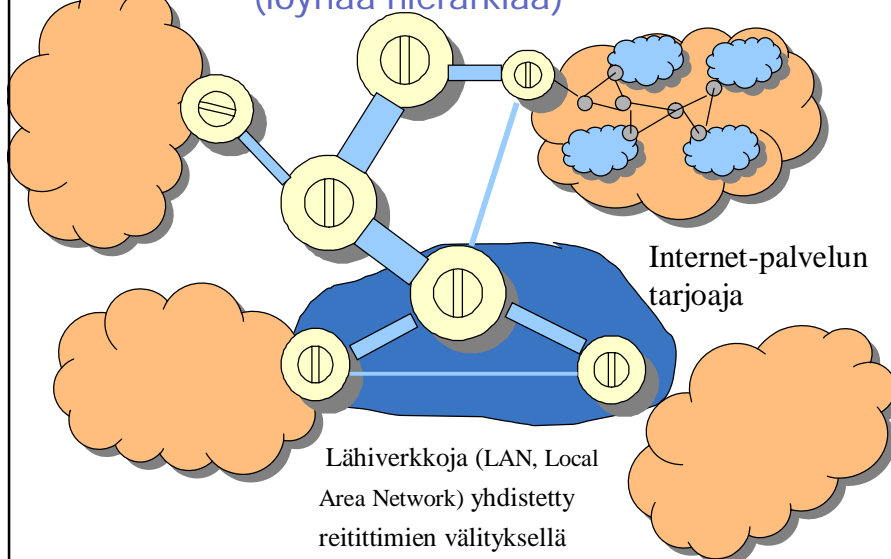


# Tietoliikenteen perusteet

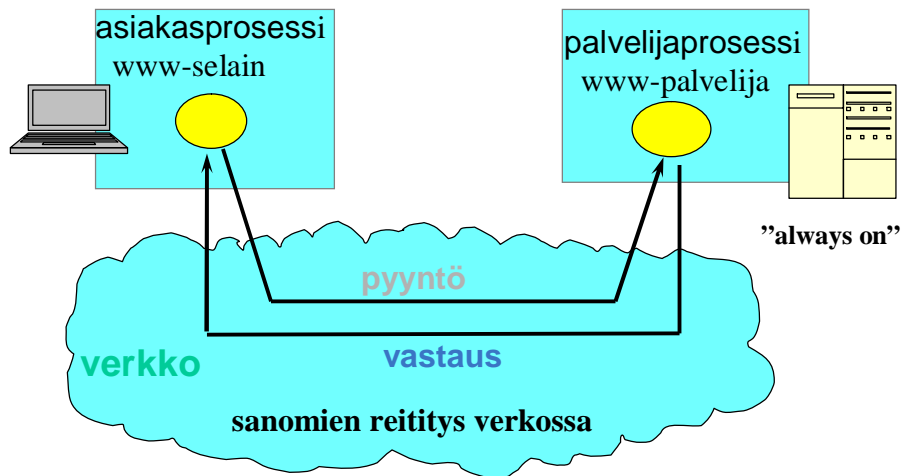
Vähän kertausta

## Internet = verkkojen verkko (löyhää hierarkiaa)





## Asiakas-palvelija-malli /vertaistoimijamalli



Tietoliikenteen perusteet /2009/ Liisa Marttinen

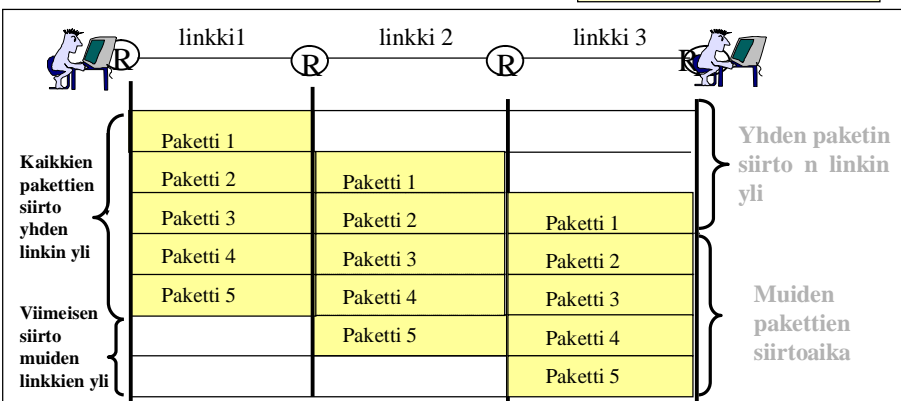
3



## Pakettivälitys siirto-aika

Olkoon siirtoaika a:

- a)  $ka+(n-1)a = (k+n-1)a$
- b)  $na+(k-1)a = (n+k-1)a$



Sanoman siirtoaika, kun sanomassa on k pakettia ja linkejä on n kappaletta

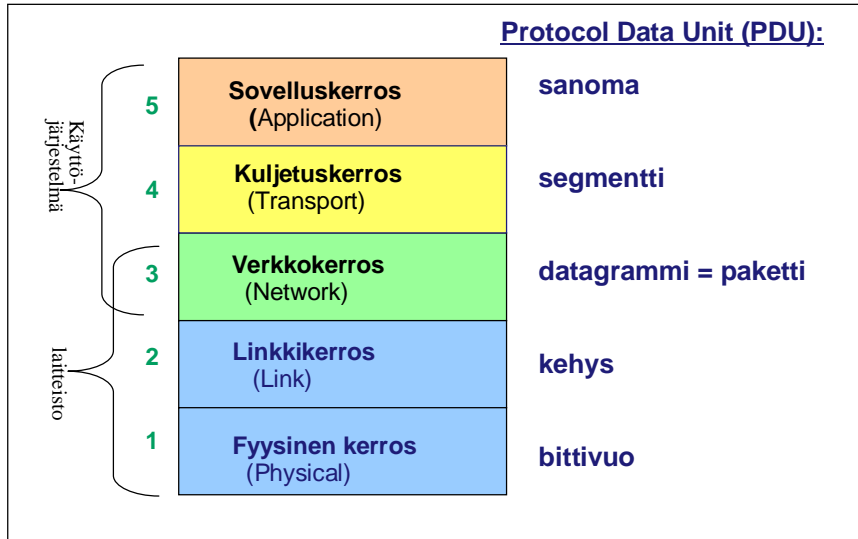
- a)  $k:n$  paketin siirto 1. linkin yli + viimeisen paketin siirto  $n-1$  linkin yli.
- b) 1. paketin siirto  $n:n$  linkin yli + muiden  $k-1$  paketin siirto yhden linkin li

Animaatio: [http://wps.aw.com/aw\\_kurose\\_network\\_4/63/16303/4173750.cw/index.html](http://wps.aw.com/aw_kurose_network_4/63/16303/4173750.cw/index.html)

Tietoliikenteen perusteet /2009/ Liisa Marttinen

4

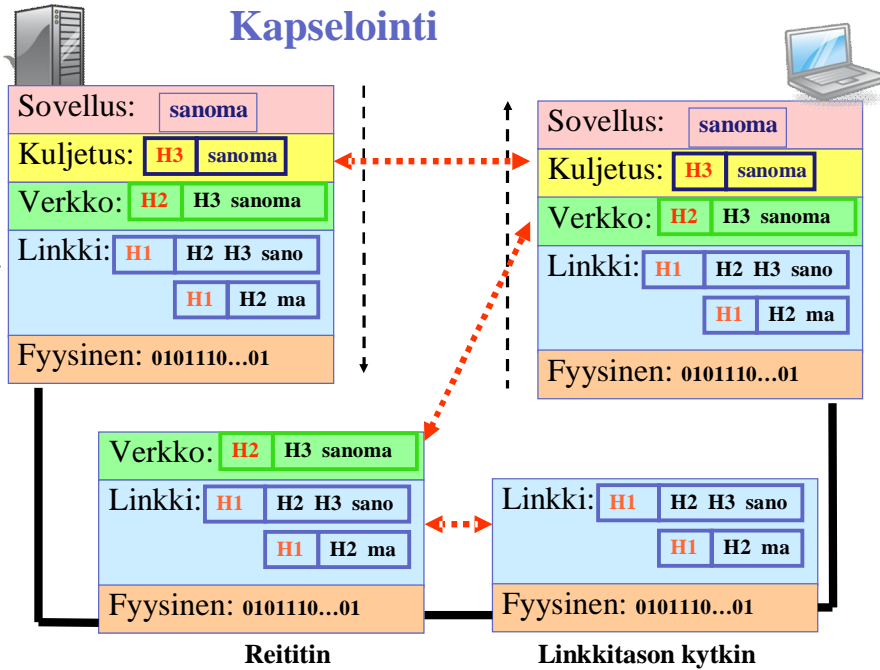
# Internet-protokollapino



Tietoliikenteen perusteet /2009/ Liisa Marttinen

5

# Kapselointi



Tietoliikenteen perusteet /2009/ Liisa Marttinen

6

# HTTP (HyperText Transfer Protocol)

```
GET /somedir/page.html
HTTP/1.1
Host: www.someschool.edu
User-Agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

PC, jossa on Explorer-selain



## WWW:N sovellusprotokolla

Tekstimuotoiset sanomat

pyyntö – vastaus

### Asiakas

Selain: **FireFox, Internet Explorer, Opera, Apple Safari, ...**

pyytää, noutaa ja näyttää objektit

### Palvelija

etsii objektin (tiedoston) koneen hakemistosta

ja lähettää sen vastauksena asiakkaalle

### Tilaton protokolla

Palvelija ei muista mitään edellisistä pyynnöistä => evästeet (cookies)

Palvelin, jossa on Apache-ww-palvelija



HTTP Request

HTTP Response

HTTP Response

HTTP Request

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 22 Feb 2007 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 29 Jan 2007 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
data data data data ...
```

