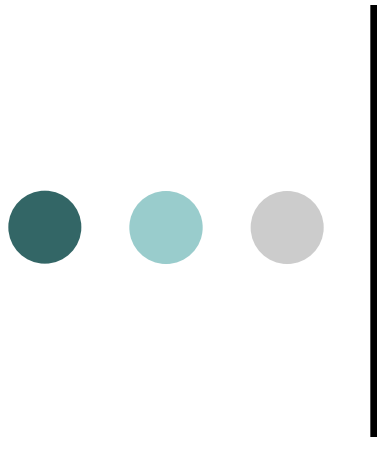


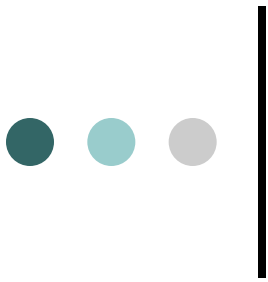
# Multimedijalno inženjerstvo – master strokovne studije



## Digitalni komunikacioni sistemi: Lekcija 7: Pouzdan prenos podataka

zima 2019/2020

Branimir M. Trenkić



## ***Pouzdan prenos podataka***

- redundantnost preko pažljive retransmisije –
  - brojanje paketa & potvrđivanje –
  - dva protokola: stop-and-wait i protokol kliznog prozora -
- vremenske kontrole i vreme odziva (RTT) -



# Problem

- Dato: ***Mreža sa ne-garantovanom uslugom prenosa*** (***best-effort network***) u kojoj:
  - Paketi mogu biti slučajno ***izgubljeni u prenosu***
  - Paketi mogu biti ***primljeni u proizvoljnom redosledu***
  - ***Kašnjenje*** paketa je ***promenljivo*** (baferovanje)
  - Paketi mogu biti čak i ***duplirani u prenosu***



# Problem

- **Predajnik *S*** u **prijemnik *R*** žele **pouzdanu komunikaciju**
  - Aplikacija (na *R*) želi prijem svih bajtova podataka u potpuno **istom redosledu** u kojem su oni i poslani (sa *S*)
  - Svaki bajt mora biti **isporučen samo jednom**
- Ove funkcionalnosti se obezbeđuju, u principu, **pouzdanim transportnim (ili sloj veze) protokolom**
  - Aplikacija je „slojevito iznad“ ovih protokola



# Predloženi plan

## ○ Predajnik

- ***Svaki paket*** sadrži ***redni broj*** (koji se sekvencijalno inkrementira)
- ***Nakon slanja*** paket, **sačuvati** (***xmit- vreme, paket***) ***na listi ne-potvrđenih paketa***
- ***Prijemom potvrde*** (ACK paket) sa odredišta sa određenim rednim brojem, **ukloniti** odgovarajući element ***iz liste ne-potvrđenih paketa***
- **Periodično pretraživati listu ne-potvrđenih** paketa kako bi se pronašli paketi koji su poslani pre određenog vremena

# Predloženi plan

## ○ Predajnik

- **Ponovo ih poslati**, ažurirati *xmit*- vreme za slučaj ponovne provere
- „pre određenog vremena“- uslov za ponovno slanje:

$$xmit < sada - \underline{timeout} (RTO)$$

(**RTO** – Retramssmission TimeOut)

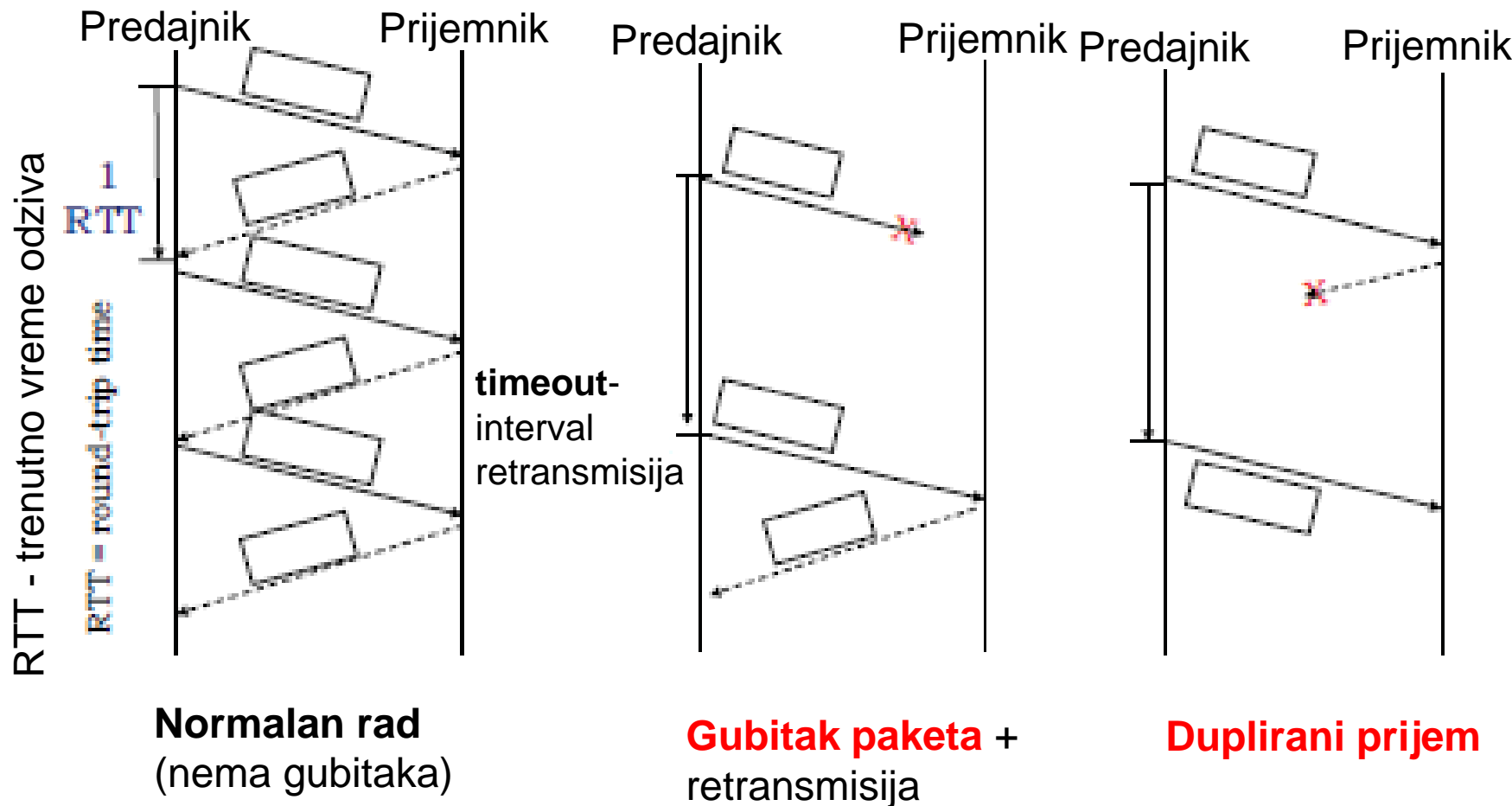


# Predloženi plan

## ○ Prijemnik

- *Slati ACK paket za svaki primljeni paket podataka, referenciran na redni broj primljenog paketa*
- *Isporučiti sadržaj paketa* (payload) prijemnoj aplikaciji

