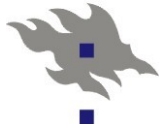




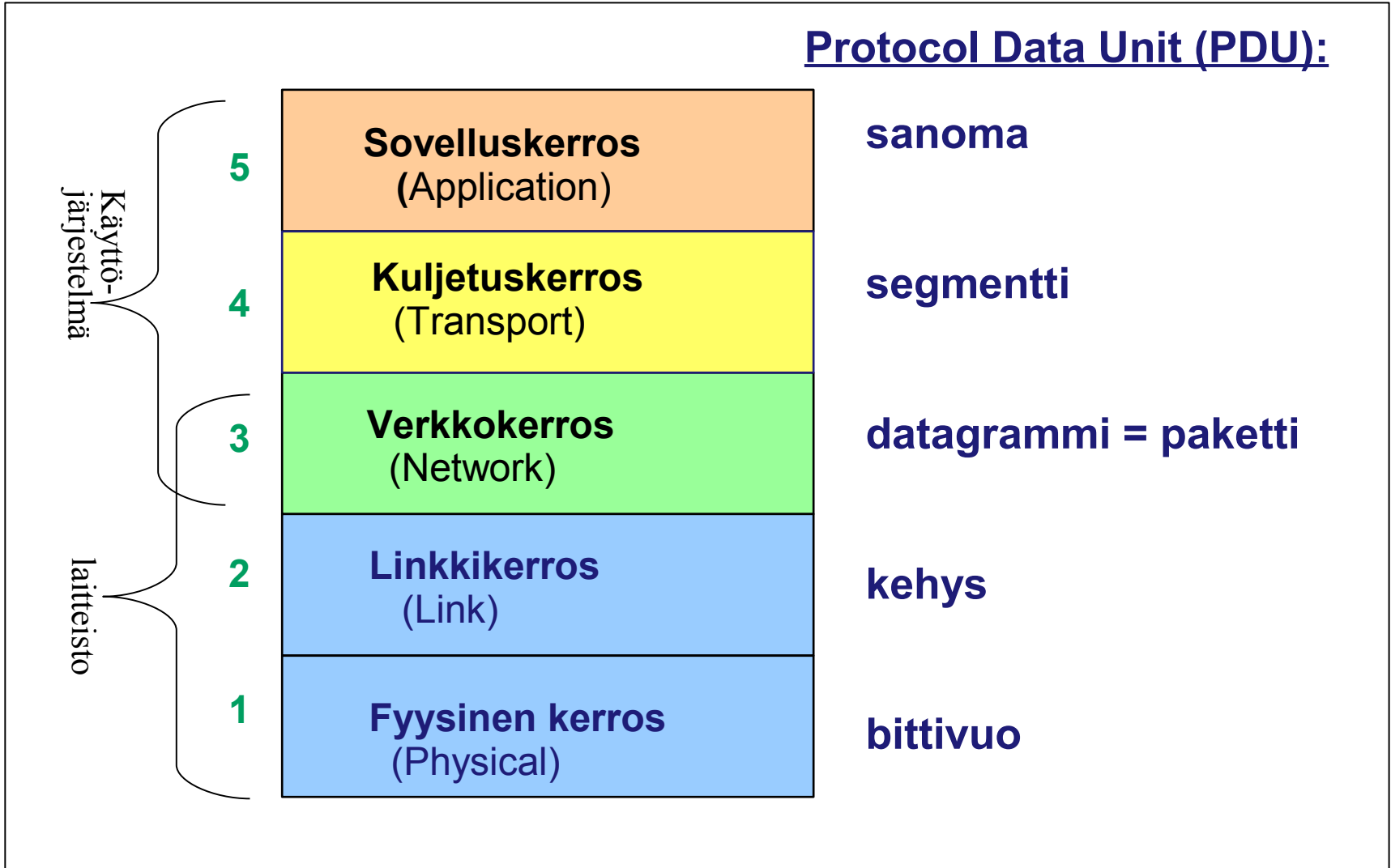
Tietoliikenteen perusteet

Vähän kertausta

