Preddiplomski studij

Računarstvo

Komunikacijske mreže

7. Transportni sloj.
   Transportni protokoli u Internetu.

Ak.g. 2011./2012.

Podsjetimo se ….

Zadaća transportnog sloja

- Transparentan prijenos transportnih jedinica podataka od izvora do odredišta
  - s kraja na kraj mreže (engl. *end-to-end*)
  - oslanja se na mrežni sloj, pruža (spojnu ili nespojnu) uslugu višem sloju (sjednički, prezentacijski, aplikacijski)

- prijenos uz zahtijevanu kvalitetu usluge:
  - pouzdana usluga - prijenos bez pogrešaka (dodatni mehanizmi za pouzdanost)
  - prijenos uz najmanje kašnjenje (minimalno procesiranje)

Sadržaj predavanja

- Usluga transportnog sloja
- Funkcionalnost
  - adresiranje
  - multipleksiranje
  - uspostava i raskid veze
  - kontrola toka i privremena pohrana
  - oporavak od prekida
- Protokoli transportnog sloja u Internetu
  - Transmission Control Protocol
  - User Datagram Protocol

Usluga transportnog sloja

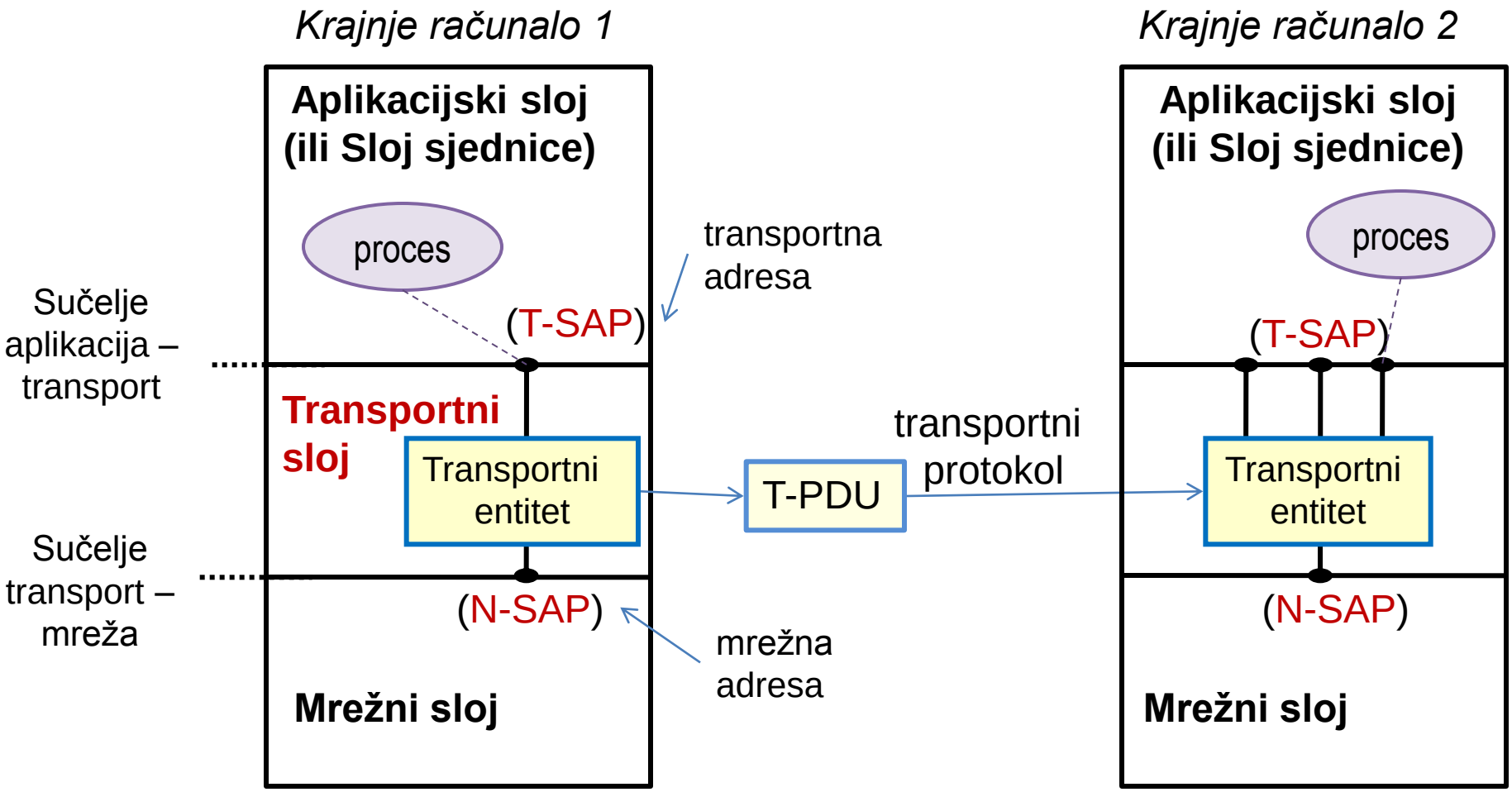
- svrha: omogućiti **logičko** povezivanje procesa na krajnjim računalima

- usluga može biti spojna i nespojna
- funkcije:
  - adresiranje (na razini transportnog sloja)
  - multipleksiranje
  - uspostava i raskid veze (za spojnu uslugu)
  - kontrola toka
  - oporavak od prekida komunikacije
- "kompenzacija nedostataka" mrežnog sloja

- izbor transportnog protokola ovisit će o parametrima kvalitete usluge koje zahtijeva aplikacija!

Logičko povezivanje procesa

logička veza – procesi koji komuniciraju ponašaju se kao da su izravno spojeni



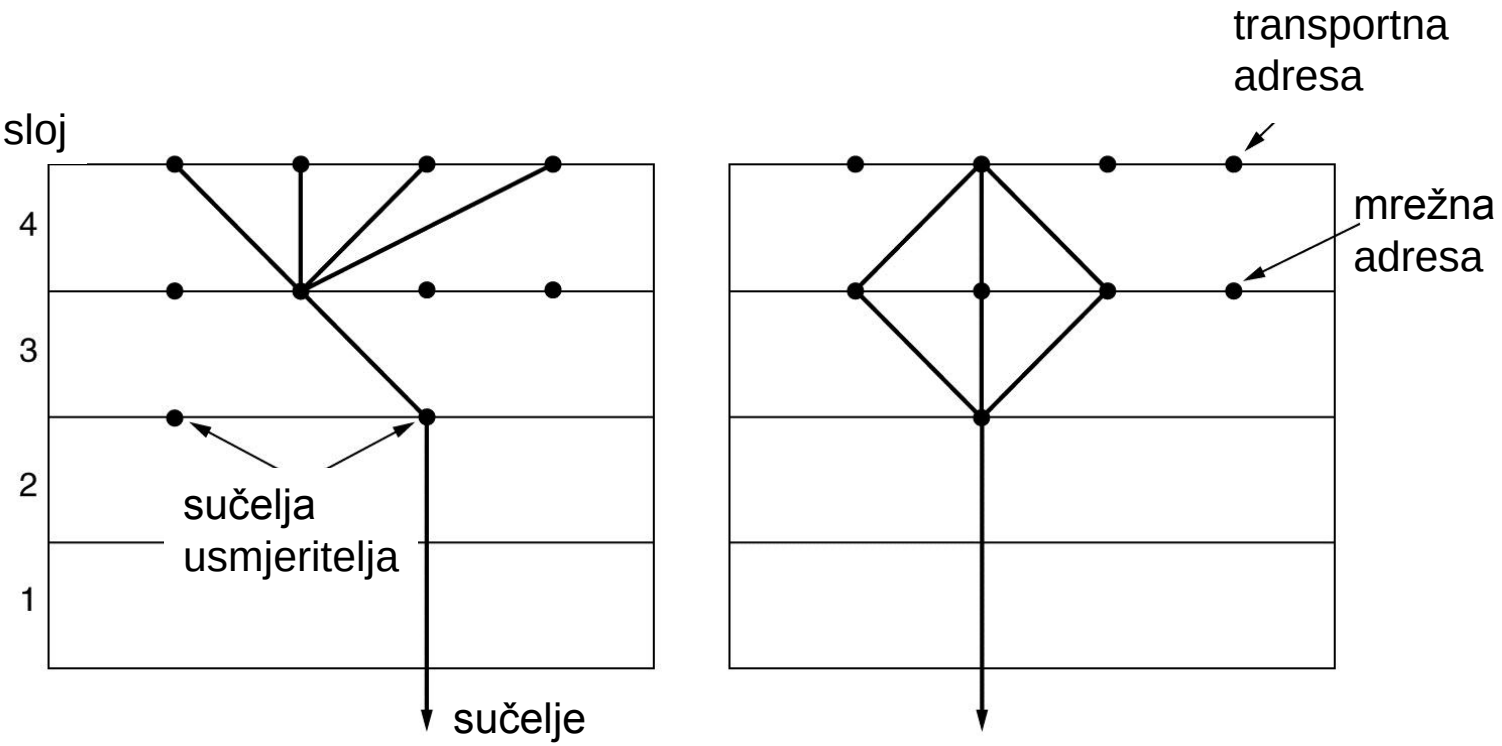
Adresiranje

- na sučelju transporta i mreže: mrežna pristupna točka usluzi – N-SAP (Network-Service Access Point)
  - adresa mrežnog sučelja
  - u internetskom modelu: IP adresa
- na sučelju transporta i aplikacije: transportna pristupna točka usluzi – T-SAP (Transport-Service Access Point)
  - adresa transportnog entiteta
  - u internetskom modelu: vrata (engl. *port*)

- krajnje točke logičke veze:
  (IP adresa izvora, vrata na izvoru) --- (IP adresa odredišta, vrata na odredištu)
  - u internetskom modelu: priključnica (engl. socket)

- multipleksiranje: način preslikavanja T-SAP:N-SAP
  - odozgo, n:1
  - odozdo, 1:n

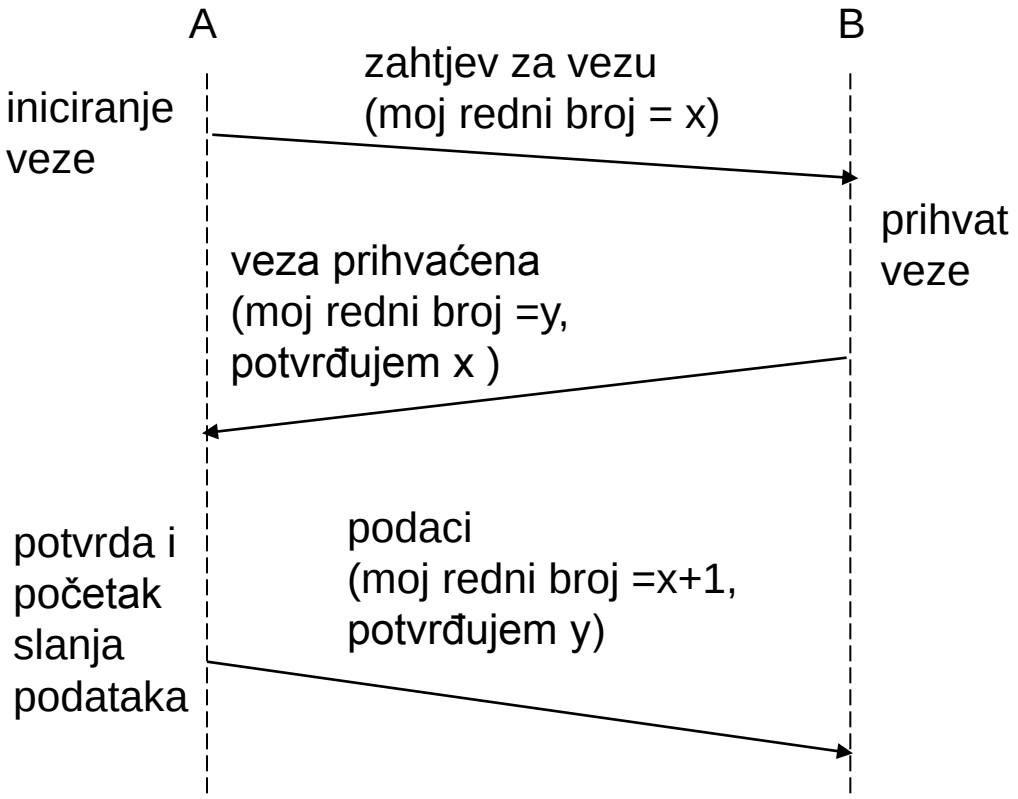
Multipleksiranje



- multipleksiranje odozgo:
  više procesa komunicira preko iste mrežne adrese
  - npr: TCP, UDP
- multipleksiranje odozdo:
  proces otvara više mrežnih veza i šalje podatke naizmjence po njima

Uspostava veze

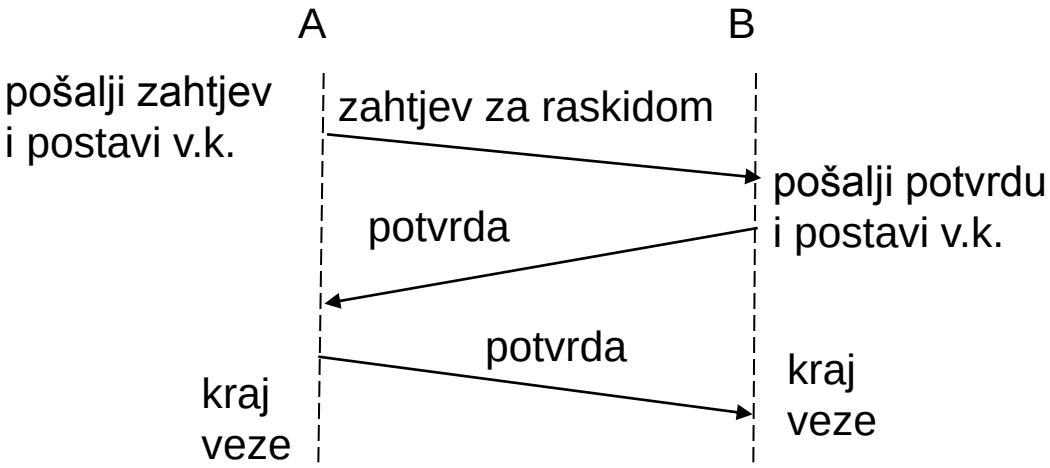
- kontrolne poruke: zahtjevi i odgovori u zadanom (dogovorenom) obliku
- mehanizmi:
  - numeracija poruka
  - (pozitivne) potvrde
  - negativne potvrde
  - vremenska kontrola
  - klizeći prozor



- vremenska kontrola (v.k.) nužna za slučajeve ispada ili gubitka zahtjeva/ potvrde

Raskid veze

- različiti scenariji urednog zatvaranja veze
- primjer:



- nužna vremenska kontrola (v.k.) za slučajeve ispada ili gubitka zahtjeva/ potvrde

Kontrola toka i privremena pohrana

- kontrola toka – sličan problem kao u sloju podatkovne poveznice:
  - usklađivanje pošiljateljeve brzine slanja i brzine primanja/obrade na strani primatelja
  - najčešći model kontrole toka – klizeći prozor!