Linux-kone, jossa on Firefox-selain



origin servers  
public Internet  
www.herkkutalo.com

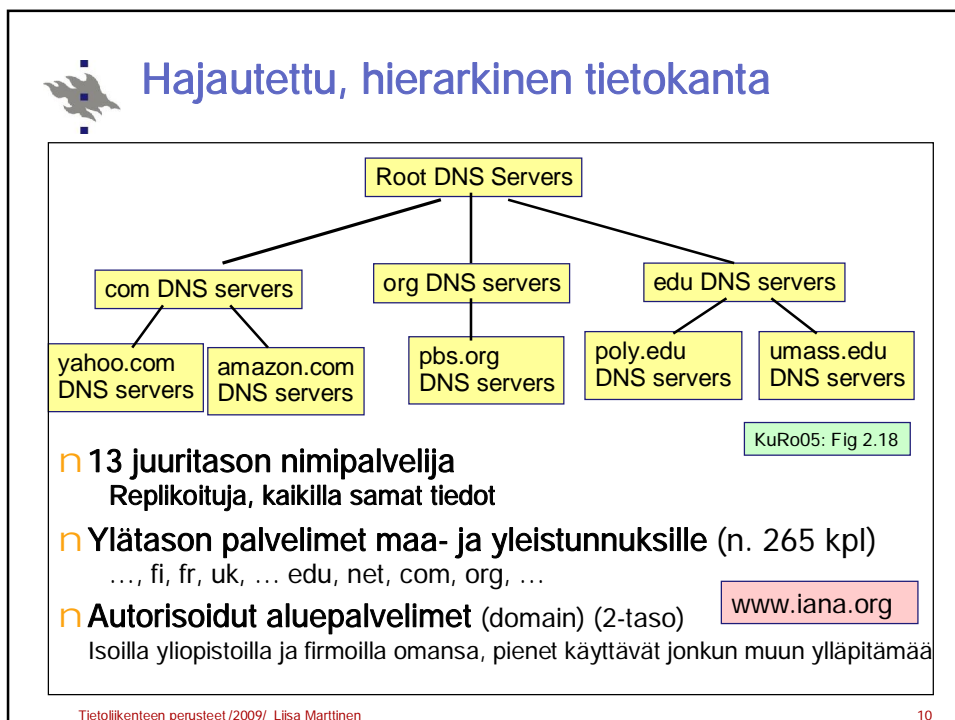
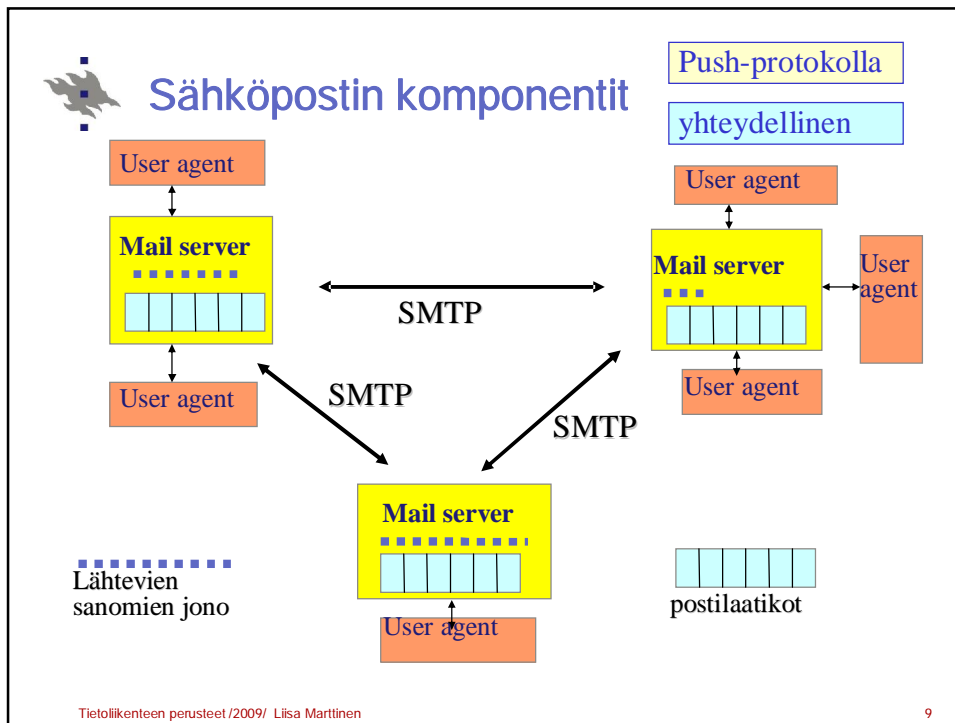
```
HTTP/1.1 304 Not Modified
Date: Thu, 14 Jul 2007 15:39:29
```

institutional network  
1.5 Mbps access link

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.herkkutalo.com
If-modified-since: Wed, 4 Jul 2007 09:23:24
```

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.herkkutalo.com
```

10 Mbps LAN  
institutional cache



# Skaalautuvuus

KuRo08: Fig. 2.24

## Asiakas-palvelinmalli:

Palvelimen siirrettävä  $n \cdot F$  bittiä => siirtoaika =  $nF/u_s$ .

Hitain asiakas  $d_{\min}$  saa tiedoston ajassa  $F/d_{\min}$

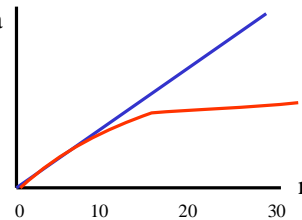
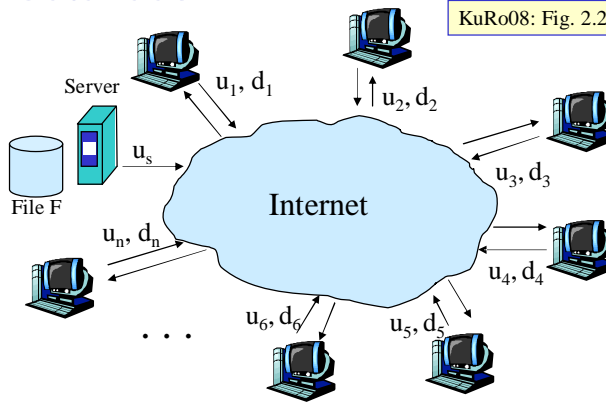
**Siirtoaika =**  
 $\max(nF/u_s, F/d_{\min})$

Kun  $n$  kasvaa, palvelimen kuorma kasvaa ja siirtoaika kasvaa.

**Vertaistojamalli** (alussa tiedosto on palvelimella)

**Siirtoaika =**  $\max(F/u_s, F/d_{\min}, nF/(u_s + V u_i))$

Summamerkki

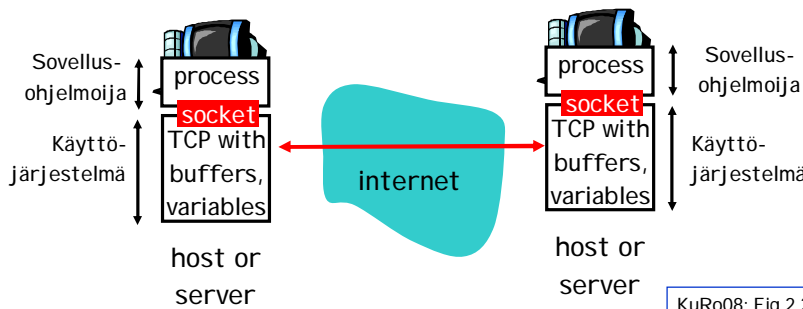


# Pistoke (socket)

**n** Kuljetuspalvelun ja sitä käyttävän sovelluksen rajapinta isäntäkoneessa

Sovelluksen tietoliikenne = KJ:n palvelupyyntöjä  
 Pistoke on "palveluluukku"

**n** Alunperin Berkeley UNIXin (BSD) mukana



KuRo08: Fig 2.26

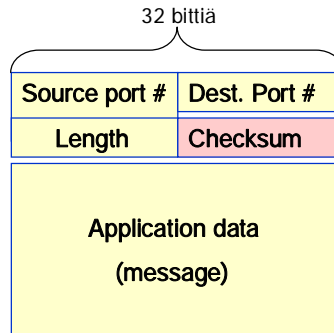
# UDP: Tarkistussumma

## Lähetys

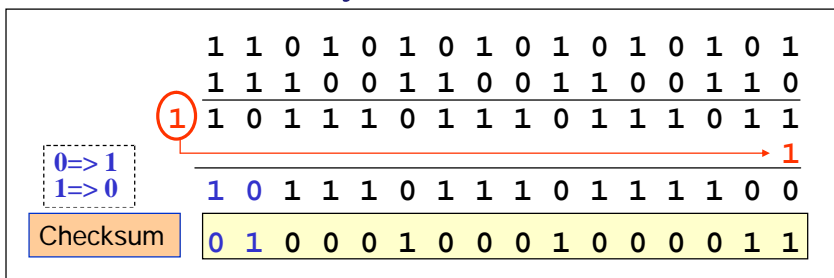
- Summaa 16 bitin kokonaisuudet (otsake + pseudo-otsake mukana), ylivuotobitit lasketaan mukaan, talleta **yhden komplementtina**

