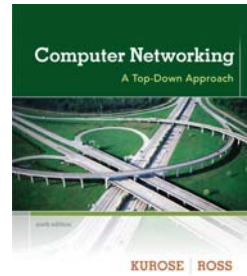


Permbledhje

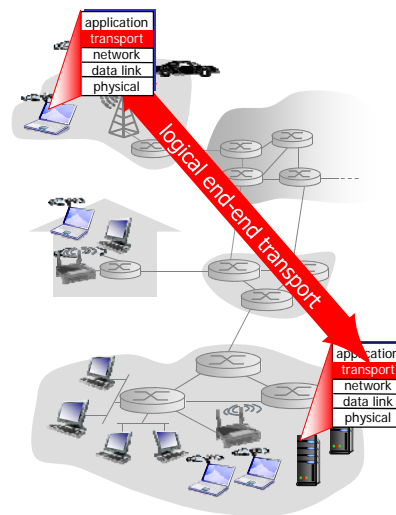


*Computer
Networking: A Top
Down Approach*
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Transport Layer 3-1

Sherbimet dhe protokollet e Transportit

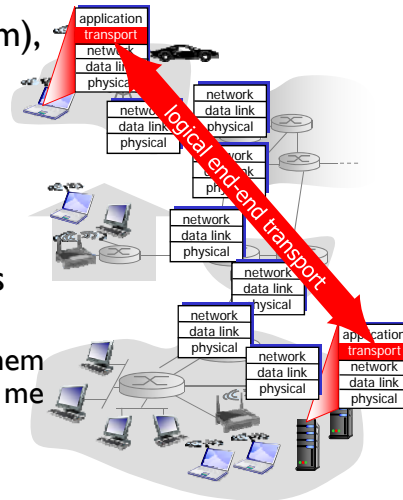
- ❖ Krijon komunikim llogjik ndermjet proceseve te aplikimeve qe hoste te ndryshem
- ❖ Protokollet e transportit funksionojne ne sistemet fundore
 - Ana derguese: ndan mesazhet e aplikimeve ne *segmente*, dhe i kalon tek shtresa e rrjetit (network)
 - Ana marrese: rinderton segmentet ne mesazhe, dhe i kalon tek shtresa e zbatimit
- ❖ Me shume se sa nje protokol transporti qe mund te perdoret nga zbatimet
 - Internet: TCP and UDP



Transport Layer 3-2

Protokollet e shtreses se transportit te Internetit

- ❖ Dergim Reliable (i besueshem), dhe sipas rradhes (TCP)
 - Kontrolli i kongestion
 - Kontrolli i rrjedhes
 - Vendosja (setup) I lidhjes
- ❖ Dergim i pabesueshem (Unreliable), jo sipas rradhes (unordered): UDP
 - Pa ndonje zgjerim te rendesishem ndaj "best-effort" (perpjekses me te mire) te IP
- ❖ Sherbime jo te pranishem:
 - Garantim vonese
 - Garantim bandwidth



Transport Layer 3-3

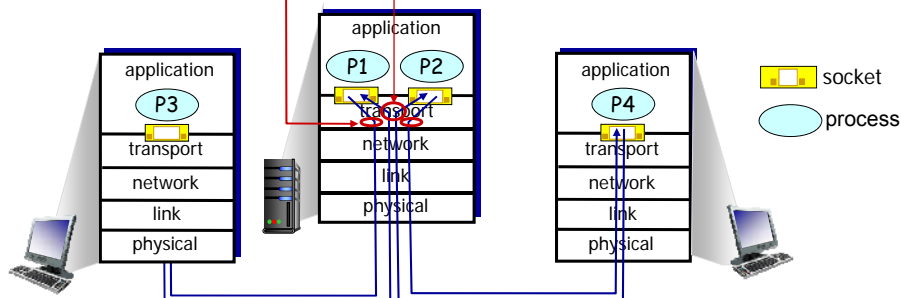
Multiplexing/demultiplexing

multiplexing ne dergues:

Trajton te dhena te dhana nga disa sockets, shton header te transportit (i perdorur me vone per demultiplexing)

demultiplexing ne marres:

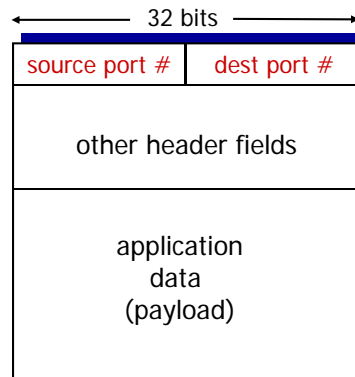
Perdor informacionin e headerin per te derguar segmentet e Marre tek socket e duhura



Transport Layer 3-4

Si funksionon demultipleximi

- ❖ Hosti merr diagramat IP
 - Cdo datagram ka adresen IP te burimit, adresen IP te destinacionit
 - Cdo datagram mban nje segment te shtreses se transport
 - Cdo segment ka nje nje numer porte te burimit dhe destinacionit
- ❖ Hosti perdor adresat IP dhe numerat e portave per te drejtuar segmentet ne portat e duhuara



TCP/UDP segment format

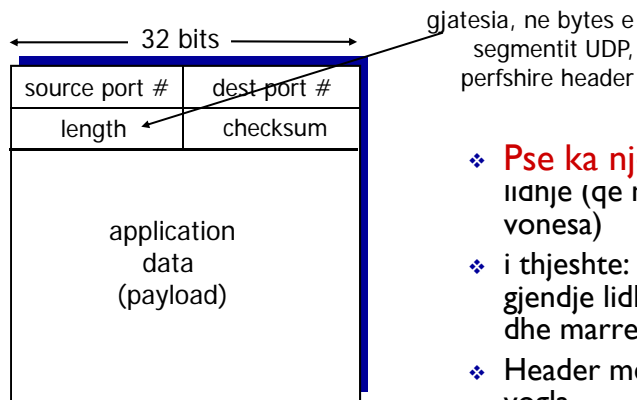
Transport Layer 3-5

UDP: User Datagram Protocol [RFC 768]

- ❖ Pa sherbime shtese, minimumi i protokollit te transportit ne Internet
- ❖ Sherbim "best effort" (perpjeken me te mire), segmentet UDP mund te:
 - humbin
 - Te dergohen jashte rradhes tek aplikimet
- ❖ **Connectionless (pa lidhje):**
 - Pa dhenie dore ndermjet derguesit dhe marresit UDP
 - Cdo segment UDP trajtohen pavaresisht nga te tjeret
- ❖ **Perdorime te UDP:**
 - streaming multimedia apps (tolerant ndaj humbjeve, sesitive ndaj shpejtesia)
 - DNS
 - SNMP
- ❖ **Transferim i besueshem mbi UDP:**
 - Shto besueshmeri (reliability) ne shtresen e aplikimit
 - Korrigjimi i gabimeve specifike te aplikimit!

Transport Layer 3-6

UDP: segment header



UDP segment format

- ❖ **Pse ka nje UDP?** dosje lidhje (qe mund te shtoje vonesa)
- ❖ i thjeshte: nuk mbahet gjendje lidhje ne dergues dhe marres
- ❖ Header me permasa te vogla
- ❖ Nuk ka congestion control: UDP mund te dergosa sa shpejte te munde

Transport Layer 3-7

UDP checksum

Qellimi: te detektoje “gabime” (psh. bitet e ndryshuar) ne segmentin e transmetuar

dergues:

- ❖ Trajton permbajtjen e segmenteve, si sequence numerash te plote 6-bite
- ❖ checksum: shton addition (one's complement sum) ne permbajtjen e segmentit
- ❖ Derguesi vendos vleren e checksum ne fushen e checksum ne UDP

marres:

- ❖ Llogarit checksum e segmentit te marre
- ❖ Kontrollon nese checksum llogaritur eshte i barabarte me vleren e fushes se checksum:
 - JO – eshte detektuar nje gabim
 - PO – nuk eshte detektuar gabim. Por *megjithate mund te kete gabime?*

Transport Layer 3-8

Internet checksum: shembull

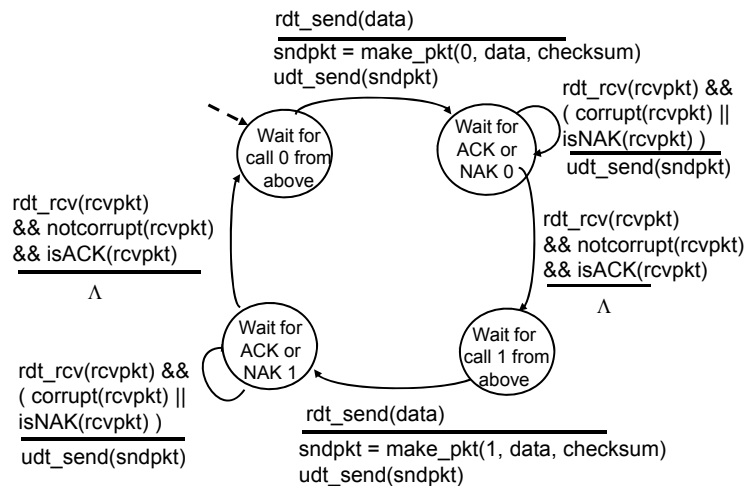
example: mblidh dy numerate plote 16-bitesh

	1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
	1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
mbartje	1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
shuma	1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0
checksum	0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

Shenim: kur blidhen numerat, nje mbartje nga bitet me me vlere duhet t'i shtohen rezultatit

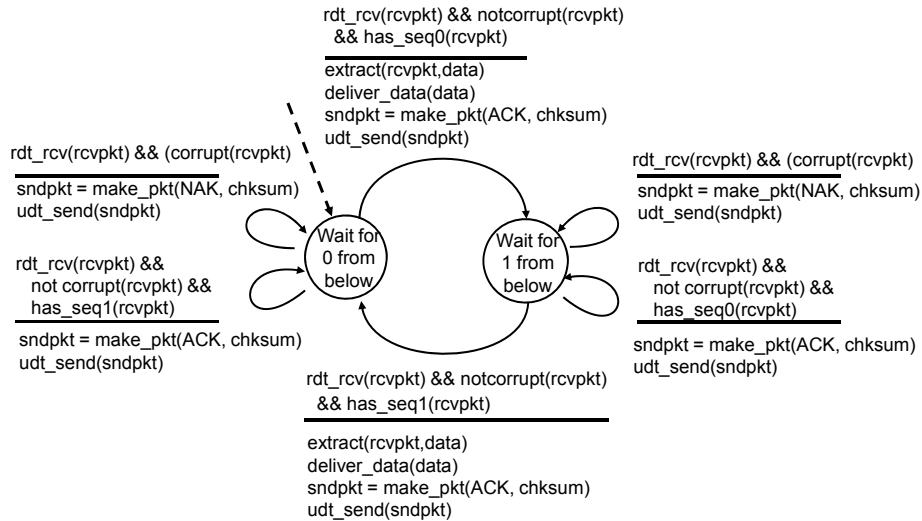
Transport Layer 3-9

rdt2.1: sender, handles garbled ACK/NAKs



Transport Layer 3-10

rdt2.1: receiver, handles garbled ACK/NAKs



Transport Layer 3-11

rdt2.1: discussion

sender:

- ❖ seq # shtuar tek pkt
- ❖ A mjaftojne dy seq. #' s (0,1)? Pse?
- ❖ Duhet te kontrolloje nese ACK/NAK e marrura jane te korruptuara (me gabime)
- ❖ Dyfish gjendjesh
 - Gjendja duhet te "kujtoje" nese pkt e "pritur" duhet te kete seq # 0 ose 1

receiver:

- ❖ Duhet te kontrolloje nese paketa e marre eshte duplicate (e perseritur)
 - Gjendja tregon nese 0 apo 1 pritet si pkt seq #
- ❖ shenim: marresi nuk mund te dije nese ACK/NAK I tij i fundit eshte marre OK ne dergues

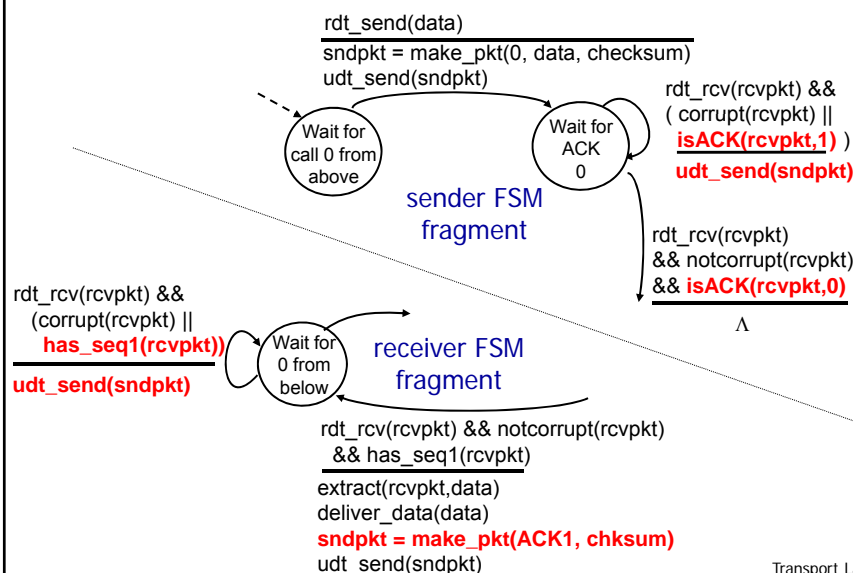
Transport Layer 3-12

rdt2.2: nje protokoll pa NAK

- ❖ Te njejtat funksions si rdt2.1, por duke perdour vetem ACKs
- ❖ Ne vend te NAK, marresi dergon ACK per pkt e fundit te marre OK
 - Marresi duhet te fuse ne menyre ekspliciteseq # e pkt qe po ACKed
- ❖ ACK e perseritur ne dergues con ne te njejtin veprim si NAK: ritranmetim i paketes ne fjale

Transport Layer 3-13

rdt2.2: derguesi dhe marresi



Transport Layer 3-14

rdt3.0: kanale me gabime dhe humbje

Kushte te reja: kanali mund gjithashtu te humbi paketa (data, ACKs)

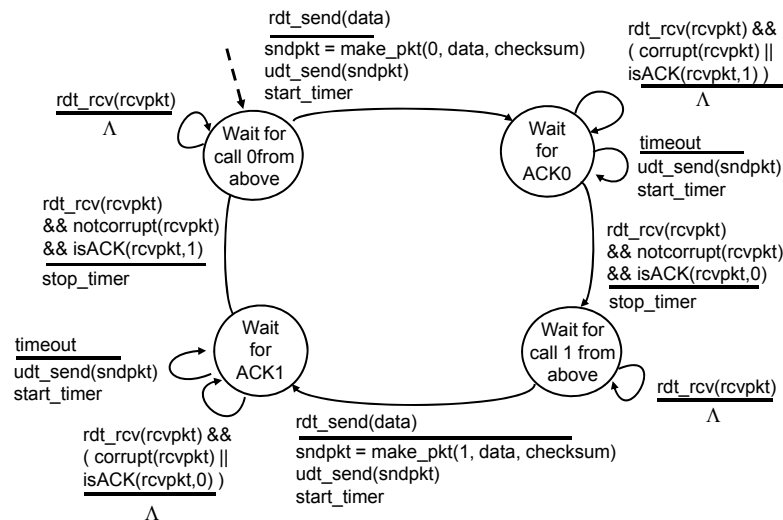
- checksum, seq. #, ACKs, ritransmetimet do te ndihmojne ... por nuk mjaftojne

qasja: derguesi pret per nje sasi kohe te “arsyeshme” per ACK

- ❖ Ritransmeton nese nuk merr ndonje ACK gjate kesaj kohe
- ❖ Nese pkt (apo ACK) nuk jana humbur por vec vonuar:
 - Ritransmetimi do te rezultojte ne nje perseritje te paketes, por seq. #’ s e zjgidhin kete problem
 - Marresi duhet te specifikoje seq # e paketes qe po ACKed
- ❖ Kerkon ore countdown

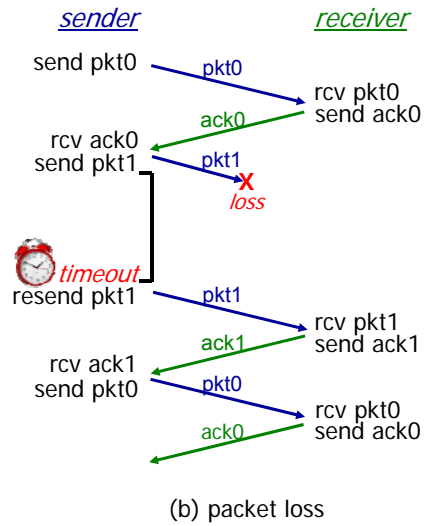
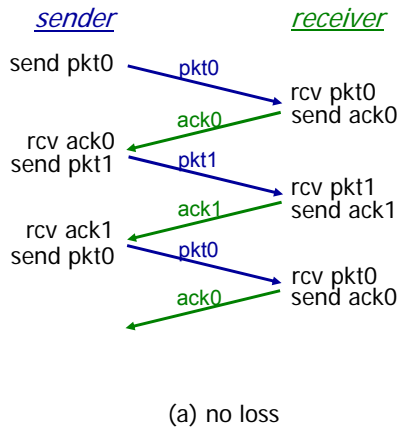
Transport Layer 3-15

rdt3.0 dergues



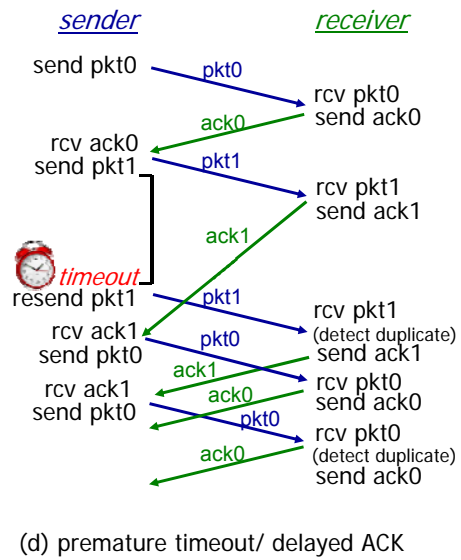
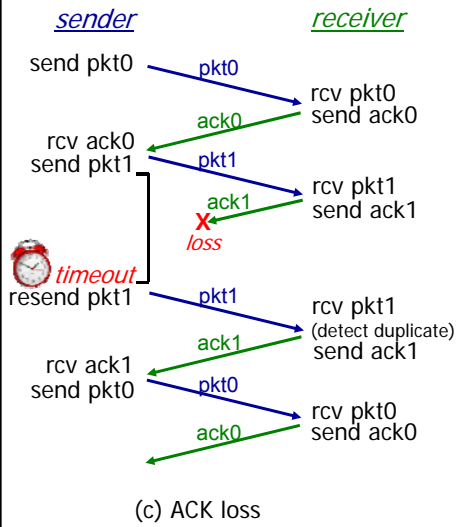
Transport Layer 3-16

rdt3.0 ne veprim



Transport Layer 3-17

rdt3.0 ne veprim



Transport Layer 3-18

Performanca e rdt3.0

- ❖ rdt3.0 eshte korrekte, por me performance te ulet
- ❖ e.g.: 1 Gbps link, 15 ms prop. delay, 8000 bit packet:

$$D_{trans} = \frac{L}{R} = \frac{8000 \text{ bits}}{10^9 \text{ bits/sec}} = 8 \text{ microsecs}$$

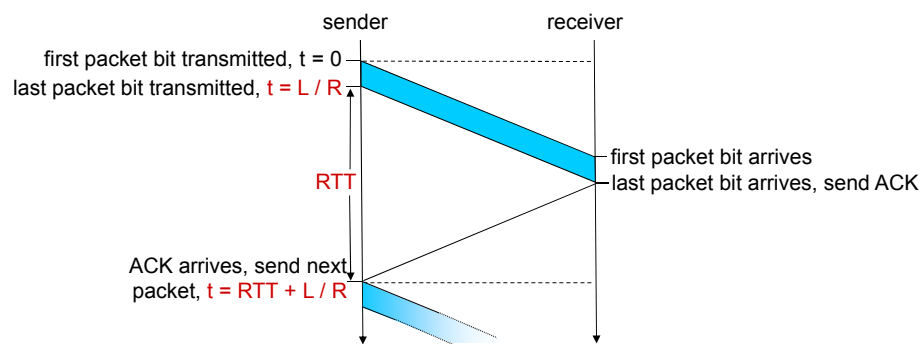
- U_{sender} : **utilization** – fraction of time sender busy sending

$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

- if RTT=30 msec, 1KB pkt every 30 msec: 33kB/sec thruput over 1 Gbps link
- ❖ Protokoli kufizon perdorimin e risurseve fizike!