- privremena pohrana – zbog nepouzdane dostave i moguće promjene redosljeda datagrama
  - treba privremeno pohranjivati T-PDU
  - pohrana i prilikom slanja, i prilikom primanja!
  - rješenje: usklađivanje veličine klizećeg prozora i dinamičko rukovanje memorijskim spremnikom

Oporavak (transportne veze) od prekida

- prekidi se događaju - moguća mjesta kvara:
  - krajnji uređaji
  - usmjeritelji
- kvar na usmjeritelju i krajnje točke sa očuvanim stanjem transportne veze – jednostavan oporavak
- ako je kvar na krajnjem uređaju - što se događa s transportnom vezom nakon ponovnog pokretanja?
  - teži problem, stanje nije sačuvano
  - općeniti zaključak: nemoguće je sasvim prikriti kvar od viših slojeva, ali vrijedi - prekid na sloju N može ispraviti sloj N+1 pod pretpostavkom da krajnje točke “znaju gdje su stale”

Izbor transportnog protokola

Izbor transportnog protokola ovisi o parametrima kvalitete usluge (pouzdanost, kašnjenje, itd.) koje zahtijeva aplikacija!

Primjeri aplikacija – zahtjevnost /oštrina kriterija

Aplikacija	Pouzdanost	Kašnjenje	Kolebanje kašnjenja	Širina pojasa
Elektronička pošta	Visoki	Niski	Niski	Niski
Transfer datoteka	Visoki	Niski	Niski	Srednji
Pristup Webu	Visoki	Srednji	Niski	Srednji
Rad na daljinu	Visoki	Srednji	Srednji	Niski
Audio na zahtjev	Niski	Niski	Visoki	Srednji
Video na zahtjev	Niski	Niski	Visoki	Visoki
Telefonija	Niski	Visoki	Visoki	Niski/Srednji
Videokonferencija	Niski	Visoki	Visoki	Visoki

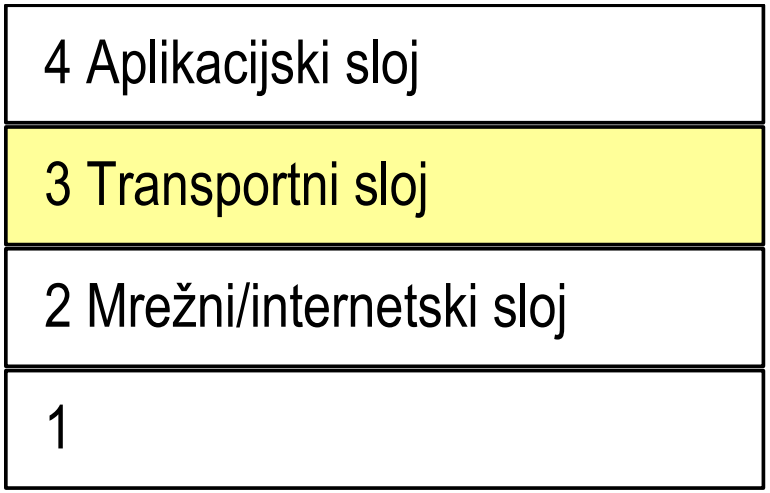
Ključne značajke protokola transportnog sloja

- dvosmjerna komunikacija
  - sposobnost istovremenog slanja i primanja
- pouzdanost transporta
  - detekcija gubitka paketa i eventualna reakcija
- transfer poruka ili niza okteta
  - dvije mogućnosti tretiranja podataka: kao blokovi/poruke, ili kao niz okteta
- očuvanje redoslijeda podataka
  - rekonstrukcija izvornog redoslijeda poruka ili okteta na odredištu za slučaj narušavanja redoslijeda pri prolasku kroz mrežu
- kontrola toka
  - usklađivanje brzina slanja i primanja podataka između krajnjih točaka (procesa)

Sadržaj predavanja

- Usluga transportnog sloja
- Funkcionalnost
  - adresiranje
  - multipleksiranje
  - uspostava i raskid veze
  - kontrola toka i privremena pohrana
  - oporavak od prekida
- Protokoli transportnog sloja u Internetu
  - Transmission Control Protocol
  - User Datagram Protocol

TCP/IP model: transportni sloj i protokoli



- transmisijski kontrolni protokol (*Transmission Control Protocol*, TCP)
  - pouzdana transportna usluga: prijenos niza okteta bez pogrešaka, uz isporuku potpune informacije u nepromijenjenom redoslijedu
- Korisnički datagramski protokol (*User Datagram Protocol*, UDP)
  - jednostavna transportna usluga: prijenos uz najmanje moguće kašnjenje informacije

Transmission Control Protocol - TCP

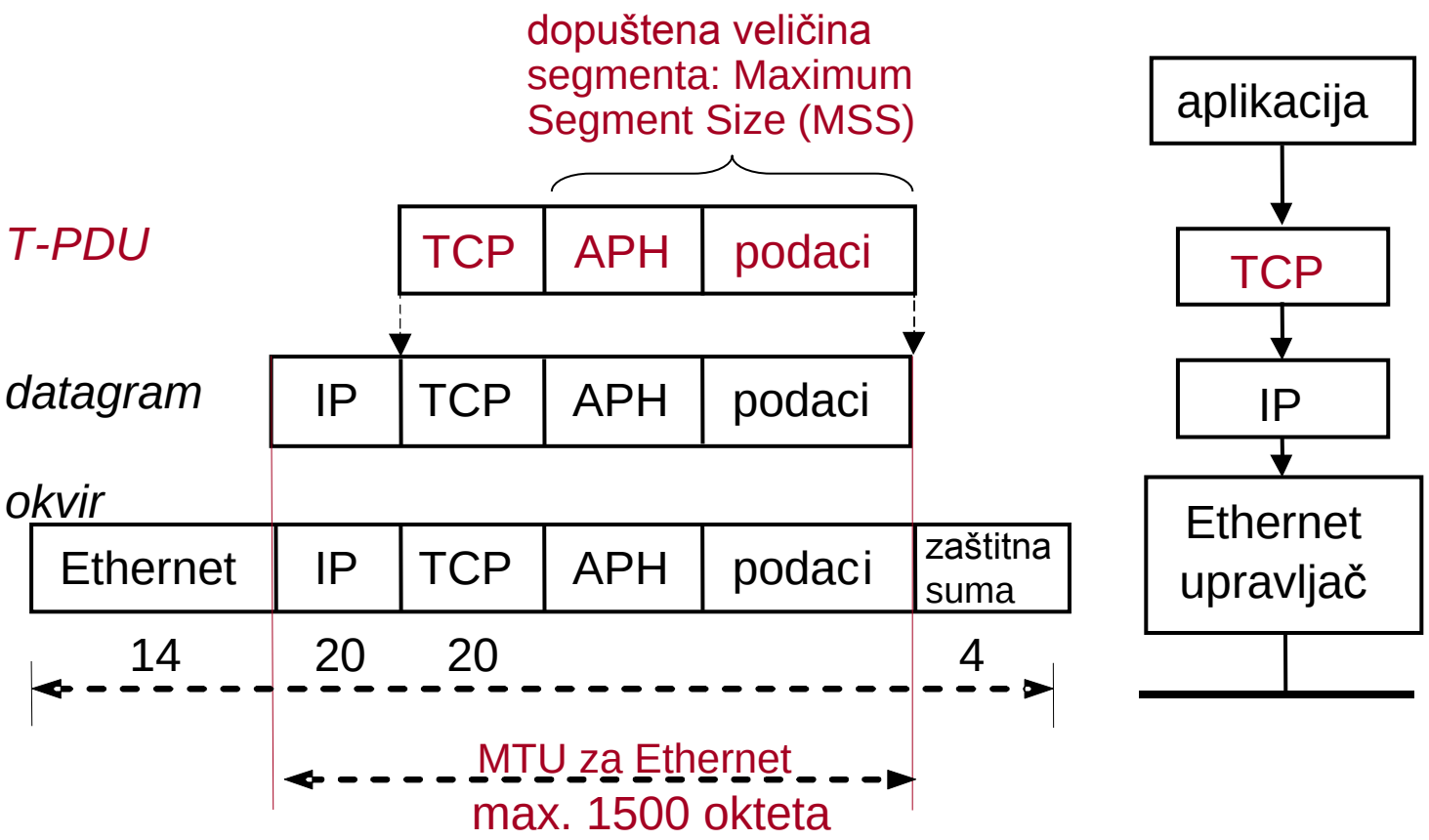
- TCP je spojno-orijentirani, pouzdani internetski protokol transportnog sloja
  - TCP pruža spojnu uslugu transporta struje okteta povrh nespojnog IP-a
  - uspostavlja logičku vezu između procesa na krajnjim računalima
  - osigurava pouzdan transport s kraja na kraj pomoću mehanizama potvrde i retransmisije, uz očuvani redoslijed struje okteta i upravljanje transportnom vezom.
  - logička veza između procesa definirana je parom 16-bitnih transportnih adresa, koje se u internetskoj terminologiji nazivaju vrata (engl. *port*).
  - TCP PDU naziva se (TCP) segment.

Funkcionalnost TCP-a

- osnovne funkcije:
  - osnovni transport podataka
  - adresiranje i multipleksiranje
  - pouzdanost
  - upravljanje tokom
  - upravljanje logičkom vezom
  - prioritet i sigurnost (logičke veze, ne podataka!)

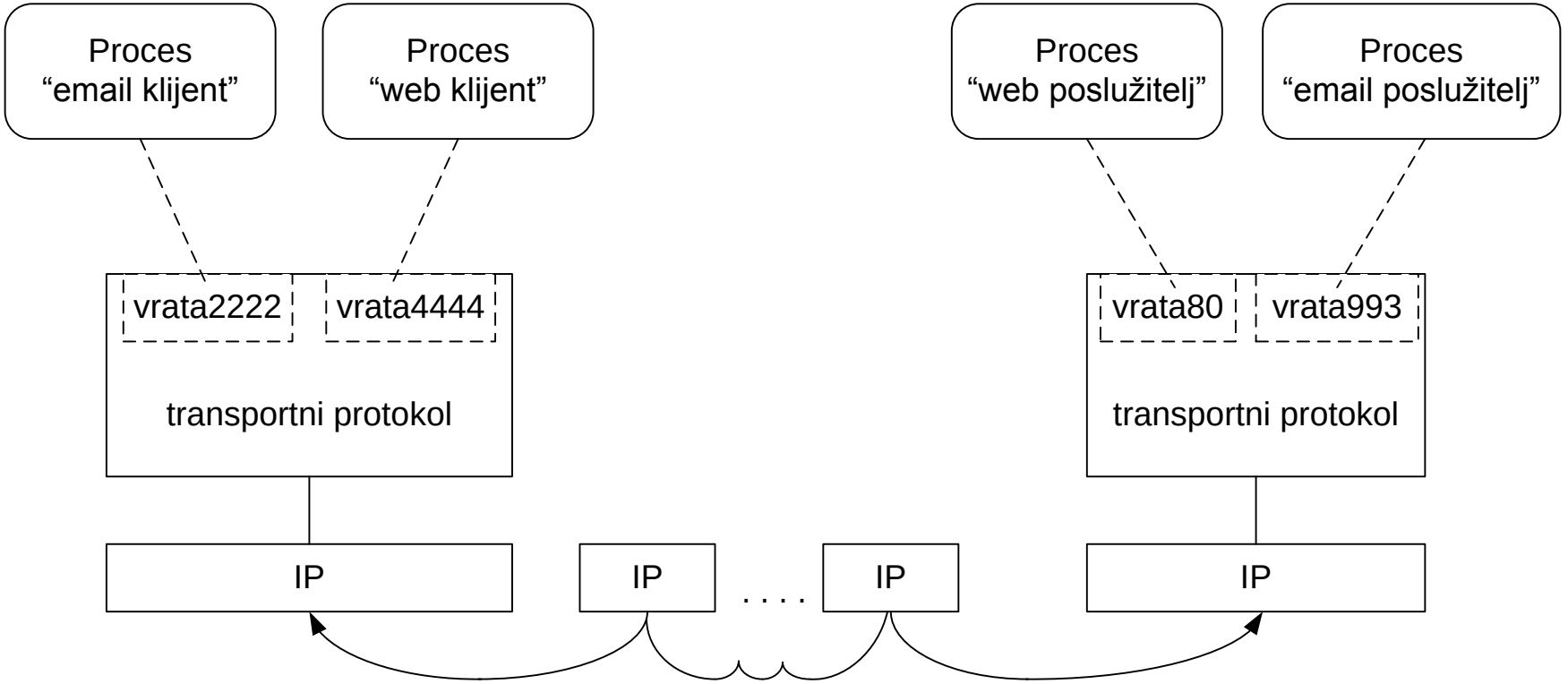
Transport podataka

- dvosmjerni transport kontinuiranog niza podataka, pakiranjem okteta podataka u *segmente,* koje potom predaje protokolu mrežnog sloja