## Vastaanotto

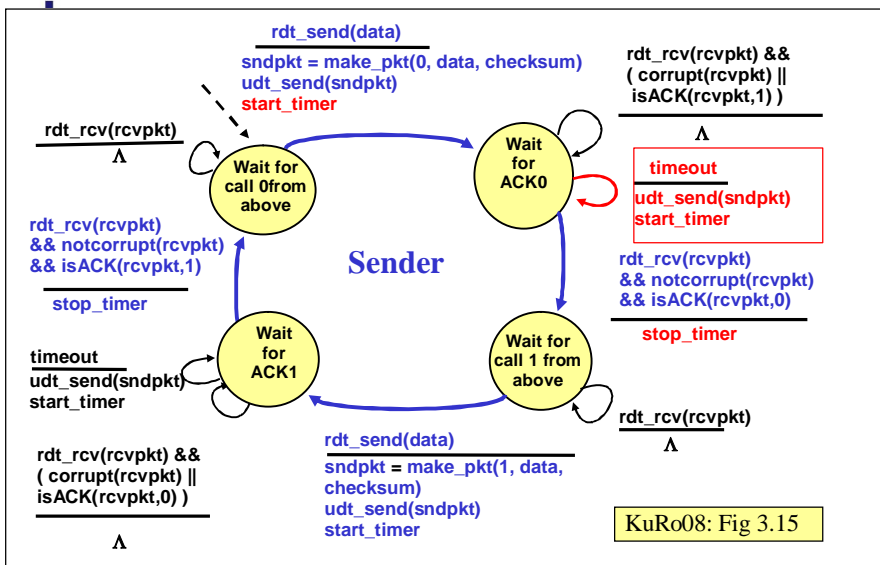
- Summaa 16 b kokonaisuudet (myös tarkistussumma).
- Jos tuloksena on 16 ykköstä, niin OK!



UDP-otsake

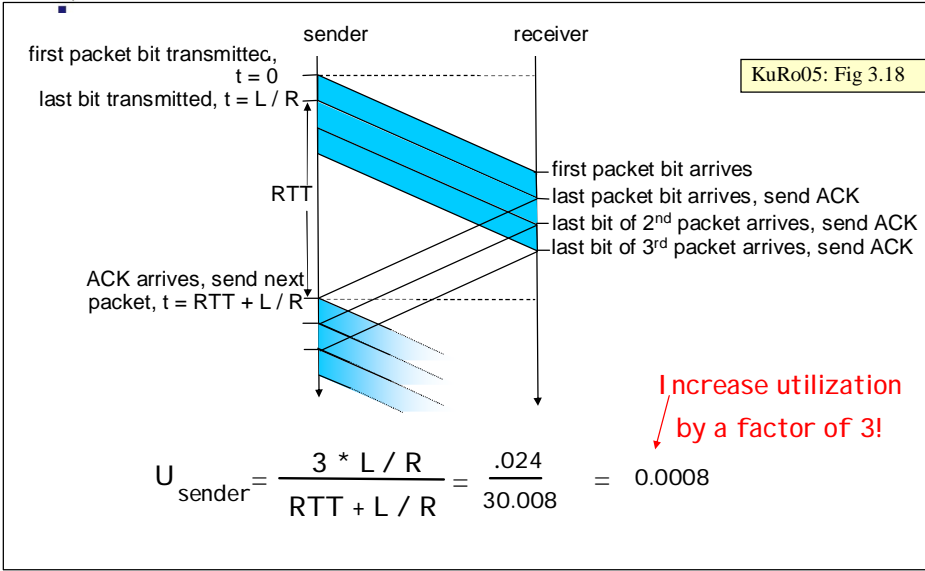


# rdt3.0

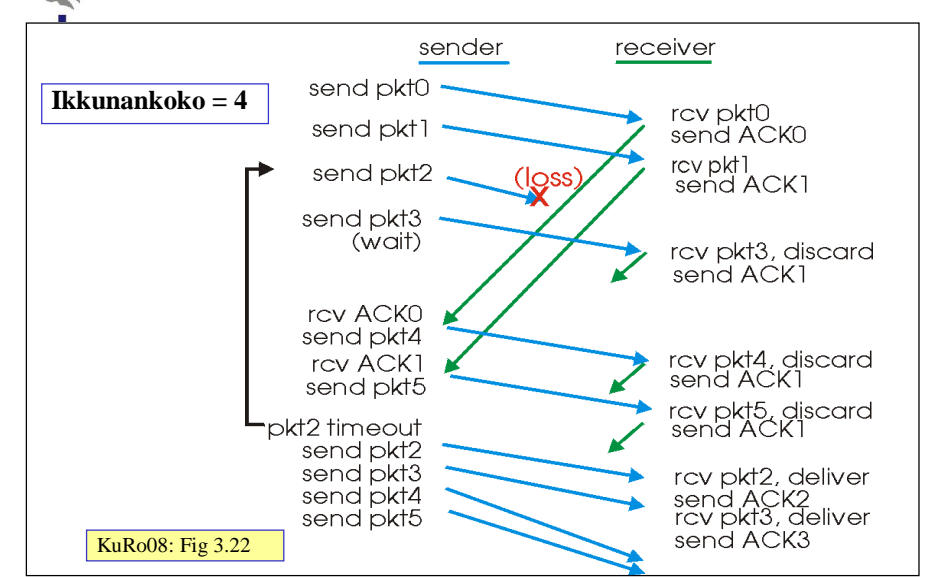




## Liukuhintoitus: käyttöasteen kasvattaminen

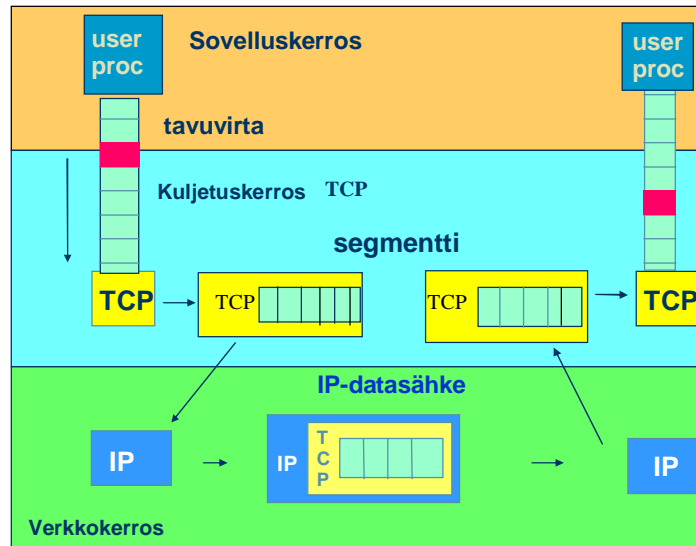


## Go-Back-N: Esimerkki





## TCP: prosessilta prosessille -tavuvirta



Tietoliikenteen perusteet /2009/ Liisa Marttinen

17

## TCP:

### Luotettava, järjestyksen säilyttävä tavuvirta

- n Ei sanomaraajoja
- n Tavunumerointi
- n Checksum-tarkistus
- n Puskurointi uudelleenlähetyksiä varten
- n Kumulatiiviset kuittaukset

### Yhteydellinen

- n Kolminkertainen käsittely, yhteyden purku

### Vuonvalvonta, ruuhkanhallinta (-valvonta)

- n Lähettäjä ei saa tukahduttaa vastaanottajaa eikä reitittämiä
- n Vuonvalvonta: Receive window
- n Ruuhkanhallinta

Tietoliikenteen perusteet /2009/ Liisa Marttinen

18



## Yhteyden muodostus

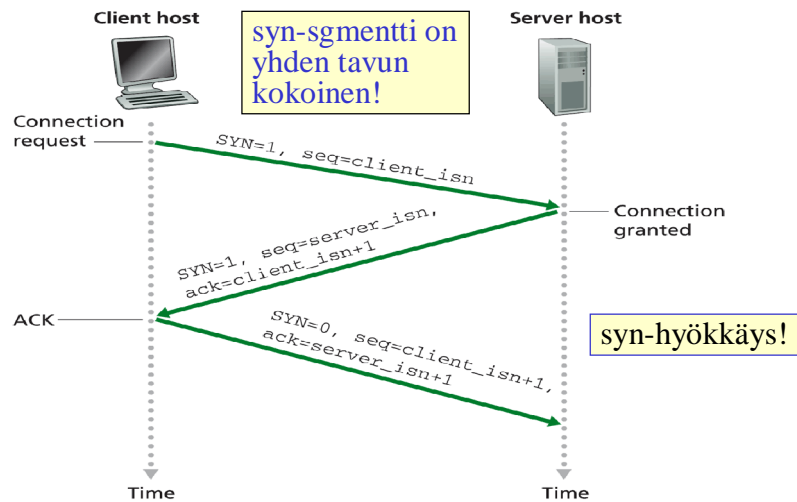


Figure 3.39 ♦ TCP three-way handshake: segment exchange

Tietoliikenteen perusteet /2009/ Liisa Marttinen

19



## TCP Reno: Hidas aloitus (slow start) ja ruuhkanvälttely (congestion avoidance)

### Aluksi ruuhkaikkuna = yksi segmentti

- Alussa hidas siirtonopeus =  $MSS/RTT$

### Kukin kuittaus kasvattaa yhdellä ruuhkaikkunan kokoa

- Eksponentiaalinen kasvu **hidas aloitus**
- Ikkuna kaksinkertaistuu yhden  $RTT$ :n aikana

