



# Tietoliikenteen perusteet

Syksy 2017

Timo Karvi

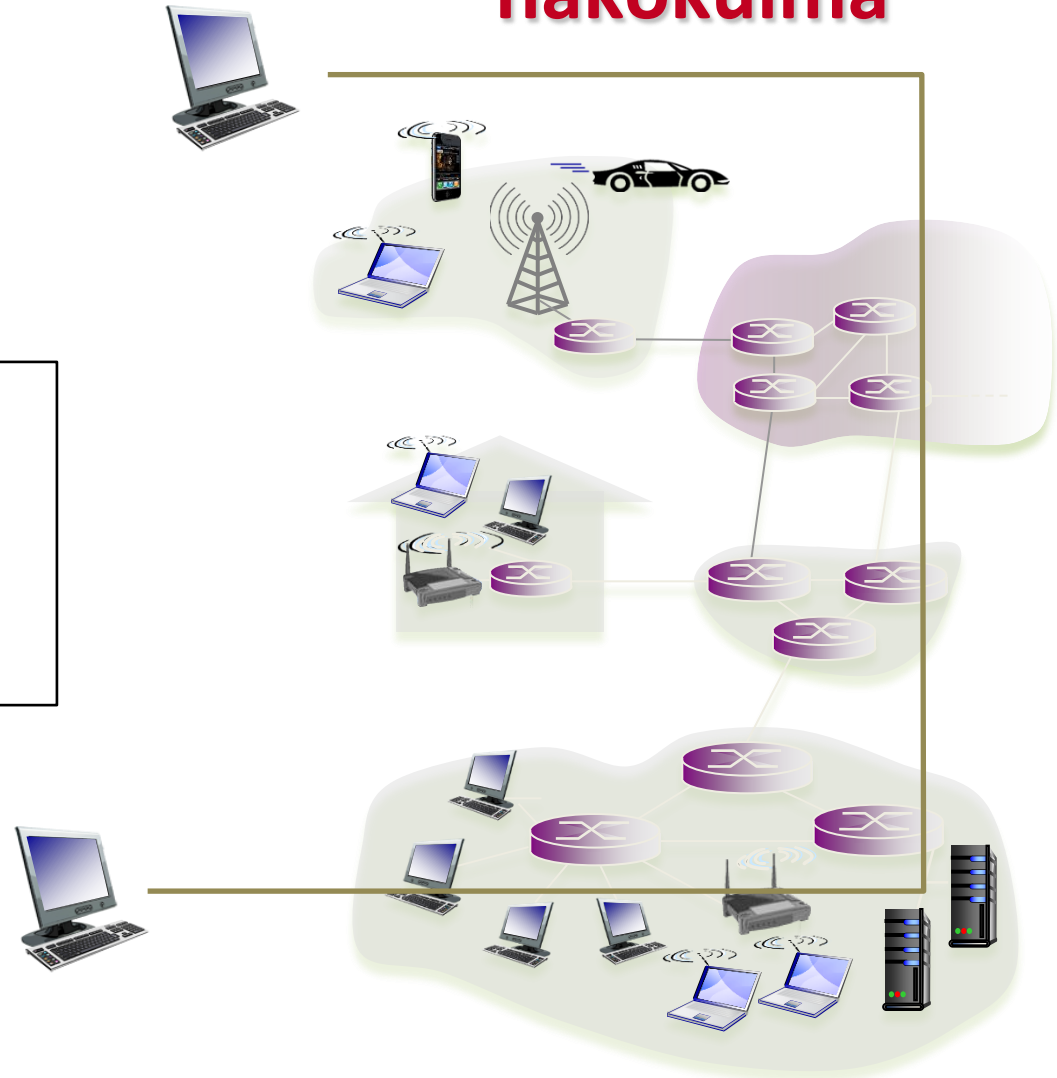
Pääasiallisesti kuvien  
© J.F Kurose and K.W. Ross, All  
Rights Reserved

# Käyttäjän näkökulma

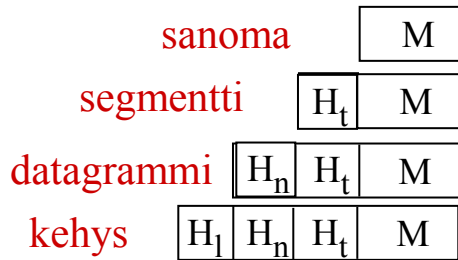
*Lähettäjä/ Lähde  
(sender, source)*

Käyttäjä ei näe  
suoraan verkon  
sisäistä toimintaa.

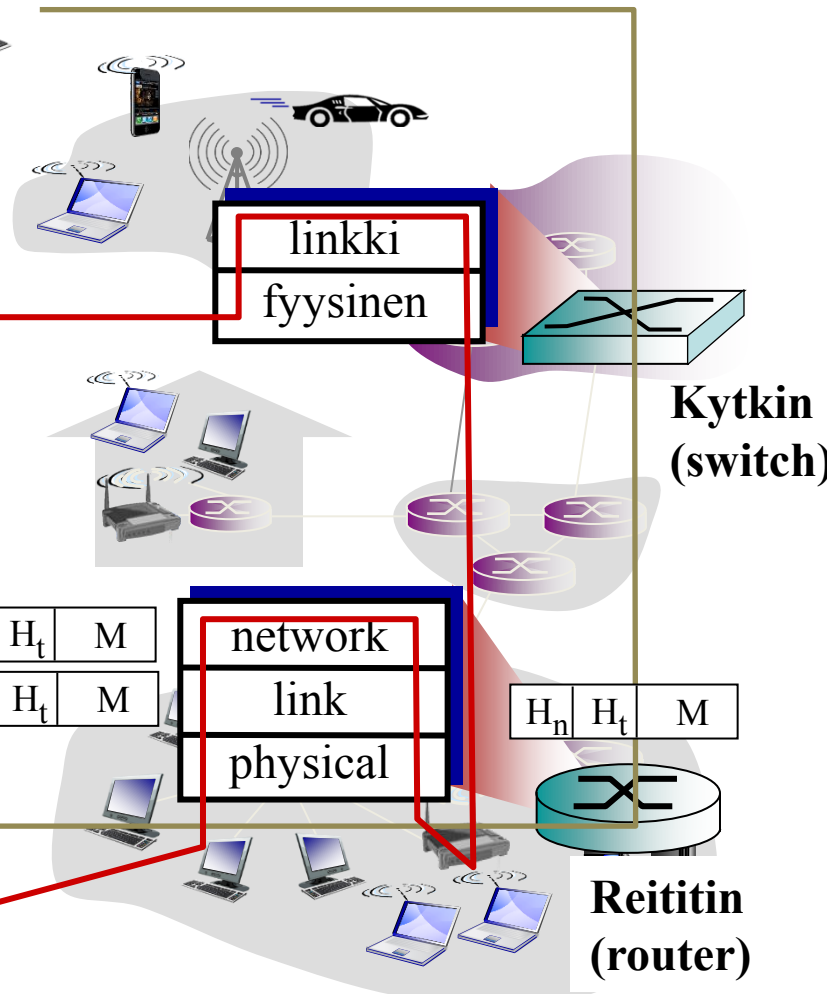
*Vastaanottaja / Kohde  
(receiver, destination)*



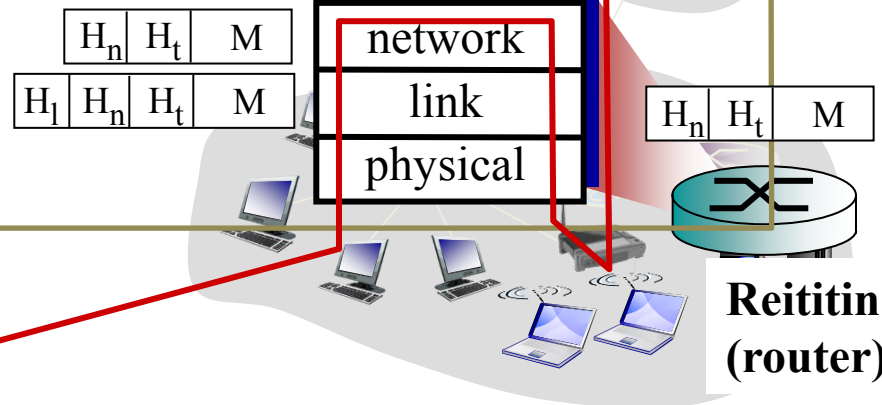
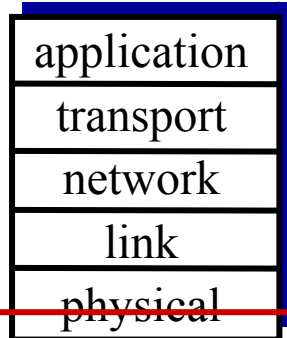
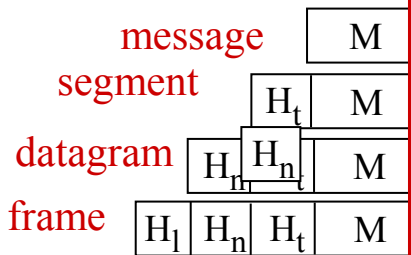
# Lähettäjä (source)



# Ammattilaisen näkökulma



# Vastaanottaja (destination)



# Kurssin oppimistavoitteita

- Käsitteet ja nimet paikoilleen!
  - Suomenkielinen alan keskeinen sanasto (*ja englannink.*)
- Internetin rakenteelliset, tekniset ja toiminnalliset periaatteet
  - Millaisia komponentteja? Niiden tehtävät? Miten datan siirto?
- TCP/IP-protokollapinon periaatteet
- Verkkosovellusten ja –protokollien (yhteis)toiminta
- TCP, IP ja Ethernet-protokollien toiminta
- Luotettavan kuljetuspalvelun periaatteet
- Tietoturvan uhkia ja suojautumiskeinoja
  - tätä on pariin aikaisempaan vuoteen verrattuna vähemmän.