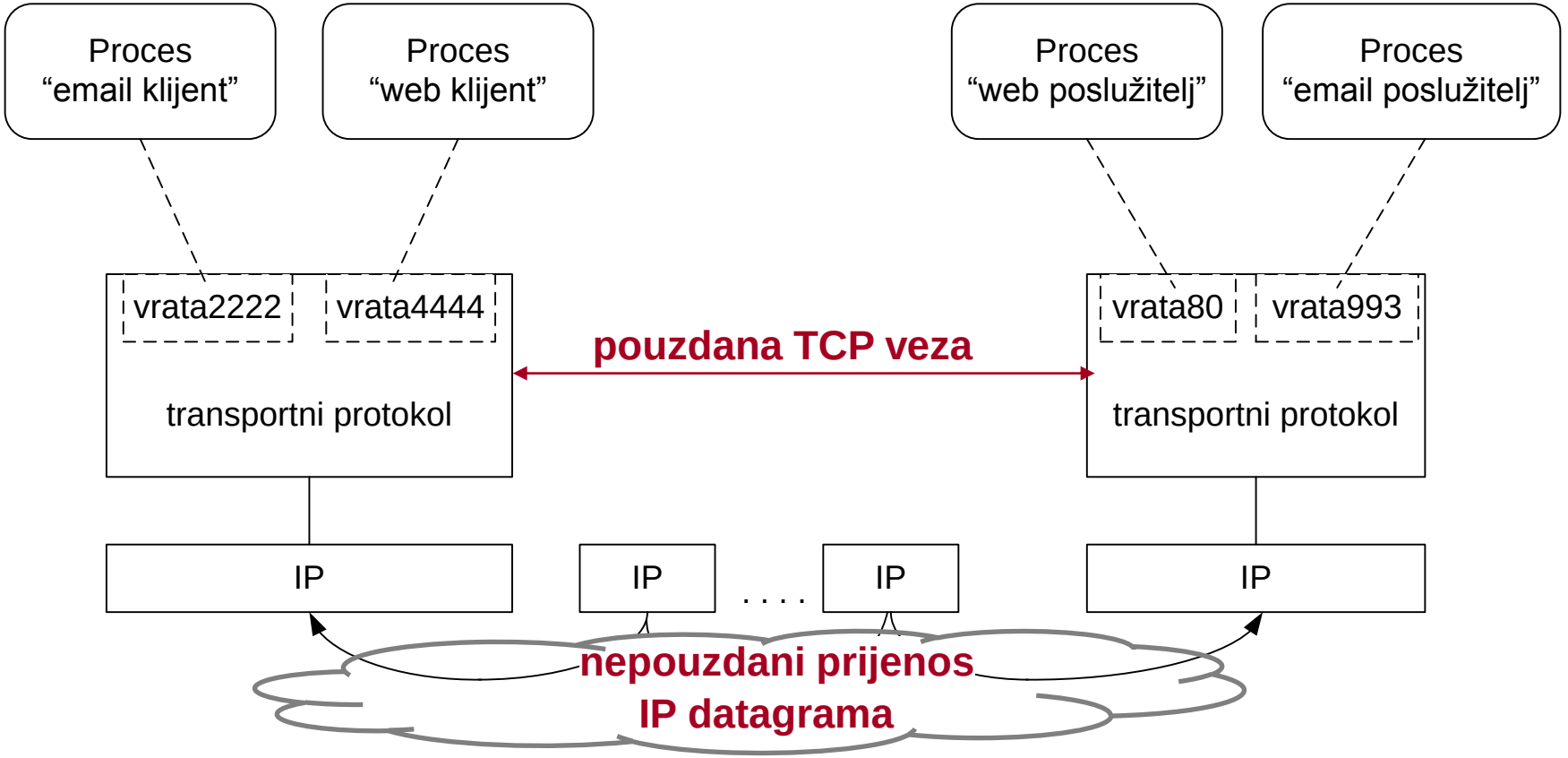
Adresiranje i multipleksiranje

- raščlanjivanje tokova podataka koji pripadaju različitim procesima
- Krajnje točke komunikacije:
  <IP adresa izvora, vrata na izvoru, IP adresa odredišta, vrata na odredištu>



TCP logička veza

- pouzdana TCP veza preko nepouzdanog datagramskog mrežnog sloja



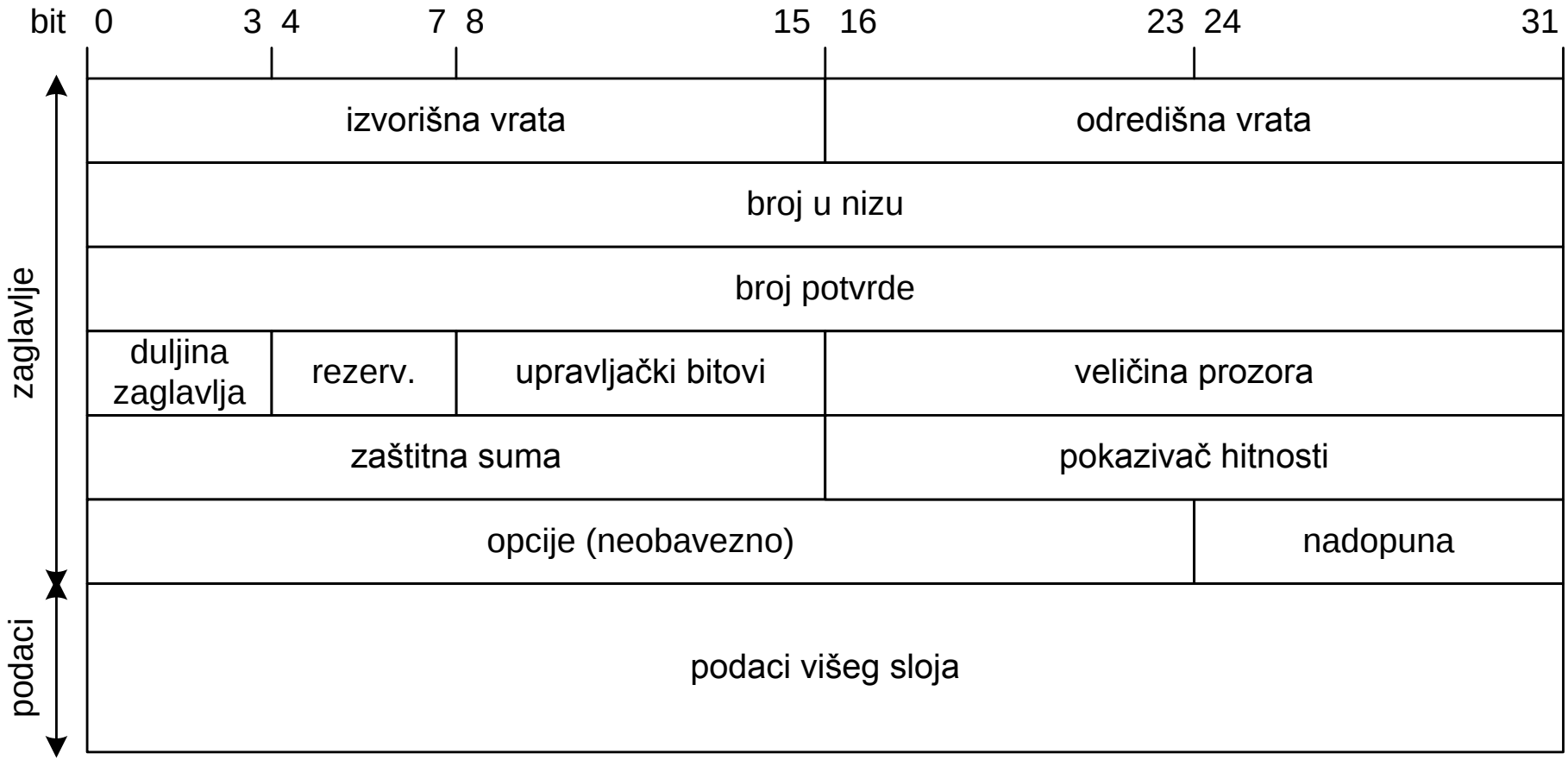
Pouzdanost veze

- mehanizmi koji osiguravaju pouzdanost:
  - detekcija pogrešaka
  - retransmisija
  - kumulativna potvrda
  - vremenska kontrola
  - polja u zaglavlju koja služe za oznake segmenta u nizu i oznake potvrda

- osnovna ideja:
  - svaki segment je numeriran
  - primatelj potvrđuje primljene segmente
  - potvrda je kumulativna – potvrđuje sve oktete do onog na kojeg se potvrda odnosi
  - pošiljatelj postavlja vremensku kontrolu prilikom slanja segmenta
  - ukoliko do isteka vremenske kontrole ne primi potvrdu, pošiljatelj smatra segment izgubljenim

važno! numeriraju se segmenti, ali taj broj NIJE redni broj segmenta, nego redni broj **okteta** koji je **prvi u promatranom segmentu!**

Struktura TCP-segmenta



Nazivi i uloge upravljačkih bitova

upravljački bitovi

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N

- uspostava i raskid veze:
  - SYN (od engl. *synchronize*)
  - FIN (od engl. *finish*)
- potvrda:
  - ACK (od engl. *acknowledgement*)
- prioritetni podaci:
  - URG (od engl. *urgent*)
  - PSH (od engl. *push*)
- poništavanje veze:
  - RST (od engl. *reset*)

Način rada protokola TCP

- Da bi ostvario svoje funkcije, TCP “pamti” niz parametara vezanih uz svaki par komunicirajućih procesa, odn. uz stanje logičke veze
  - ponašanje se može opisati automatom stanja
  - automat stanja opisuje jedan kraj veze, odn. jedan TCP proces (na primjer, proces na strani klijenta i li na strani poslužitelja – uočimo da oni ne moraju biti u istom stanju!)
  - stanje – karakterizira ga skup internih varijabli procesa
  - prijelaz – karakteriziraju ga ulazni događaj koji izaziva promjenu stanja i izlazni signal ili poruka prema drugoj krajnjoj točki

Stanja i prijelazi

- Stanja
  - Listen
  - Syn_Rcvd
  - Syn_Sent

  } uspostavljanje veze

  - Established → veza uspostavljena
  - Close_wait
  - Last_ack

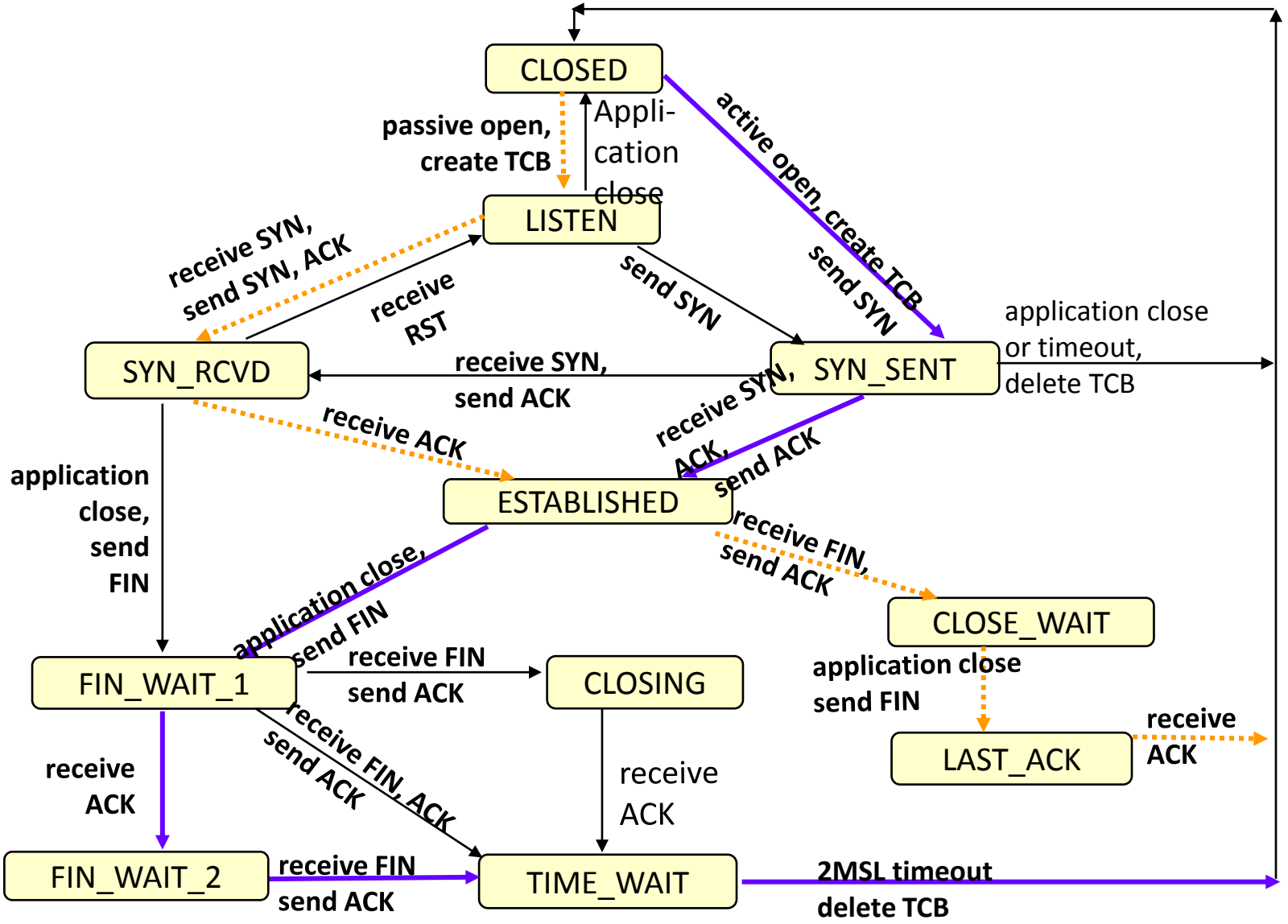
  } čekanje na završetak pasivno zatvaranje (poslužitelj)

  - Fin_wait1
  - Fin_wait2
  - Time_wait
  - Close_wait

  } čekanje na završetak aktivno zatvaranje (klijent)

- Prijelazi – TCP segmenti (upravljačke poruke ili podaci)

Automat stanja - TCP

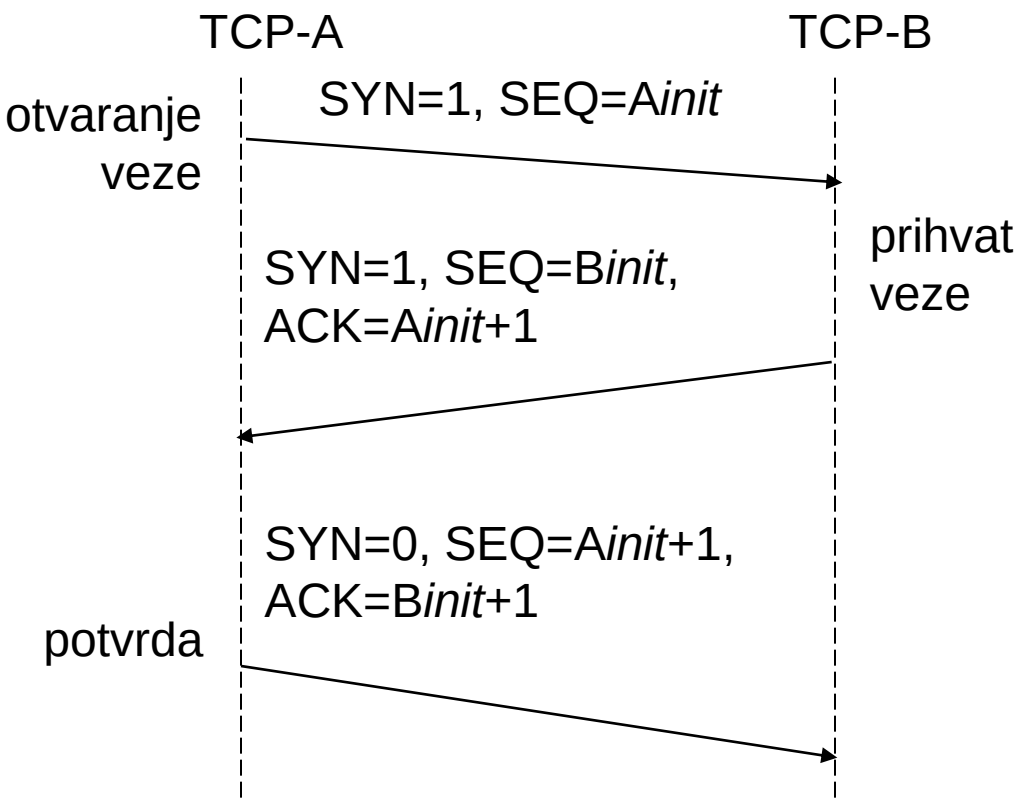


Uspostava TCP-veze

- Procedura uspostave veze:
  - svaka od strana mora poslati tzv. SYN segment u kojem objavljuje parametre bitne za vezu, npr.:
    - početni broj za numeraciju okteta
    - veličinu prozora
    - izvorišna i odredišna vrata
    - (može biti i drugih parametara, u opcijama)
  - strana koja primi SYN segment mora poslati potvrdu (ACK) da ga je primila