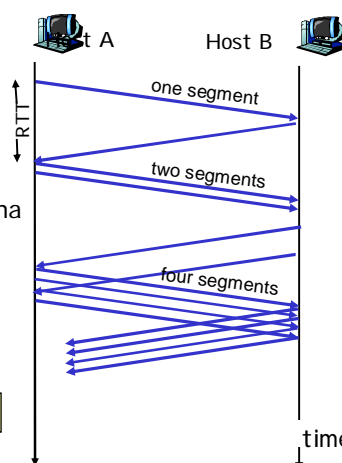
### Jos uudelleenlähetys, puolita ruuhkaikkunan koko

- Multiplicative decrease

### Sen jälkeen kasvata ikkunaa yksi segmentti/RTT

- Lineaarinen kasvu (Additive increase) **ruuhkanvälttely**
- Ruuhkan välttely (congestion avoidance)

### Siirtonopeus = $CongWin / RTT$ tavua/sek

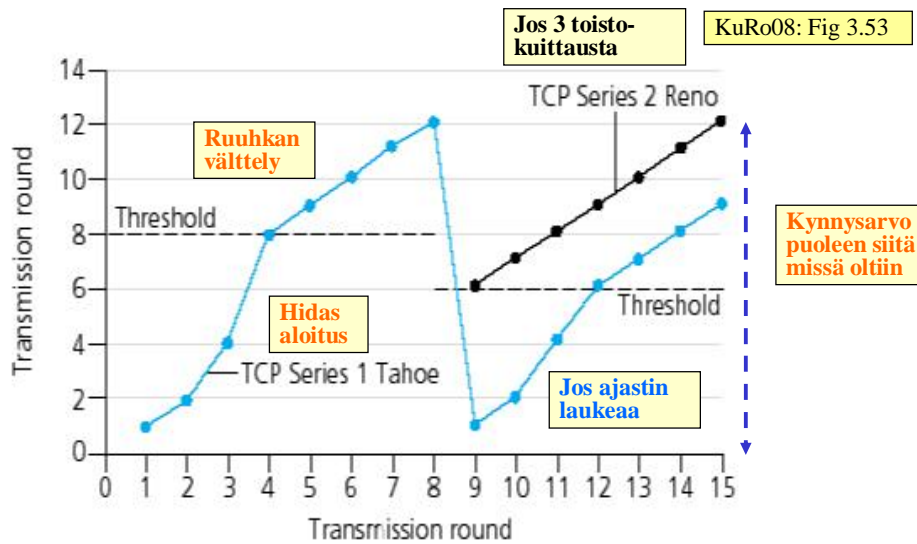


Tietoliikenteen perusteet /2009/ Liisa Marttinen

20



## TCP Tahoe vs. TCP Reno



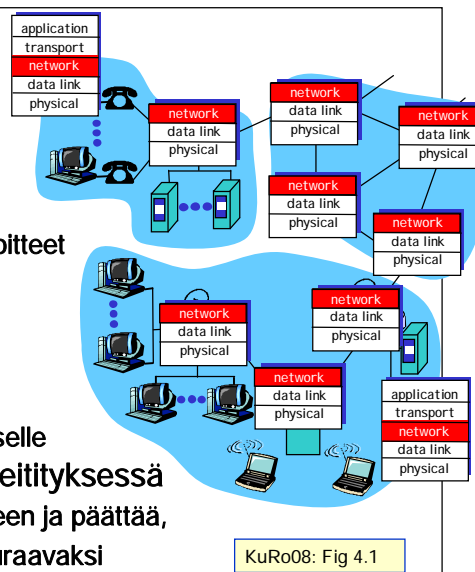
Tietoliikenteen perusteet /2009/ Liisa Marttinen

21



## Verkkokerros

- n Toimittaa kuljetuskerroksen segmentit vastaanottajalle
- n Lähettäjä
  - n luo segmenteistä verkkokerroksen IP-paketteja
  - n Lisää otsaketietoja: mm. IP-osoitteet
- n Reitittäminen
  - n Isäntä - reititin ... reititin - isäntä
- n Vastaanotto
  - n Poista otsake
  - n Anna segmentti kuljetuskerrokselle
- n Verkkokerros toimii etenkin reitityksessä
  - n Reititin tutkii IP-paketin otsakkeen ja päättää, mihin linkkiin se lähetetään seuraavaksi



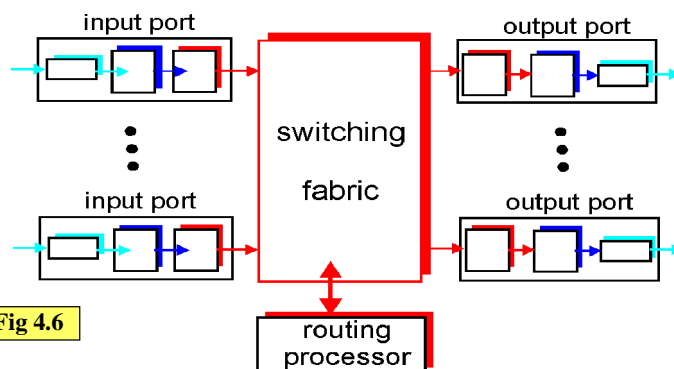
Tietoliikenteen perusteet /2009/ Liisa Marttinen

22



## Reitittimen arkkitehtuuri

- n Kaksi tehtävää
  - n Välitä paketteja tulolinkeistä ulosmenolinkkeihin
  - n Suorita reititys algoritmia / -protokollaa
- n Portti ~verkkokortti
  - n Useita portteja niputettu yhteen linjakortiksi (line card)



KuRo08:Fig 4.6

Tietoliikenteen perusteet /2009/ Liisa Marttinen

23



## IP-paketin rakenne (IPv4)

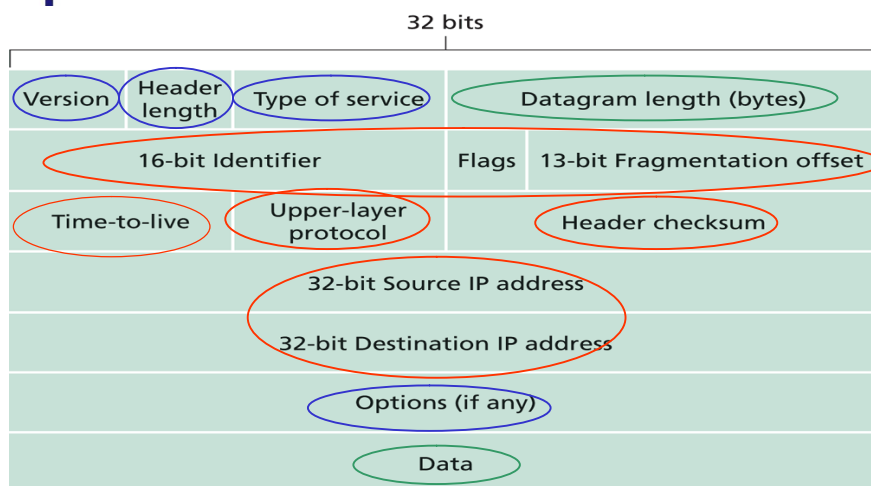


Figure 4.13 ♦ IPv4 datagram format

Tietoliikenteen perusteet /2009/ Liisa Marttinen

24

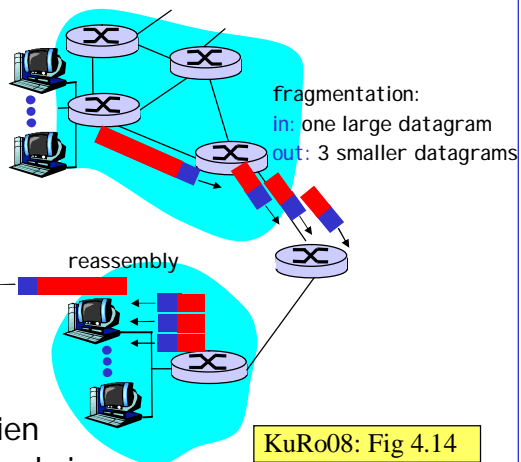
## IP-pakettien paloittelu (fragmentointi)

### Maximum transfer Unit (MTU)

suurin mahdollinen IP-paketti  
eri linkeillä eri koko  
Esim. Ethernet 1500 B

Liian iso paketti pilkottava  
reitittimessä pienemmiksi  
paketeiksi (fragmenteiksi),  
jotka kohdekone kokoaa  
voivat kukin kulkea eri reittiä

IP-otsakkeessa kentät  
yhteenkuuluvien fragmenttien  
tunnistamiseksi ja kokoamiseksi