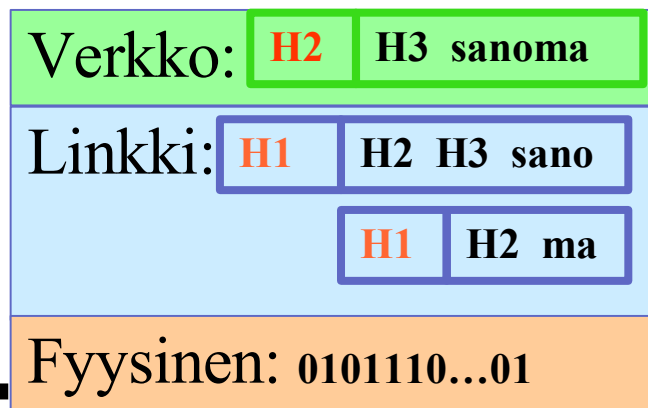
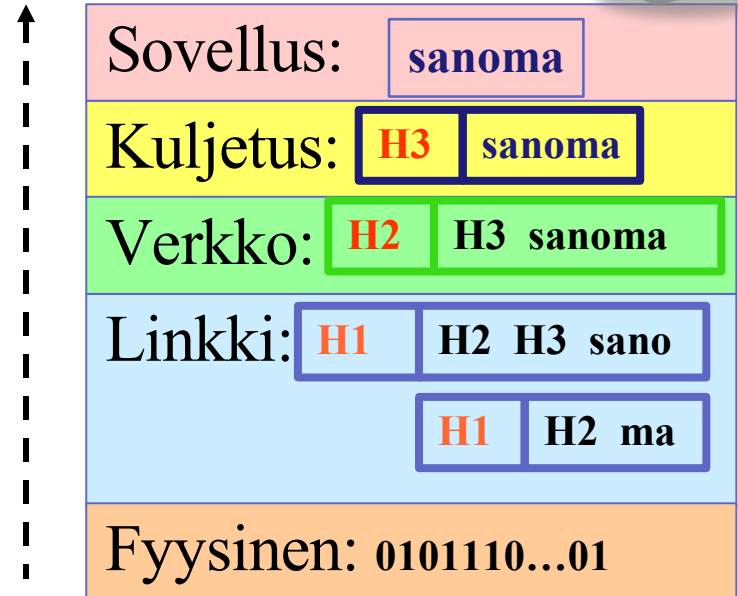
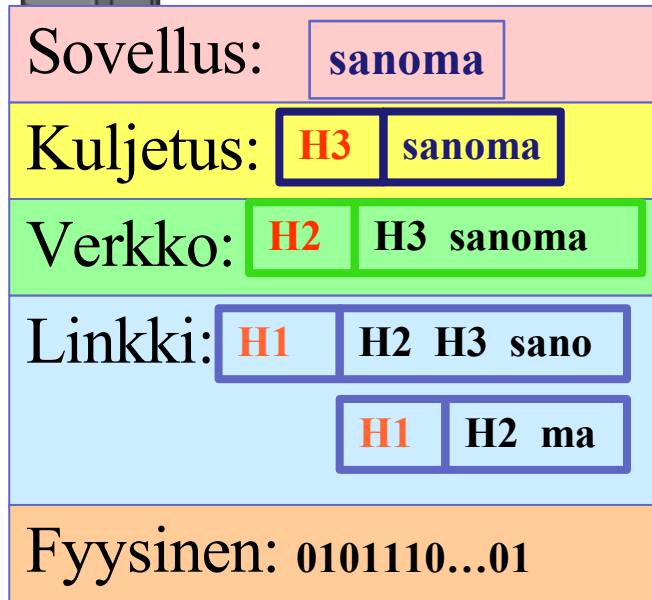
Internet-protokollapino