Uspostava TCP-veze

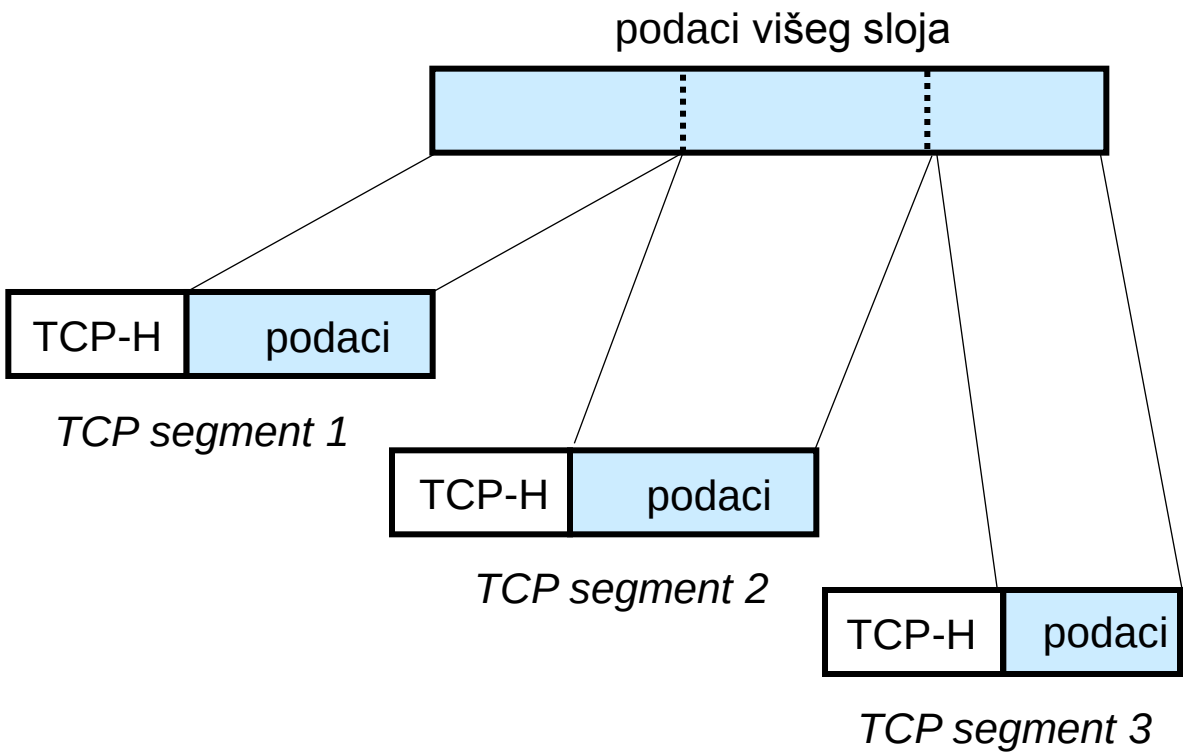
- prilikom uspostave veze, inicijaliziraju se parametri komunikacije

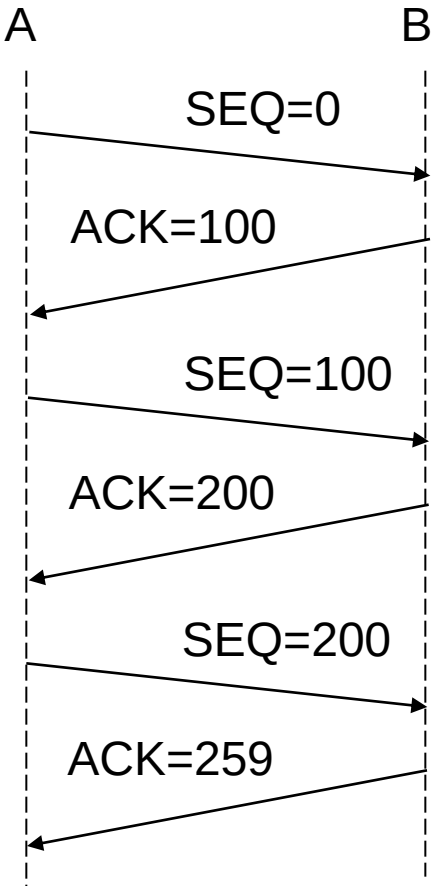


- nakon uspješne uspostave veze, slijedi razmjena podataka aplikacijskih procesa

Transport niza podatkovnih okteta - primjer

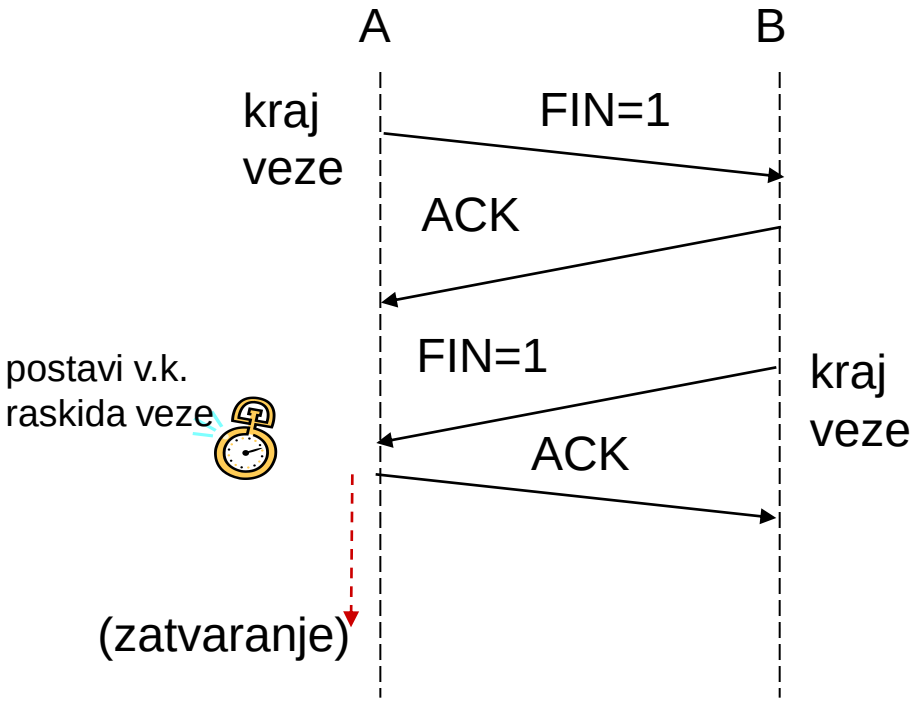
podaci 259 okteta
MSS = 100 okteta
segment 1: okteti 0-99 -> redni broj = 0
segment 2: okteti 100-199 -> redni broj = 100
segment 3: okteti 200-258 -> redni broj = 200





Raskid TCP-veze

• kad aplikacije nemaju više podataka za poslati, započinju “dogovor” o raskidu veze



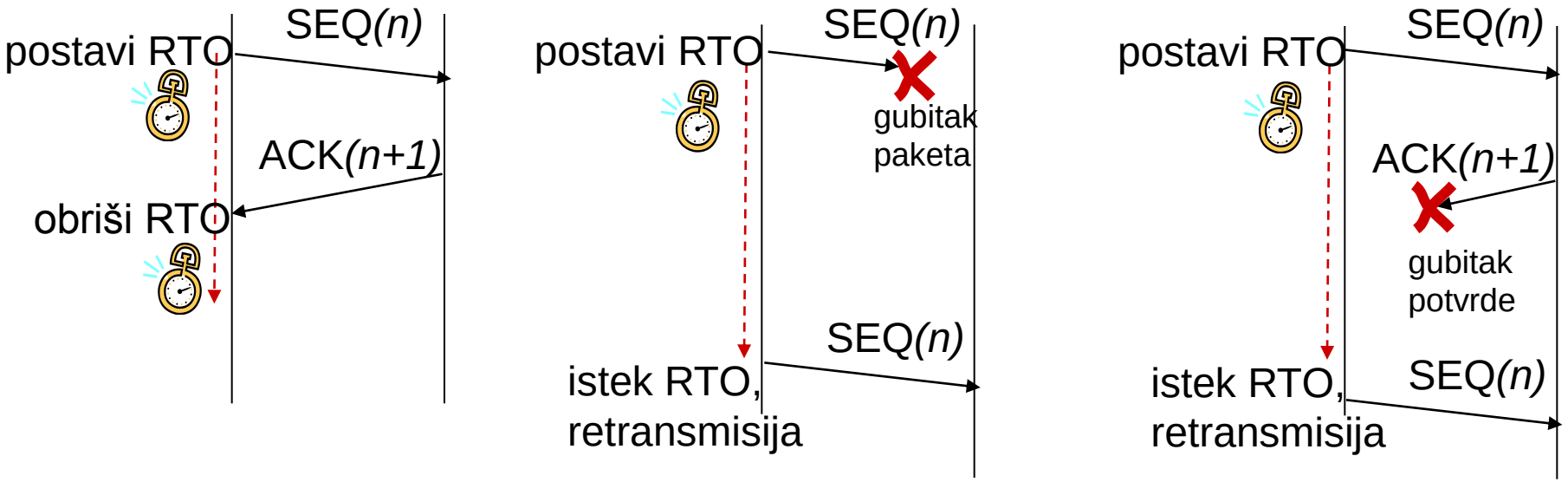
TCP – pouzdan prijenos

- Segmenti se u mreži mogu izgubiti iz više razloga
  - greške u prijenosu koje se ne daju ispraviti
  - usmjerivači ispuštaju pakete pri preopterećenosti linkova
  - petlje u usmjeravanju i TTL mehanizam

- TCP garantira da će poduzeti sve mjere da nadomjesti oktete izgubljene u mreži
  - ponavljanje slanja nakon određenog vremenskog perioda (*retransmission timeout, RTO*)
  - eksponencijalno produljivanje vremenskog intervala za ponavljanje slanja (*exponential backoff*)

Vremenska kontrola retransmisije

- TCP pošiljatelj postavlja RTO prilikom slanja segmenta
- ako potvrda za segment ne stigne do trenutka isteka RTO, pošiljatelj smatra segment izgubljenim i šalje ga ponovno

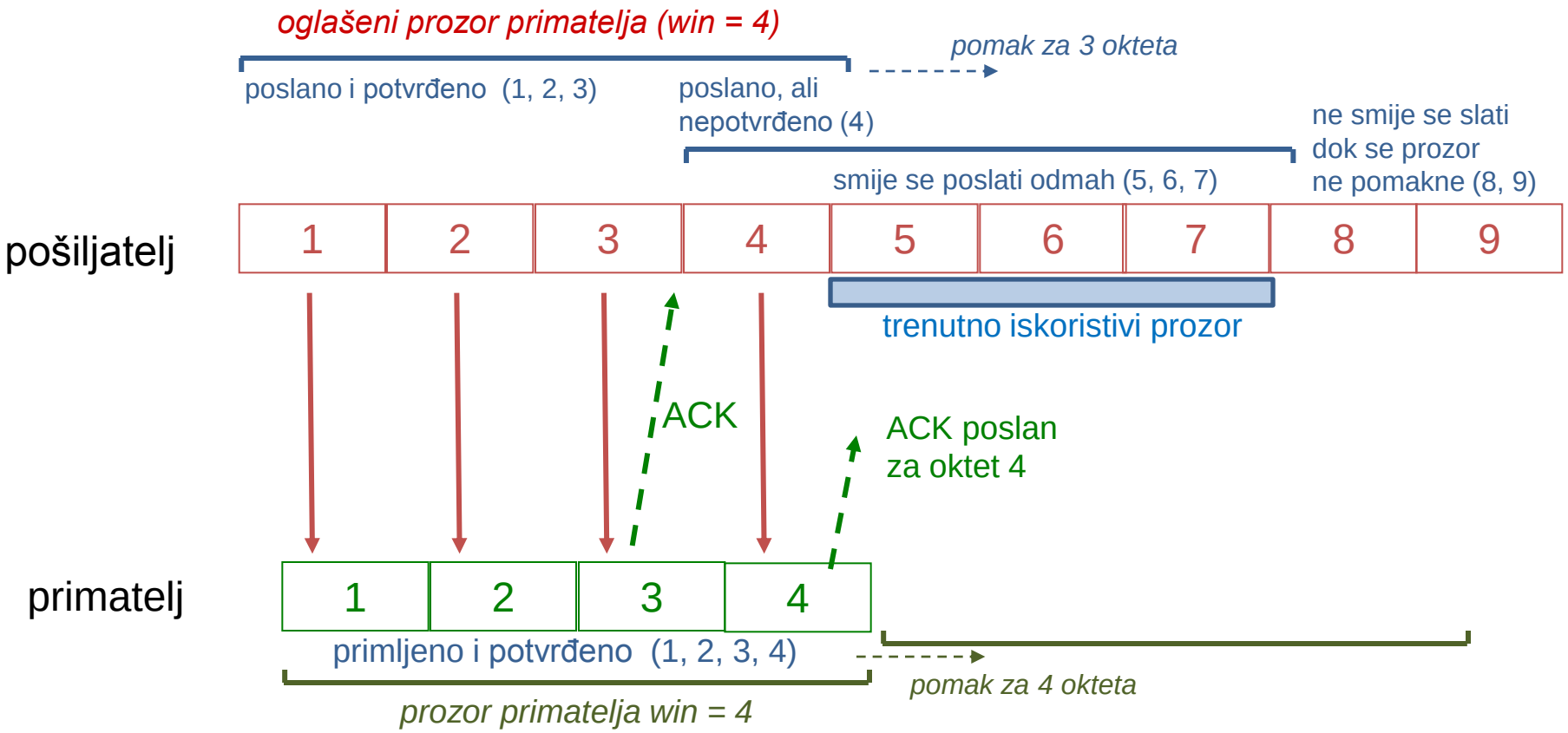


TCP – kontrola toka

- Usklađivanje brzine slanja i primanja
  - što ako pošiljatelj šalje brže nego što primatelj može obrađivati?
- TCP koristi mehanizam klizećeg prozora
  - prozor je broj okteta koje pošiljatelj u danom trenutku smije poslati a da ne čeka potvrdu bilo kojeg od tih okteta
  - prozor objavljuju obje strane i to u sklopu potvrda koje šalju
- Primjer
  - pretpostavimo da je pošiljatelj primio potvrdu za oktet N-1 i u potvrdi je objavljen prozor W
  - pošiljatelj smije poslati sve oktete do N+W bez ikakve daljnje potvrde
  - međutim, ne smije poslati oktete od N+W nadalje sve dok mu se barem jedan dio okteta prije N+W ne potvrdi