Käyttäjän  
näkökulma

# Oppimistavoitematriisin pääteemat

- Tietokoneverkon rakenne ja TCP/IP-arkkitehtuuri
- Verkkosovellukset, sovellusprotokollat
- **Kuljetuskerros, luotettava tiedonsiirto prosessien välillä epäluotettavan verkon yli käyttäen TCP- ja UDP-protokollia**
- **Verkkokerros, pakettien siirto verkossa** lähdekoneelta kohdekoneelle, IP-osoitteet, **IP-protokollat**, reitittimen toiminta, reititysprotokollat
- Linkkikerros, ethernet-lähiverkko, datan siirto siirtolinkin yli koneelta toiselle
- Tietoliikenteen turvallisuus

# Asema opetuksessa

Aineopintojen 2. vuoden pakollinen kurssi

## Esitietoja

- Algoritmien lukutaito
- Bittitason esityksen tunteminen
- Järjestelmän hierarkkinen rakenne
- Kuinka sovellus saa käyttöjärjestelmältä (KJ) palvelua? = palvelupyyntö
- Kuinka laitteisto saa KJ:ltä palvelua? = keskeytysmekanismi

## Mitä kurssin jälkeen?

- Aineopintojen harjoitustyö: Tietoliikenne
- Verkkosovellusten toteuttaminen
- Web-sovellusohjelmointi
- Internet-protokollat
- Hajautetut järjestelmät
- Maisteriopinnoissa syventäviä erikoiskursseja eri teemoista

TiTo

# Syksyn 2017 kurssisuunnitelma

## Luentoviikkojen teemat

1. Tietokoneverkot ja Internet
2. Verkkosovelluksia ja sovellusprotokollia
3. Kuljetuskerros: TCP, UDP
4. Verkkokerros: IP
5. Linkkikerros, lähiverkot
6. Tietoturvaa (ehkä vähän)
7. Langatonta ja kertaus

## Aikataulu

- Luennot
  - ma 12-14 B123
  - to 12-14 B123
- Harjoitukset
  - ti 12-14 B222 (TK)
  - ti 16-18 C222 (IR)
  - to 10-12 C222 (IR)
  - to 14-16 B120 (TK)

# Kurssimateriaali

- Kurssikirja
  - Kurose J.F., Ross K.W., Computer Networking. A Top-Down Approach. (6th ed.) Addison –Wesley, 2012.
  - Kirjan omat www-sivut:  
[http://www.awl.com/kurose\\_ross/](http://www.awl.com/kurose_ross/)
  - (3-5 editiot käyvät, mutta lukunumeroinnissa ja sisällössä eroja)
  - Vastaavien kirjoittajia: Tanenbaum, Stallings ja Halsall
- Kurssin www-sivut
  - Luentokalvot - yleensä ilmestyvät ennen luentoa
  - Harjoitustehtävät - noin viikkoa ennen harjoituksia
  - Tiedotteita - tarvittaessa



# Tietoliikenteessä runsaasti lyhenteitä - Älä eksy lyhenneviidaksoon

LAN MAN ATM ISDN WAN SAP ISO TCP UDP ACK  
ITU-T PPP ARP CRC RFC FDDI P2P DoD SMTP  
TDM RSA IEEE URL IPv6 PSTN QAM MIME FTP  
IPS VC FDM X.25 FUNET IMAP POTS DSL WDM  
CDN NIC OAM GSM PCN DNS HDLC DoS SLIP  
MAC ALOHA CSMA LLC FEC DES HEC IETF  
ADSL HFC IANA NAP HTML NAK API GBN MTU  
HTTP CSMA/CD ITU POP3 WAP UTP POP ARPA  
RTT TLD ARQ SR MSS SYN TTL ICANN HDCP LS

# Eksymisen välttäminen: Tee töitä!

- Muodosta asiasta mielekäs kokonaisuus
  - Pysyttele hereillä luennoilla ja kirjaa lukiessasi!
  - Yritä koko ajan ymmärtää ja jäsentää
  - Tee omia muistiinpanoja
  - Kirjaa omat ajatukset ja kysymykset saman tien paperille
- Kysymykset eivät koskaan ole tyhmiä
  - Kysy luennoilla luennoijalta ja harjoituksissa ohjaajalta (vaikka luennon väliajalla tai harjoitusten loppuksi)
  - Kysy kavereilta
- Kerää kaveriporukka => ”opintopiiri”
  - Pohtikaa yhdessä harjoitustehtäviä
  - Selvittäkää kurssin ongelmakohtia



# Hyödynnä harjoitukset

- Tee tehtävät huolellisesti etukäteen
  - Yksin tai porukalla;
  - Mieti ja yritä ratkaista tehtäviä parina eri päivänä.
  - Vaikeat tehtävät jäävät 'hautumaan' ja aivot tekevät taustatyötä.
- Kysele ongelmakohtista
  - Kun ratkaisuja esitetään
  - Tilaisuuden jälkeen ohjaajalta tai jopa seuraavalla harjoituskerralla
- Yritä ratkaista ennen harjoituksia epäselväksi jääneet tehtävät itsenäisesti harjoitusten jälkeen
  - Jos ei onnistu, kysele lisää

# Sisältöä

- **Internet**
- **Verkon reunalla:**
  - asiakkaat ja palvelimet,
  - yhteydetön ja yhteydellinen palvelu
- **Pääsy Internetiin, fyysinen media**
- **Verkon sisällä**
  - Piirikytkentäinen, pakettikytkentäinen verkko
  - Datasähkeverkko, virtuaalipiiriverkko
- **Viivytykset ja katoamiset siirrossa**
  - Mitä viipeitä? Miksi dataa katoaa
- **Protokolla ja protokollapino**
- **Kerrosarkkitehtuuri**
- **Internet-protokollapino: kerrokset ja sano...**
- **Internetin rakenne**
- **Tietoturva: hyökkäyksiä**

## Oppimistavoitteet:

- Perusterminologia tutuksi
- Yleiskuva Internetistä
  - rakenne
  - toiminnallisuus
- Internetin protokollapino ja sen eri kerrosten tehtävät

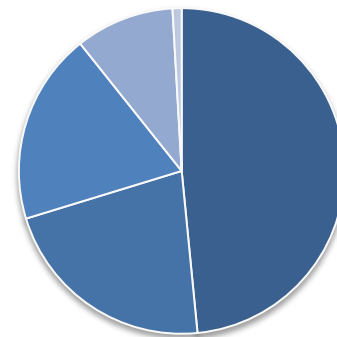


# Internet

- 1969: 4 konetta (ARPAnet)
- 1972: 30 konetta, sähköposti
- 1979: 200 konetta
- 1985: 2000 konetta (1983: TCP/IP )
- 1989: 160 000 konetta (1990: Web)
- 1995: 6 miljoonaa konetta,  
44,8 miljoonaa käyttäjää
- 2006: 450 miljoonaa konetta,  
1157 miljoonaa käyttäjää
- 2012: 2405 miljoonaa käyttäjää  
**34% maailman väestöstä (6/2012)**

<http://www.internetlvestats.com/>  
<http://www.internetworldstats.com/>

## Usage by Region 2013



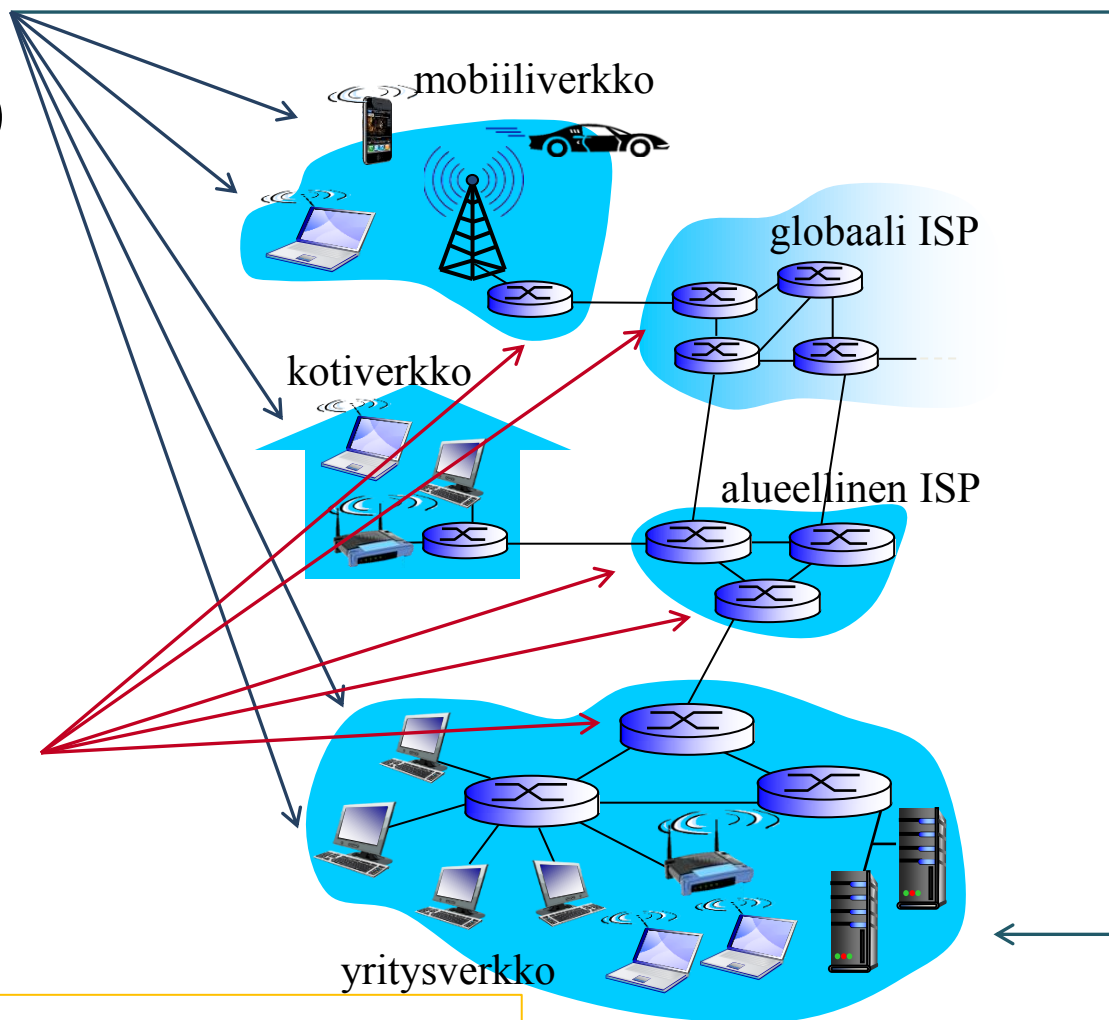
■ Asia      ■ Americas ■ Europe  
 ■ Africa      ■ Oceania