Transport Layer 3-19

rdt3.0: stop-and-wait operation



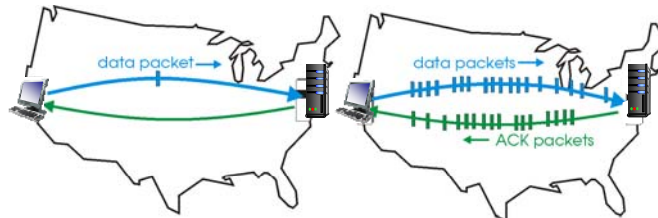
$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

Transport Layer 3-20

Pipelined protocols

pipelining: derguesi lejon disa, “in-flight”, yet-to-be-acknowledged pkts

- Hapesira e numerave te sequencave duhet rritur
- buffering at sender and/or receiver



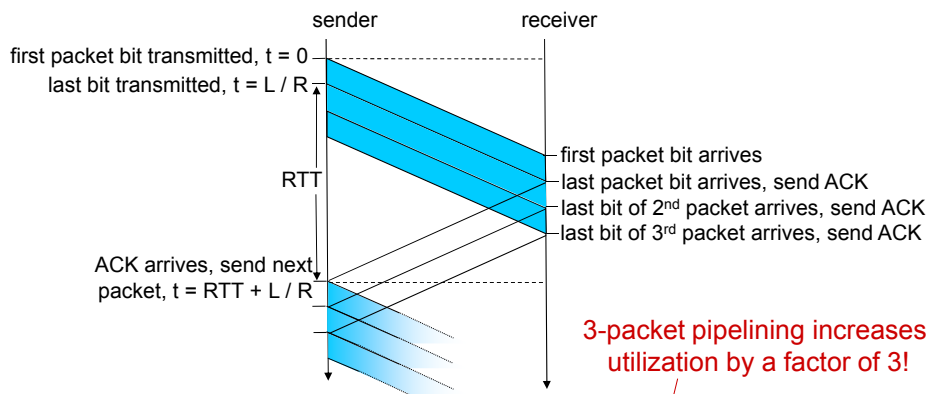
(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

❖ Dy forma te pergjithshme te pipelined protocols: *go-Back-N, selective repeat*

Transport Layer 3-21

Pipelining: increased utilization



$$U_{\text{sender}} = \frac{3L / R}{RTT + L / R} = \frac{.0024}{30.008} = 0.00081$$

Transport Layer 3-22

Pipelined protocols: overview

Go-back-N:

- ❖ Derguesi mund te kete deri ne N paketa unacked (te pa konfirmuara) ne pipeline
- ❖ Marresi dergon vetem *cumulative (grumbullues) ack*
 - Nuk e ack paketen nese ka nje gap
- ❖ Derguesi ka timer per paketen me te vjeter te pa konfirmuar
 - kur timer mbaron, ritransmeton te gjitha kapetat e pakonfirmuara

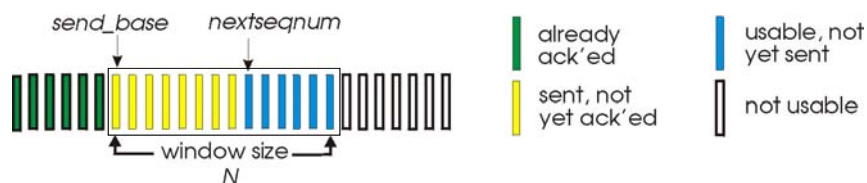
Selective Repeat:

- ❖ Derguesi mund te kete deri ne N paketa te pa konfirmuara ne pipeline
- ❖ marresi con *individual ack* per cdo pakete
- ❖ Derguesi mban nje timer per cdo pakete te pakonfirmuar
 - kur timer mbaron, ritransmeton vetem ate pakete te pa konfirmuar

Transport Layer 3-23

Go-Back-N: derguesi

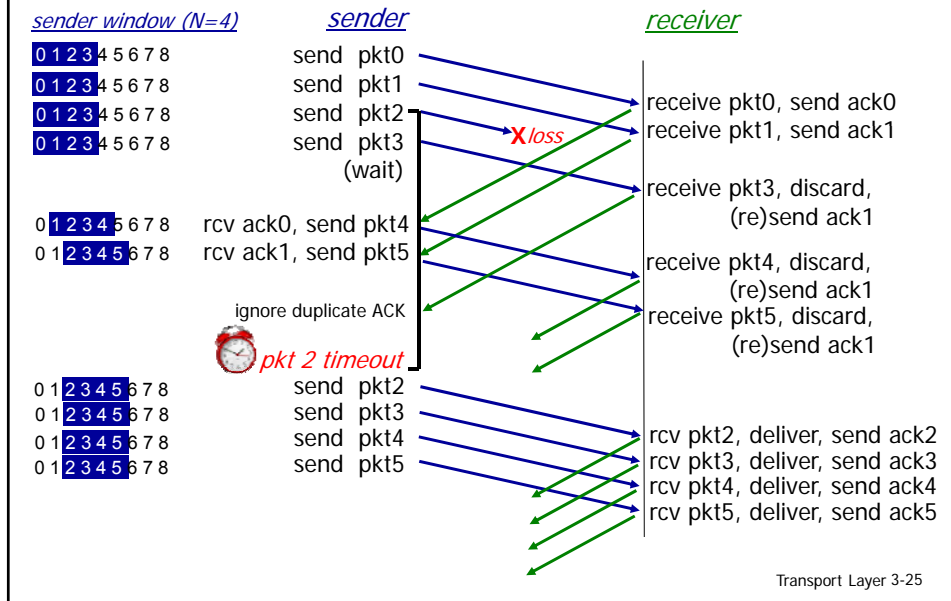
- ❖ k-bit seq # in pkt header
- ❖ “dritarja” gjer ne N, lejohen paketa te pakonfirmuara njera pas tjetres



- ❖ ACK(n): konfirmon te gjitha paketa deri, perfshire seq # n - *“cumulative ACK” (ACK grumbullues)*
 - Muns te marre ACKs te perseritura (shiko marresin)
- ❖ timer per paketen me vjeter qe eshte ne udhetim
- ❖ *timeout(n)*: ritransmeton paketen n dhe te gjitha paketat e tjera me seq # me te madh ne dritare

Transport Layer 3-24

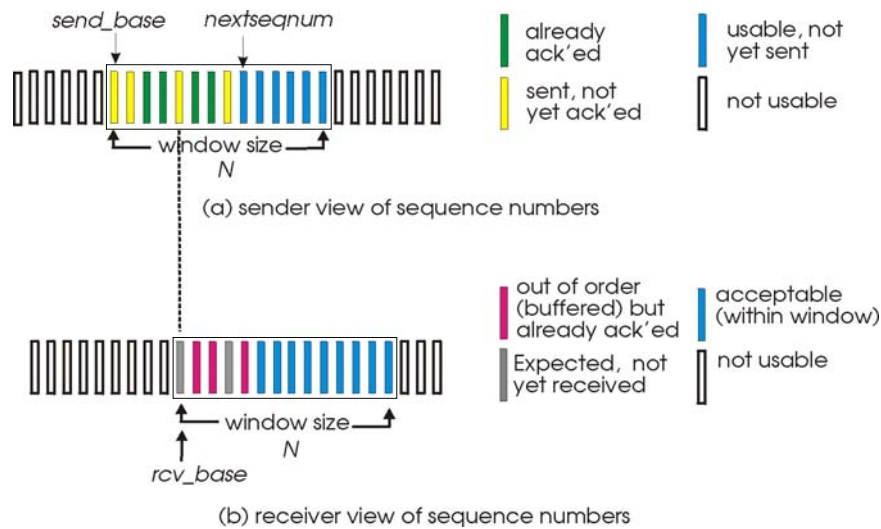
GBN ne veprim



Selective repeat

- ❖ Marresi ne menyre individuale konfirmon te gjitha paketat e marra ne rregull
 - Ruan pkts ne bufera, sipas nevojës, per tu derguar simbas rradhes tek shtresa e lartme
- ❖ Derguesi ridwergon vetem paketat per te cilat nuk eshte marre konfirmim (ACK)
 - Derguesi ka timer per cdo pakete te pakonfirmuar
- ❖ Dritarja e derguesit
 - N seq # te njepasnjeshme
 - Kufizon seq #s e paketave te derguara dhe te te atyre te pakonfirmuara

Selective repeat: sender, receiver windows



Selective repeat

dergues

Te dhena nga lart:

- ❖ Nqs ka seq # in window, dergo pkt

timeout(n):

- ❖ ridergo pkt n, rifillo timer

ACK(n) in [sendbase, sendbase+N]:

- ❖ shenjo pkt n si te marre
- ❖ Nqs n eshte paketa me sec # me te vogel e pakonfirmuar, perparo window base tel seq # I paketes tjeter te pakonfirmuar

marres

pkt n in [rcvbase, rcvbase+N-1]

- ❖ dergo ACK(n)
- ❖ out-of-order (jashte rradhes) : buffer
- ❖ in-order: dergo tek aplikimi (gjithashtu dergo paketat e ruajtura simbas rradhes), perparo dritaren paketa pasardhese qe nuk eshte marre akoma

pkt n in [rcvbase-N, rcvbase-1]

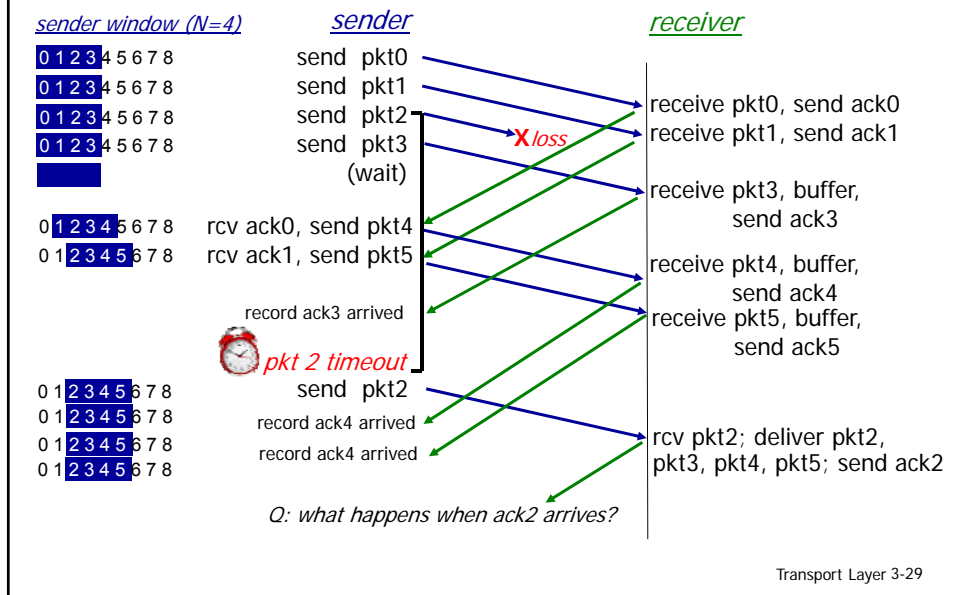
- ❖ ACK(n)

perdryshe:

- ❖ Mos bej gje

Transport Layer 3-28

Selective repeat in action

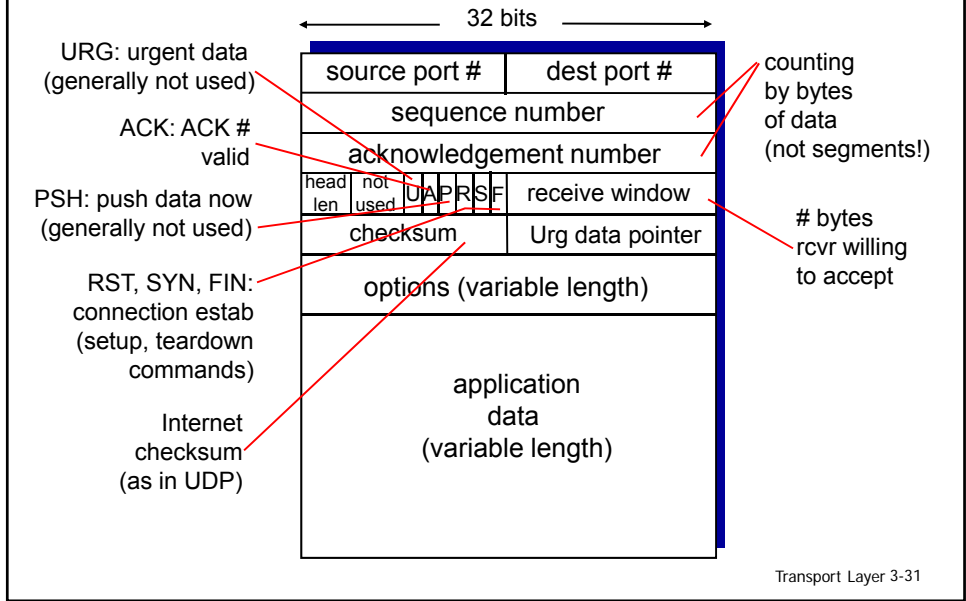


TCP: Overview RFCs: 793, 1122, 1323, 2018, 2581

- ❖ **point-to-point:**
 - Nje dergues, nje marres
- ❖ **reliable, in-order varg byte:**
 - no "message boundaries"
- ❖ **pipelined:**
 - TCP congestion and flow control vendosin per madhesine e dritares

- ❖ **full duplex data:**
 - Rrjedhje dy drejtimeshe e te dhenave ne te njejten lidhje (connection)
 - MSS: gjatesia me e madhe e segmentit
- ❖ **connection-oriented (me lidhje):**
 - handshaking (pershendetje) (shkembim mesazhesh) inits sender, receiver state before data exchange
- ❖ **flow controlled (kontroll i rrjedhjes):**
 - Derguesi nuk do te dergoje me shume se sa marresi mund te pranoje

TCP segment structure



TCP seq. numbers, ACKs

sequence numbers:

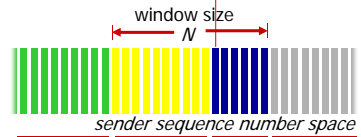
- Numeri i bytit te pare ne te dhenat e segmentit

acknowledgements:

- seq # e bytit pasardhes qe pritet nga ana tjetër
- cumulative ACK (grumbullues)

outgoing segment from sender

source port #	dest port #
sequence number	
acknowledgement number	
rwnd	
checksum	urg pointer

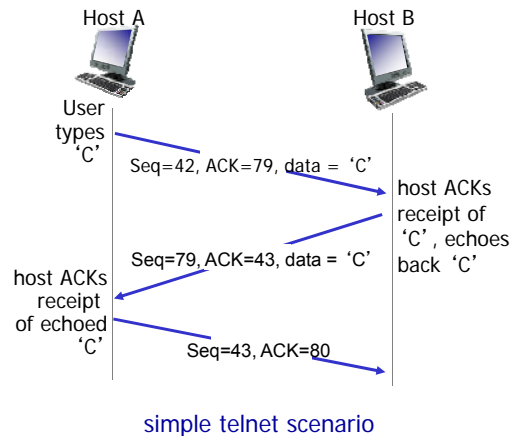


incoming segment to sender

source port #	dest port #
sequence number	
acknowledgement number	
checksum	urg pointer

Transport Layer 3-32

TCP seq. numbers, ACKs



Transport Layer 3-33

TCP round trip time, timeout

Q: Si te vendoset vlera e timeout e TCP?

- ❖ Me e gjate se RTT
 - por RTT ndryshon
- ❖ *Shume e shkurter:* timeout te parakohshem, ritransmetim te panevojshes
- ❖ *Shume te gjate:* reagim te ngadalte ndaj humbjes se segmentit

Q: si ta vleresojme RTT?

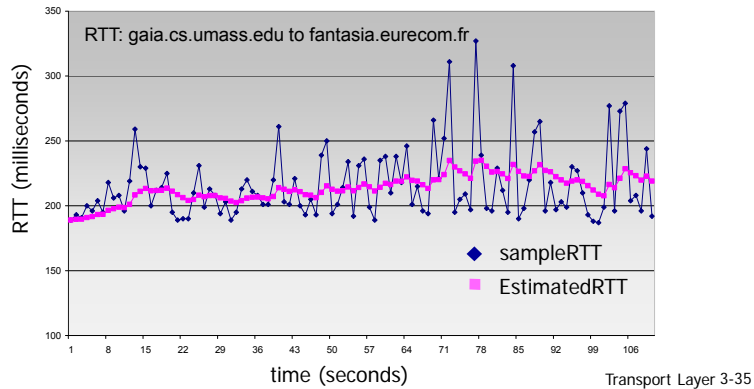
- ❖ **SampleRTT:** matet qe nga transmetimi i segmentit gjer ne marrjen e ACK
 - Nuk konsiderohen retransmissions
- ❖ **SampleRTT** ndryshon
 - Behet nje mesatare midis matjeve

Transport Layer 3-34

TCP round trip time, timeout

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

- ❖ exponential weighted moving average
- ❖ influence of past sample decreases exponentially fast
- ❖ typical value: $\alpha = 0.125$

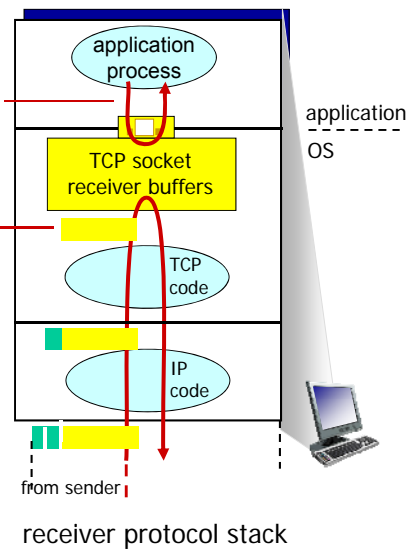


TCP flow control

application may remove data from TCP socket buffers ...

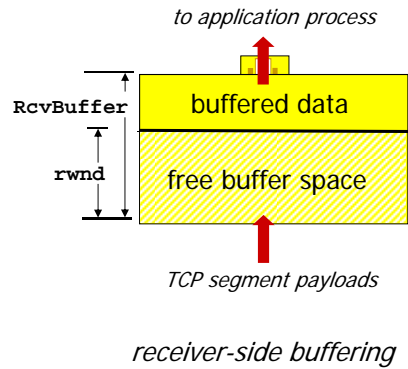
... slower than TCP receiver is delivering (sender is sending)

flow control
 Marresi kontrollon derguesin, pra derguesi nuk mund te permyse buferin e marresit duke transmetuar shume dhe shume shpejt



TCP flow control

- ❖ Marresi “reklamon” hapësirën e lirë të buferit duke përfshirë vlerën e **rwnd** në TCP header të segmenteve receiver-tek-sender
- ❖ Dërguesi kufizon sasinë e paketave të pakonfirmuara (në udhëtim) sac është **rwnd** e marresit
- ❖ Garanton që buferi i marresit nuk përmbetet

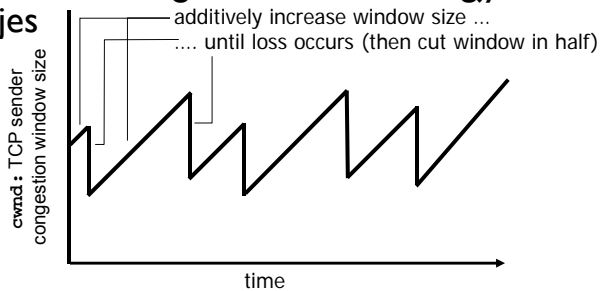


Transport Layer 3-37

TCP congestion control: additive increase multiplicative decrease

- ❖ **qasje**: dërguesi rrit transmission rate (window size), duke testuar për bandwidth që mund të përdoret, gjerisa ndodhë ndonjë humbje
 - **additive increase**: rritje e **cwnd** me 1 MSS për çdo RTT gjersa dedektohen humbje
 - **multiplicative decrease**: ngushton **cwnd** në gjysëm mbas humbjes

AIMD saw tooth behavior: probing for bandwidth



Transport Layer 3-38

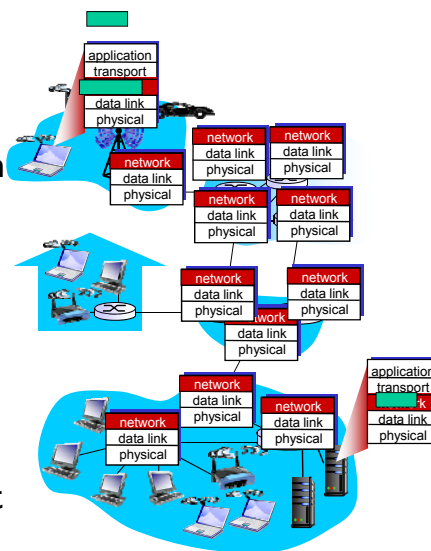
Chapter 3: summary

- ❖ principles behind transport layer services:
 - multiplexing, demultiplexing
 - reliable data transfer
 - flow control
 - congestion control
 - ❖ instantiation, implementation in the Internet
 - UDP
 - TCP
- next:**
- ❖ leaving the network “edge” (application, transport layers)
 - ❖ into the network “core”

Transport Layer 3-39

Shtresa e Rrjetit

- ❖ Transporton segmentet nga derguesi tek marresi
- ❖ ne anen derguese i mbeshtjellin segmentet ne datagrama
- ❖ Ne anen marrese, con segmentet tek shtresa e transportit
- ❖ Protokollet e shtreses rrjetit ne cdo *every* host, routera
- ❖ routerat examinojne fushen e header ne te gjithat datagramat IP e kalojne nder ta