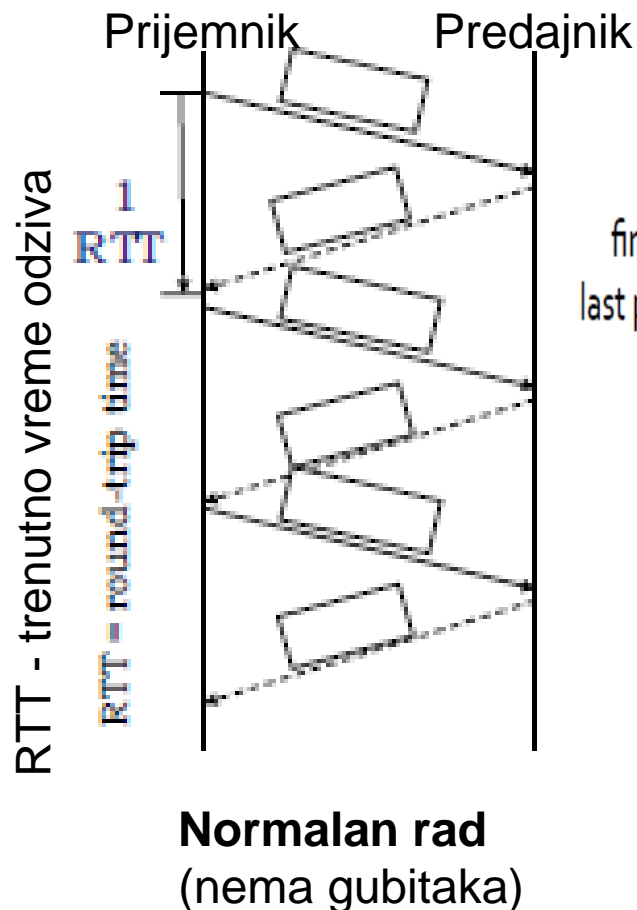
# Stop-and-Wait protokol



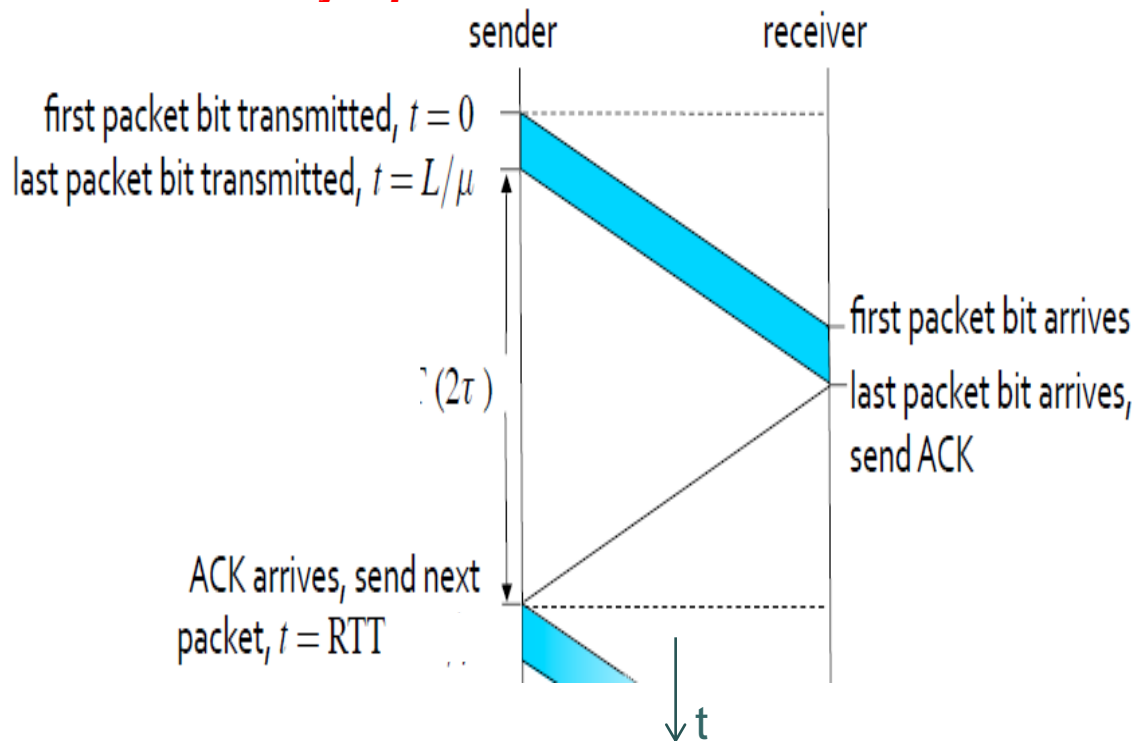
Traži “tačno jedan” dobije “najmanje jedan”



# Stop-and-Wait protokol



## *Precizniji opis RTT:*



# Revizija plana

## ○ Predajnik

- ***Svaki paket*** sadrži ***redni broj*** (koji se sekvencijalno inkrementira)
- ***Nakon slanja*** paket, ***sačuvati*** (***xmit-*** vreme, paket) ***na listi ne-potvrđenih paketa***
- ***Prijemom potvrde*** (ACK) sa odredišta sa određenim rednim brojem, ***ukloniti*** odgovarajući element ***iz liste ne-potvrđenih paketa***
- ***Periodično pretraživati listu ne-potvrđenih*** paketa kako bi se pronašli paketi koji su poslani pre određenog vremena

# Revizija plana

## ○ Predajnik

- *Ponovo ih poslati*, ažurirati *xmit- vreme* za slučaj ponovne provere
- „pre određenog vremena“- uslov za ponovno slanje:

$$xmit < sada - \underline{timeout} (RTO)$$

(**RTO** – Retramssmission TimeOut)



# Revizija plana

## ○ Prijemnik

- *Slati ACK za svaki primljeni paket, referenciran na redni broj primljenog paketa*
- *Isporučiti sadržaj paketa (payload) prijemnoj aplikaciji u redosledu definisanom rednim brojevima*
  - *Dodatak:*
  - *Pratiti redni broj sledećeg paketa čiji sadržaj treba isporučiti aplikaciji;* na taj način je jednostavno ***prepoznati duplikat u prijemu*** koji ne treba isporučiti po drugi put



# Dileme

- ***Protokol mora*** na korektan način ***upravljati gubicima***
  - ***Izgubljeni podaci: retransmisija*** će obezbediti izgubljene podatke
  - ***Izgubljeni ACK: retransmisija*** će pokrenuti drugi ACK od prijemnika
- ***Dužina bafera***
  - *Na strani predajnika*
    - Bafer ***čuva ne-potvrđene pakete***
    - Prekinuti prenos ako je prostor u baferu problem



# Dileme

## ○ *Dužina bafera*

### ● Na strani prijemnika

- Bafer **čuva pakete primljene van redosleda**
- Prekinuti potvrđivanje ako je prostor u baferu problem

## ○ *Izbor vremenske kontrole (RTO): u odnosu na RTT*

- **Previše malo:** nepotrebne retransmisije
- **Previše veliko:** loša propusnost
  - Isporuka je zaustavljena dok se čekaju nedostajući paketi

# Podešavanje retransmisionog tajmera (RTO)

- **Izbor RTO** vremena zasnovan je **na proceni** trenutnog **vremena odziva**, **RTT vremena** (*Round-Trip Time*)
- Međutim, **izračunavanje RTT** vremena je **složen proces**
- **Izmereno RTT**. Izmereno RTT vreme (u oznaci  $RTT_M$ ) **svakog paketa** je vreme od trenutka slanja do trenutka prijema potvrde za poslati paket
- U toku trajanja konekcije, protokol neprekidno meri i ažurira  $RTT_M$

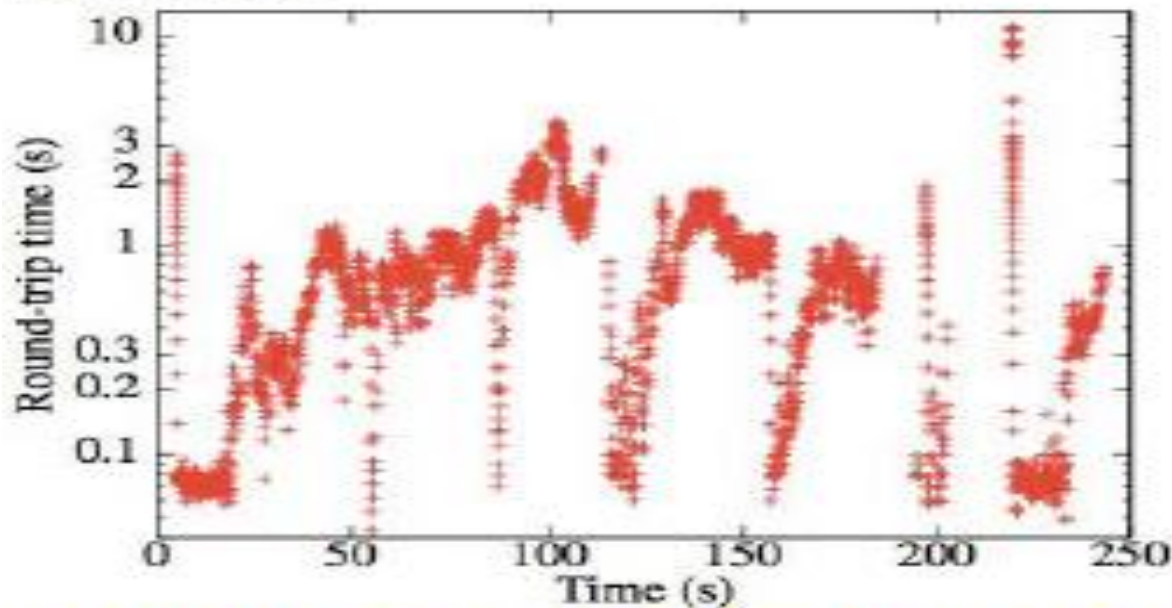
# Podešavanje retransmisionog tajmera (RTO)

- Uravnoteženo RTT.  $RTT_M$  se menja od merenja do merenja
- Na savremenom Internetu **fluktuacije ovog vremena su toliko velike** da se ono ne može koristiti za podešavanje retransmisionog tajmera  
(vidi grafikon na sledećem slajdu!)
- Problem filtriranja
- **Nisko-propusni filter (LPF)**:- *ne uračunavati promene koje nisu postojane*



# Podešavanje retransmisionog tajmera (RTO)

**Figure 1: Round-trip time during a TCP download on the Verizon LTE network in Cambridge, Mass., Oct. 14, 2011 at 3 p.m.**



<http://nms.csail.mit.edu/papers/index.php?detail=208>

# Podešavanje retransmisionog tajmera (RTO)

- Uravnoteženo RTT

- Transportni protokoli koriste **uravnoteženo RTT**, u oznaci  $RTT_s$ , koje se izračunava na sledeći način:

Početno  nema vrednost

Posle prvog merenja   $RTT_s = RTT_M$

Posle svakog sledećeg merenja (LPF) 

$$RTT_s = (1-\alpha)RTT_s + \alpha RTT_M$$




Parametar  $\alpha$  određuje kolika težina se daje staroj vrednosti  $RTT_s$ , a kolika novoizmerenom  $RTT_M$ . Tipično se usvaja  $\alpha=1/8$ .

Ako je  $RTT_M$  manje od trenutnog  $RTT_s$ ,  **$RTT_s$  će biti smanjeno** i obrnuto. Međutim, ova promena neće biti nagla, zbog  $\alpha$ , već postepena

# Podešavanje retransmisionog tajmera (RTO)

- **RTT devijacija**



- Pored  $RTT_S$ , izračunava se i devijacija RTT-a, u oznaci  **$RTT_D$** , na sledeći način:

- Početno  **nema vrednost**
- Posle prvog merenja   **$RTT_D = RTT_M/2$**
- Posle svakog sledećeg merenja 

$$RTT_D = (1-\beta)RTT_D + \beta |RTT_S - RTT_M|$$

- Tipično se usvaja,  **$\beta=1/4$**

# Podešavanje retransmisionog tajmera (RTO)

- **RTO vreme.**
  - Konačno, vrednost RTO se izračunava na osnovu **(I) uravnoteženog** RTT i **(II) njegove devijacije**
  - Koristi se sledeća formula:
    - Početno  **Početna vrednost**
    - Posle svakog merenja 
- $$\text{RTO} = \text{RTT}_s + k \cdot \text{RTT}_D$$
- Tipično se usvaja, ***k=4***

# Podešavanje retransmisionog tajmera (RTO)

## *Eksponencijalni backoff*

- Postavlja se pitanje: *koju vrednost za RTO izabрати nakon retransmisije paketa?*
- Većina implementacija koristi strategiju *eksponencijalnog backoff-a*
- *RTO vrednost se duplira* sa svakom *uzastopnom retransmisijom*
- Nakon prve retransmisije paketa, **RTO** postaje  **$2 * RTO$** , posle druge uzastopne retransmisije  **$4 * RTO$**  itd.....