## World internet usage June 2015

<u>Region</u>	<u>Penetration</u>
Africa	27.0%
Asia	38.8%
Europe	73.5%
Middle East	49.0%
North America	87.9%
Latin America / Caribbean	53.9%
Oceania (incl. Australia)	72.9%
<i>World total</i>	<i>45.0%</i>

# Tietoliikenneverkon osat

- Verkon reunoilla
  - Isäntäkoneet (hosts)
  - Asiakkaat (clients)
  - Palvelimet (servers)
  - Palvelimet usein palvelinkeskuksissa (data center)
- Pääsy Internetiin
- Verkon syövereissä
  - Verkkoja yhdistävät reitittimet
  - Verkkojen verkko



Kaavakuva Internetistä vuodelta 1999:

<http://www.cheswick.com/ches/map/gallery/isp-ss.gif>

# Verkon rakenneosat



PC



palvelin



kannettava tietokone



älypuhelin



langaton linkki

— johdollinen yhteys

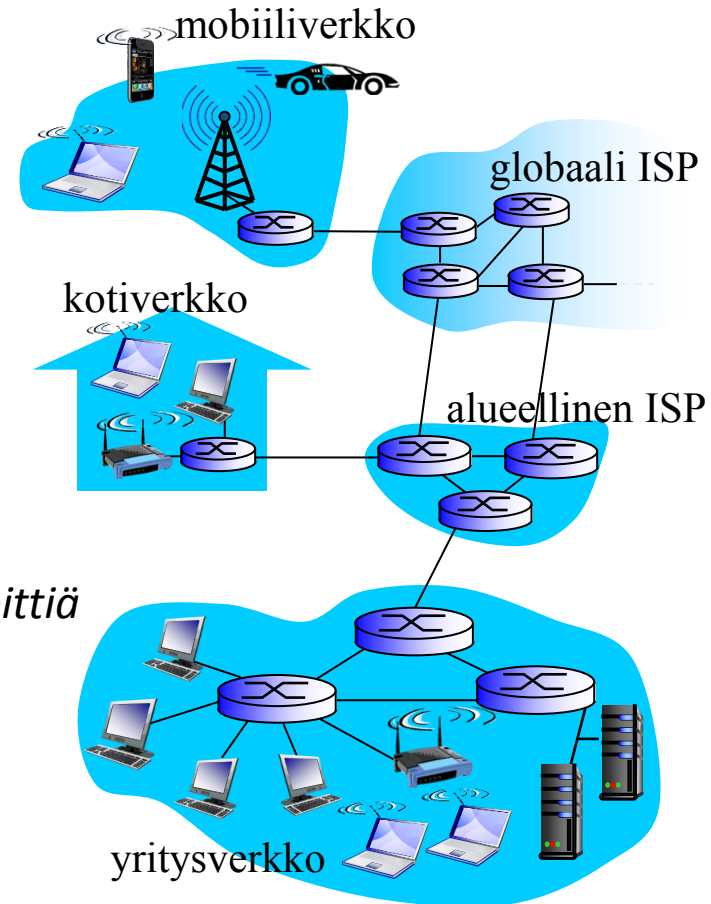


reititin



kytkin

- Miljoonia isäntäkoneita (hosts, end systems)
  - suorittavat hajautettuja sovelluksia (network apps)
- Tietoliikennelinkkejä (communication links)
  - optinen kuitu, kuparijohto, elektromagneettiset aallot (radio, infrapuna, satelliitti)
  - Siirtonopeus (transmission rate): *bittinä sekunnissa* (bps)
  - Kaistanleveys (bandwidth)
- Pakettikytkentäinen (Packet switches):
  - Siirtää paketteja (bittijono)
  - Reitittimiä (routers) ja kytkimiä (switches)



# Internet: eri näkökulmia

## Rakenne:

- verkkojen verkko (löyhä kytkentä)
  - Internet-palveluntarjoajien (Internet service provider, ISP) verkot yhdistetty
  - Julkinen Internet vs. rajattu intranet ja extranet
- Päästä-päähän suunnittelumalli
  - tila ja toiminnot reunoilla
- Protokollat - kommunikointisäännöt
- Standardeja
  - RFC – request for comments
  - IETF – Internet Engineering Task Force

## Palvelu:

- Infrastrukturi, joka tarjoaa palveluja sovelluksille:
  - Web, VoIP, sähköposti, some, verkkopelit, verkkokauppa, ...
- Tarjoaa ohjelmointirajapinnan sovelluksille (application programming interface, API)
  - koukkuja, joiden avulla sovellus voi lähettää ja vastaanottaa viestejä
- Tarjoaa viestintäpalvelua, hyvä analogia: posti
  - kirje postilaatikkoon



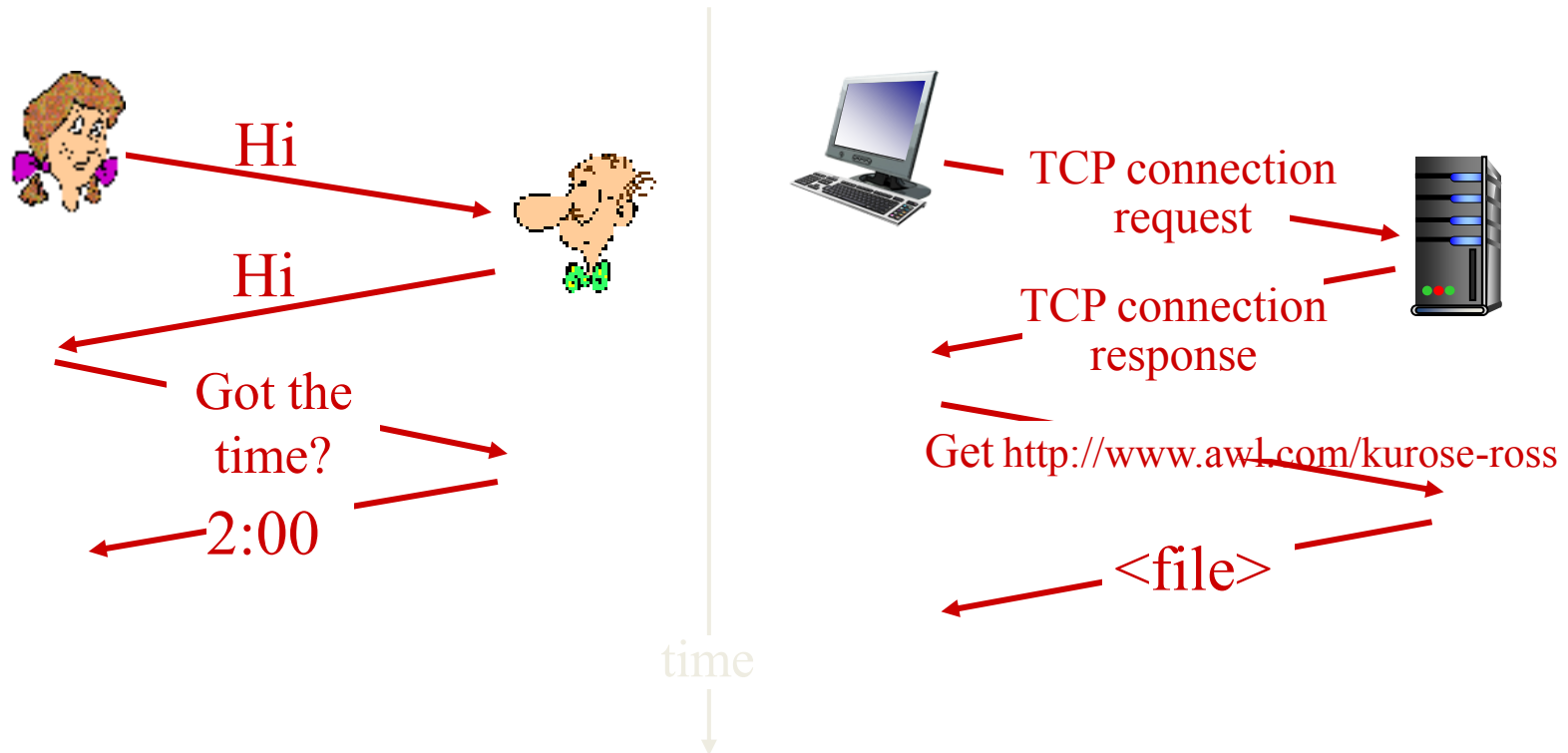
# Palvelu vs protokolla

- **Palvelu:** joukko toimintoja, jotka ovat käytettävissä
  - Internetin kuljetuspalvelu,
  - API = miten ohjelma pääsee käyttämään Internetin infrastruktuurin palveluja
- **Protokolla:** säännöt, jotka määräävät, miten sanomia vaihdetaan palvelun toteuttamiseksi
  - Sanomien muoto, sanomien järjestys, ..
  - Itse asiassa hajautettu algoritmi!
- Päästä-päähän-protokolla (end-to-end)
  - sovelluksen prosessilta toisen sovelluksen prosessille