Tietoliikenteen perusteet /2009/ Liisa Marttinen

25

## Esimerkki

length	ID	fragflag	offset
=4000	=x	=0	=0

4000 tavun IP-paketti:  
dataa 3980 B  
MTU 1500 B

Yhdestä IP-paketista tulee  
3 pienempää IP-pakettia

1480 B dataa  
20 B IP-otsaketta

offset = 1480/8

length	ID	fragflag	offset
=1500	=x	=1	=0
=1500	=x	=1	=185
=1040	=x	=0	=370

0	1480	2860
1. Pala: 1480 tavua	2. Pala: 1480 tavua	3. Pala: 1020 tavua

Tietoliikenteen perusteet /2009/ Liisa Marttinen

26



## CIDR: Classless InterDomain Routing

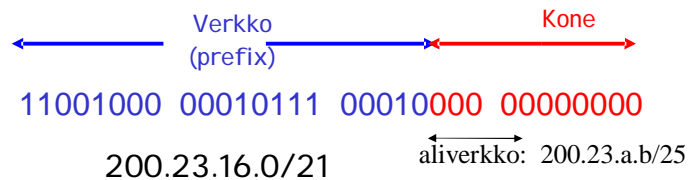
### Verkko-osa voi olla minkä tahansa kokoinen

Vanha luokallinen osoite: A-luokka 8 b, B-luokka 16 b, C-luokka 24 b

### Formaatti: a.b.c.d/x

x ilmoittaa verkko-osan bittienlukumäärän (prefix)

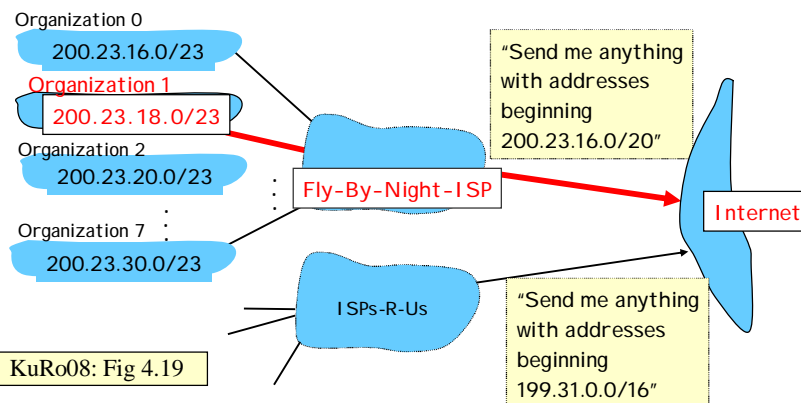
Esim. Organisaatio, jolla 2000 konetta varaa  $2024 = 2^{11}$  konenumeroa, jolloin verkko-osaa varten jää 21 bittiä  
Yritys voi vielä itse jakaa viimeiset 11 bittiä aliverkko-osoitteeksi ja koneosoitteeksi. Tämä jako ei näy ulkopuolelle.



## Hierarkkinen osoite

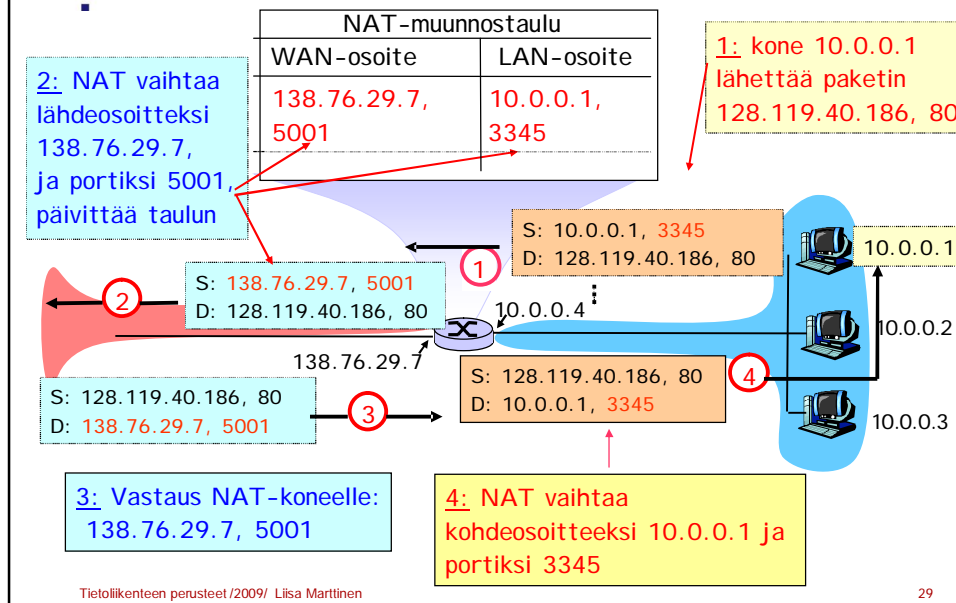
### CIDR luo reititystä helpottavan hierarkian

Aggregointi (yhdistäminen): yhteinen alkuosa => samaan suuntaan





## NAT: Esimerkki



## Reititysalgoritmi

- o Etsii edullisimmat reitit lähdekoneelta kohdekoneille
  - o Käytetään reititystaulun muodostamiseen
    - Mille linkille paketti seuraavaksi siirretään tältä reititimeltä
- o Reititysalgoritmi, joka tarvitsee täydellisen tiedon verkosta:
  - o Ennen laskentaa käytössä koko kuva verkosta:
    - Kaikki linkkiyhteydet solmujen välillä ja niiden kustannukset
    - Käytännössä vain tietyistä autonomisesta alueesta
  - o Parhaat reitit lasketaan joko keskitetysti tai hajautetusti
  - o **Linkkitila-algoritmi** (link-state algorithm)
- o Reititysalgoritmi, jolle riittää epätäydellinen kuva verkosta
  - o Aluksi reitin tietää vain niistä koneista, joihin itse on yhdistetty
  - o Iteratiivinen algoritmi: reitin vaihtaa tietoja naapuriensa kanssa ja saa tietoa muusta verkosta
  - o **Etäisyysvektorialgoritmi** (distance vector algorithm)

## Dijkstran algoritmi

$D(v)=2, D(w) = 5, \mathbf{D(x)=1}$   
 $D(y) = \infty, D(z) = \infty$

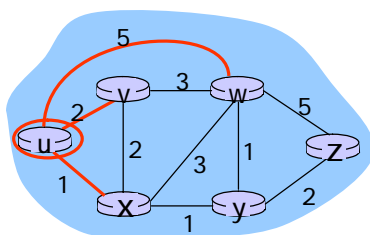
- 1 **Initialization:**
- 2  $N' = \{u\}$
- 3 for all nodes  $v$
- 4 if  $v$  adjacent to  $u$
- 5 then  $D(v) = c(u,v)$
- 6 else  $D(v) = \infty$
- 7
- 8 **Loop**
- 9 find  $w$  not in  $N'$  such that  $D(w)$  is a minimum
- 10 add  $w$  to  $N'$
- 11 update  $D(v)$  for all  $v$  adjacent to  $w$  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  $N'$**

Tietoliikenteen perusteet /2009/ Liisa Marttinen

31

## Etäisyysvektoreititys:

### Esimerkki 1



Kohde	kust.	linkki
<b>Z</b>	<b>4</b>	<b>X:ään</b>

Jos on jo saatu selville  
(= naapurit kertoneet), että  
 $D_v(z) = 5, D_x(z) = 3, D_w(z) = 3$

$$\begin{aligned}
 D_u(z) &= \min \{ c(u,v) + d_v(z), \\
 &\quad c(u,x) + d_x(z), \\
 &\quad c(u,w) + d_w(z) \} \\
 &= \min \{ 2 + 5, \\
 &\quad 1 + 3, \\
 &\quad 5 + 3 \} = 4
 \end{aligned}$$

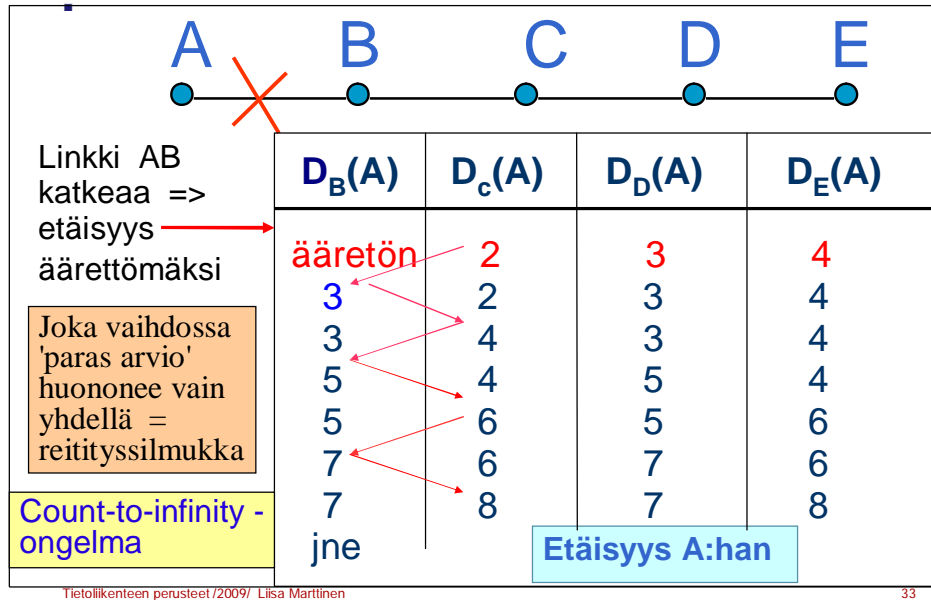
Kun paketti on matkalla solmusta  $u$  solmuun  $z$ , se tulee seuraavaksi lähettää solmuun  $x$ , joka tuotti tuon minimin => talleta tieto omaan etäisyysvektoriin (= reititystauluun)