# Performanse S&W protokola

- ***Efikasnost protokola***

- ***Iskorišćenost*** prenosnog ***kanala***

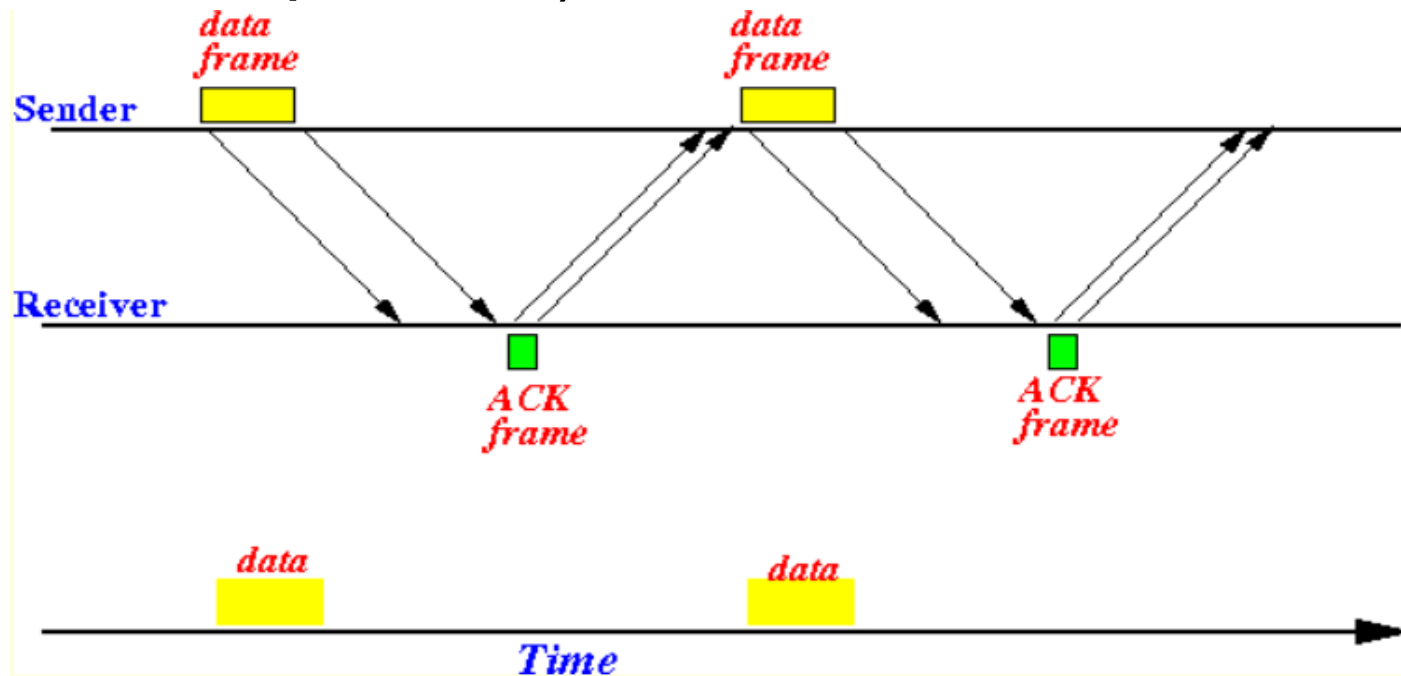
- Mera efektivnog korišćenja
    - Deo prenosnog kapaciteta kanala iskorišćenog za prenos podataka

- ***Propusnost protokola***

- Predpostavka: ***prenos podataka je bez grešaka***

# Performanse S&W protokola

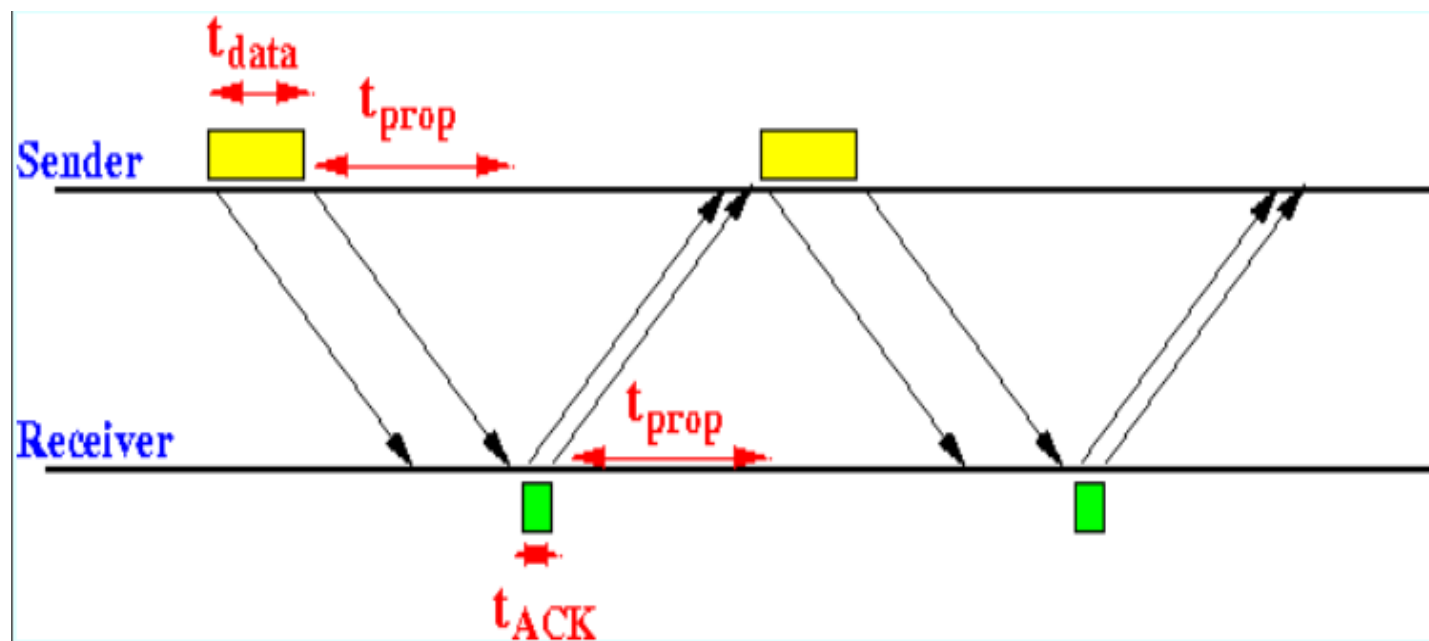
- Maksimalna iskorišćenost prenosnog kanala (bez grešaka u prenosu)



- Iskorišćenost = deo vremena na vremenskoj osi žuto obojen (prenos okvira)

# Performanse S&W protokola

- Maksimalna iskorišćenost prenosnog kanala



$t_{data}$  = vreme slanja poruke

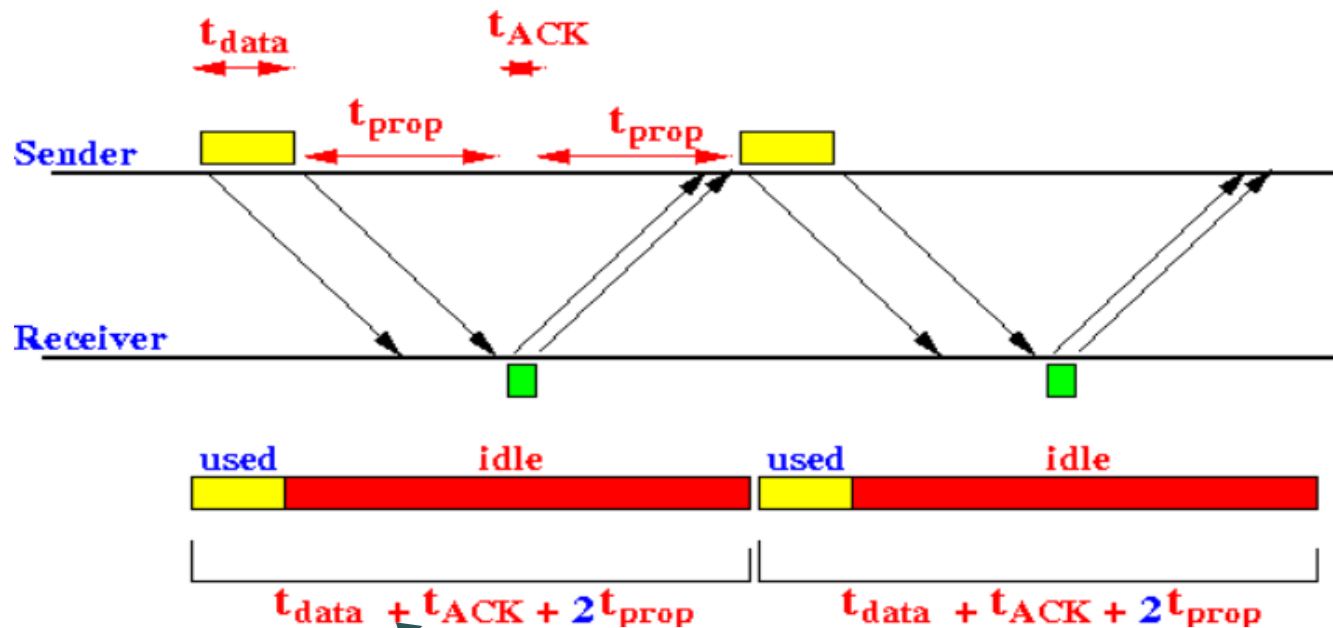
$t_{prop}$  = vreme prostiranja signala od predajnika do prijemnika

$t_{ACK}$  = vreme obrade poruke i slanje ACK poruke



# Performanse S&W protokola

- Maksimalna iskorišćenost prenosnog kanala



Trajanje ciklusa =

$$\begin{aligned} & t_{data} + t_{prop} + t_{ACK} + t_{prop} \\ & = t_{data} + t_{ACK} + 2 \times t_{prop} \end{aligned}$$

# Performanse S&W protokola

- Maksimalna iskorišćenost prenosnog kanala (U)
- Iskorišćenost = deo vremena iskorišćen na slanje poruke u odnosu na dužinu ciklusa

Maksimalna iskorišćenost (efikasnost kanala), U:

$$U = \frac{t_{data}}{t_{data} + t_{ACK} + 2 t_{prop}}$$
$$\approx \frac{t_{data}}{t_{data} + 2 t_{prop}}$$

( $t_{ACK}$  je vrlo malo pa se može zanemariti u analizi!)