# Protokolla = käyttäytymissäännöt

Ihmisten välinen prot.  
protokolla

tietokoneiden välinen



# Palvelumalleja

- Asiakas/palvelija-malli
  - pyyntö-vastaus-protokolla
  - Asiakas lähettää pyynnön, johon palvelija vastaa
  - Yksinkertainen ja usein riittävä (perinteinen malli)
  - Oletus, että palvelija on aina valmiina
  - www-selain / www-palvelin, postisovellus / postipalvelija, ....
- Vertaistoimijamalli (peer-to-peer, P2P)
  - Kukin isäntäkone voi toimia sekä asiakkaana että palvelijana
  - Ei oletusta laitteiden valmiudesta
  - Napster, Gnutella, KaZaA (FastTrack), EDonkey, eMule, BitTorrent, Mute, ...
  - Internet-puhelin: Skype

# Sanomien välitys

Viestinvälityspalvelu sovellukselta toiselle voi olla:

- **yhteydellinen** (connection-oriented) tai **yhteydetön** (connectionless)
  - Yhteydellinen: Yhteyden muodostus – yhteyden käyttö – yhteyden purku (~puhelu)
  - Yhteydetön: yhteyden käyttö (~posti)
- **luotettava** (reliable) tai **epäluotettava** (unreliable)
  - luotettava pyrkii estämään, havaitsemaan ja paikkaamaan virheet)
  - epäluotettava = 'hälläväliä'

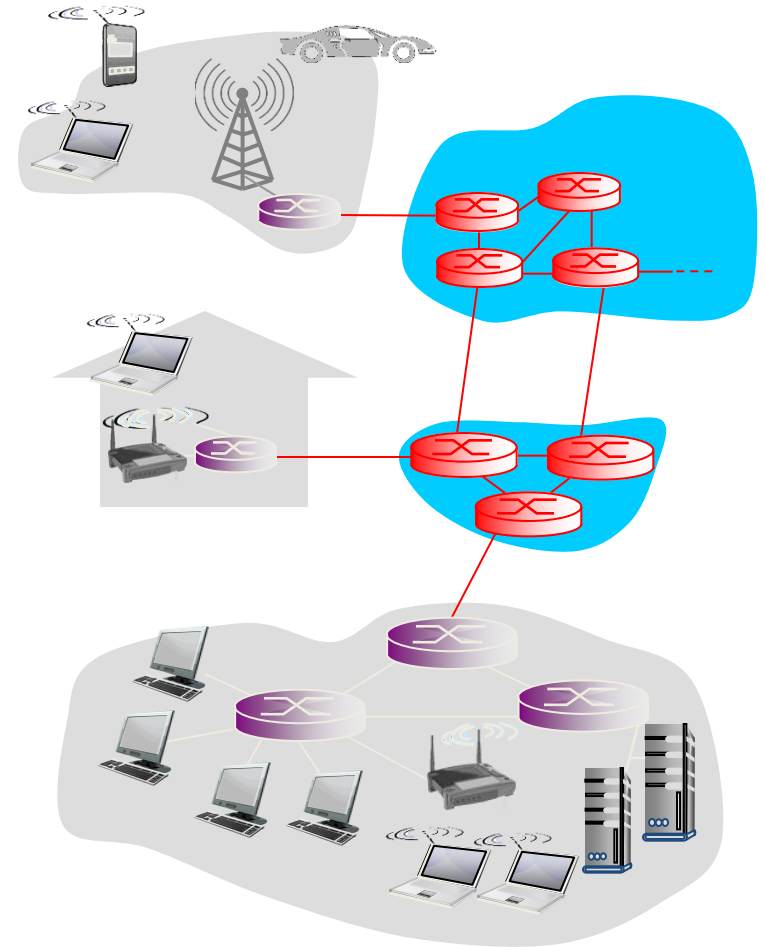
Internetissä:

- TCP-protokolla => yhteydellinen ja luotettava
- UDP-protokolla => yhteydetön ja epäluotettava

# VERKON SYÖVEREISSÄ

# Verkon ydin (*core*)

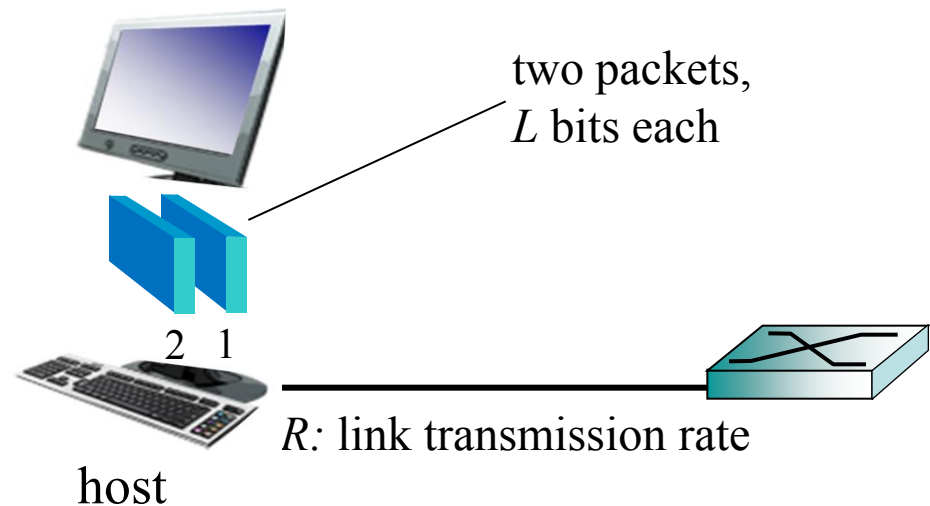
- Toisiinsa liitettyjen reittimien verkko (*mesh*)
- **Pakettikytkentäinen:** isäntäkoneet pilkkovat sovelluskerroksen sanomat paketeiksi (*packets*)
  - Reitittimet siirrävät paketteja linkki kerrallaan reittimeltä toiselle, näin muodostuu polku (*path*) lähettäjältä vastaanottajalle
  - Linkki siirtää aina yhden paketin kerrallaan täydellä teholla / kapasiteetilla



# Isäntäkone lähettää datapaketteja

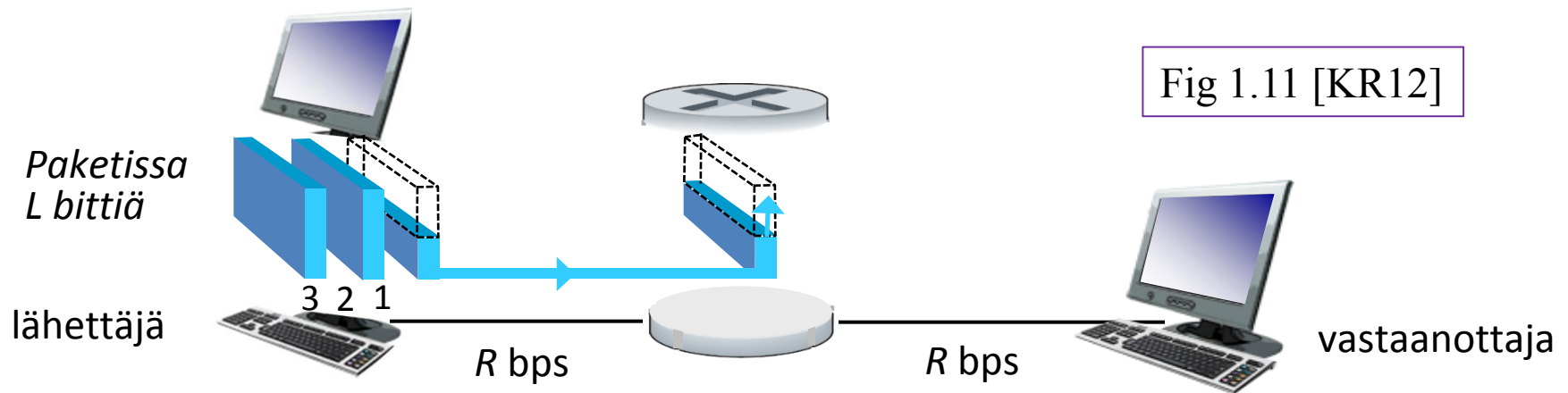
Isäntäkoneen prosessi:

- Vastanota sovelluksen viesti
- Pilko  $L$ -bitin mittaisiin osiin, *paketteihin (packet)*
- Lähetä paketti verkkoon, linkin *siirtonopeudella (transmission rate)  $R$* 
  - (*engl. Myös capacity, bandwidth*)



$$\text{Paketin lähetysviive} = \text{Aika, joka tarvitaan } L\text{-bitin paketin lähettämiseen} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