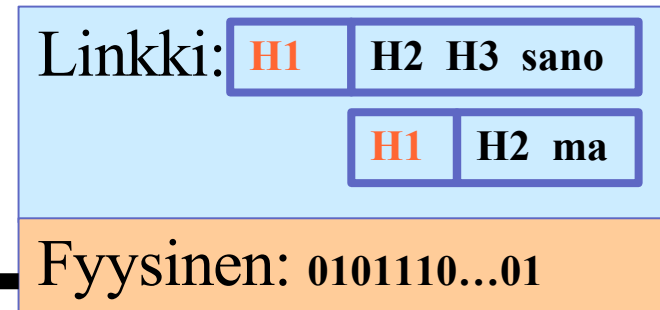
Protocol Data Unit (PDU):



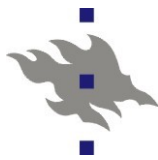
Kapselointi



Reititin



Linkkitason kytkin



HTTP (HyperText Transfer Protocol)

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



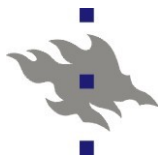
Palvelin, jossa on Apache-www-palvelija

HTTP Request
HTTP Response

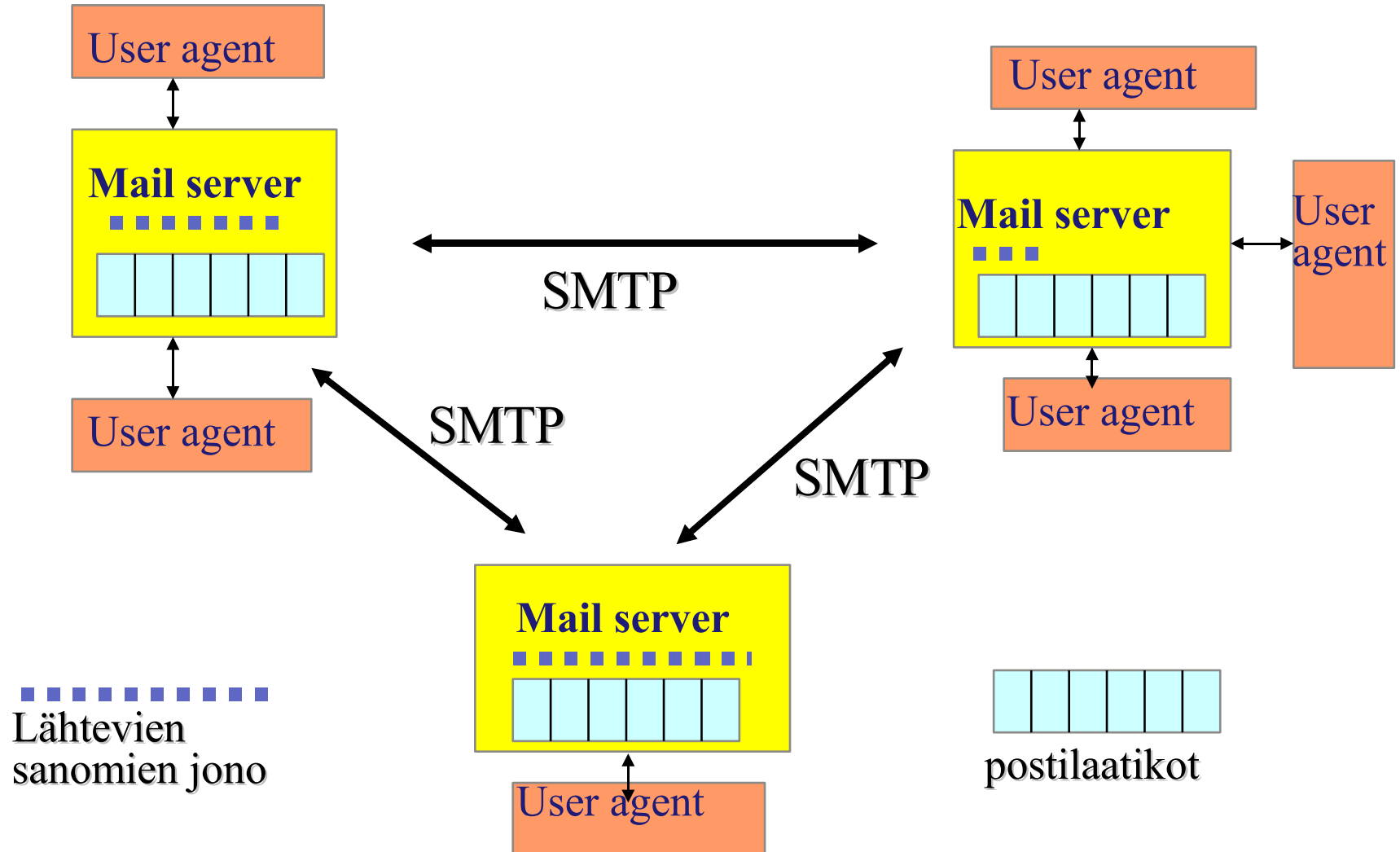
HTTP Response
HTTP Request



Linux-kone, jossa on Firefox-selain