Tietoliikenteen perusteet /2009/ Liisa Marttinen

32

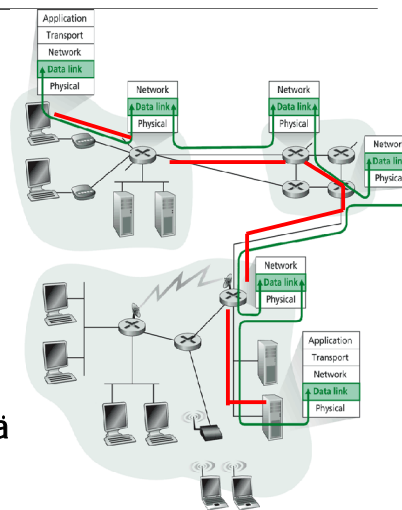


## Huono uutinen etenee hitaasti!



## Linkkikerros

- n Laitetoimintoa
- n Siirtää paketin fyysistä linkkiä pitkin koneelta (solmulta (node)) toiselle
  - langallinen / langaton
  - bitit sisään, bitit ulos
- n Kapseloi paketin siirtoon sopivaan muotoon
  - n Siirtokehys (frame)
- n Lähiverkossa linkejä voi yhdistää keskittimillä tai kytkimillä
  - n Käytetään fyysisiä osoitteita
  - n 'reititystä' ilman IP-osoitteita

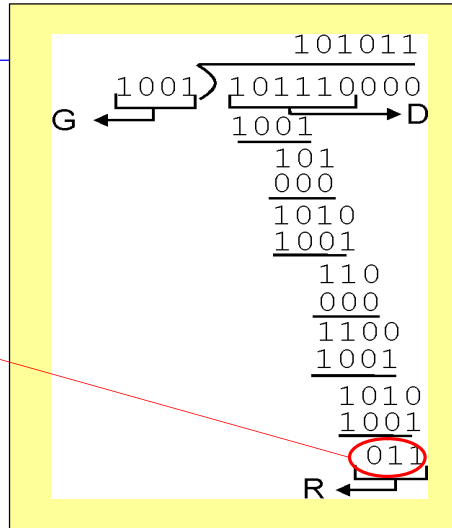


## CRC-esimerkki

Data: 101110  
 G: 1001, polynomina  
 $1*x^3 + 0*x^2 + 0*x^1 + 1*x^0$   
 <D,R>: 101110???

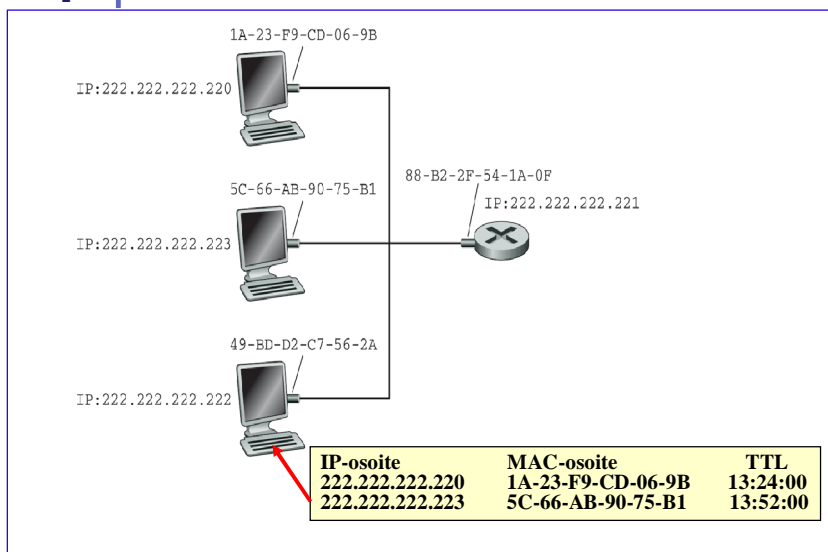
Lähetä: 101110**011**

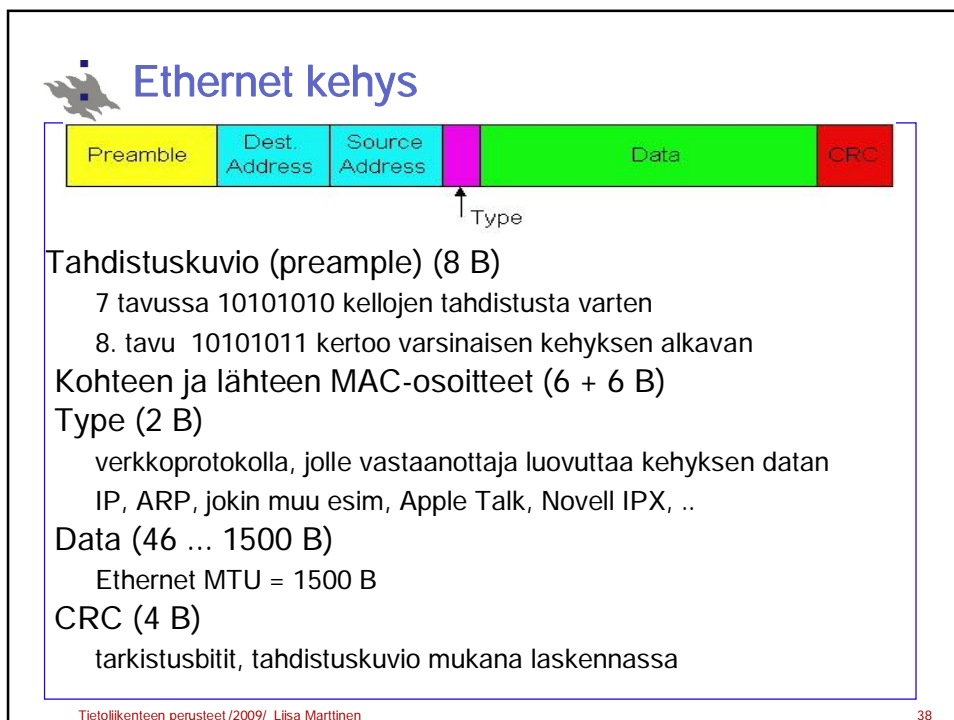
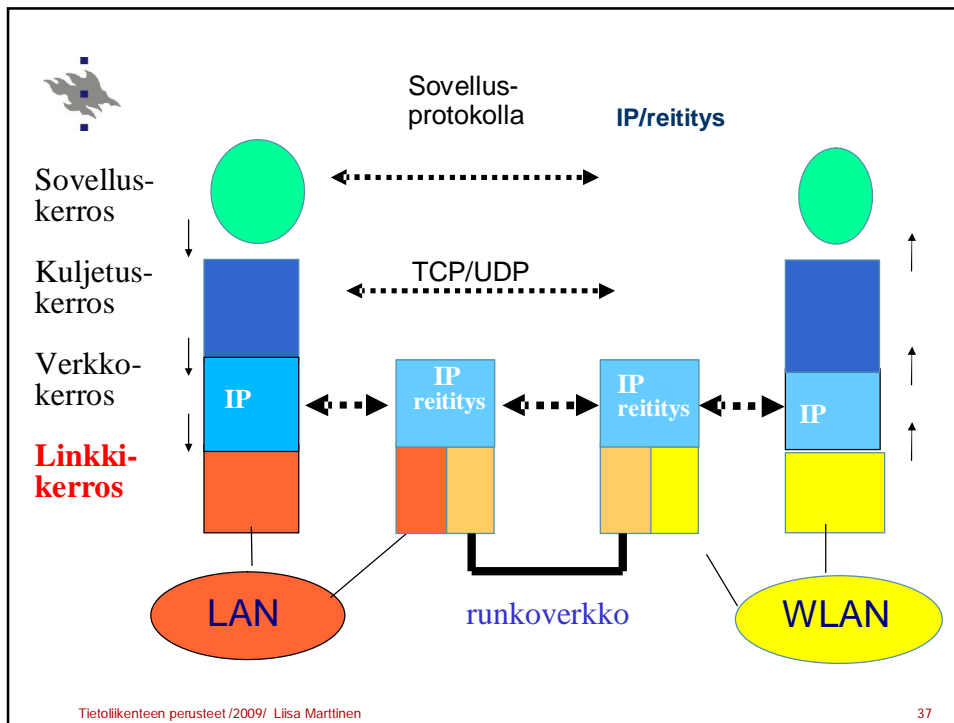
Modulo 2-aritmetiikka  
 vähennyslasku yhteenlaskuja  
 ei lainaamista, ei muistinumeroita  
 = bittitason XOR  
 $1+1=0, 1+0=0+1=1, 0+0=0$



KuRo08:Fig 5.8

## MAC-osoitteet ja ARP-taulu, ARP-protokolla







## CSMA/CD (with Collision Detection)

### Asema kuuntelee myös lähettämisen jälkeen

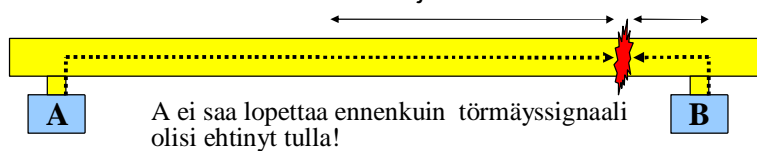
- n Langallinen LAN: signaalin voimakkuus muuttuu
  - Esim. Ethernet
- n Langaton LAN: hankalaa

### Jos törmäys

- n Niin keskeytä heti lähettäminen
- n ja yritä uudestaan satunnaisen ajan kuluttua
- n Näin törmäyksen aiheuttama hukka-aika pienenee