# Pakettikytkentä (packet-switching): etappivälitys (store-and-forward)



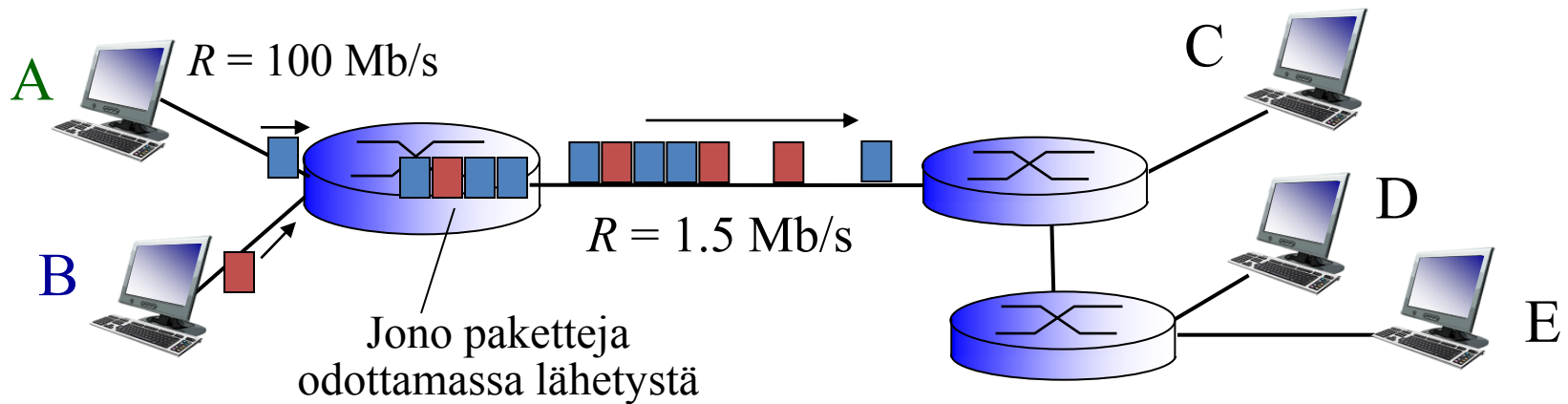
- **Etappivälitys:** paketti vastaanotetaan kokonaan ja vasta sitten lähetetään eteenpäin
- Kestää  $L/R$  sekuntia siirtää  $L$ -bitin paketti linkistä, jonka nopeus on  $R$  bps

*Esimerkki yhden linkin yli:*

- $L = 7.5$  Mbits
- $R = 1.5$  Mbps
- Siirtoaika linkin yli = 5 sec



# Pakettikytkentä: jonotus ja pakettien katoaminen



## Jonotus (queuing) ja katoaminen (loss):

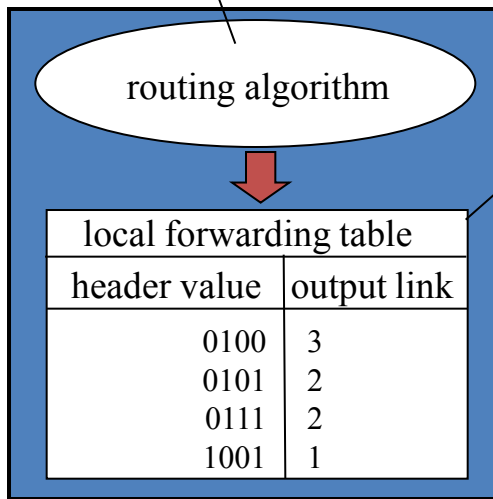
- ❖ Jos paketteja saapuu hetkellisesti nopeammin kuin niitä ehditään lähettää eli saapumistiheys (arrival rate)  $>$  lähetystiheys (transmission rate)
  - Paketit jonottavat lähetysvuoroa
  - Jos reitittimen jono (=puskuri) täyttyy, se kadottaa saapuvia paketteja

Ruuhka  
(congestion)

# Reititys ja lähetys

## Reititys (routing):

- Reititysalgoritmit laskevat parhaat reitit ja päivittävät taulukkoa

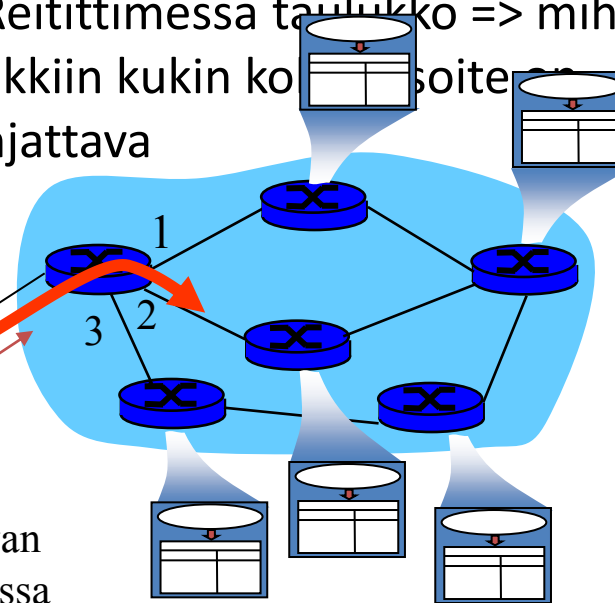


## edelleenlähetys (forwarding):

- Reititin siirtää vastaanottamansa paketin oikean linkin lähetysjonoon ja lähettää sen vuorollaan
- Reitittimessä taulukko => mihin linkkiin kukin ko... soite... ohjattava

Kohdeosoite saapuvan paketin otsaketiedoissa

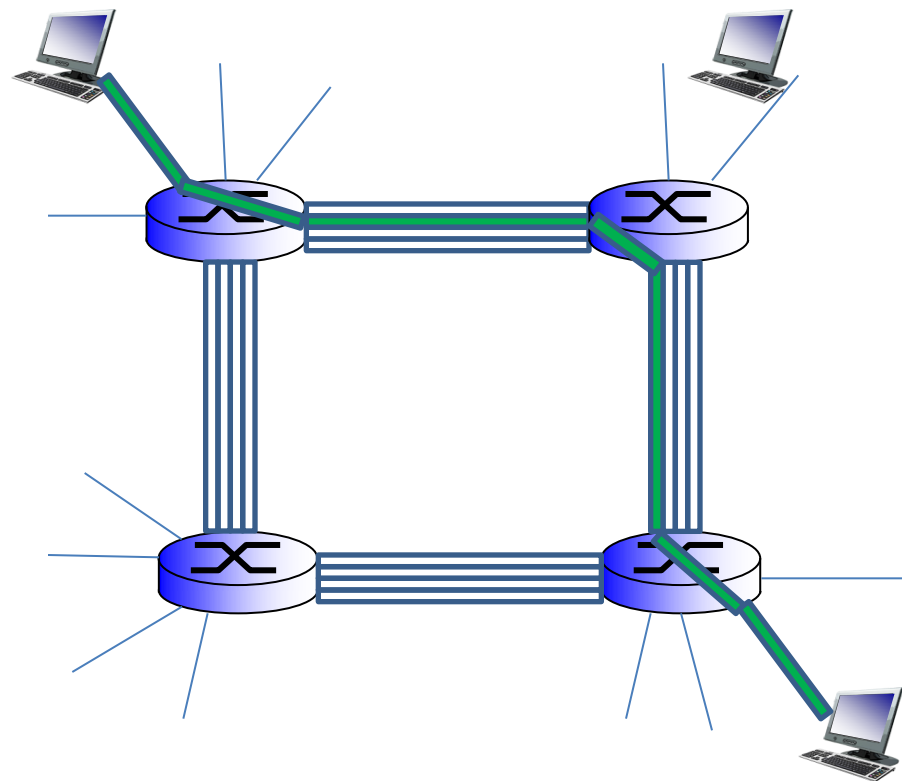
Timo Karvi



# Vaihtoehtoinen rakenne: piirikytkentä (circuit switching)

Varaa ensin kaikki polun linkit.  
Kaikki data näitä pitkin

- Kuvassa, kussakin linkissä tilaa neljälle piirille.
  - Yhteys varaa piirin 2 ylälinkistä ja piirin 1 oikeasta linkistä.
- Resurssit varattuna vaikka ei käytetä: niitä ei jaeta (*no sharing*)
- Takaa tasaisen nopeuden
- Kuten perinteinen puhelinverkko



vrt: vesipisteiden  
yhdistäminen letkuilla ja  
veden valutus

# Piirikytkentä: kanavointi (multiplexing)

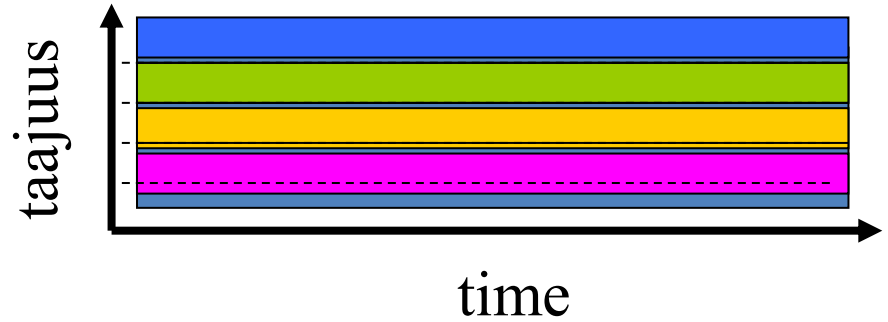
Linkille on limitetty usean yhteyden sanomia

Esimerkki:

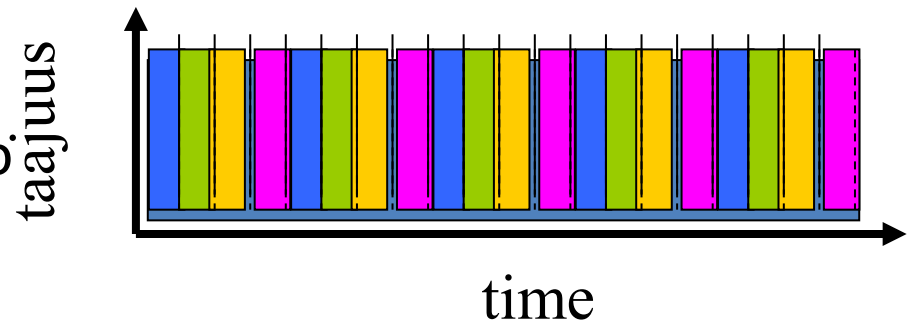
4 käyttäjää 

- **Taajuusjako** (*frequency-division multiplexing, FDM*): linkin kaistanleveys (taajuudet) jaettu käyttäjien kesken
- **Aikajako** (*time-division multiplexing, TDM*): jokainen saa käyttöönsä koko kaistanleveyden tietyn aikajakson ajaksi

FDM



TDM

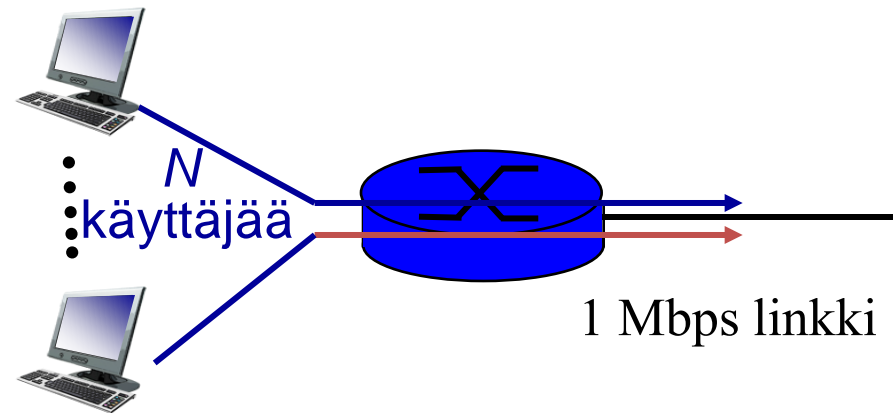


# Pakettikytkentä vs. piirikytkentä

*Pakettikytkentä sallii enemmän samanaikaisia käyttäjiä!*

## Esimerkki:

- 1 Mb/s linkki
- jokainen käyttäjä:
  - 100 kb/s kun “aktiivinen”
  - aktiivinen 10% ajasta
- *piirikytkentä:*
  - 10 käyttäjää
- *pakettikytkentä:*
  - Kun 35 käyttäjää, tod.näk. että > 10 aktiivista samaan aikaan on vähemmän kuin 0.0004 \*



**Q:** Mistä tulee tuo arvo 0.0004?

**A:** Binomijakauman kertymäfunktio

$$F(k; n, p) = \Pr(X \leq k) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$$

**Q:** Entä jos yli 35 käyttäjää?

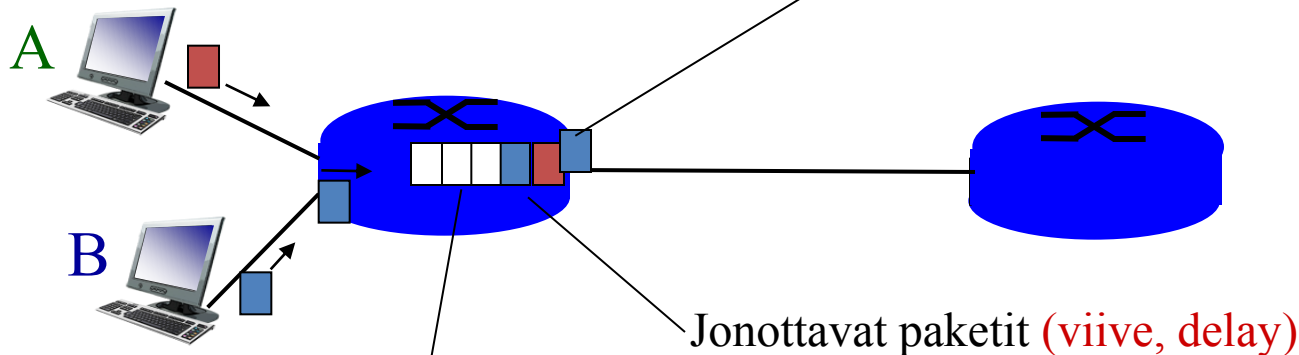
# VIIVETTÄ SIIRTOTIELLÄ

# Mistä pakettien viivästymisen ja katoaminen johtuu?

Paketit jonottavat (*queue*) reittimien puskureissa (*buffers*)

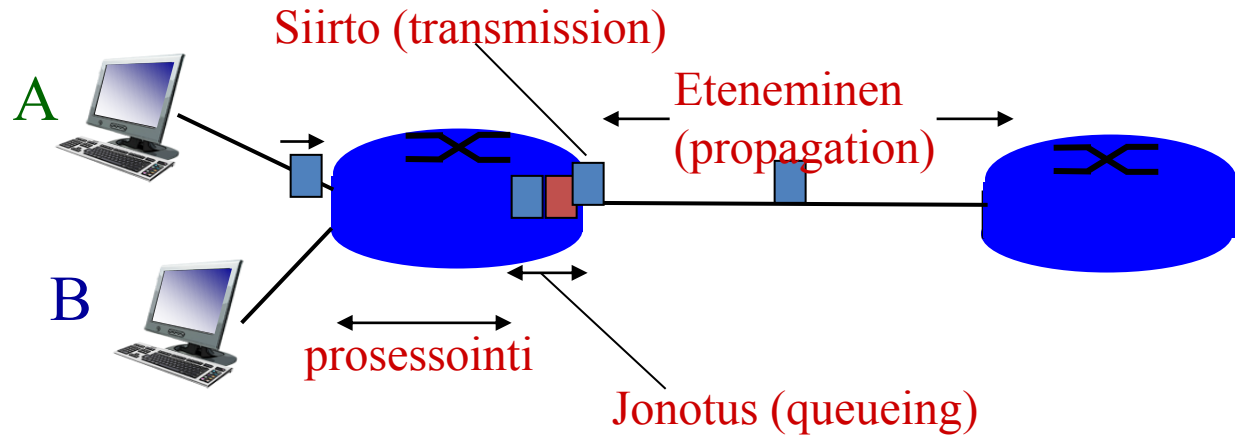
- Pakettien saapumistiheys (*arrival rate*) ylittää (tilapäisesti) ulosmenevän linkin kapasiteetin
- Paketit jonottavat ja odottavat lähetysvuoroaan

Lähetysvuorossa oleva paketti (*viive, delay*)



Vapaata puskuritilaa: jos puskuritila loppuu, saapuvat paketit **kadotetaan**, koska niille ei ole tilaa (**loss**)

# Viipeen neljä syytä



- **Prosessointiviive**

- Tarkista bittivirheet, hylkää virheelliset paketit
- Tutki paketin otsake, selvitä reititystaulusta, mille linkille menossa

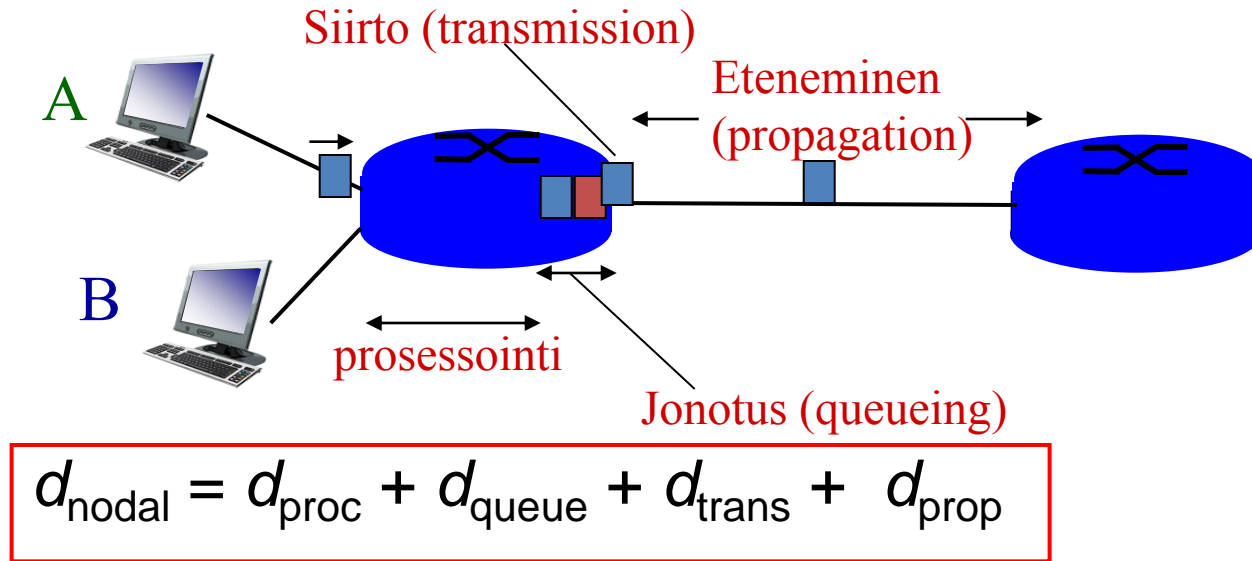
- **Jonotusviive (queuing delay)**

- **Siirtoviive + etenemisviive**

- Siirtoviive = paketin lähettämiseen (linkille siirtämiseen) kuluva aika
- Etenemisviive = bittien etenemiseen linkillä kuluva aika