Network Layer 4-40

Dy funksionet kryesore te shtreses se rrjetit

❖ *Forwarding (shtyrja)*: leviz paketat nga hyrja e router ne dalje te duhur te router

❖ *Routing (rrugezimi)*: percakton rrugen e paketer nga burimi ne destinacion

- Algoritmat e routing

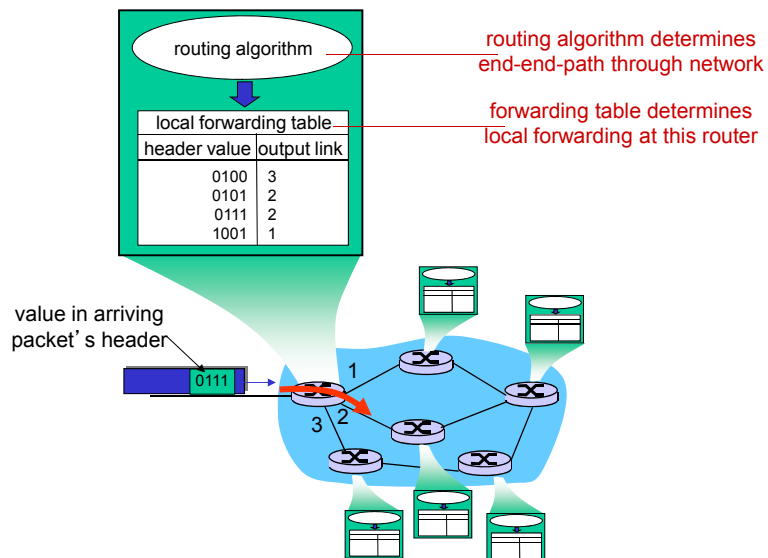
analogji:

❖ *routing*: procesi I planifikimit te udhetimit nga burimi ne destinacion

❖ *forwarding*: procesi i kalimit ne nje nje lidhese

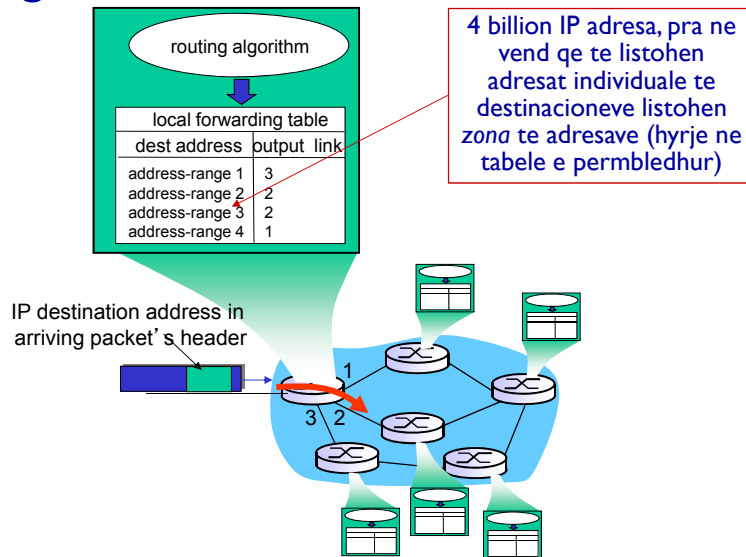
Network Layer 4-41

Bahkveprimi ndermjet routing dhe forwarding



Network Layer 4-42

Tabela e forwarding se datagramit



Datagram forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Network Layer 4-44

Perputhja me e gjate e prefiksit

Perputhja me e gjate e prefiksit

Kur shikon per hyrjen ne tabelen e forwarding per nje adrese destinacioni te dhene, perdor prefiksin me te gjate te adreses qe perputhet me adresean e destinacionit

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

examples:

DA: 11001000 00010111 00010**110** 10100001

which interface?

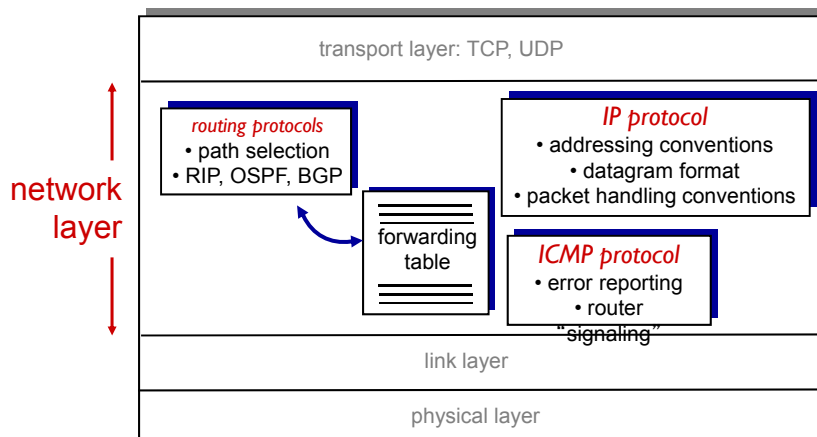
DA: 11001000 00010111 00011**000** 10101010

which interface?

Network Layer 4-45

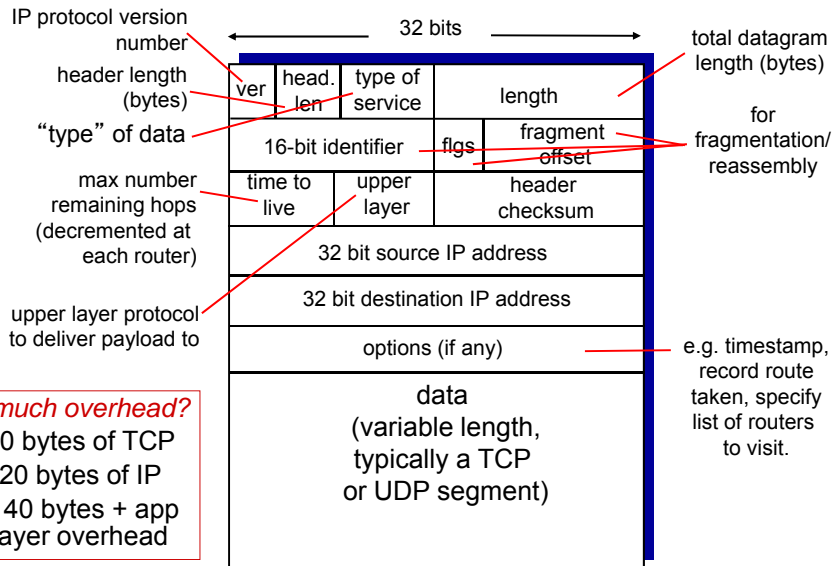
Shtresa e rrjetit te Internetit

Funksione te shtreses se rrjetit ne host, router:



Network Layer 4-46

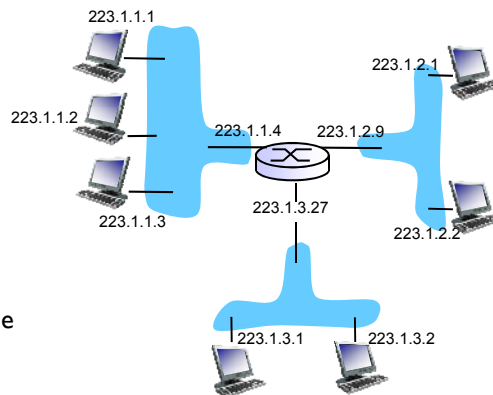
IP datagram format



Network Layer 4-47

Hyrje ne adresimin IP

- ❖ **IP address:** 32-bit identifikues per host, nderfaqen e routerit
- ❖ **nderfaqe:** lidhje ndermjet host/router dhe linkut fizik
 - Routerat zakonisht kane shume nderfaqe
 - Hostet kane zakonisht nje apo dy nderfaqe (psh, wired Ethernet, wireless 802.11)
- ❖ **Adresat IP te asociuara me cdo nderfaqe**



223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

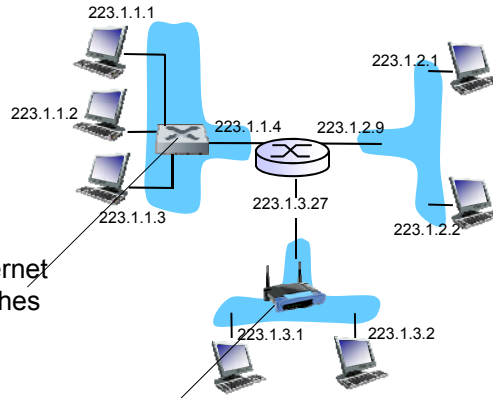
Network Layer 4-48

IP addressing: introduction

Q: si jane lidhur nderfaqet?

A: chapter 5, 6.

A: nderfaqer kablore Ethernet lidhen me Ethernet switches



A: wireless WiFi interfaces connected by WiFi base station

Network Layer 4-49

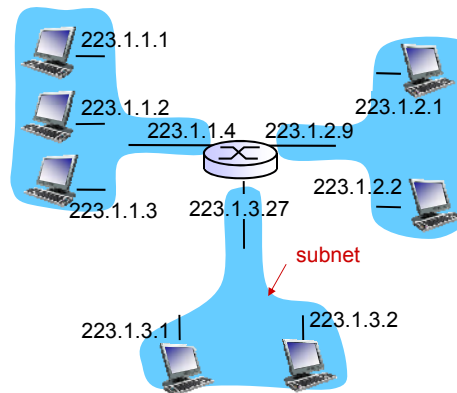
Subnets

❖ **IP address:**

- subnet part – bitet e pare
- host part – bitet e fundit

❖ **Cfare eshte nje subnet ?**

- Nderfaqe paisjesh me pjese subnet te njejte te adreses IP
- Mund t elidhen fizikisht me njera tjetren **pa nderhyrjen e routerave**