### Kauanko kuunneltava?

- n  $2^*$  maksimi etenemisviive solmujen välillä

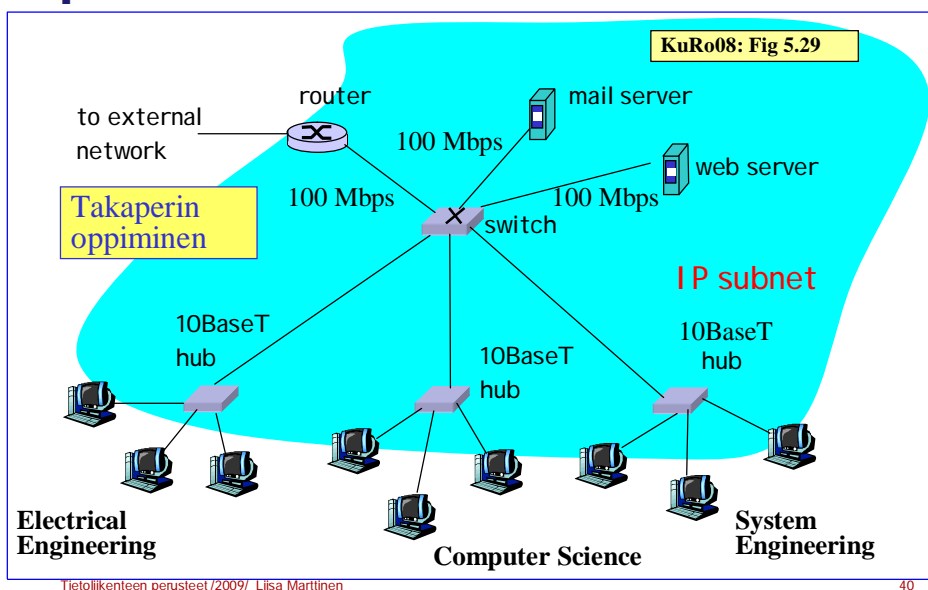


Tietoliikenteen perusteet /2009/ Liisa Marttinen

39



## LAN, verkkosegmentit



Tietoliikenteen perusteet /2009/ Liisa Marttinen

40



# 802.11: CSMA/CA

## Lähetys

### 1. Jos kanava vapaa

Kuuntele DIFS aikayksikköä  
Lähetä kehys kokonaan

### 2. Jos kanava varattu

→ Käynnistä peruutuslaskuri (backoff)  
random(max), jota vähennetään vain  
kun kanava on vapaa,  
Lähetä, kun laskuri nollassa  
Jos ei tule kuittausta, niin yritä  
uudestaan max = 2\*max

## Vastaanotto

Jos kehys OK

Odota SIFS aikayksikköä

Lähetä ACK (linkkikerroksen ACK)

KuRo08: Fig 6.10

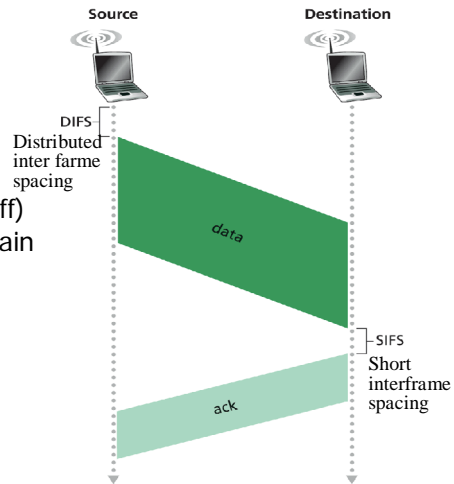


Figure 6.8 ♦ 802.11 uses link-layer acknowledgments

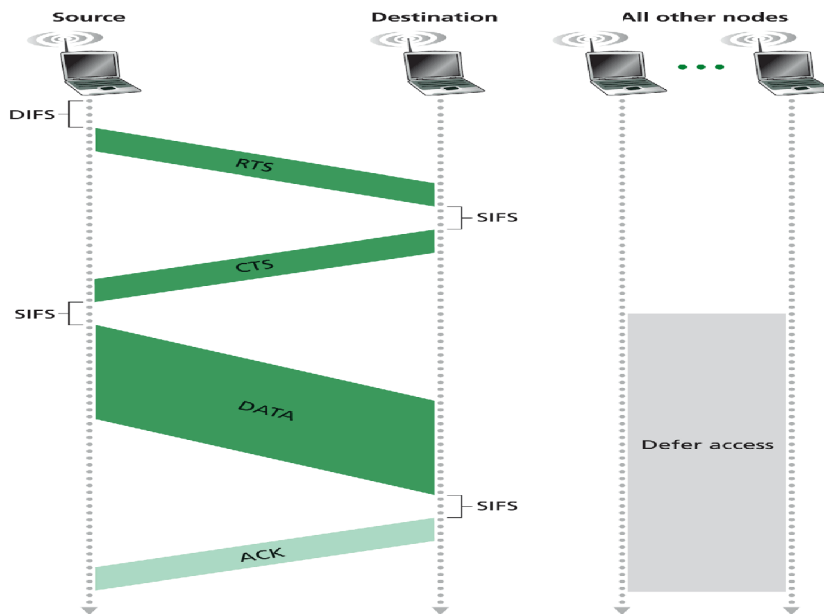


Figure 6.10 ♦ Collision avoidance using the RTS and CTS frames

## Hajautettu DoS-hyökkäys (DDoS)

Hyökkääjä ottaa ensin haltuun ison joukon koneita niiden omistajien huomaamatta

- Koputtelee ja löytää turva-aukot
- Asentaa hyökkäysohjelman, joka vain odottelee käskyä /kellolyömaa