# Performanse S&W protokola

- Značajne činjenice fizičkih karakteristika prenosnih medijuma:

***Brzina prostiranja*** svetlosti u vakumu (prostoru)  $\approx 3 \times 10^8 \text{ m/s}$

Brzina prostiranja električnog signala kroz provodnik  
 $\approx 2 \times 10^8 \text{ m/s}$

# ● ● ● | Performanse S&W protokola

○ Dalje.... (***parametar a***)

$$\mathbf{U} = t_{\text{data}} / (t_{\text{data}} + 2 t_{\text{prop}}) = \mathbf{1/(1 + 2a)}$$

$$\mathbf{a} = t_{\text{prop}} / t_{\text{data}}$$

$$\begin{aligned} \mathbf{a} &= \text{vreme prostiranja} / \text{vreme slanja} = (d/V)/(L/C) \\ &= \mathbf{Cd/VL} \end{aligned}$$

d – razdaljina između stanica

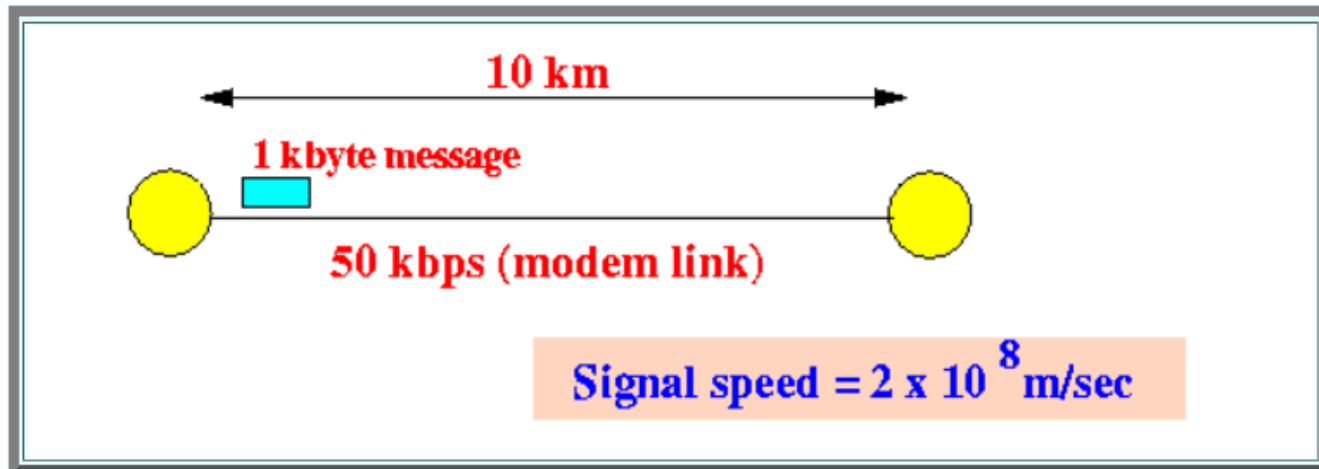
V – brzina prostiranja signala

L – dužina poruke u bitima

C – brzina prenosa u b/s

# Performanse S&W protokola

## ○ *Primer* (*spori* prenosni kanal):



$t_{data}$ :

1 data frame = 1 kbytes  $\approx$  1000 bytes = 8000 bits

Transmission speed = 50 kps  $\approx$  50000 bits in one sec

$\implies$  time to transmit 1 data frame =  $8000/50000$   
 $= 0.16$  sec

$t_{data} = 0.16$  sec

$t_{prop}$ :

Signal speed =  $2 \times 10^8$  m/sec  
 $= 2 \times 10^5$  km/sec

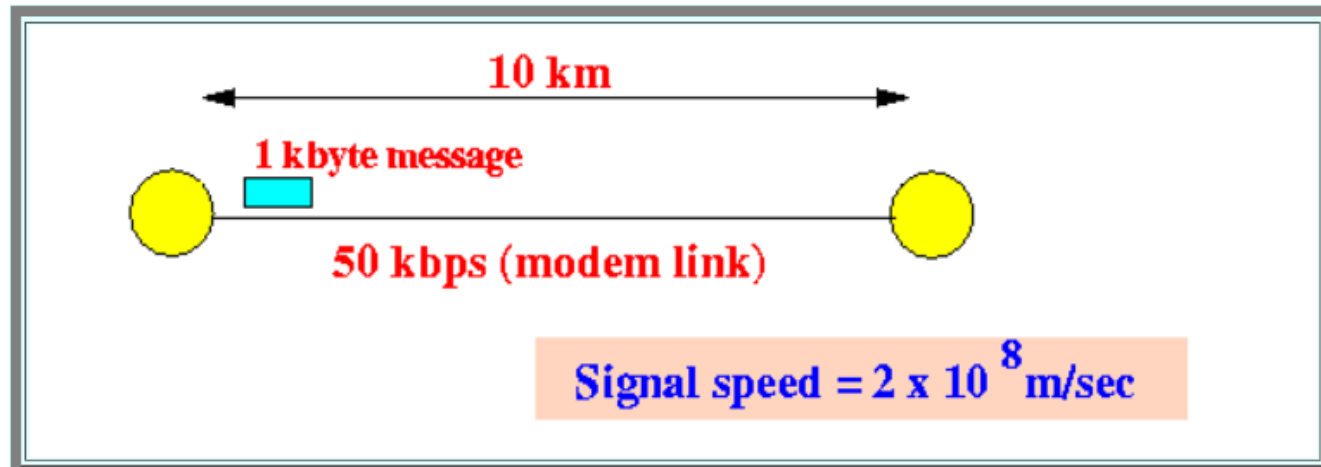
Distance between sender and receiver = 10 km

$\implies$  time for signal to travel 10 km =  $10 / (2 \times 10^5)$   
 $= 5 \times 10^{-5}$

$t_{prop} = 5 \times 10^{-5}$  sec

# Performanse S&W protokola

- Primer (spori prenosni kanal):



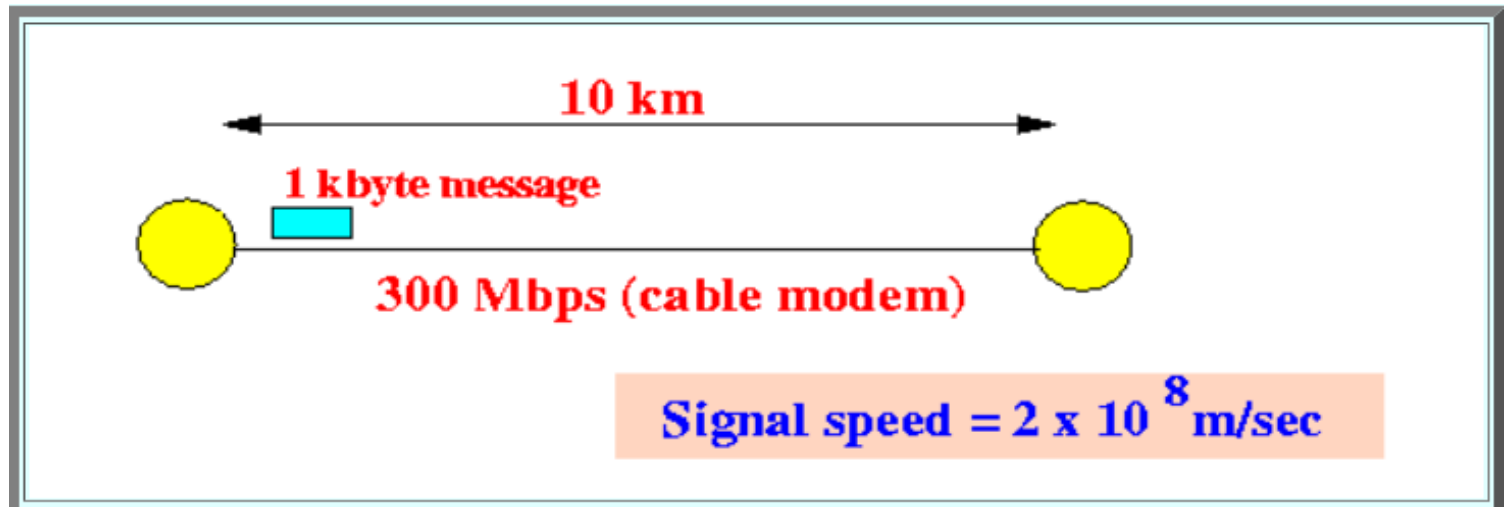
Iskorišćenost (efikasnost kanala),

U

$$\begin{aligned} U &= \frac{0.16}{0.16 + 2 \times 0.00005} \\ &= \frac{0.16}{0.1601} \\ &= 99.94 \% \end{aligned}$$

# Performanse S&W protokola

- *Primer* (**brzi** prenosni kanal):



Iskorišćenost (efikasnost kanala),

U

$$\begin{aligned} U &= \frac{2.67 \times 10^{-5}}{2.67 \times 10^{-5} + 2 \times 5 \times 10^{-5}} \\ &= \frac{2.67}{2.67 + 10} = \frac{2.67}{12.67} \\ &= 21.07 \% \end{aligned}$$



