa fajlova, poslovne aplikacije, baze podataka...	Primena: Kada nije bitna pouzdanost: <ul style="list-style-type: none">- jednostavne aplikacije- Periodična komunikacija Kada je bitna brzina, malo kašnjenje: <ul style="list-style-type: none">- real-time saobraćaj (IPTV, IP telefonija, video konferencije...)

QUIC protokol

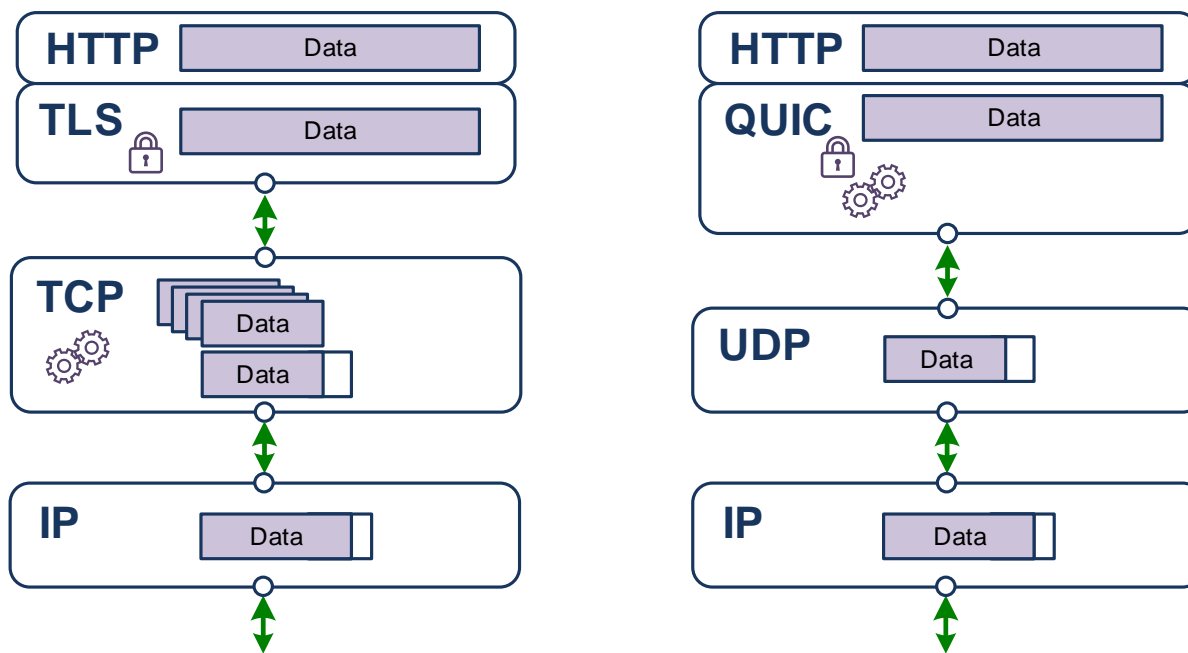
- Specifičnosti veb saobraćaja
 - Sigurnost - **TLS** - *Transport Layer Security*
 - Koriste ga aplikacije za šifrovanje saobraćaja
 - Oslanja se na TCP
 - **QUIC** – Google-ov protokol za sigurnu veb komunikaciju
 - Umesto sadašnje kombinacije TCP+TLS (HTTPS)
 - Objedinjuje funkcionalnost TCP i TLS, korišćenjem UDP, port 443



QUIC protokol

- Osobine

- Brzo uspostavljanje veza, bolji subjektivni osećaj korisnika
- Moguće multipleksiranje više tokova preko jedne veze (sve veb sesije)
- Efikasniji i fleksibilniji mehanizmi kontrole toka
- Mogućnost zadržavanja QUIC veze i prilikom promene IP adrese klijenta
 - Značajno za mobilnost – npr. sa WLAN na 4G/5G
- Tendencija porasta (10-20%) - najviše google, youtube



Literatura

- Wendell Odom
„CCNA - Cisco official exam certification guide“
Cisco Press
- James Kurose, Keith Ross
„Computer Network - A Top-Down Approach“
- James Kurose, Keith Ross
„Umrežavanje računara: Od vrha ka dnu“
prevod 7. izdanja
CET