Kaapatut koneet aloittavat samaan aikaan hyökkäyksen uhrin kimppuun

- Hajautetusti
- IP-osoitteet peukaloituina

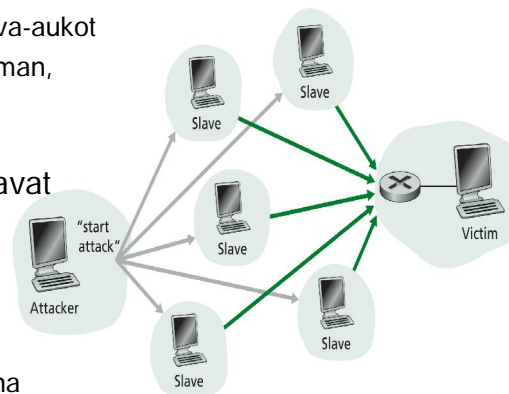


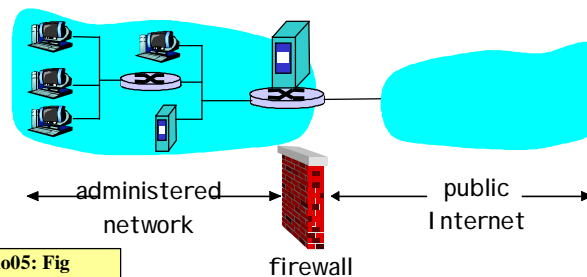
Figure 8.26 ♦ A DDoS attack

## Palomuri (firewall)

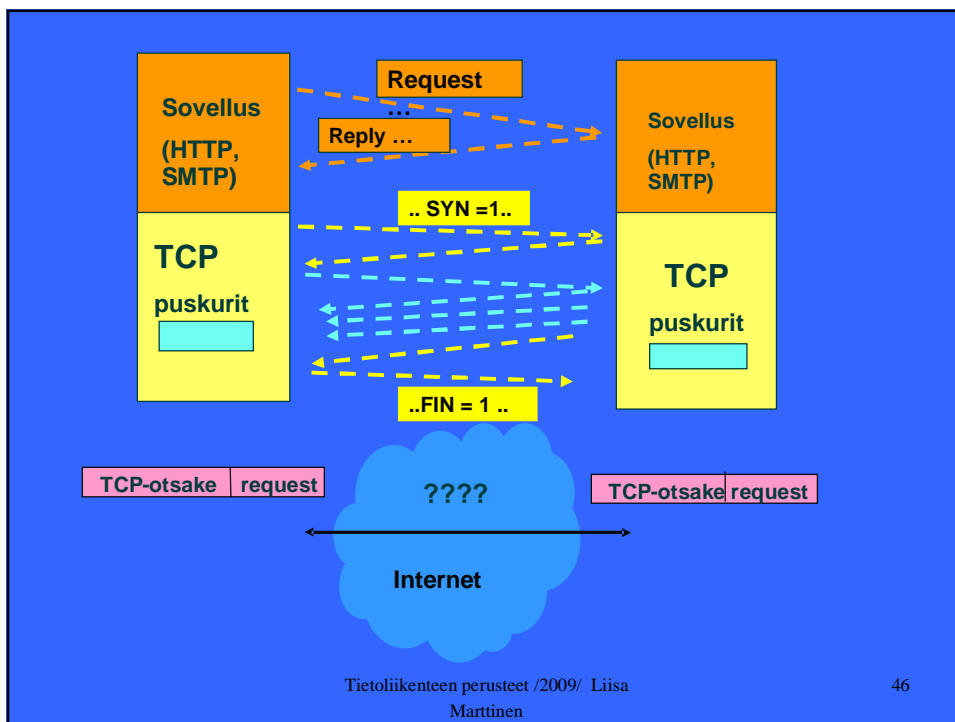
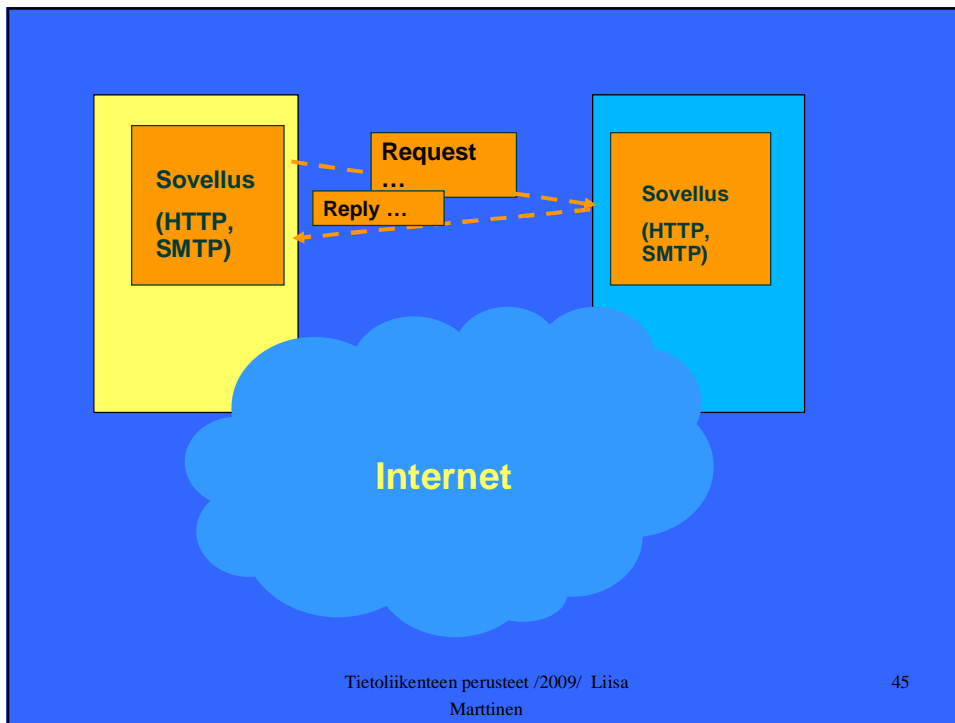
Ohjelmisto + laitteisto

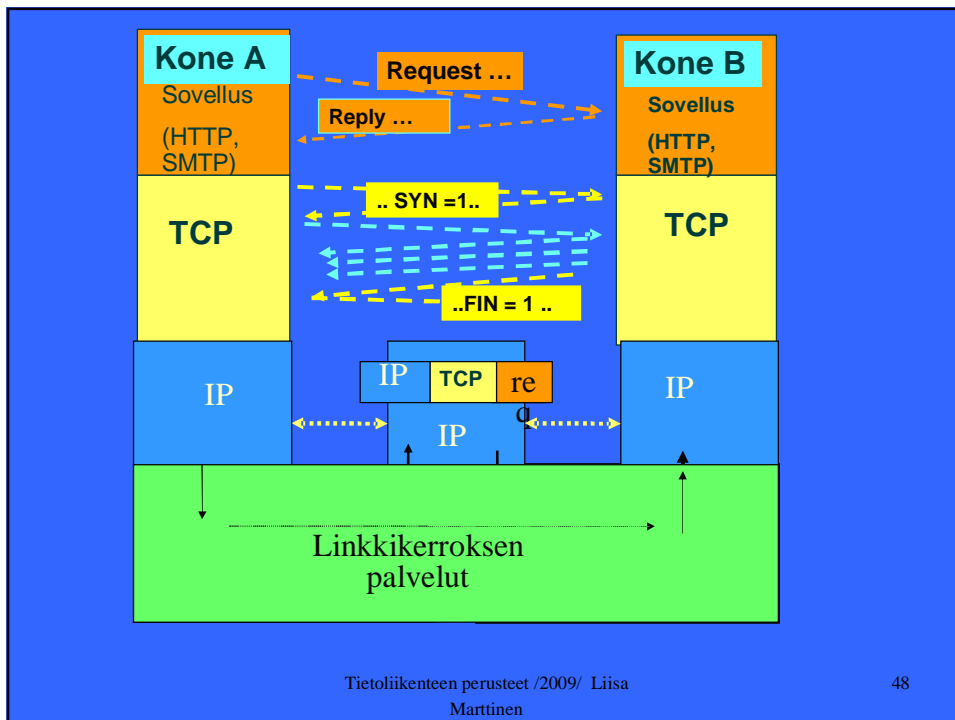
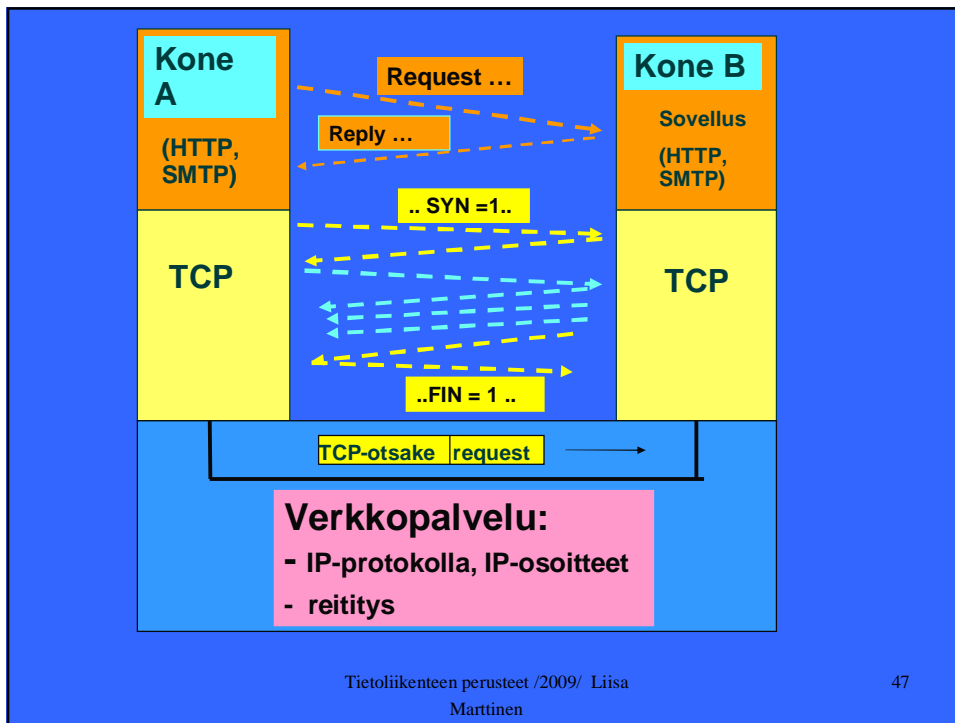
Suodattaa (filteroi) liikennettä organisaation oman verkon (intranet) ja julkisen Internetin välillä

- Osa IP-paketeista pääsee palomuurin läpi, osa ei

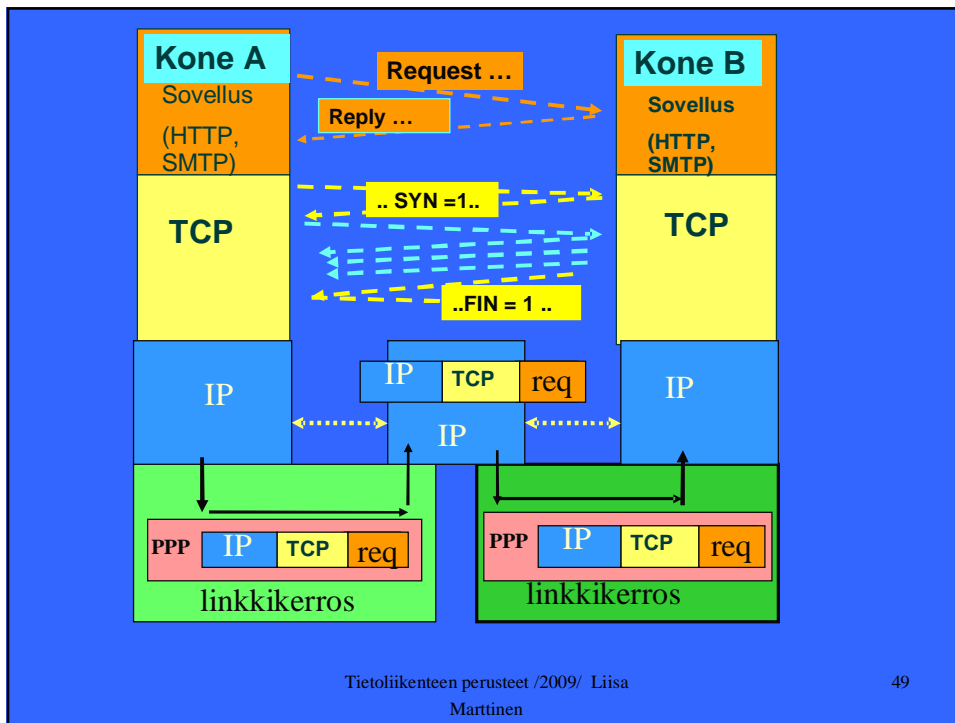


KuRo05: Fig 8.23









# · Tietoliikenteen perusteet

Siinäpä se!

Tietoliikenteen perusteet /2009/ Liisa Marttinen

50