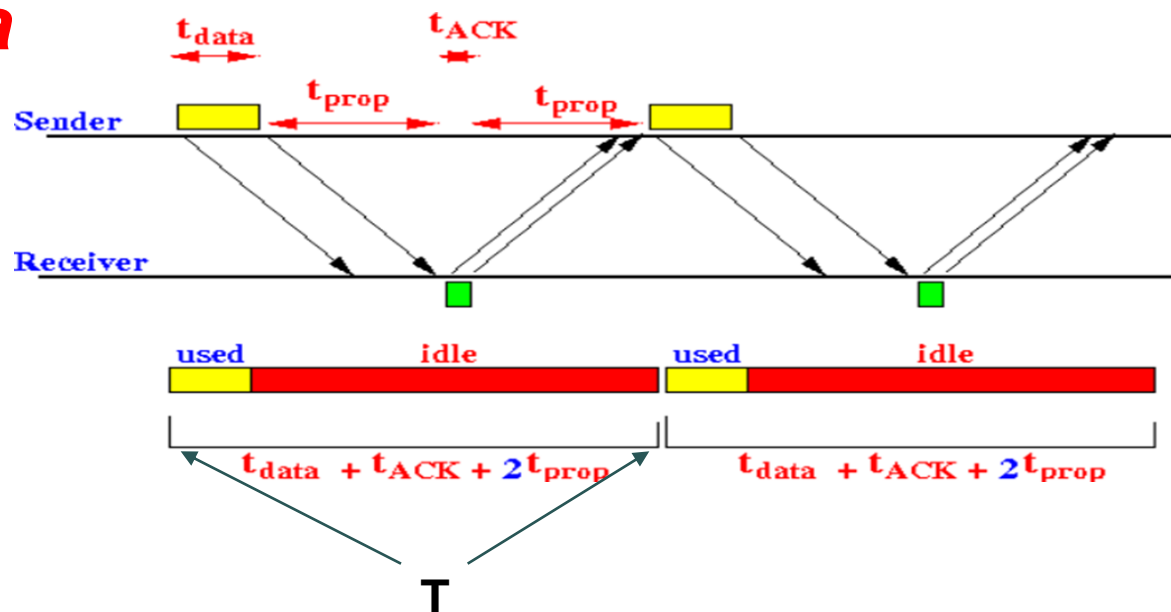
# Performanse S&W protokola

- Zaključak:
- *Stop-and-Wait protokol obezbeđuje efikasno korišćenje prenosnog kanala u slučaju sporijih prenosnih kanala*
- *Vrlo je neefikasan u slučaju brzih linkova!*



# Performanse S&W protokola

- Propusnost
- Želimo da izračunamo *srednje vreme*,  $T$  (u sek.) između uspešno isporučenih paketa – *vreme trajanja ciklusa*



$$Propusnost = 1/T = 1/(t_{data} + 2t_{prop})$$

# Performanse S&W protokola

- Propusnost
- **Ne može se** samo **predpostaviti** da je  $T = t_{\text{data}} + 2t_{\text{prop}} = \text{RTT}$ , jer se **paketi mogu izgubiti**
- Uvodimo novu predpostavku: ***moguć je gubitak paketa (greške) u prenosu***

# Propusnost S-and-W protokola

- Propusnost - moguć je gubitak paketa (greške) u prenosu
  - Predpostavka: Neka **postoji  $N$  linkova** između predajnika i prijemnika
  - Ako je **verovatnoća gubitaka paketa/ACK po linku** jednak  $p$ , onda je  $(1 - p)$  **verovatnoća isporuke** paketa po tom linku
  - **Verovatnoća** njegove **isporuke preko  $N$  linkova** jednaka je  $(1 - p)^N$
  - Dakle, **verovatnoća da će paket** sa podacima ili potvrdom **biti izgubljen** je  $P = 1 - (1 - p)^N$

# Propusnost S-and-W protokola

- Sada možemo ***napisati jednačinu za  $T$  (srednje vreme trajanja jednog ciklusa)*** u zavisnosti od  **$RTT(= t_{\text{data}} + 2t_{\text{prop}})$**  i  **$RTO$**  (dužine vremenske kontrole):

$$\begin{aligned} T &= (1 - P)RTT + P(RTO + T) \\ &= RTT + (P/(1 - P))RTO \end{aligned}$$

- Dakle, ***očekivana propusna moć*** protokola je  **$1/T$**

## Najbolji slučaj

- Dešava se kada je **RTT isti za svaki paket**, tako da je **dužina vremenske kontrole** nešto veća od **RTT**

$$T = RTT + (P/(1-P))RTT = (1/(1-P))RTT$$

$$\text{Propusnost} = (1 - P)/(t_{\text{data}} + 2t_{\text{prop}})$$

- Ako **link** koji je usko grlo može podržati **100 paketa/sek.** i **RTT je jednako 100ms**, tada koristeći stop-and-wait protokol, maksimalna **propusnost je najviše 10 paketa/sek.**



## Najbolji slučaj

- Ako link koji je usko grlo može podržati 100 paketa/sek i RTT je jednako 100ms, tada koristeći stop-and-wait protokol, maksimalna propusnost je najviše 10 paketa/sek.
- ***Iskorišćenost - samo 10%***
- Potreban nam je neki ***bolji pouzdani transportni protokol.....***

# Ideja: protokol kliznog prozora

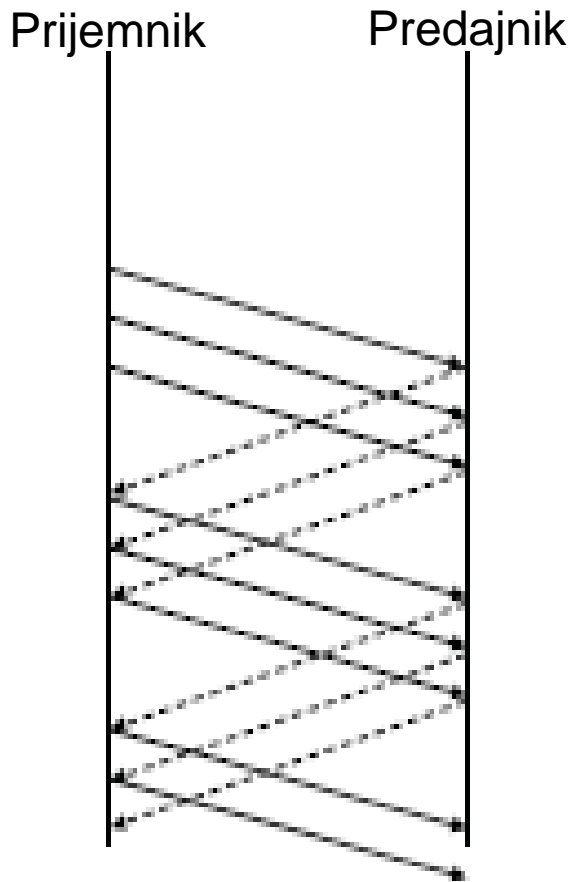
- Koristiti prozor

- Omogućuje da ***u mreži postoji najviše  $W$  nepotvrđenih paketa*** u komunikaciji
- ***$W$***  se naziva ***dužina prozora***
- ***Slanje*** paketa ***se preklapa*** sa ACK (potvrdama)

- Predajnik ***pomera prozor za 1 prijemom*** (u sekvenci) ***svake potvrde*** (ACK)

- t.j. ***prozor klizi***
- ***redukuje se slobodni period***

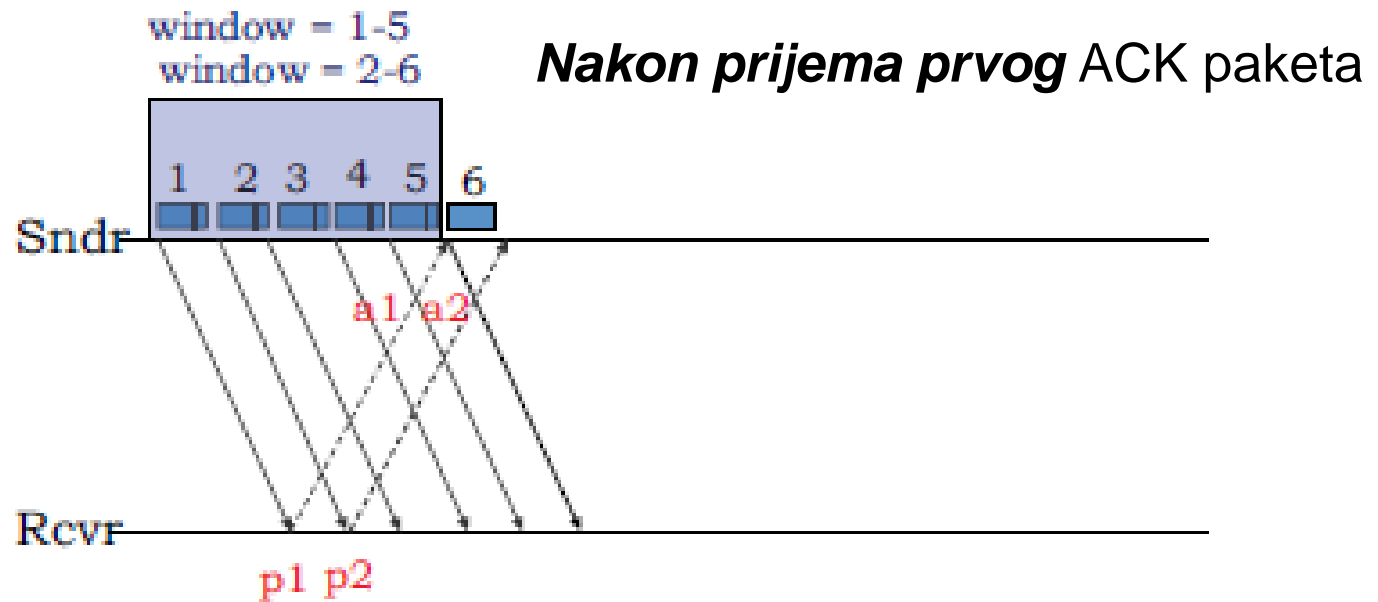
# Ideja: protokol kliznog prozora



- Predpostavimo da je ***dužina prozora,  $W$ , fiksna i poznata***
- Kasnije ćemo diskutovati ***o načinu kako ga treba setovati***
- **$W = 3$**  u primeru levo