Mehanizmi upravljanja tokom

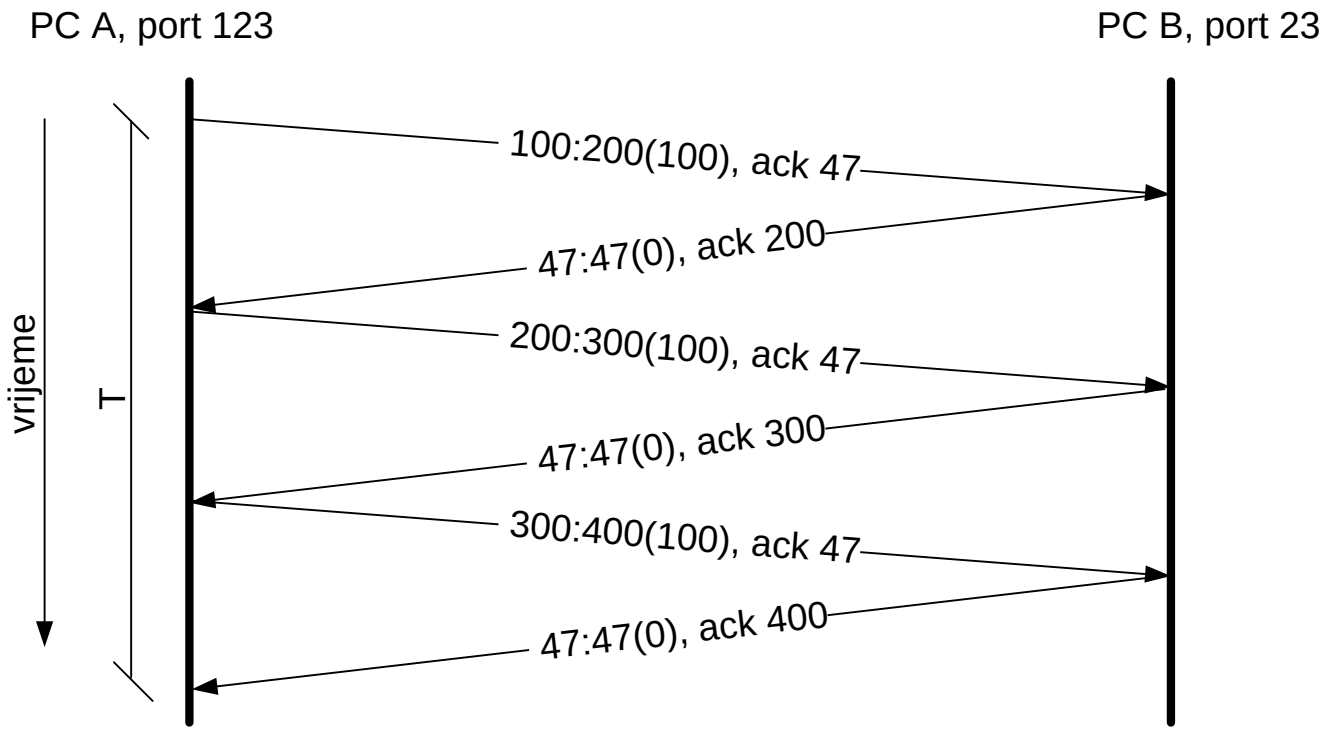
- prilagodba brzine slanja prema brzini primanja primatelja i stanju u mreži
- klizeći prozor - veličinom prozora je definiran max. broj nepotvrđenih okteta koje pošiljatelj može poslati odjednom



TCP – kontrola toka

- Primjer:

Oznake: od:do(duljina), ack “oktet koji se potvrđuje”

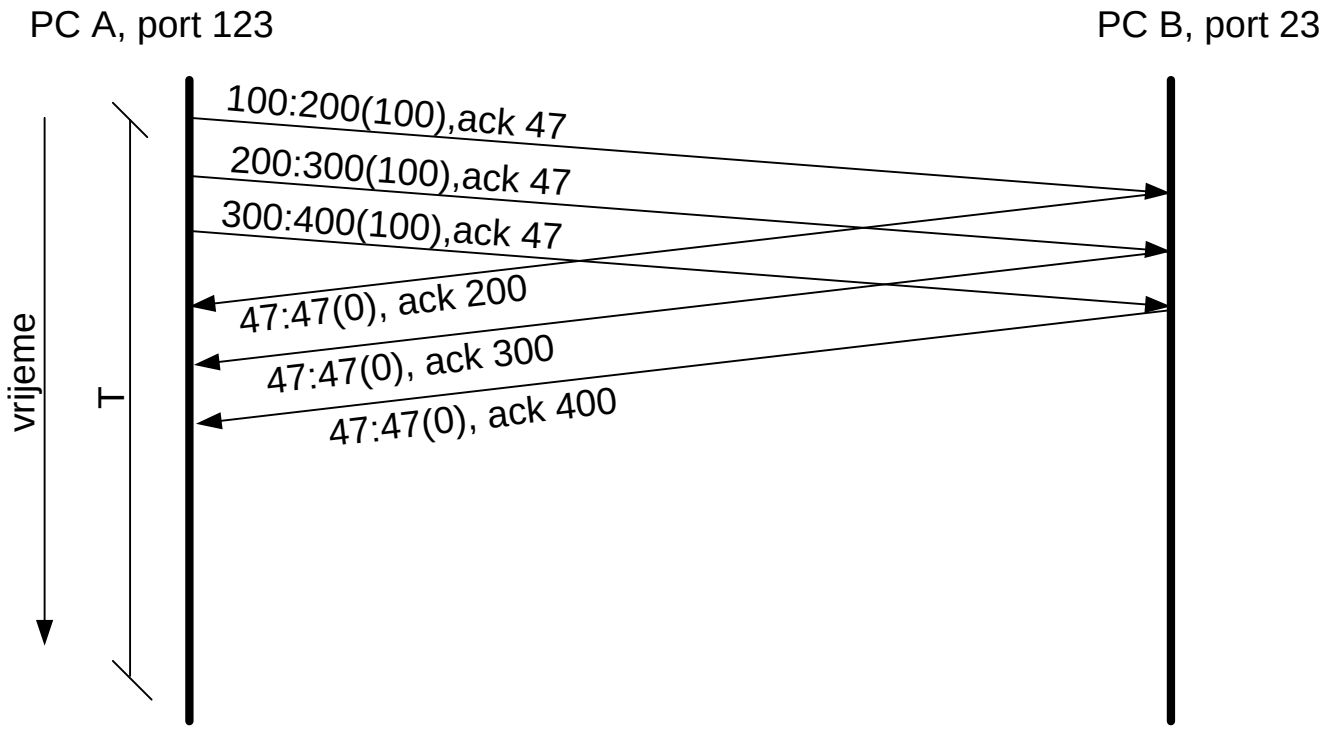


Je li moguće prenijeti segmente sa slike u još kraćem vremenu?

TCP – kontrola toka

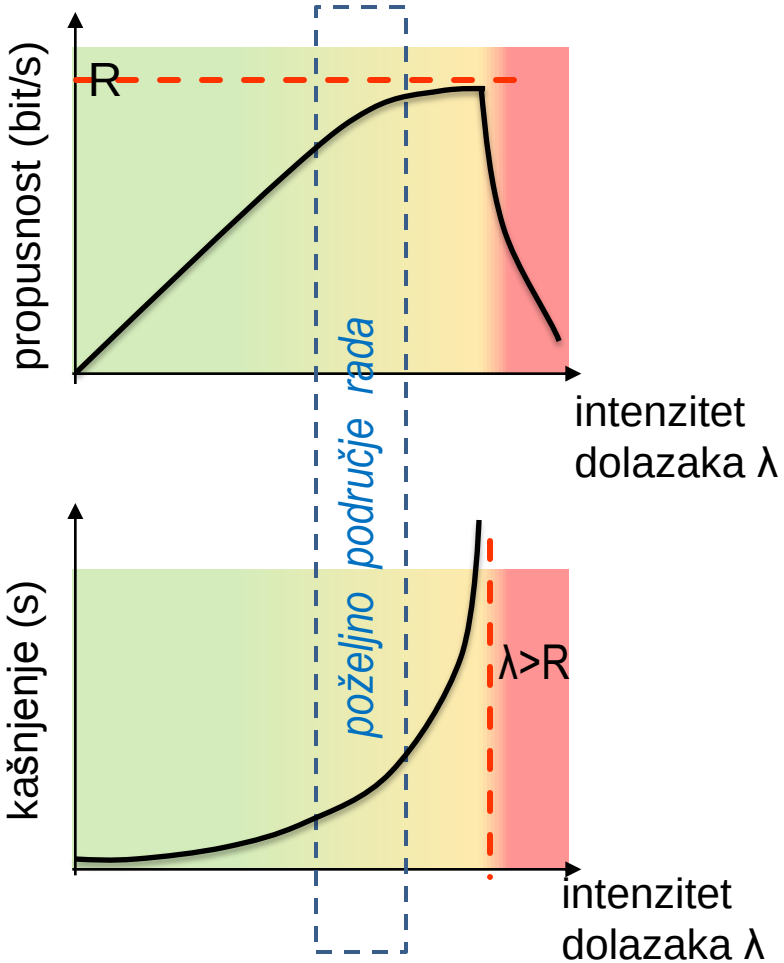
- Pošiljatelj smije poslati više segmenata odjednom!

Oznake: od:do(duljina), ack “oktet koji se potvrđuje”



TCP – kontrola zagušenja

- uvode se složeni mehanizmi kojima se TCP veza prilagođava (pretpostavljenom) zagušenju u mreži



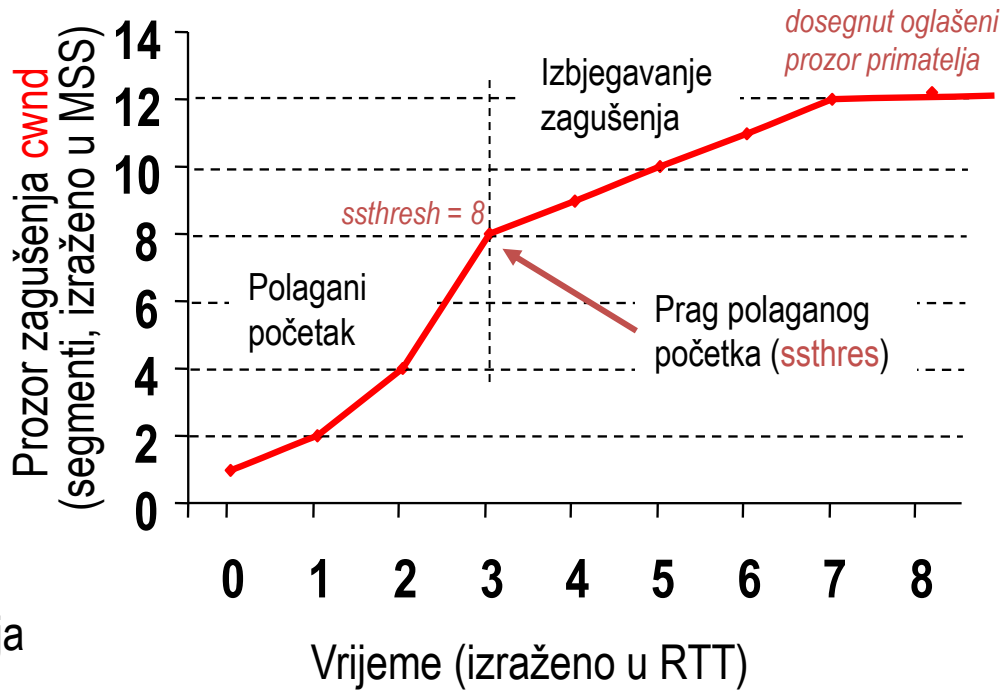
- faze nastanka zagušenja:
  - normalan rad: λ dovoljno manje od R
  - početak zagušenja: λ se približava R
  - nastupa zagušenje: λ>R
    - drastičan pad performansi!
- pitanja za TCP:
  - kako detektirati zagušenje?

  ideja: učestali gubici, učestali istek RTO
  - što napraviti?

  ideja: naglo smanjiti brzinu slanja, a zatim je postupno povećavati

TCP – kontrola zagušenja

- TCP veza se nastoji dinamički prilagoditi raspoloživoj širini pojasa, uz "poštenu" podjelu s drugim vezama
  - problem je što nijedan pošiljatelj ne zna, niti može odrediti "svoj dio"
  - ideja – svatko za sebe postupno povećava brzinu slanja i prati stanje – cilj je da se opće stanje "stabilizira" i zagušenje popusti
  - mehanizam: pošiljatelj uvodi još jedan "prozor": prozor zagušenja
  - efektivni prozor pošiljatelja = min{oglašeni prozor primatelja, prozor zagušenja}
  - rukovanje prozorom zagušenja:
    - eksponencijalni rast u početku (faza "polagani početak")
    - linearni rast kasnije (faza "izbjegavanje zagušenja")
  - postoje i drugi mehanizmi upravljanja zagušenjem (i dalje se razvijaju!)



TCP – kontrola zagušenja

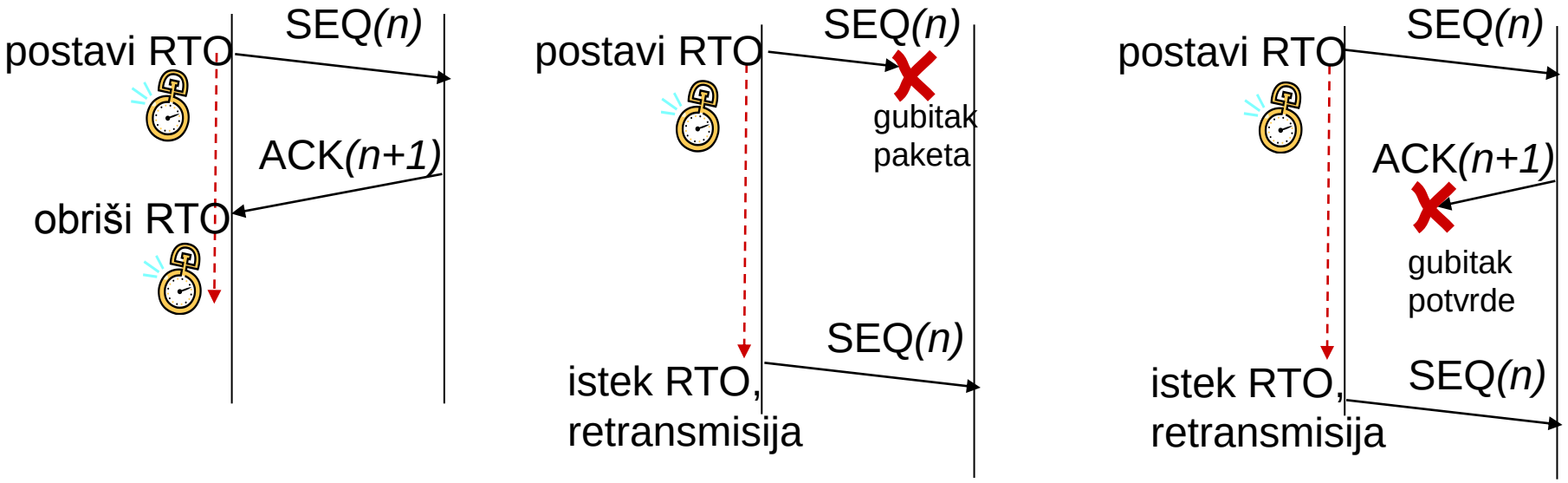
- Dva upravljačka mehanizma
  - polagani početak (*Slow Start*)
  - izbjegavanje zagušenja (*Congestion avoidance*)
- Upravljanje zagušenjem odvija se pomoću prozora
  - prozor zagušenja *cwnd* (*congestion window*)
  - prozor primatelja *rwnd* (*receiver advertised window*)
  - prag polaganog početka *ssthresh* (*slow start threshold size*) služi za osvježavanje vrijednosti *cwnd*
- Prozor pošiljatelja *swnd* (*sender window*)
  - swnd = min{rwnd, cwnd}

Kako TCP ustanovljuje gubitak paketa?