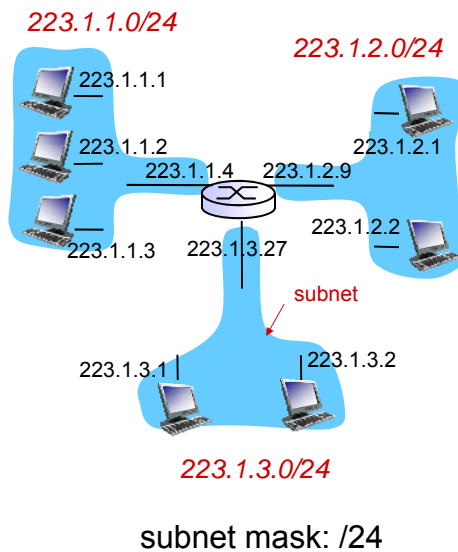
Rrjet qe perbehet nga 3 subnete

Network Layer 4-50

Subnets

recipe

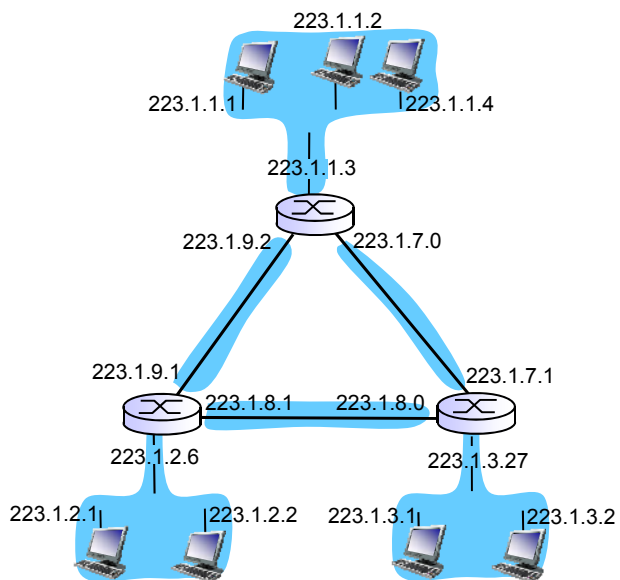
- ❖ Per te percaktuar subnetet, zgjidh cdo nderfaqe nga hosti apo routeri i saj, duke krijuar ishuj rrjetash te izoluara
- ❖ Cdo rrjet i izoluar qohet nje **subnet**



Network Layer 4-51

Subnets

how many?



Network Layer 4-52

DHCP: Dynamic Host Configuration Protocol

Qellim: lejon hostin qe ne menyre dinamike marre IP address nga serveri I rrjetit kur i bashkohet rrjetit

- Mund ta rinovoje mbajtjen e adreses ne perdorim
- Lejon ripordorimin e adresave (i mban adresat vetem kur jane te lidhur)
- Suporton perdoruesit mobile qe duan t'i bashkohen rrjetit

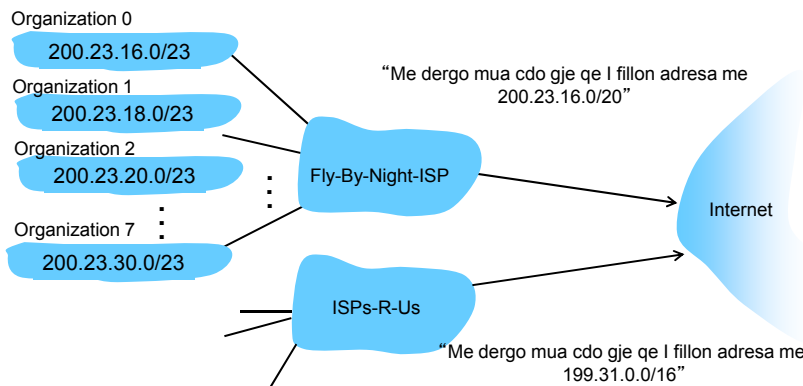
DHCP overview:

- host broadcasts “DHCP discover” msg [optional]
- DHCP server pergjigjet me “DHCP offer” msg [optional]
- host kerkon IP adres: “DHCP request” msg
- DHCP server dergon adresen: “DHCP ack” msg

Network Layer 4-53

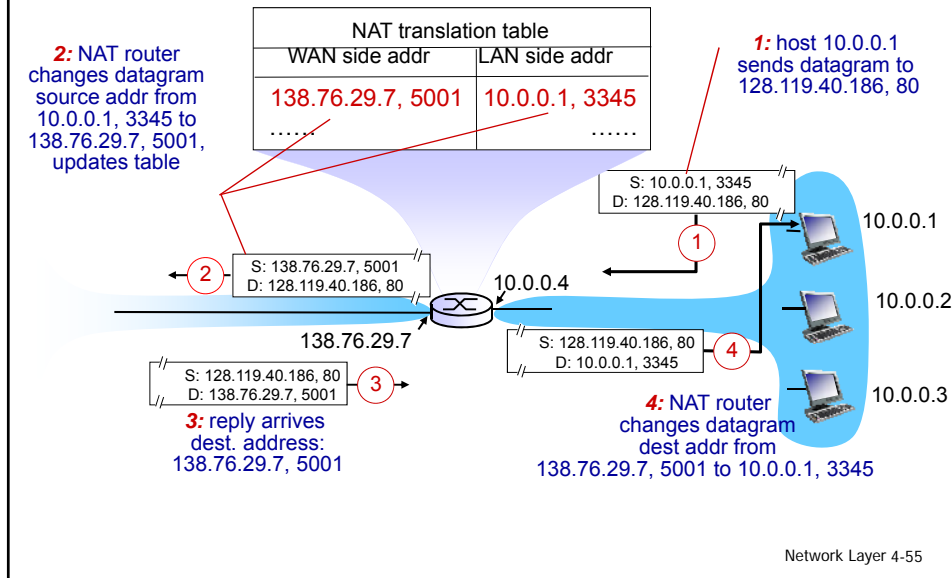
Adresimi hierarkik: permbledhja e rrugëve

Adresimi hierarkik lejon reklamim eficient te informacionit te rrugëve:



Network Layer 4-54

NAT: network address translation



Dijkstra's Algorithm

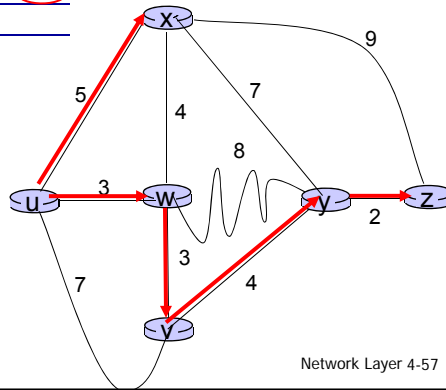
- 1 **Initialization:**
- 2 $N' = \{u\}$
- 3 for all nodes v
- 4 if v adjacent to u
- 5 then $D(v) = c(u,v)$
- 6 else $D(v) = \infty$
- 7
- 8 **Loop**
- 9 find w not in N' such that $D(w)$ is a minimum
- 10 add w to N'
- 11 update $D(v)$ for all v adjacent to w and not in N' :
 - 12 $D(v) = \min(D(v), D(w) + c(w,v))$
 - 13 /* new cost to v is either old cost to v or known
 - 14 shortest path cost to w plus cost from w to v */
 - 15 **until all nodes in N'**

Dijkstra's algorithm: example

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w	5,u	11,w	∞	∞
2	uwx	6,w		11,w	14,x	
3	uwxv			10,v	14,x	
4	uwxvy				12,y	
5	uwxvyz					

notes:

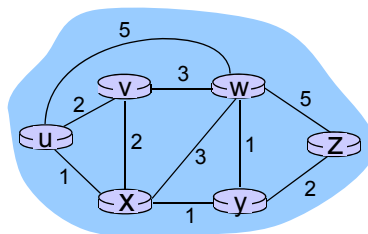
- ❖ construct shortest path tree by tracing predecessor nodes
- ❖ ties can exist (can be broken arbitrarily)



Network Layer 4-57

Dijkstra's algorithm: another example

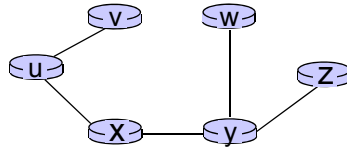
Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x	2,x	∞	∞
2	uxy	2,u	3,y		4,y	
3	uxyv		3,y		4,y	
4	uxyvw				4,y	
5	uxyvwz					



Network Layer 4-58

Dijkstra's algorithm: example (2)

resulting shortest-path tree from u:



resulting forwarding table in u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Network Layer 4-59

Distance vector algorithm

Bellman-Ford equation (dynamic programming)

let

$d_x(y) :=$ cost of least-cost path from x to y

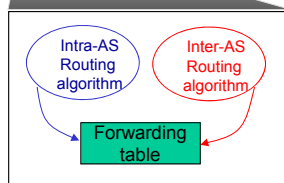
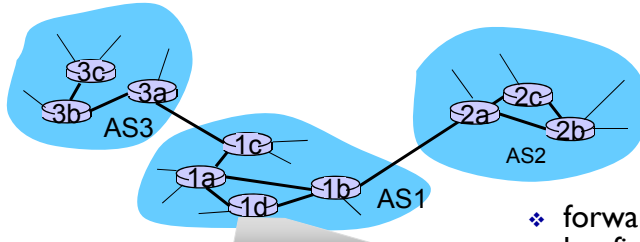
then

$$d_x(y) = \min \{ c(x,v) + d_v(y) \}$$

\min taken over all neighbors v of x
cost to neighbor v
cost from neighbor v to destination y

Network Layer 4-60

Autonomous Systems (Ases) e nderlidhur



- ❖ forwarding table te konfiguruar nga te dy algoritmet intra- and inter-AS routing
 - intra-AS vendos hyrje per destinacione te brendeshme
 - inter-AS & intra-AS vendos hyrje per destinacione te jashteme

Network Layer 4-61

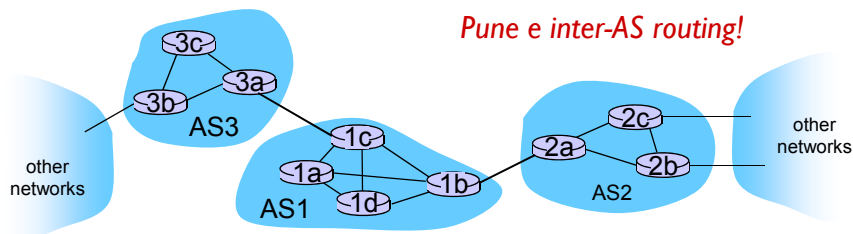
Inter-AS tasks

- ❖ Supozo routeri AS1 merr datagram te destinuar jashte ASI:
 - router duhet te dergoje paketen drejt gateway router, por te cili?

ASI duhet:

1. Mesoje cilet destinacione jane te arritshme nepermjet AS2, e cilet nepermjet AS3
2. Shperndaje kete informacion arritshmerie tek te gjithe routerat ne ASI

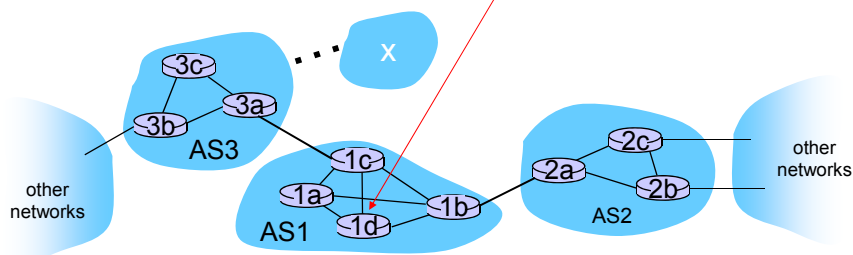
Pune e inter-AS routing!