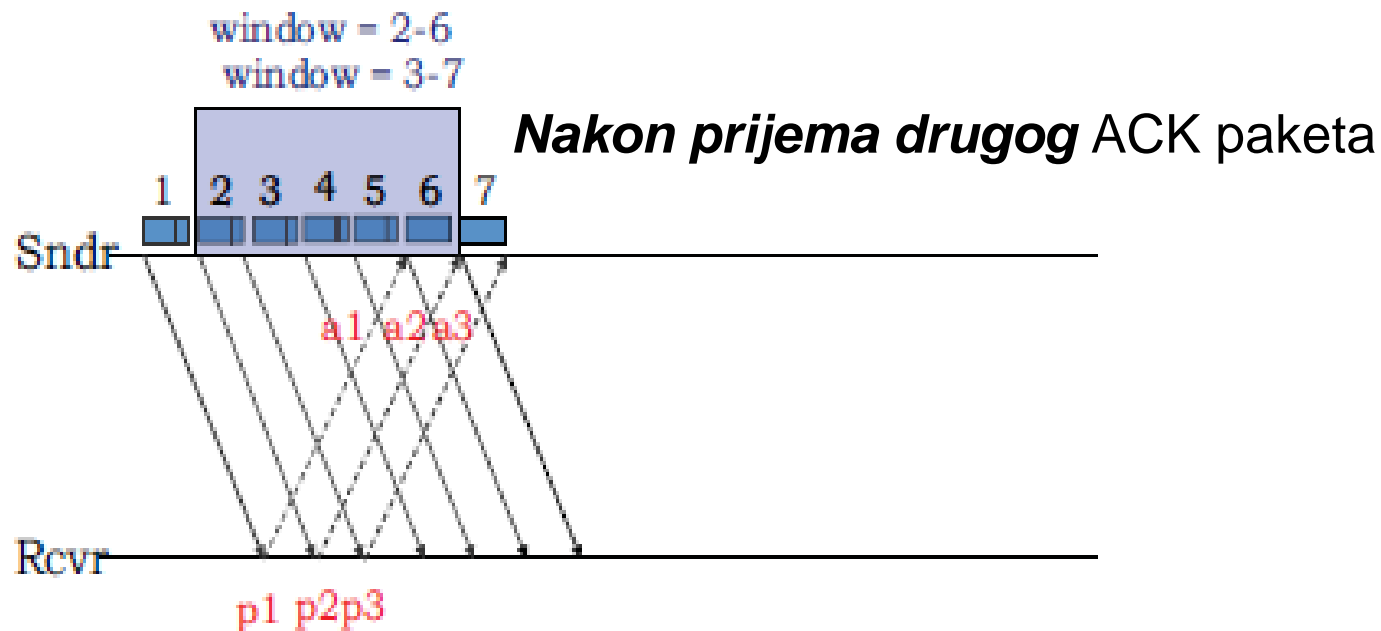


# Protokol kliznog prozora u akciji



**$W = 5$**  u ovom primeru

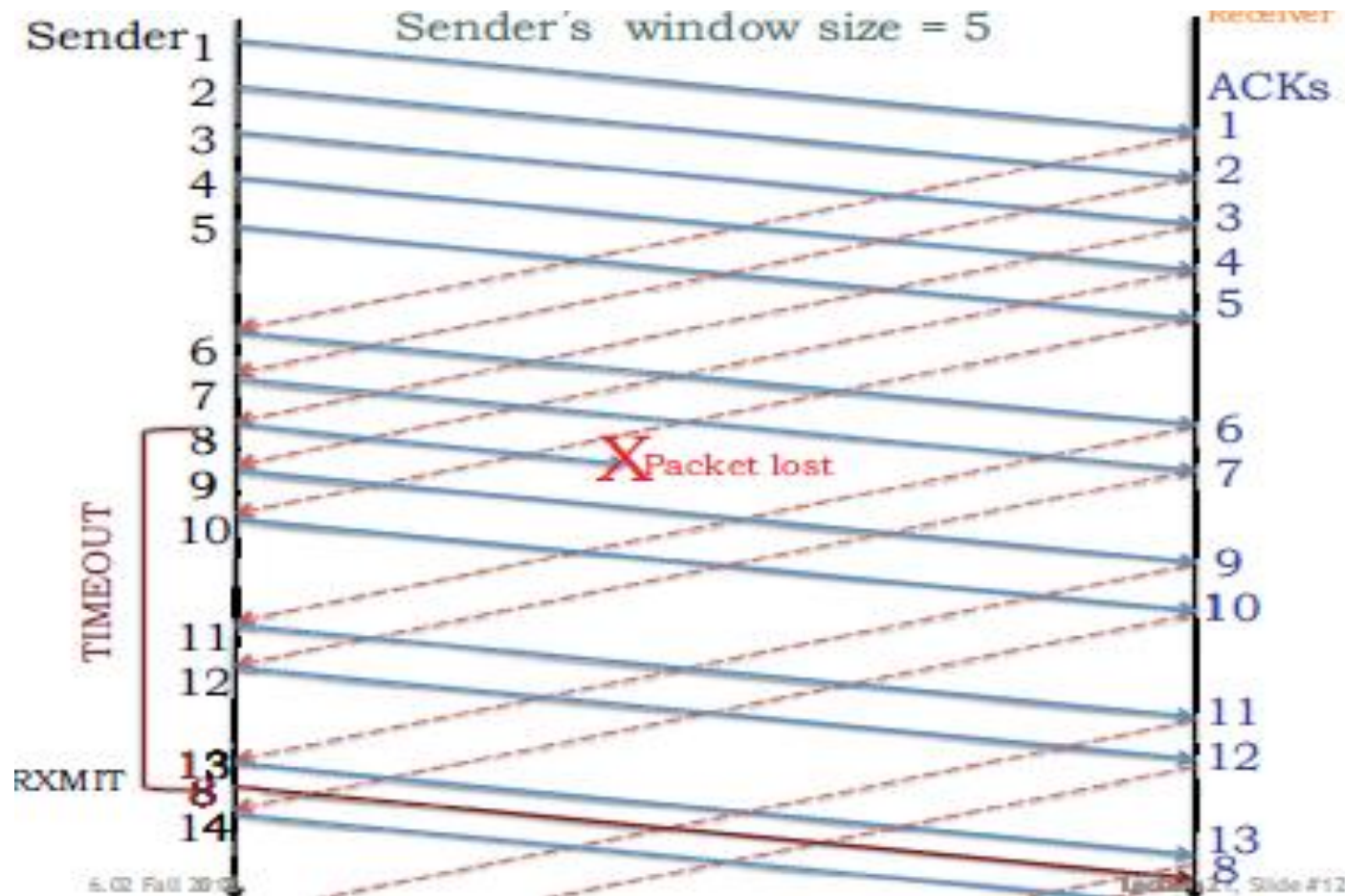
# Protokol kliznog prozora u akciji



**Definicija prozora:** ako je **dužina prozora  $W$** , tada je **maksimalni broj nepotvrđenih paketa jednak  $W$**

Klizni prozor fiksne dužine

# Protokol kliznog prozora u akciji



# Implementacija kliznog prozora

## ○ Predajnik

- *Svaki paket sadrži redni broj* (koji se sekvencijalno inkrementira)
- *Nakon slanja paket, **sačuvati** (xmit- vreme, paket) **na listi ne-potvrđenih paketa***
- *Paket se šalje ako je*

$\text{len}(\text{lista ne-potvrđenih paketa}) < \text{dužina prozora, } W$

- *Prijemom potvrde (ACK) sa odredišta sa određenim rednim brojem, **ukloniti** odgovarajući element **iz liste nepotvrđenih paketa***



# Implementacija kliznog prozora

## ○ Predajnik

- ***Periodično pretraživati listu nepotvrđenih paketa*** kako bi se pronašli paketi koji su poslani pre određenog vremena
- ***Ponovo ih poslati, ažurirati xmit vreme*** za slučaj ponovne provere
- „pre određenog vremena“ : ***xmit < sada – timeout***



# Implementacija kliznog prozora

## ○ Prijemnik

- ***Slati ACK za svaki primljeni paket, referenciran na*** redni broj primljenog paketa
- ***Isporučiti sadržaj paketa*** (payload) prijemnoj aplikaciji ***u redosledu*** definisanom rednim brojevima
- ***Sačuvati isporučene pakete*** primljene u redosledu ***u lokalnom baferu*** (***ukloniti duplikate***).
  - Odbaciti dolazne pakete koji su već isporučeni (izazvano retransmisijom zbog gubitka ACK)



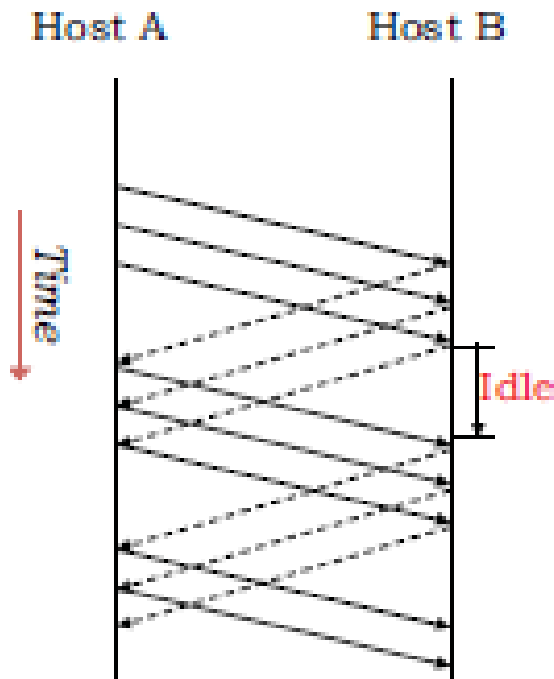
# Implementacija kliznog prozora

## ○ Prijemnik

- ***Voditi računa o sledećem paketu*** koji aplikacija očekuje
- ***Posle svakog prijema, aplikaciji isporučiti što više paketa u redosledu***

# Određivanje dužine kliznog prozora

- Kako odrediti dužinu kliznog prozora ( $W$ ) tako da se propusnost maksimizira?
- Odgovor: Primenom Little-ovog zakona!