- Vremenska kontrola retransmisije (*Retransmission Time-Out*, RTO)
  - slučaj kad nema potvrde do isteka vremenske kontrole
  - detekcija gubitka (paketa ili potvrde)

Vremenska kontrola retransmisije

- TCP pošiljatelj postavlja RTO prilikom slanja segmenta
- ako potvrda za segment ne stigne do trenutka isteka RTO, pošiljatelj smatra segment izgubljenim i šalje ga ponovno



Izračunavanje RTO za uspješno poslane pakete

- vrijednost RTO se izračunava dinamički

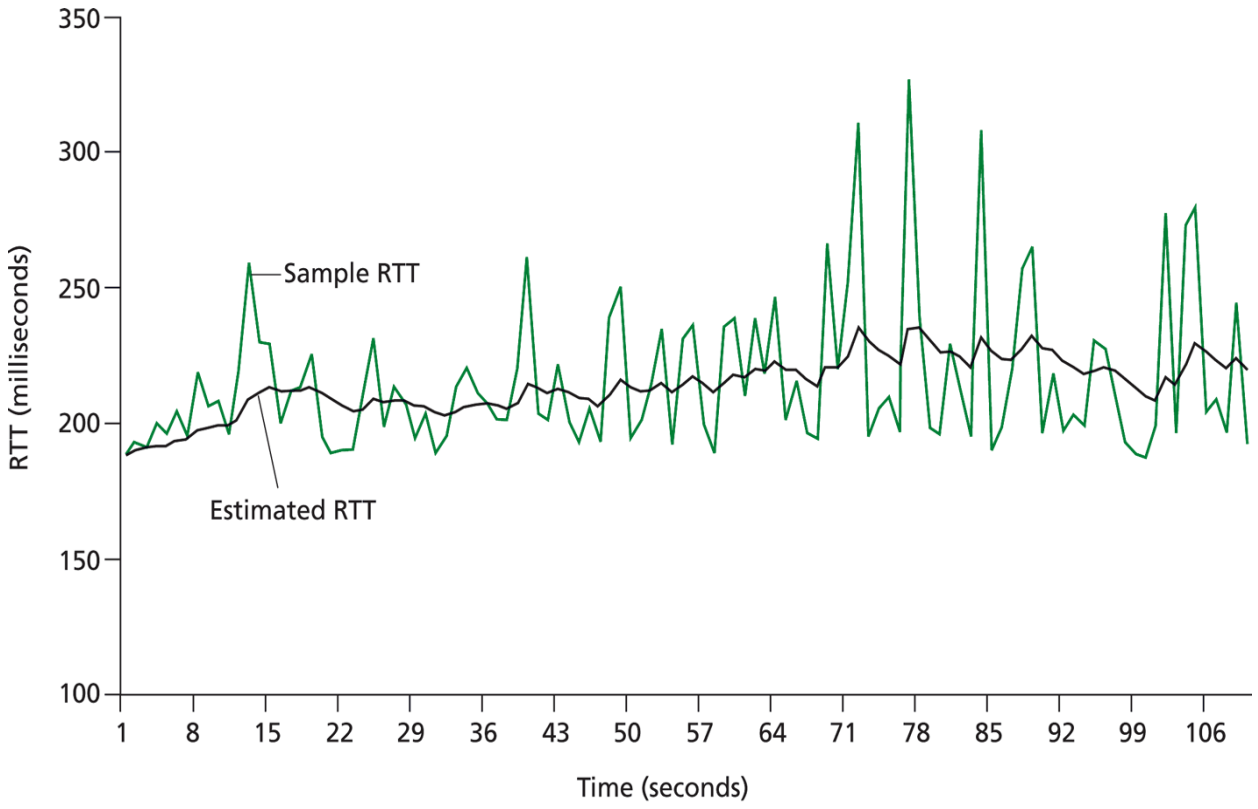
$$RTO = RTT + 4 * D$$

  - RTT – varijabla koja služi kao procjena prosječnog obilaznog vremena (*Round Trip Time)* do odredišta
  - D – srednja devijacija RTT-a, računa se kao
    D = srednja vrijednost $| x_i - x |$

Praćenje vrijednosti RTT

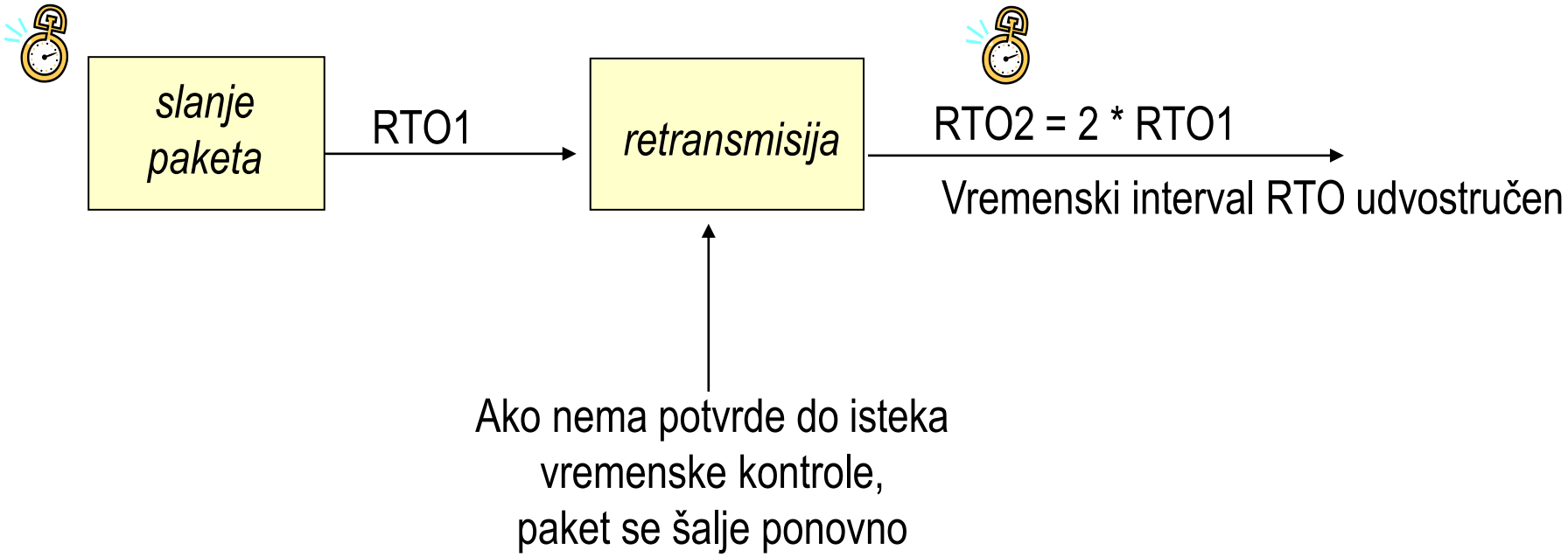
- velike vrijednosti RTT povećavaju srednju vrijednost
- velika odstupanja povećavaju devijaciju D

- rješenje: pri izračunu se uzimaju se samo vrijednosti RTT za uspješno potvrđene pakete



RTO za ponovljene pakete

- Karnov algoritam:
  - **ne** osvježavaj RTT za segmente koji se šalju ponovno (~~RTO = RTT + 4 * D~~)
  - umjesto toga, udvostručuj RTO prilikom svakog isteka vremenske kontrole (RTO2 = 2 * RTO1), sve dok neki sljedeći segment uspješno ne prođe od prve



Varijable stanja TCP-a

- najveća veličina segmenta (*Maximum Segment Size, MSS*)
  - max. broj podatkovnih okteta koje pošiljatelj smije poslati u jednom TCP segmentu
  - vrijednost MSS ovisi o mrežnoj tehnologiji; odgovara duljini podatkovnog polja okvira na sloju linka, umanjenoj za duljinu IP zaglavlja i TCP zaglavlja
- prozor primatelja (receiver window, *rwnd*)
  - broj okteta koje primatelj objavljuje da može primiti
- prozor zagušenja (congestion window, *cwnd*)
  - inicijalno 1 MSS, povećava se sa svakom potvrdom; najprije eksponencijalno (do vrijednosti praga polaganog početka (*ssthres*) u fazi polaganog početka), a zatim linearno (u fazi izbjegavanja zagušenja)
- prozor pošiljatelja (sender window, *swnd*)
  - najveći broj segmenata koje pošiljatelj smije poslati; definira se kao manja od vrijednosti *rwnd* i *cwnd*
- prag polaganog početka (slow start threshold, *ssthres*)

Upravljanje zagušenjem

- Kad ustanovi gubitak paketa, TCP pošiljatelj pretpostavlja da je to posljedica zagušenja i drastično smanjuje prozor zagušenja, čime usporava brzinu slanja

- ako dođe do isteka vremenske kontrole retransmisije:
  - prag polaganog početka ssthres postavlja se na polovicu veličine prozora prije gubitka paketa, odnosno:

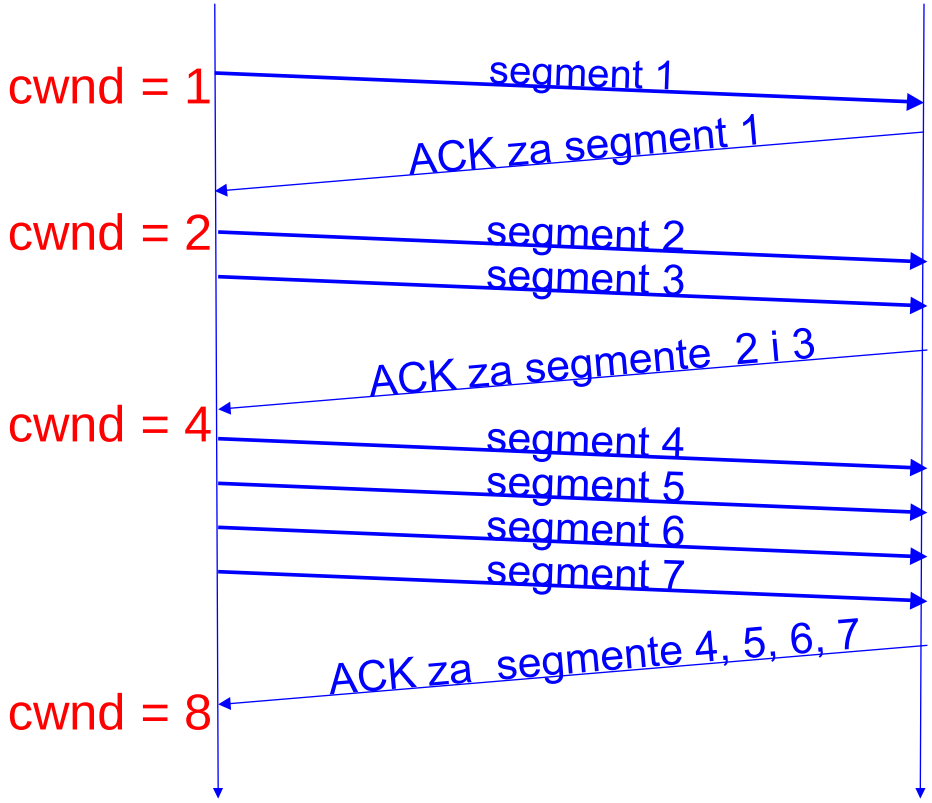
    ssthresh = max { swnd/2 ; 2*MSS }
  - prozor zagušenja cwnd se postavlja na **1 MSS**

- pošiljatelj se vraća u fazu **polaganog početka**

Polagani početak

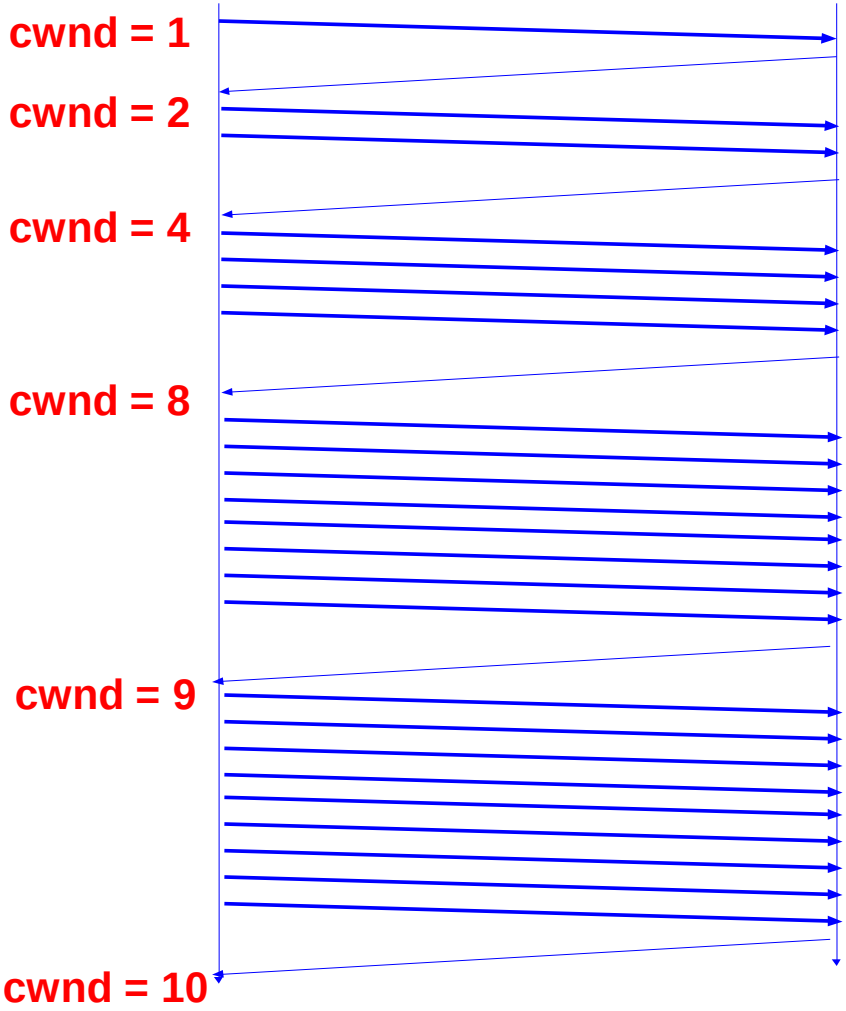
- veličina cwnd se postavlja na cwnd = 1 MSS
- veličina cwnd se povećava za 1 MSS po primitku svake nove potvrde
- cwnd raste **eksponencijalno**
- faza polaganog početka završava kada veličina prozora dosegne prag polaganog početka (ssthresh)