# Viipeen neljä syytä



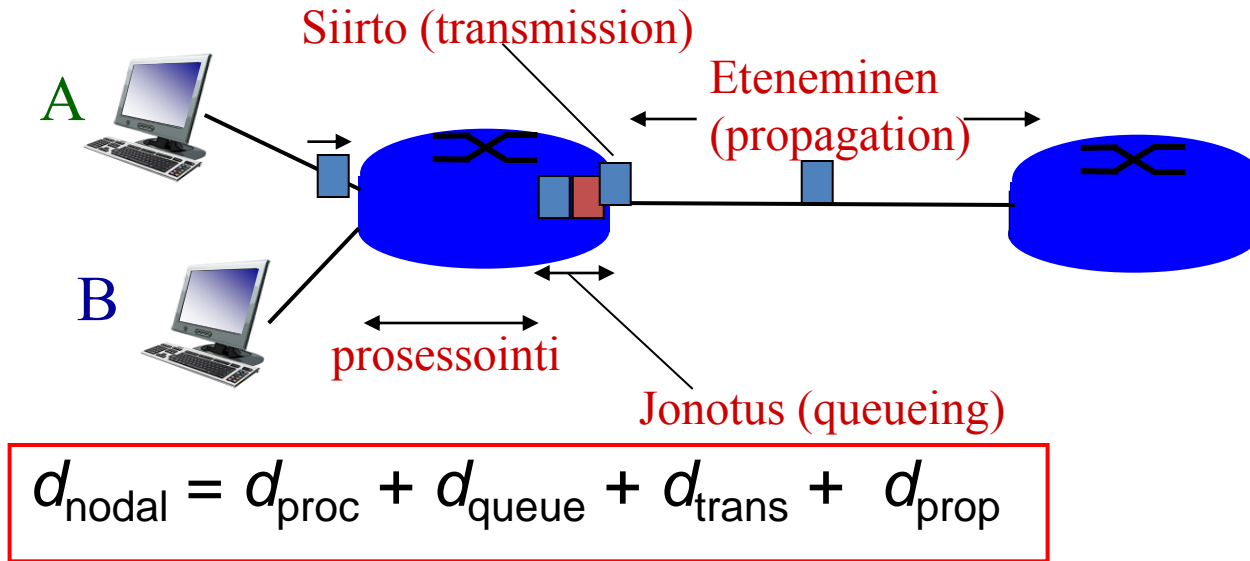
## $d_{\text{proc}}$ : Prosessointiviive

- Tarkista bittivirheet, hylkää virheelliset paketit
- Tutki paketin otsake, selvitä reititystaulusta, mille linkille menossa
- Tyypillisesti < msec

## $d_{\text{queue}}$ : Jonotusviive

- Aika, jonka odottaa ulosmenopuskurissa lähetysvuoroa
- Kesto riippuu ruuhkautumisasteesta (congestion level)

# Viipeen neljä syytä



$d_{\text{trans}}$ : siirtoviive

- L – paketin koko bitteinä
- R – linkin nopeus (bps)
- $d_{\text{trans}} = L/R$

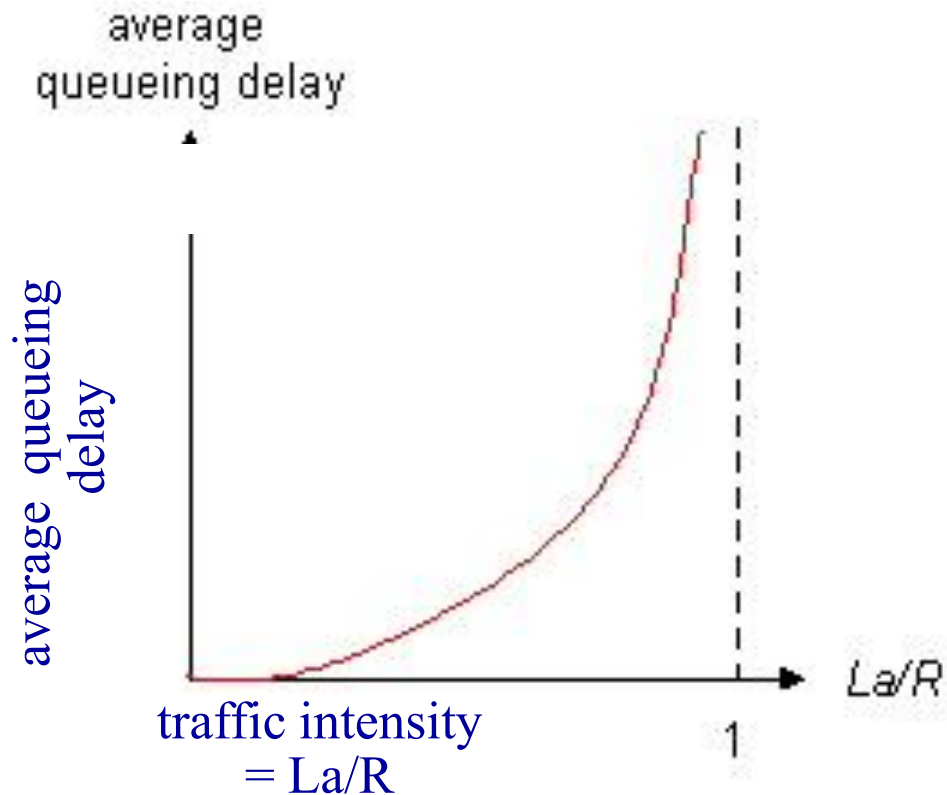
$d_{\text{prop}}$ : etenemisviive

- d – fyysisen linkin pituus
- s: etenemisnopeus ( $\sim 2 \times 10^8$  m/sec)
- $d_{\text{prop}} = d/s$

$d_{\text{trans}}$  ja  $d_{\text{prop}}$  eri suuruusluokkaa!

# Jonotusviive

- $R$ : linkin nopeus (bps)
- $L$ : paketin pituus (bits)
- $a$ : keskim. pakettien saapumistiheys (*arrival rate*)



- ❖  $La/R \sim 0$ : keskim. pieni jonotusviive
- ❖  $La/R \rightarrow 1$ : keskim. suuri jonotusviive
- ❖  $La/R > 1$ : saapuu enemmän tehtäviä kuin ehtii palvella, ääretön jonotusviive!



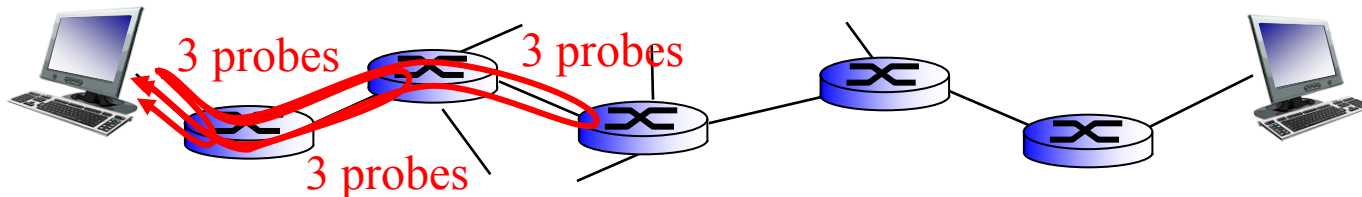
$La/R \sim 0$



$La/R \rightarrow 1$

# Internetin viipeitä ja reittejä


- Miltä Internetin viipeet ja reitit näyttävät?
- `traceroute` : mittaa viipeen lähettäjältä polulla oleville reitittimille. provides delay measurement from source to router along end-end Internet path towards destination. Kullekin reitittimelle  $i$ :
  - Lähettää kolme pakettia reitittimelle  $i$ , joka on vastaanottajalle johtavan polun varrella
  - reititin  $i$  palauttaa paketit lähettäjälle (lähettää vastausviestin)
  - Lähettäjä laskee lähetyksen ja vastaanottamisen välisen aikaeron.



# Traceroute esimerkki


**traceroute:** gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from  
gaia.cs.umass.edu to cs-gw.cs.umass.edu




1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms  
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms  
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms  
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms  
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms  
6 abilene-vbns.abilene.ucaid.edu (198.32.11.9) 22 ms 18 ms 22 ms  
7 nycm-wash.abilene.ucaid.edu (198.32.8.46) 22 ms 22 ms 22 ms  
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms  
9 de2-1.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms  
10 de.fr1.fr.geant.net (62.40.96.50) 113 ms 121 ms 114 ms  
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms  
12 nio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms  
13 nice.cssi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms  
14 r3t2-nice.cssi.renater.fr (195.220.98.110) 126 ms 126 ms 124 ms  
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54) 135 ms 128 ms 133 ms  
16 194.214.211.25 (194.214.211.25) 126 ms 128 ms 126 ms  
17 \* \* \*  
18 \* \* \*  
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms

trans-oceanic link

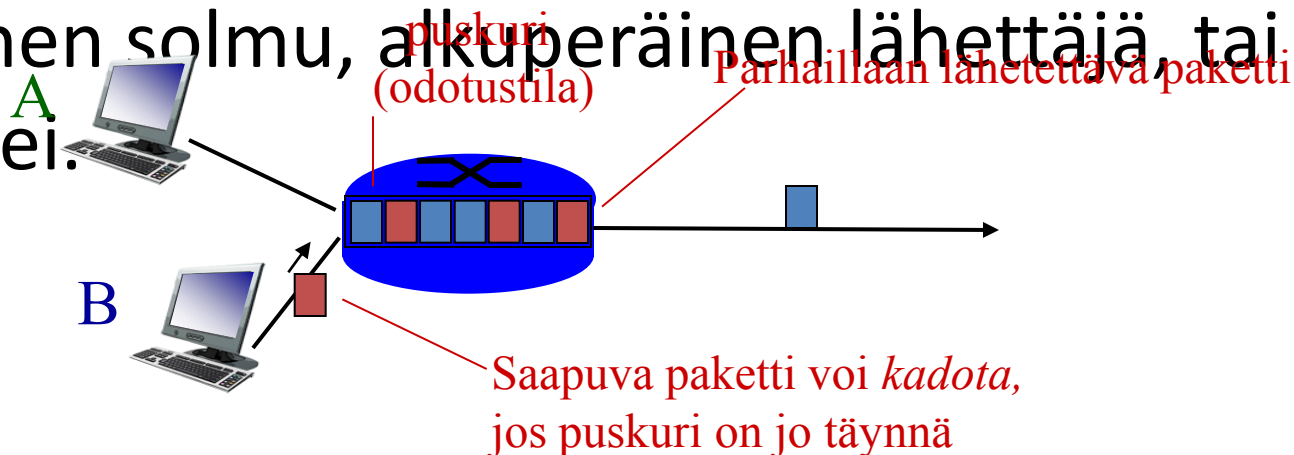


\* means no response (probe lost, router not replying)