Network Layer 4-62

Shembull: vendosja e forwarding table ne routerin Id

- ❖ supozo AS1 meson (nepermjet inter-AS protocol) qe subnet *x* arrihet nepermjet AS3 (gateway Ic), por jo nepermjet AS2
 - inter-AS protocol shperndan info e arritshmerise tek te gjithe routerat e brendeshem
- ❖ router Id percakton nga intra-AS routing info qe nderfaqia e tij *I* eshte ne rrugen me te lire tek Ic
 - instalon forwarding table entry (*x,I*)



Network Layer 4-63

Internet inter-AS routing: BGP

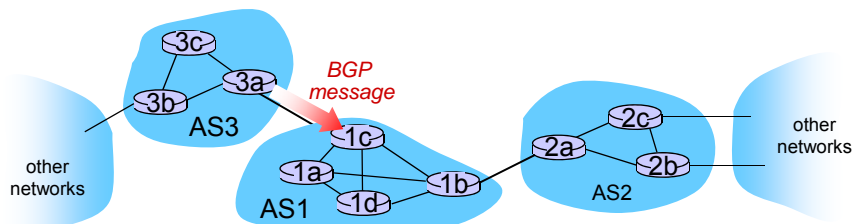
- ❖ **BGP (Border Gateway Protocol):** the de facto inter-domain routing protocol
 - “ngjitesi qe mban Internetin bashke”
- ❖ BGP i jep cdo AS nje menyre per:
 - **eBGP:** perfoje informacionin e arrtshmerise se subneteve nga Ass fqinje.
 - **iBGP:** perhap informacionin e arrtshmerise tek te gjithe ruterat brenda AS.
 - Percakton rruget e “mira” tek rrjetat e tjera bazuar tek informacioni I arrtshmerise dhe and rregullat.
- ❖ Lejon subnet te reklamoje eksistencen e vet tek Interneti: “*Jam ketu*”

Network Layer 4-64

BGP basics

- ❖ **BGP session:** dy routera BGP (“peers”) shkëmbejnë BGP messages:
 - Duke reklamuar *paths (rruge)* tek prefikset e rrjetave të ndryshme destinacion (“path vector” protocol)

- ❖ kur AS3 reklamon një prefiks tek AS1:
 - AS3 *premtton* që do të drejtojë datagramet drejt atij prefiksi
 - AS3 mund të përmbledhë prefikset në reklamimet e veta

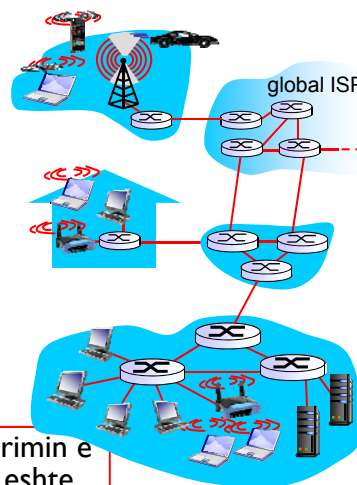


Network Layer 4-65

Shtresa e Linkut: hyrje

terminology:

- ❖ Hoste dhe routera: **nyje**
- ❖ Kanale komunikimi që lidhin nyje fqinje gjatë rrugëve të komunikimit: **links**
 - Links të kablluar
 - wireless links
 - LANs
- ❖ layer-2 packet: **frame**, mbështjell datagram



data-link layer ka përgjegjësi për transferimin e datagram nga një nyje tek një nyje që është fizikisht fqinje në një link

Link Layer 5-66

Link layer: context

- ❖ datagram te transferuara nga protokolle te ndryshem ne linke te ndryshem:
 - psh, Ethernet ne linkun e pare, frame relay ne linket e ndermjetem, 802.11 linkun e fundit
- ❖ Cdo protokoll linkusiguron sherbime te ndryshme
 - psh, mundet ose jo te siguroje transmerim te besueshem mbi link

Link Layer 5-67

Link layer - sherbime

- ❖ *framing, link access:*
 - Mbeshtjell datagram ne frame, shton header, trailer
 - channel access nqs eshte mjedis i perbashket
 - “MAC” addresses perdoren ne headerat e frame per te identifikuar burimin, dest
 - Te ndryshem nga IP address!
- ❖ *Transmetim i besueshem ndermjet nyjeve fqinje*
 - Kete e kemi diskutuar tek shtresa e transmetimit

Link Layer 5-68

Link layer services (more)

❖ *flow control:*

- Nyja me e ngadalte percakton ritmin e rrjedhes

❖ *error detection:*

- Gabimet e shkaktuara nga signal attenuation, zhurma.
- Marresi detekton pranine e gabimeve:
 - I sinjalizon derguesit per ritransmetim apo framet e humbur

❖ *error correction:*

- Marresi identifikon dhe rregullon bit error(s) pa bere ritransmetim

❖ *half-duplex and full-duplex*

- me half duplex, nyjet ne fund te linkut mund te transmetojne, por jo ne te njejten kohe

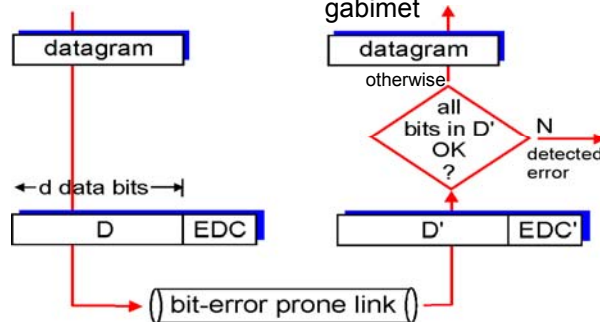
Link Layer 5-69

Error detection

EDC= Error Detection and Correction bits (redundancy)

D = Data protected by error checking, may include header fields

- Error detection jo 100% I besueshem!
- protocollit mund t'i shpetojne disa gabime, por rradhe
- sa me e madhe fusha EDC aq me mire detektohen dhe korrigojen gabimet



Link Layer 5-70

Multiple access links, protocols

Dy tipe "links":

❖ **point-to-point**

- PPP for dial-up access
- point-to-point link ndermjet Ethernet switch, host

❖ **broadcast (shared wire or medium)**

- Ethernet tip i vjeter
- upstream HFC
- 802.11 wireless LAN



shared wire (e.g.,
cabled Ethernet)



shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)



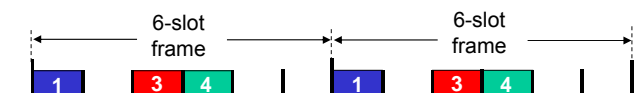
humans at a
cocktail party
(shared air, acoustical)

Link Layer 5-71

Protokolle te ndarjes se kanaleve: TDMA

TDMA: time division multiple access

- ❖ Kapja e kanalit "me rradhe"
- ❖ Cdo stacion merr intervale fikse kohe (length = pkt trans time) ne cdo rradhe
- ❖ Intervallet e papedrorur shkojne bosh
- ❖ shembull: 6-station LAN, 1,3,4 kane pkt, intervallet 2,5,6 bosh

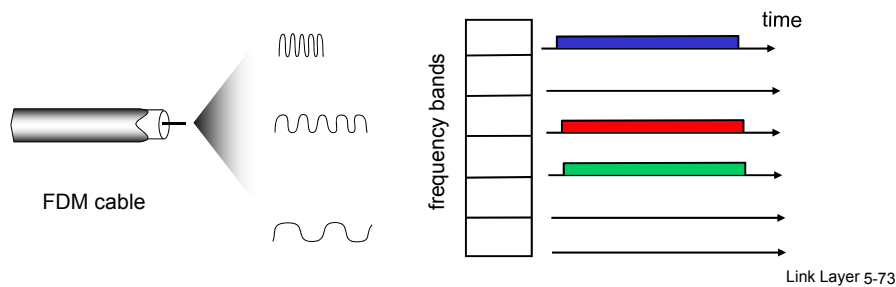


Link Layer 5-72

Channel partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- ❖ Spektri i kanalit ndahet ne zona frekuencash
- ❖ Cdo stacioni i jepet nje zone frekencash
- ❖ Kur nuk ka transmetim banda e frekencave eshte bosh
- ❖ example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



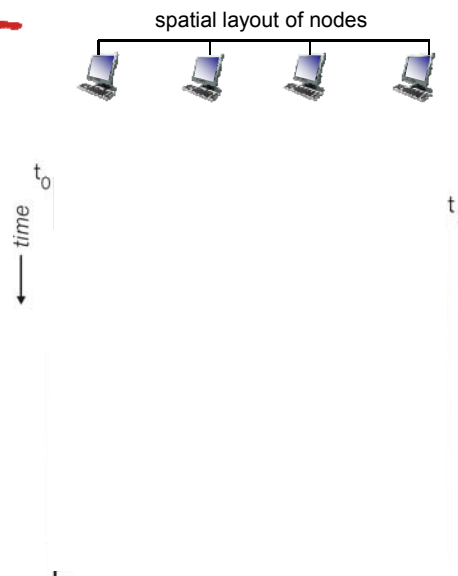
Random access protocols

- ❖ Kur nyja ka pakete per te derguar
 - Transmeton me te gjithe shpejtesine e transmetimit te kanalit R.
 - Pa koordinim paraprak midis nyjeve
- ❖ Dy ose me shume nyje transmetimi → “collision (perplasje)”,
- ❖ **random access MAC protocol** percakton:
 - Si te detektohen collisions
 - Si te kalohet collisions (e.g., duke vonuar ritransmetimet)
- ❖ Shembuj te random access MAC protocols:
 - slotted ALOHA
 - ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Link Layer 5-74

CSMA collisions

- ❖ **collisions mund te ndodhin:** vonesa e perhapjes ben qe dy nyje te mos mund te degjojne transmetimet e anasjellta
- ❖ **collision:** e gjithë paketa e shkuar dem
 - distanca & vonesa e perhapjes luajne rol ne percaktimin e probabilitetit te collision



Link Layer 5-75

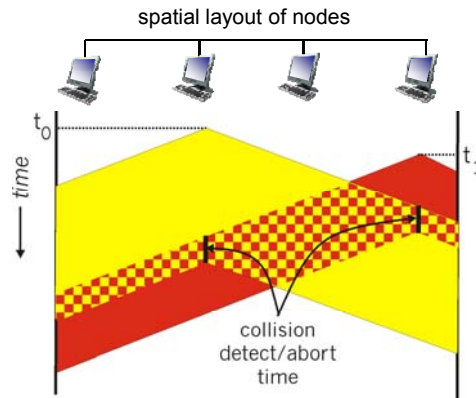
CSMA/CD (detektim i collision)