(U ovom primjeru, ssthresh = 8)

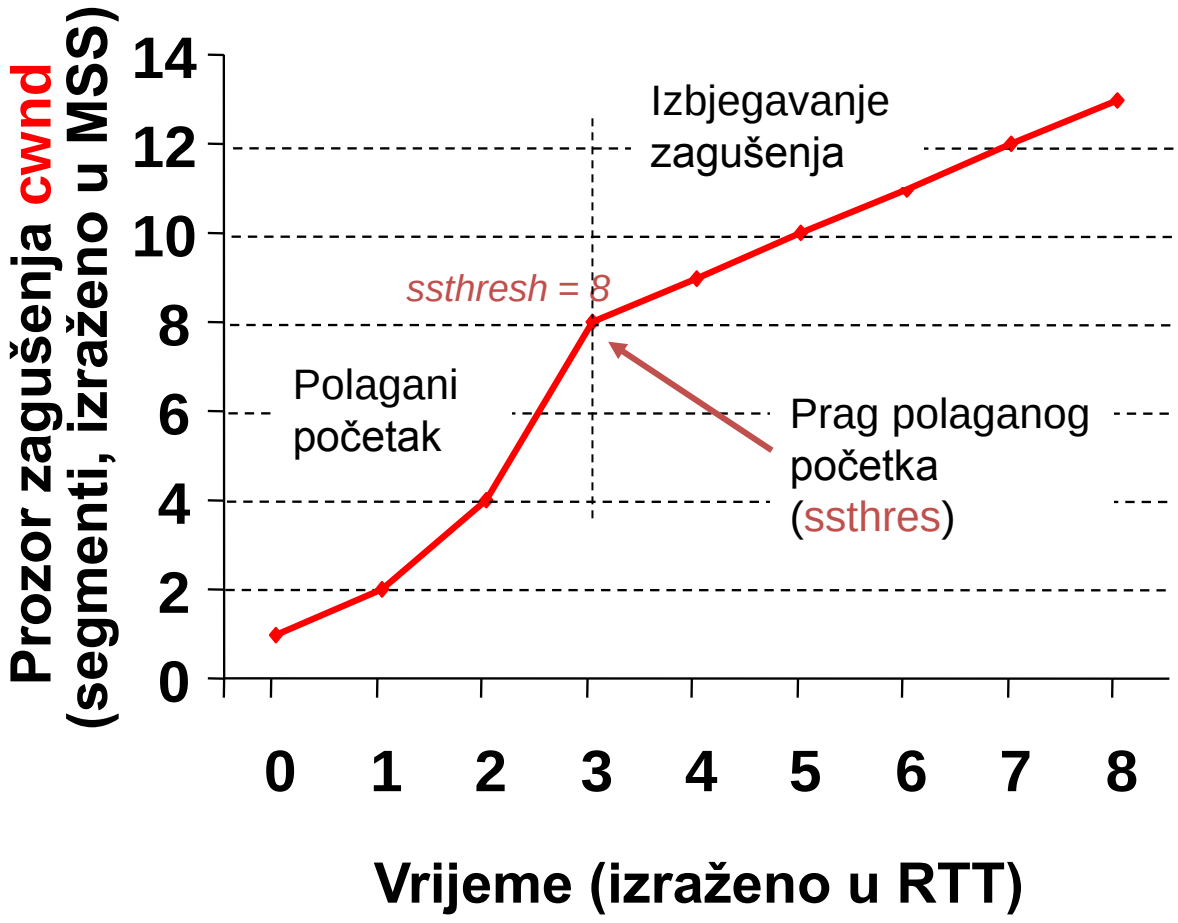


Izbjegavanje zagušenja

- nakon što cwnd dosegne prag polaganog početka (ssthresh) TCP prelazi u fazu **izbjegavanja zagušenja**
- s primitkom svake nove potvrde, pošiljatelj povećava cwnd za 1/cwnd paketa
- za vrijeme izbjegavanja zagušenja cwnd raste **linearno**
  - 1/2 MSS za RTT ako se potvrdi svaki drugi paket
  - 1 MSS za RTT ako se potvrdi svaki paket



Polagani početak i izbjegavanje zagušenja



Daljnja poboljšanja upravljanja zagušenjem

- Brza retransmisija (*Fast Retransmission*)
- Brzi oporavak (*Fast Recovery*)

Brza retransmisija

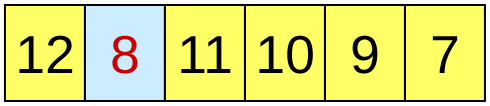
- Čekanje na istek vremenske kontrole RTO može predugo potrajati
  - ideja: može li se retransmisija pokrenuti prije?
- Da, to je *brza retransmisija*

Upravljanje zagušenjem - brza retransmisija

- brza retransmisija pokreće se kada pošiljatelj primi više dvostrukih potvrda (obično, 3)
- daljnje poboljšanje nakon brze retransmisije je brzi oporavak (*Fast recovery*)
- uočimo razliku u odnosu na istek vremenske kontrole (za takav slučaj slijedi polagani početak)
  - istek vremenske kontrole događa se kada paketi više uopće ne prolaze
  - brza retransmisija događa se kad je neki od paketa izgubljen, ali kasniji paketi prolaze
  - brza transmisija se radi dok vremenska kontrola potvrde još nije istekla jer nema potrebe za polagani početak

Detekcija gubitka paketa na temelju dvostrukih potvrda i brza retransmisija

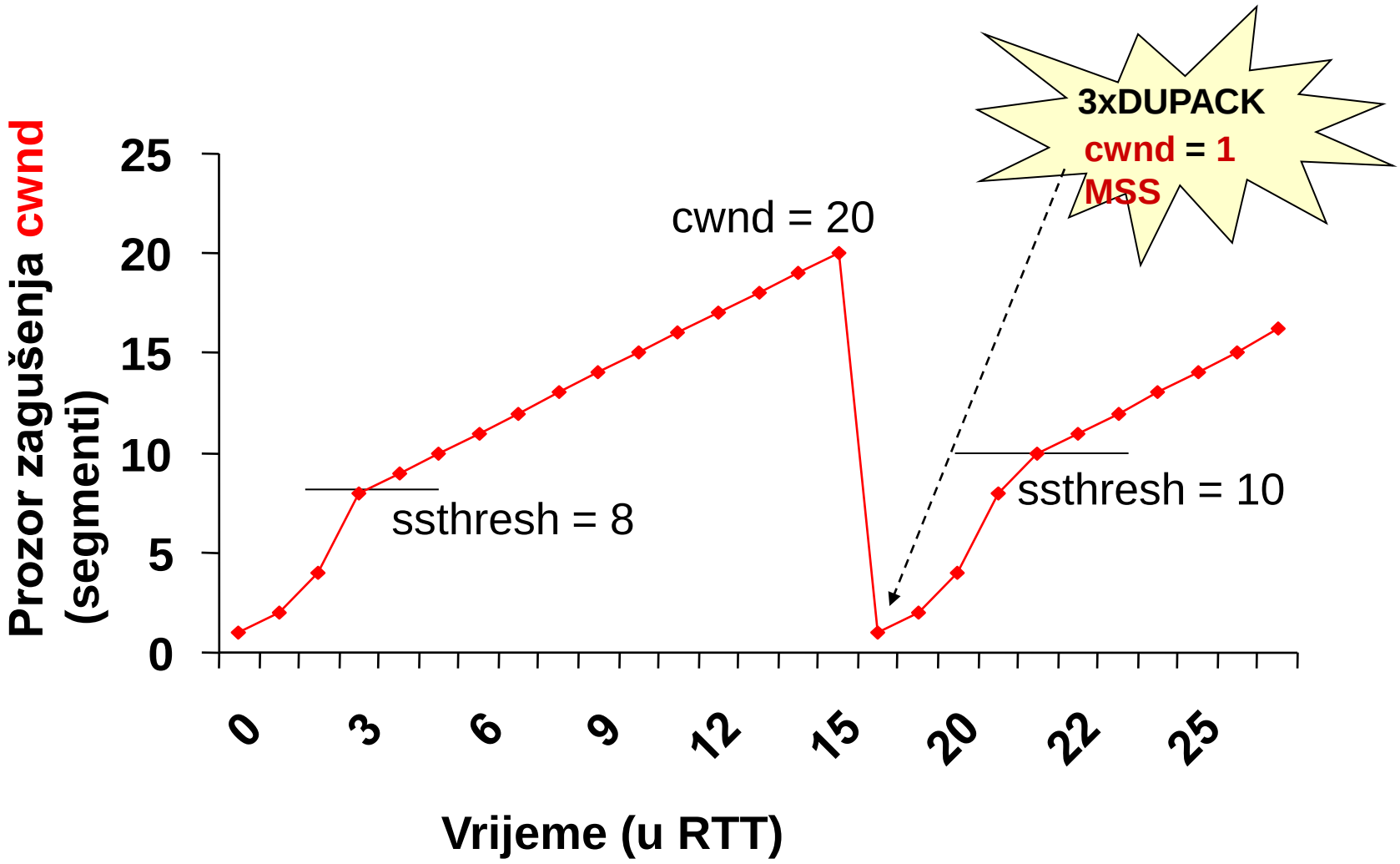
- Dvostruke potvrde (***dup****licate* ***ack****nowledgement,* "DUPACK") se šalju u slučajevima:
  - gubitka paketa, ili
  - dostave paketa izvan redosljeda
- TCP pošiljatelj pretpostavlja da je došlo do gubitka paketa ako primi **tri dvostruke ("već viđene") potvrde zaredom**



3 dvostruke potvrde se isto šalju u slučaju kada je paket isporučen najmanje tri mjesta dalje od svog predviđenog položaja

Napomena: brza retransmisija ima smisla samo ako paketi stižu uz uglavnom očuvani redosljed.

Upravljanje zagušenjem (bez brzog oporavka)



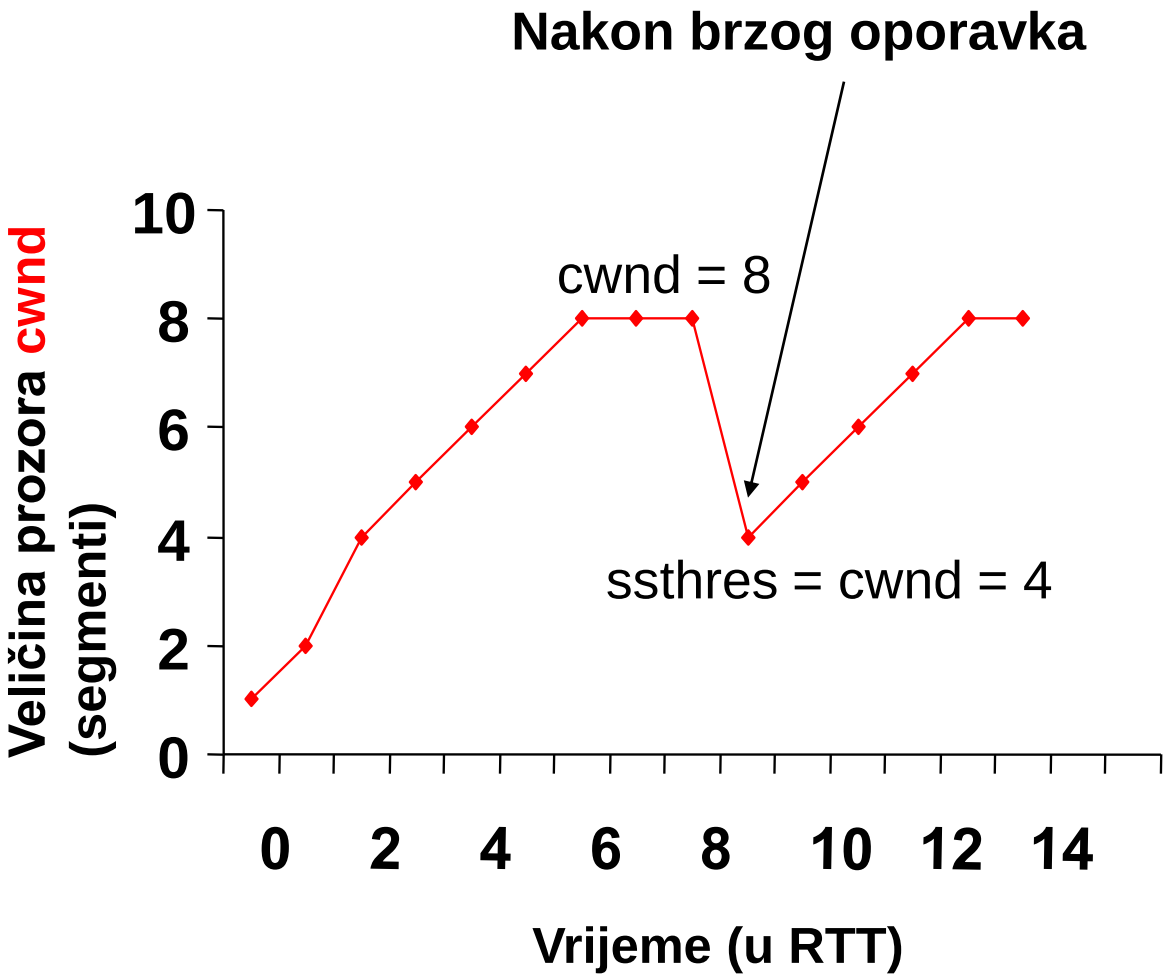
Brzi oporavak

- ssthresh = cwnd/2, a minimalno 2*MSS
- ponovno slanje segmenta koji nedostaje (to je brza retransmisija)
- postavljanje cwnd

  cwnd = ssthresh + broj dvostrukih potvrda (3) *MSS
- kad stigne nova potvrda:

  cwnd = ssthresh
  - ulazi se u fazu izbjegavanja zagušenja (linearni rast)
  - prozor zagušenja je prepolovljen (a ne vraćen na 1 MSS kao prije!)