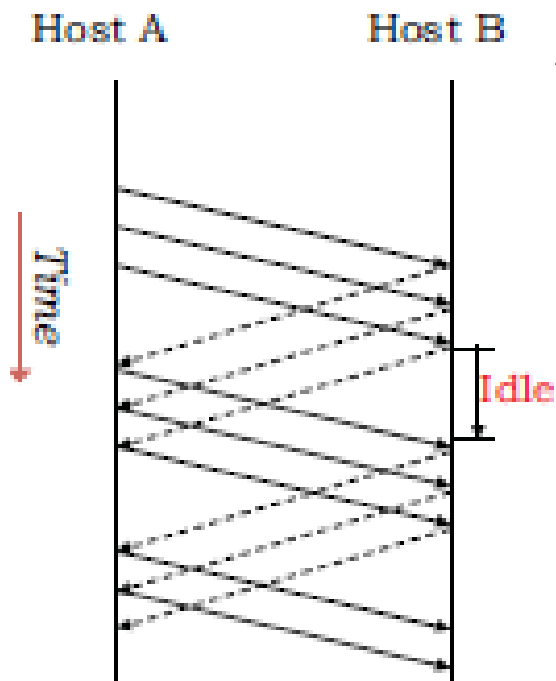
Ako možemo dobiti:  **$Idle = 0$**   
- **postigli smo željeni cilj**

- **$W$**  = broj paketa u prozoru
- **$B$**  = prenosni **kapacitet na linku** koji predstavlja „**usko grlo**“ na putanji (**paketi/sek.**)
- **$RTT_{min}$**  = **minimalno  $RTT$**  na putanji, bez ikakvog baferovanja paketa u prenosu



# Određivanje dužine kliznog prozora

- Kako odrediti dužinu kliznog prozora ( $W$ ) tako da se propusnost maksimizira?
- Odgovor: Primenom Little-ovog zakona!

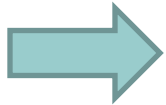


Ako je  $W = B \cdot RTT_{min}$ , tada je putanja – **potpuno iskorišćena** (uz uslov da nema gubitaka u prenosu)

- $B \cdot RTT_{min}$  se naziva “**proizvod opseg-odziv**”
- Ključni koncept svih transportnih protokola koji se baziraju na kliznom prozoru

# Propusnost SW protokola

- ***Ako nema gubitaka*** paketa u prenosu – protokol isporučuje ***W paketa svakih RTT sekundi***



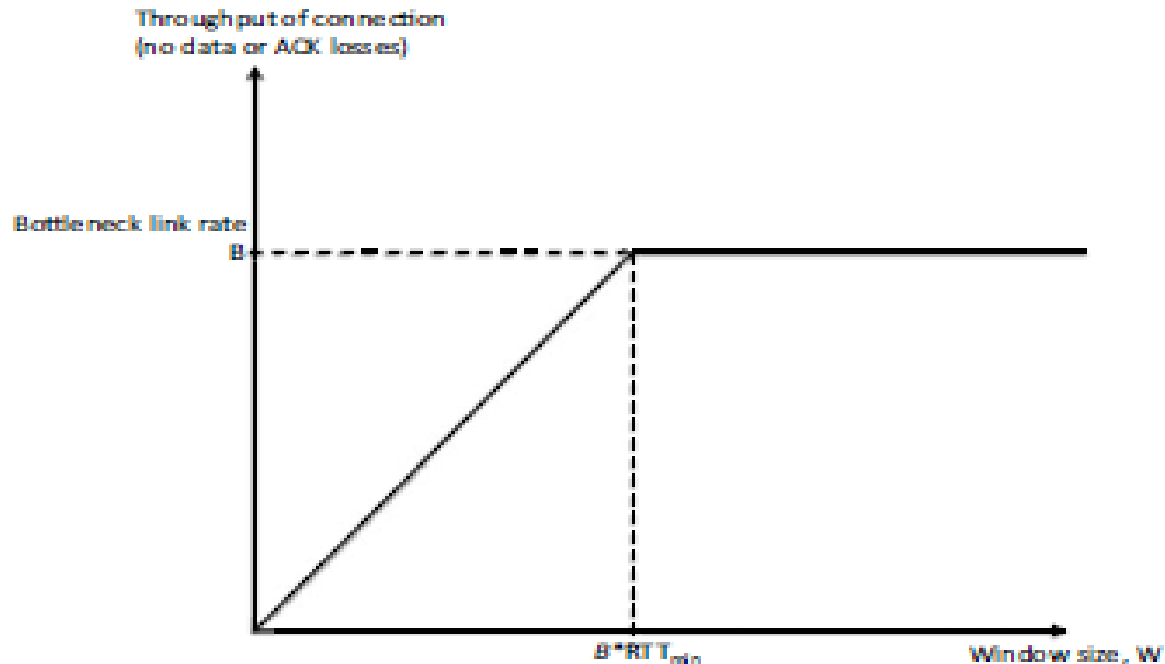
***Propusnost*** je jednaka ***W/RTT***

- **Cilj**: ***dostići što veću iskorišćenost*** – postaviti ***vrednost za W*** tako da ***link*** koji predstavlja „***usko grlo***“ - ***nikada ne bude slobodan*** (***Idle***) usled nedostatka paketa

# Propusnost SW protokola

- **Bez gubitaka paketa** u prenosu:

$$\text{Propusnost} = W/RTT_{min}, \text{ ako } W \leq B \cdot RTT_{min},$$
$$= B, \text{ ina\u0107e}$$



# Propusnost SW protokola

- **Bez gubitaka paketa** u prenosu:
- Ako  $W > B \cdot RTT_{min}$ ,  $RTT$  vreme na konekciji **će uključiti i kašnjenje usled baferovanja**
  - RTT neće biti konstantno i nezavisno od  $W$ !
  - Povećanje  $W$  će prouzrokovati povećanje  $RTT$ , ali **propusnost neće rasti iznad  $B$**
- **Kolika je propusnost u tom slučaju?**
- Na ovo pitanje ćemo odgovoriti dvostrukom primenom Little-ovog zakona

# Propusnost SW protokola

- **Bez gubitaka paketa** u prenosu:
- Izložićemo **intuitivno rešenje** da ako je  $W > B \cdot RTT_{min}$ , tada je **propusnost jednaka  $B$**  paketa u sekundi

# Propusnost SW protokola

- Prvo, neka je ***prosečan broj paketa u baferu*** na linku koji predstavlja „usko grlo“ na putanji, ***jednak***  $Q$
- Po Little-ovoj formuli primenjenoj na ovom linku,  
$$Q = B \cdot \tau$$
- ( $B$  - ***intezitet pražnjenja*** reda čekanja,
- $\tau$  - ***prosečno kašnjenje*** u redu čekanja,  $\tau = Q/B$ )
- Odatle,

$$RTT = RTT_{min} + \tau = RTT_{min} + Q/B$$

# Propusnost SW protokola

- Sada, *razmotrimo dužinu prozora  $W$* , odnosno, broj nepotvrđenih paketa
- Znamo da svi ti *paketi mogu biti*
  - A. U redu čekanja** na linku koji predstavlja „usko grlo“ na putanji ili
  - B. U delu sistema bez baferovanja**
- Dakle,

$$W = Q + B \cdot RTT_{min}$$

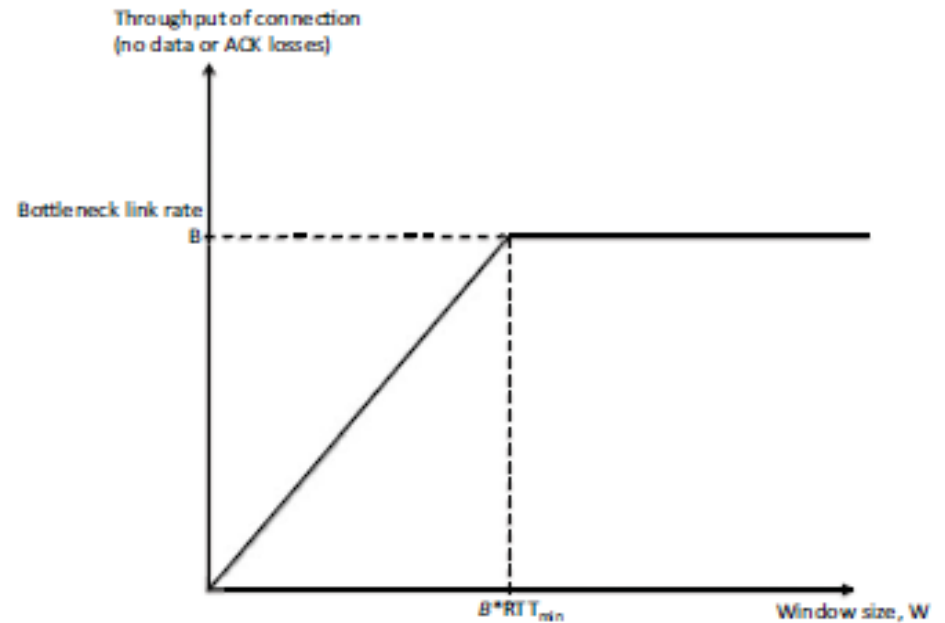
# Propusnost SW protokola

- Na kraju, *iz Little-ove formule* primenjene na čitavoj dvosmernoj putanji:

$$\text{Propusnost} = W/RTT$$

$$= (B \cdot RTT_{\min} + Q) / (RTT_{\min} + Q/B)$$

$$= B.$$





# Propusnost SW protokola

- *Sa gubicima paketa u prenosu:*
- Izabrati  $W > B \cdot RTT_{min}$  tako da se obezbedi da *link* koji predstavlja „**usko grlo**“ – **bude uvek zauzet** čak i pri gubitku paketa
- Očekujući (*srednji*) **broj slanja paketa**,  $T$ , **do uspešne isporuke** kako paketa tako i ACK za taj paket

$$T = (1 - L) \cdot 1 + L \cdot (1 + T), \text{ tako da}$$

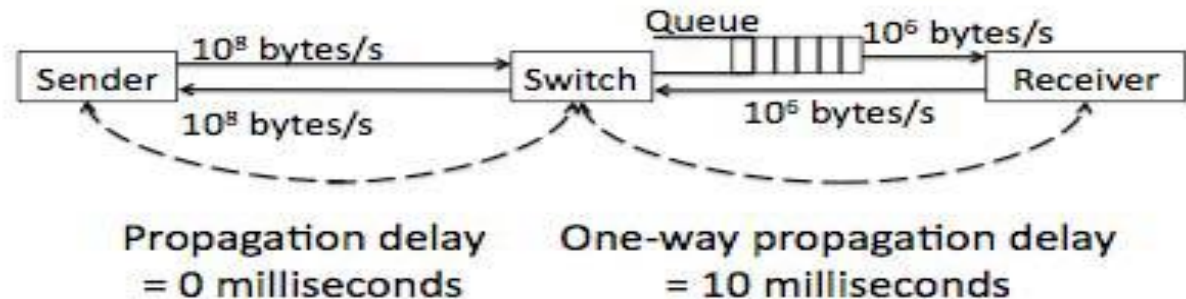
$$T = 1/(1 - L)$$

(gde je  $L = P(\text{ili paket ili ACK je izgubljen})$ )