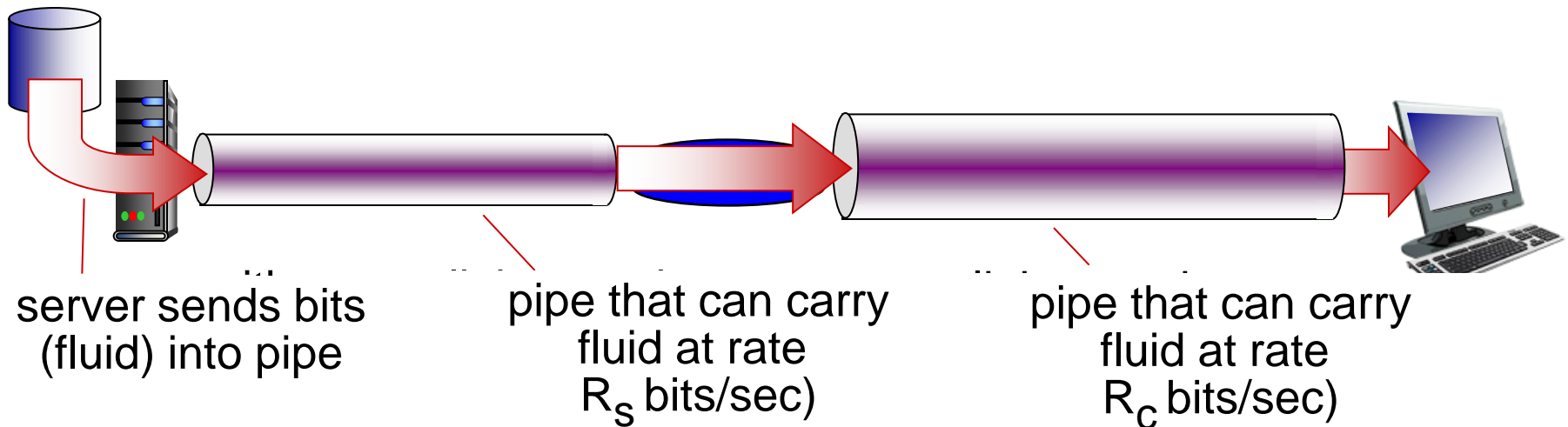
# Pakettien katoaminen

- Edeltävän linkin jonon pituus (siis puskurin koko) on äärellinen
- Täyteen jonoon saapuvat paketit pudotetaan (eli kadotetaan), koska niille ei ole tilaa
- Kadonneen paketin voi lähettää uudelleen edellinen solmu, alkuperäinen lähettäjä, tai sitten ei.



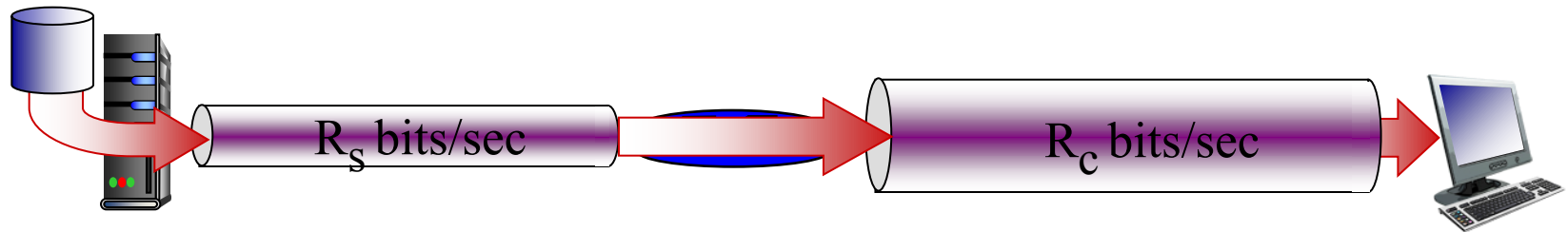
# Throughput

- **throughput**: rate (bits/time unit) at which bits transferred between sender/receiver
  - **instantaneous**: rate at given point in time
  - **average**: rate over longer period of time

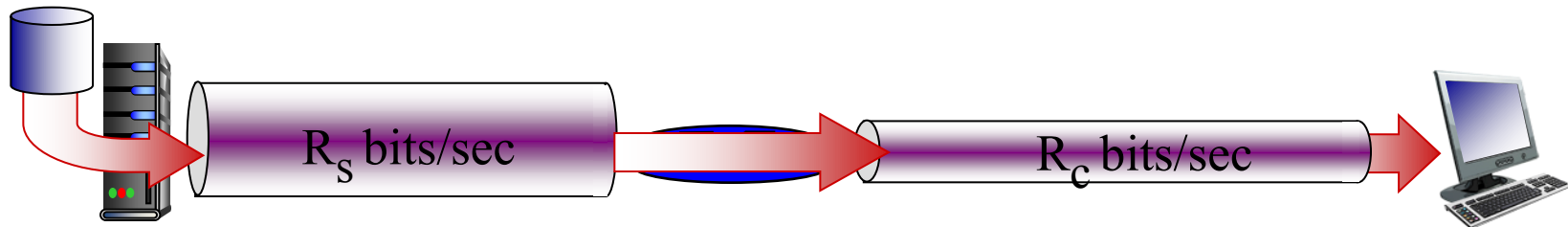


# Läpäisy (Throughput)

- $R_s < R_c$  Mikä on keskimääräinen läpäisy päästä-päähän?



- ❖  $R_s > R_c$  Entä nyt?



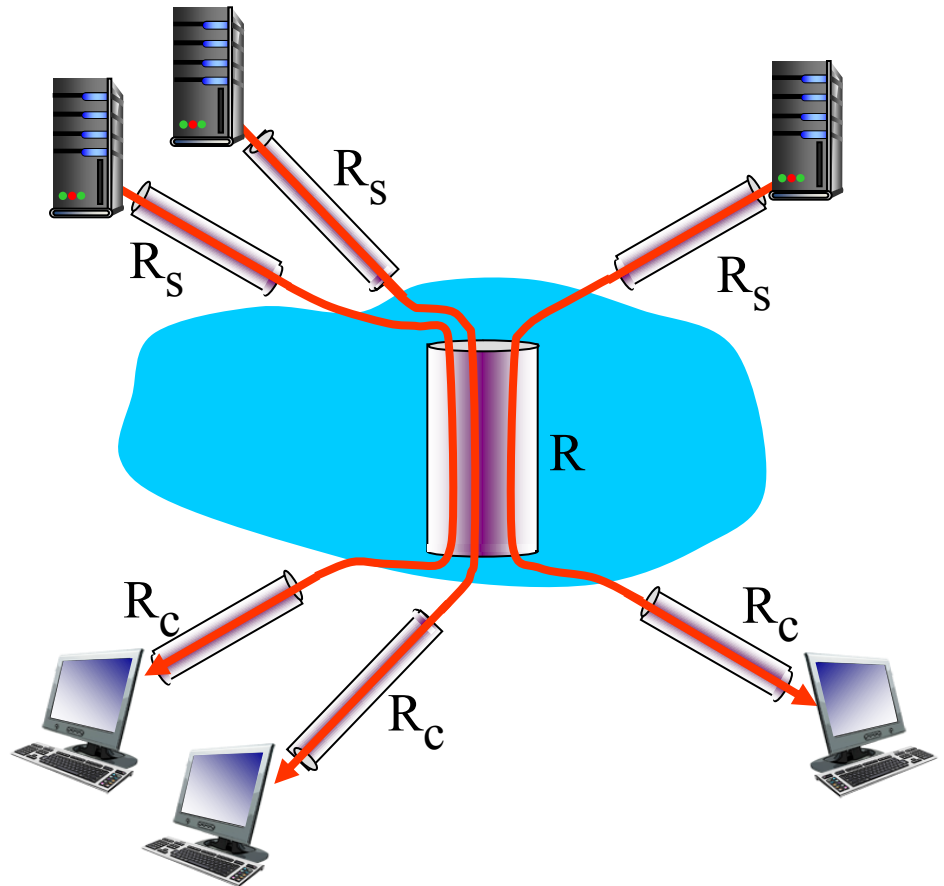
## *Pullonkaula (bottleneck)*

Linkki, joka rajoittaa läpäisyä päästä-päähän yhteydelle



# Throughput: Internet scenario

- Koko päästä-päähän yhteyden teho:  
 $\min(R_c, R_s, R/10)$
- Käytännössä: joko  $R_c$  tai  $R_s$  on usein pullonkaula



10 yhteyttä jakavat (reilusti) yhden ja saman linkin  $R$  bits/sec