Upravljanje zagušenjem – brzi oporavak

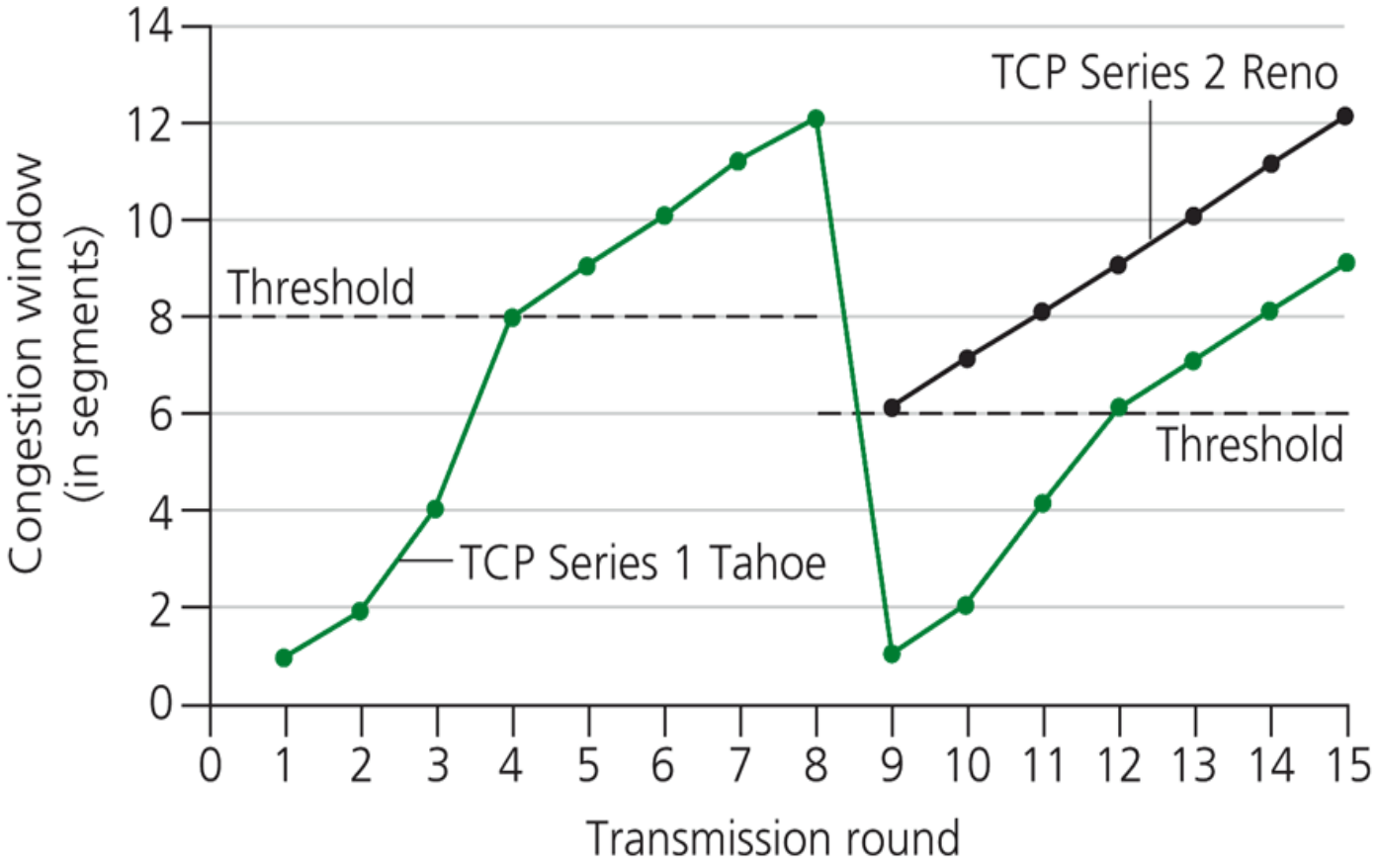


Nakon brze retransmisije i brzog opravka, veličina prozora se prepolavlja.

Razne izvedbe TCP-a

- TCP Tahoe
  - polagani početak (*slow start*)
  - izbjegavanje zagušenja (*congestion avoidance*)
  - brza retransmisija (*fast retransmit*)
- TCP Reno (temelj za većinu današnjih izvedbi TCP-a)
  - TCP Tahoe + brzi oporavak (*fast recovery*)
- TCP New-Reno
  - TCP Reno + modificirani brzi oporavak (RFC 3782); ostaje u brzom oporavku dok ne povrati sve izgubljene pakete u prozoru
- TCP-Selective Acknowledgements (TCP-SACK)
  - TCP Reno + selektivne potvrde
  - TCP-SACK omogućuje primatelju da javi pošiljatelju detaljniju informaciju o svim segmentima koji su uspješno primljeni, na temelju čega pošiljatelj može ponovno poslati samo one segmente koji su bili izgubljeni

Usporedba raznih izvedbi TCP-a



IZVOR: J. Kurose, K. Ross, "Computer Networking: A Top Down Approach Featuring the Internet", Pearson Addison-Wesley 2005.

Ograničenja protokola TCP

- Što TCP ne radi?
  - nema mehanizme za sigurnost i privatnost podataka
    - postoje razna rješenja na raznim slojevima
  - ne vodi računa o granicama poruke
    - isporučuje niz okteta, neovisno o tome kako aplikacija pošiljatelja grupira podatke (ne vidi granice poruke)
  - ne garantira isporuku višem sloju
    - ali se potrudi prije nego konačno odustane

Primjena protokola TCP

- tamo gdje je aplikaciji najvažnija pouzdanost
  - transfer datoteka
  - elektronička pošta
  - Web
  - transakcijske primjene
  - rad na udaljenom računalu

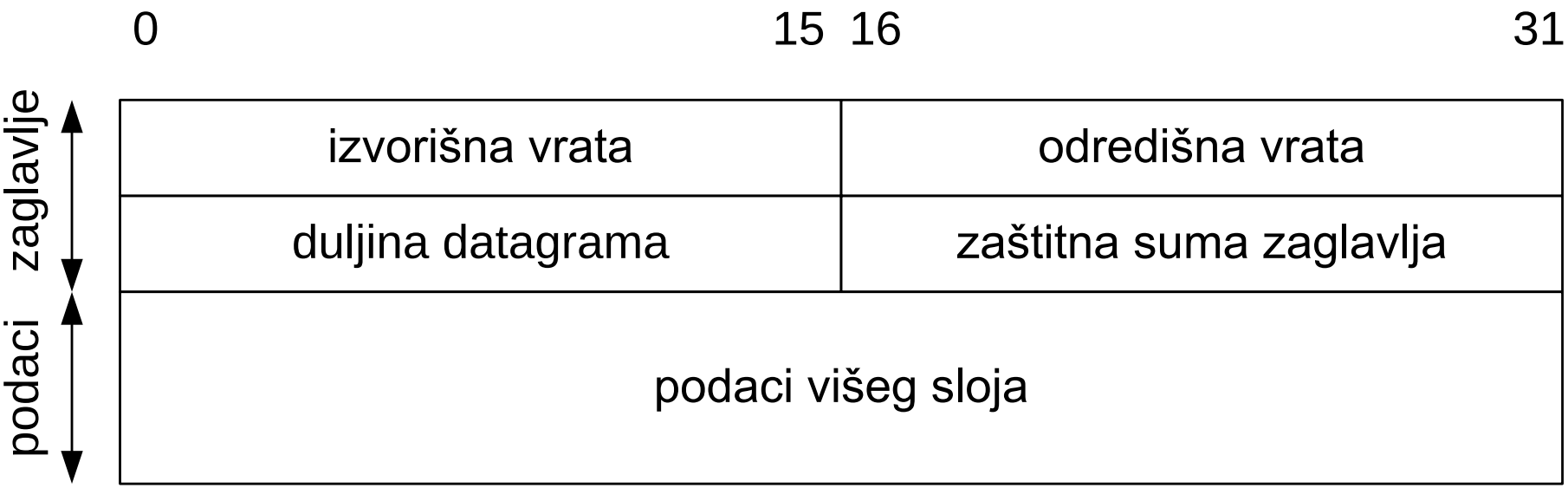
Sadržaj predavanja

- Usluga transportnog sloja
- Funkcionalnost
  - adresiranje
  - multipleksiranje
  - uspostava i raskid veze
  - kontrola toka i privremena pohrana
  - oporavak od prekida
- Protokoli transportnog sloja u Internetu
  - Transmission Control Protocol
  - User Datagram Protocol

Protokol *User Datagram Protocol*

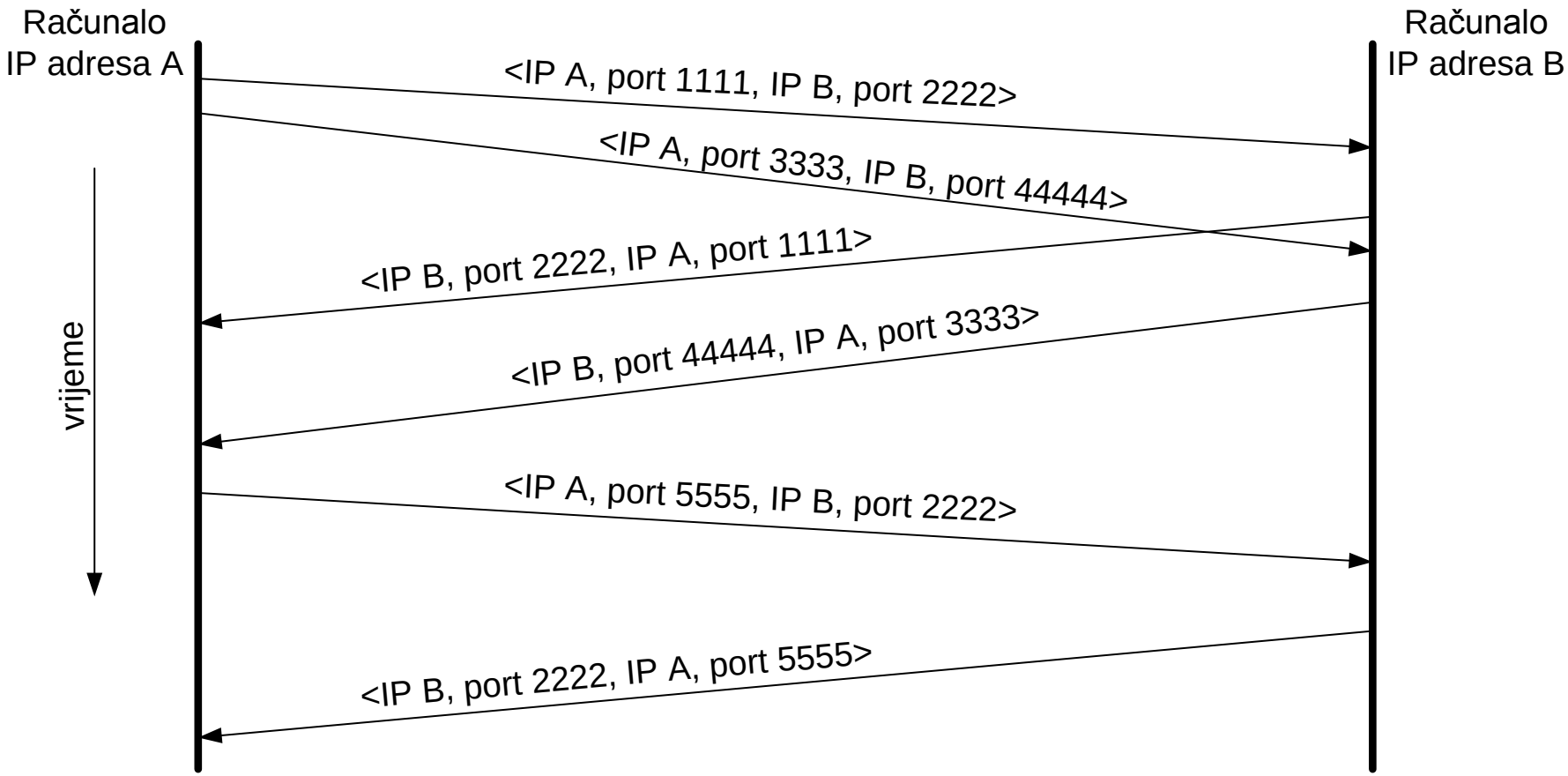
- Jednostavan transportni protokol
- Funkcije:
  - prima podatke od višeg sloja, omata ih u UDP datagram i prosljeđuje mrežnom sloju
  - minimalna funkcionalnost iznad IP-a: multipleksiranje
  - (opcionalno) radi zaštitnu sumu cijelog datagrama
- Ostale značajke:
  - nepouzdan prijenos
  - transfer blokova okteta (datagrami)
  - nema očuvanja redoslijeda
    - datagrami se isporučuju aplikaciji onim redoslijedom kojim su primljeni
  - ne pruža kontrolu toka - ako pošiljatelj prebrzo šalje, datagrami se gube

Format UDP datagrama



Napomena: brojevi UDP-vrata neovisni od brojeva TCP-vrata!

UDP – primjer multipleksiranja tokova



Oznake: <IP adresa izvora, vrata na izvoru, IP adresa odredišta, vrata na odredištu>

Ograničenja protokola UDP

- Što UDP ne radi?
  - ne uspostavlja vezu prije slanja podataka
  - ne potvrđuje primitak podataka
  - ne garantira isporuku podataka
  - ne otkriva gubitak paketa, niti radi retransmisiju izgubljenih paketa
  - ne garantira očuvanje redoslijeda
  - ne pruža kontrolu toka niti kontrolu zagušenja

OK, gdje se UDP koristi?

Primjene protokola UDP

- tamo gdje je aplikaciji dostava podataka *na vrijeme* važnija od dostave *svih* poslanih podataka (prije ili kasnije)
  - višemedijske aplikacije u stvarnom vremenu
  - na primjer: internetska telefonija, višekorisničke igre
- pogodan za kratku komunikaciju (tamo gdje je *overhead* uspostave veze neprihvatljiv)
  - brzi zahtjev/odgovor
  - na primjer: upiti za razlučivanje adrese (DNS), dinamička dodjela adrese (DHCP)
- višeodredišne primjene i difuzija
  - način komunikacije 1:n ili n:m

Osvrt na izbor transportnog sloja u aplikacijama

- Kako transportni sloj u Internet mreži utječe na maksimalnu brzinu kojom aplikacije mogu međusobno razmijenjivati podatke?

- Ako TCP pošiljatelj želi poslati veliku količinu podataka primatelju, o čemu sve ovisi brzina kojom će podaci biti preneseni?
  - ponavljanje slanja
  - kontrola toka
  - kontrola zagušenja

- A kod UDP-a?
  - aplikacije moraju same voditi računa o ispuštenim podacima i same moraju prilagođavati brzinu uvjetima u mreži