CSMA/CD: ndjen carrier (sinjalin), vonohet transmetimi si ne CSMA

- collisions *detektohen* ne kohe te shkurter
- Transmetimet qe po perplasen ndalohen, qe t ezvogelohet humbja e kanalit
- ❖ collision detection:
 - I lehte ne LANs kablore: mat nivelin e sinjalit, krahason sinjalet e tranmetuara e te marre
 - I veshtire ne wireless LANs: sinjali I marre eshte shume me I vogel se sa ai i transmetimit local
 -

Link Layer 5-76

CSMA/CD (collision detection)



Link Layer 5-77

Ethernet CSMA/CD algorithm

1. NIC merr datagram shtresa e rjetit, krijon frame
2. Nqs NIC e ndjen kanaln e lire, fillon transmetimin e frame. Nqs NIC kanaln e zene, pret sa kanali te behet i lire, pastaj transmeton.
3. Nqs NIC transmeton te gjithë frame pa detektuar nje tjetër transmetim, NIC e ka mbaruar punen me kete frame !
4. Nqs NIC detekton nje transmetim tjetër gjate transmetimit, ndalon dhe dergon sinjalin jam
5. Mbas ndalimit, NIC fillon *binary (exponential) backoff*:
 - Zgjedh ne menyre te rastin kohën për të ritransmetuar

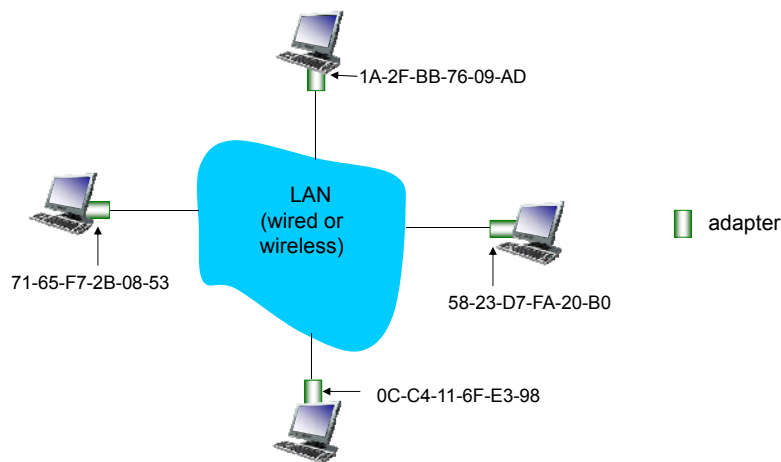
Link Layer 5-78

MAC addresses and ARP

- ❖ 32-bit IP address:
 - *network-layer* address for interface
 - used for layer 3 (network layer) forwarding
- ❖ MAC (or LAN or physical or Ethernet) address:
 - funksion: *perdoret 'lokalisht' per te cuar frame nga nje nderfaqe tek nje tjetër nderfaqe te lidhura fizikisht (i njejtë rrjet, në kuptimin e IP)*
 - 48 bit MAC address (for most LANs) te djegur në NIC ROM, nganjehere në software te ndryshueshme
 - e.g.: 1A-2F-BB-76-09-AD hexadecimal (base 16) notation (each "number" represents 4 bits)

LAN addresses and ARP

Cdo adapter në LAN ka një adresë unike në **LAN**



Link Layer 5-80

Ethernet frame structure

Adaptteri dergues mbeshtjell IP datagram (ose pakete te ndonje protokolli tjetër te rrjetit) ne **Ethernet frame**^{type}



preamble:

- ❖ 7 bytes me pattern 10101010 ndjekur nga 10101011
- ❖ per te sinkonizuar

Link Layer 5-81

Ethernet frame structure (more)

- ❖ **addresses:** 6 byte burimi, destinacioni MAC addresses
 - Nqs adaptteri merr frame me adrese destinite te tij, ose me broadcast address (e.g. ARP packet), i kalon te dhenat e frame tek protokolli i shtreses se rrjetit
 - perndryshe, adaptteri nuk merret me frame
- ❖ **type:** tregon tipin e protokollit me lart (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- ❖ **CRC:** cyclic redundancy check ne marres
 - error detektohet: frame hidhet



Link Layer 5-82

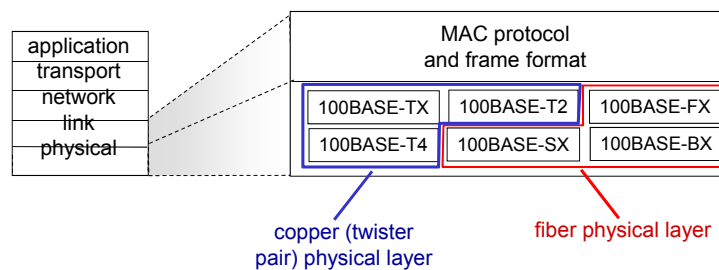
Ethernet: unreliable (pa besueshmeri), connectionless (pa lidhje)

- ❖ **connectionless**: nuk ka pershendetje ndermjet derguesit dhe marresit
- ❖ **unreliable**: NIC i marresit nuk dergon acks ose nacks tek NIC i derguesit
 - Nqs ka probleme merren protokollet e ma larte
- ❖ Ethernet' s MAC protocol: unslotted **CSMA/CD with binary backoff**

Link Layer 5-83

802.3 Ethernet standards: link & physical layers

- ❖ **shume** standarte te ndryshem Ethernet
 - Te njejtinMAC protocol dhe format frame
 - Shpejtesi te ndryshme: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
 - Shtrese fizike te ndryshme: fiber, cable



Link Layer 5-84

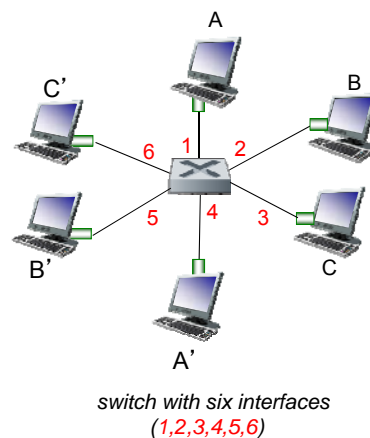
Ethernet switch

- ❖ **Paisje e link-layer: ka rol aktiv**
 - Ruan, forward Ethernet frames
 - Analizom adresen MAC ne framen ardhese, ne menyre selective dergon frame tek nje apo me shume linke dales kur frame duhet te dergohet ne nje segment, perdor CSMA/CD per te akses segmentin
- ❖ **transparent**
 - hosts nuk e dine pranine e switcheve
- ❖ **plug-and-play, self-learning**
 - Switchet nuk ka nevojte te configurohen

Link Layer 5-85

Switch: shume transmetime te njekohshem

- ❖ Hostet kane lidhje direkte, te dedikuara tek switch
- ❖ switchet buffer (ruajne) packets
- ❖ Ethernet protocol I perdorur ne cdo link hyres, per pa collisions; full duplex
 - Cdo link eshte collision domain me vehte
- ❖ **switching**: A-to-A' and B-to-B' mund t e transmetojne njekohesisht pa humbje



Link Layer 5-86

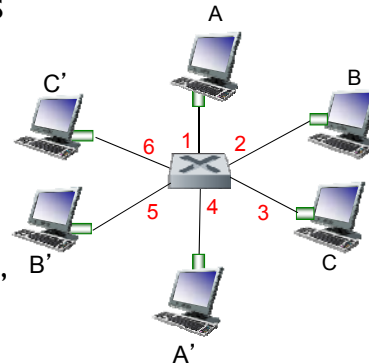
Switch forwarding table

Q: si e di switchi qe A' arrihet nepermjet nderfaqes 4, B' arrihet nepermjet nderfaqen 5?

❖ **A:** cdo switch ka nje **switch table**, ku cdo hyrje:

- (MAC address e hostit, nderfaqeja per te arritur hostin, time stamp)
- Ngjan si nje tabele routing

Q: si krojohen e mbahenhyrjet ne switch table?



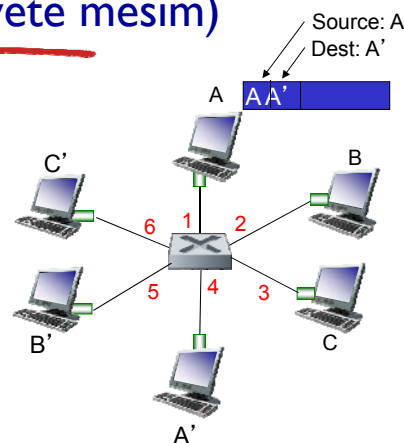
switch with six interfaces (1,2,3,4,5,6)

Link Layer 5-87

Switch: self-learning (vete mesim)

❖ Switchi **meson** cilet hoste mund te arrihen nepermjet ciles nderfaqe

- Kur merr nje frame, switchi "meson" vendosjen e derguesit: segmentin hyres LAN
- Shenon ciftin dergues/vendosje ne switch table



MAC addr	interface	TL
A	1	60

Switch table (initially empty)

Link Layer 5-88

Switch: frame filtering/forwarding

Kur nje frame merret ne switch:

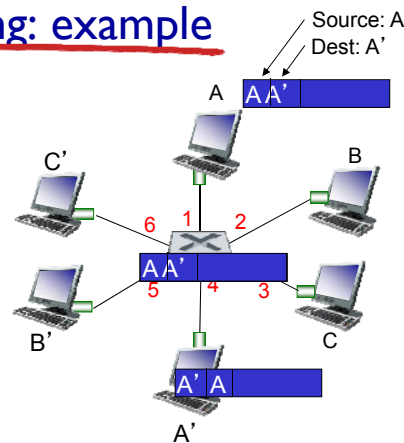
1. Shenon linkun hyres, MAC address te hostit dergues
2. indekson switch table duke perdorur MAC destination address
3. if entry gjendet per destinacionin
 - then {
 - if destinacioni ne segmentin nga i cili erdhi frame
 - then hedh frame
 - else dergon frame ne nderfaqen e treguar nga entry
 - }
 - else flood /* forward on all interfaces except arriving interface */

Link Layer 5-89

Self-learning, forwarding: example

- ❖ frame destinacon, A', vendodhja e panjohur: *flood*
- ❖ Vendodhja e A e njohur:

ne menyre selektive dergo ne nje link te vetem



MAC addr	interface	TTL
A	1	60
A'	4	60

switch table (initially empty)

Link Layer 5-90