# Propusnost SW protokola

- $T$ : ovoliki broj slanja – *jedan paket će biti uspešno primljen i potvrđen*
- Dakle, **koeficijent iskorišćenosti sistema** je  $(1/T)$ ,  
$$= 1 - L$$
- Sledi,  
 **$Propusnost = (1 - L) \cdot B$**
- Ako je  $W \gg B \cdot RTT_{min}$ , tada je kašnjenje preveliko, vremenske kontrole vrlo duge, tako da **negativno utiče na druge konekcije**

# Primer



Max queue size = 30 packets

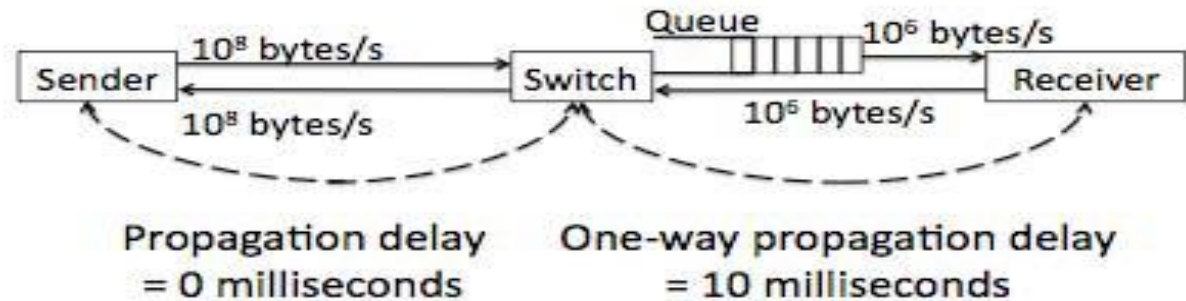
Packet size = 1000 bytes

ACK size = 40 bytes

Initial sender window size = 10 packets

- **Pitanje:** Kojim će intezitetom (u paketima po sekundi) protokol isporučiti više-gigabajtsku datoteku od predajnika do prijemnika?
  - Predpostavka je da ne postoji drugi saobraćaj u mreži i paketi mogu biti izgubljeni samo usled preopterećenja bafera

# Primer



Max queue size = 30 packets

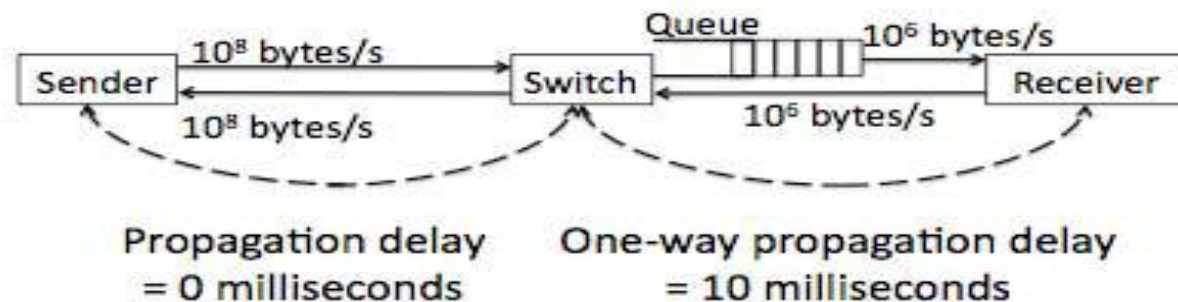
Packet size = 1000 bytes

ACK size = 40 bytes

Initial sender window size = 10 packets

- **Odgovor:** RTT je oko 20 milisekundi plus vreme slanja paketa, koje je oko 1 milisekunde (1000 bajtova intezitetom slanja od 1Mb/s)
- Iz toga sledi da je  **$propusna\ moć = W/RTT =$**   
 **$= 10\ paketa / 21\ ms = 476\ paketa\ po\ sekundi$**

# Primer (nastavak...)



Max queue size = 30 packets

Packet size = 1000 bytes

ACK size = 40 bytes

Initial sender window size = 10 packets

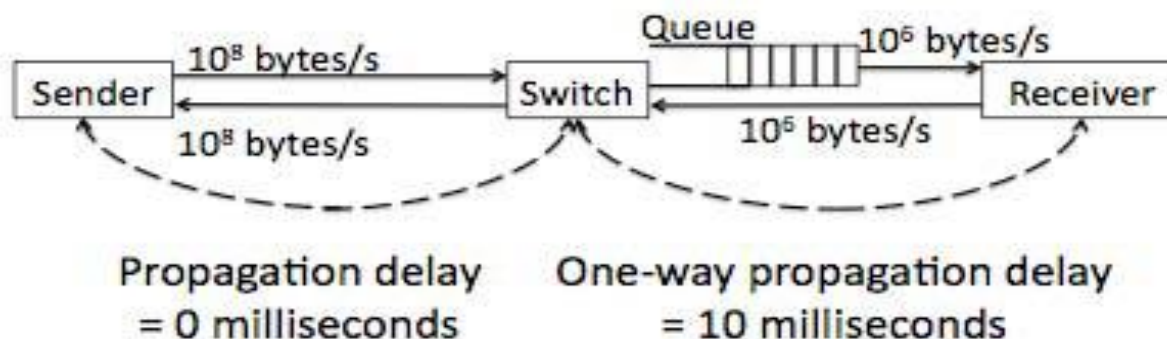
- Želimo da **dupliramo propusnost** transportnog protokola sa kliznim prozorom. Možemo primeniti **samo jednu od sledećih tehnika**:
  - Duplirati dužinu prozora.
  - Poloviti vreme prostiranja signala po linkovima.
  - Duplirati brzinu prenosa linka između komutatora i prijemnika

# Primer (nastavak...)

- Za svaku od sledećih dužina prozora ( $W = 10, 30$  i  $50$ ) na predajnoj strani, koje od gornjih tehnika, aproksimativno dupliraju propusnu moć
  - Ako ne postoji tehnika kojom je to moguće učiniti – odgovor je „Ne postoji“
  - Može postojati više od jednog odgovora za svaku dužinu prozora, u tom slučaju navesti sve odgovore
  - Napomenimo da se tehnike primenjuju izolovano
- Objasniti odgovore

# Rešenje

- Napomena: **opseg-odziv proizvod** na ovoj putanji je **20 paketa**



Max queue size = 30 packets

Packet size = 1000 bytes

ACK size = 40 bytes

Initial sender window size = 10 packets

# Rešenje

$W = 10$ .

○ Odgovor: **a. i b.**

- U slučaju kada je  $W = 10$ , propusna moć je **oko 476 paketa/sekundi**
- Ako dupliramo dužinu prozora, propusnost će biti **duplirana na 952 paketa/s**
- Ako se **redukuje vreme prostiranja** signala kroz linkove, propusnost će, takođe, biti duplirana
- Nova propusnost će još uvek biti manja od kapaciteta linka koji predstavlja „usko grlo“ u mreži koja je jednaka 1000 paketa/s.





# Rešenje

$W = 50$ .

- Odgovor: **c.**
  - Kada je  $W = 50$ , **propusnost je već 1000 paketa/s.**
  - U tom stanju, dupliranjem dužine prozora ili polovljenjem vremena prostiranja (RTT) **ne povećava se propusnost**
  - Ako se brzina prenosa duplira na linku između komutatora i prijemnika, **kapacitet „uskog grla“ postaje 2000 paketa/s**



# Rešenje

**$W = 50$ .**

- Odgovor: **c.**
  - nastavak....
  - ***Dužina prozora od 50 paketa sa  $RTT$  od 20 ili 21 milisekundi omogućuje propusnost veću od 2000 paketa/s.***
  - Dupliranjem brzine prenosa linka između komutatora i prijemnika („usko grlo“) će se duplirati propusnost sa dužinom prozora od 50 paketa



# Rešenje

$W = 50$ .

- Odgovor: **c.**
  - nastavak....
  - Sa dužinom bafera u komutatoru od 30 paketa i  $W = 50$  paketa, paketi iz inicijalnog prozora će prouzrokovati preopterećenje bafera i gubitak paketa
  - Međutim, to neće prouzrokovati pad propusne moći u ravnotežnom stanju



# Rešenje

**$W = 30$ .**

- Odgovor: ***Ne postoji.***
  - Kada je  **$W = 30$** , propusna moć je ***već 1000 paketa/s.***
  - Ako dupliramo dužinu prozora ili polovimo RTT, propusnost se neće promeniti
  - Interesantna situacija nastaje ***kada dupliramo brzinu prenosa, link*** („usko grlo“) će sada biti u mogućnosti da ***isporučuje 2000 paketa/s.***



# Rešenje

**$W = 30$ .**

- Odgovor: ***Ne postoji.***
  - nastavak....
  - Ali dužina prozora od ***30 paketa*** i ***RTT od 21*** milisekunde, daje propusnost od oko ***1500 paketa/s*** (ako koristimo 21 ms, dobijamo 1428 paketa/s.)
  - Dakle, ***poboljšanje je oko 50%***, što je daleko od dupliranog što smo želeli
  - Ne postoji tehnika (od ponuđenih) koja to može ostvariti