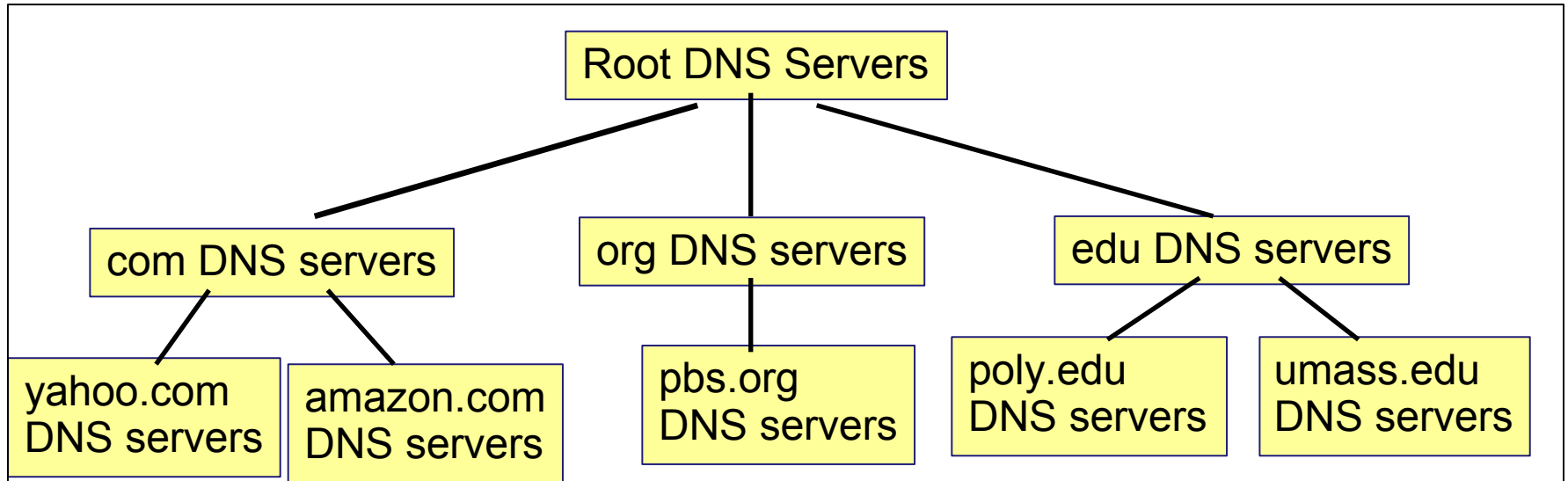


Sähköpostin komponentit





Hajautettu, hierarkinen tietokanta



KuRo05: Fig 2.18

- **13 juuritason nimipalvelija**

Replikoituja, kaikilla samat tiedot

- **Ylätason palvelimet maa- ja yleistunnuksille (n. 265 kpl)**

..., fi, fr, uk, ... edu, net, com, org, ...

- **Autorisoidut aluepalvelimet (domain) (2-taso)**

www.iana.org

Isoilla yliopistoilla ja firmoilla omansa, pienet käyttävät jonkun muun ylläpitämää



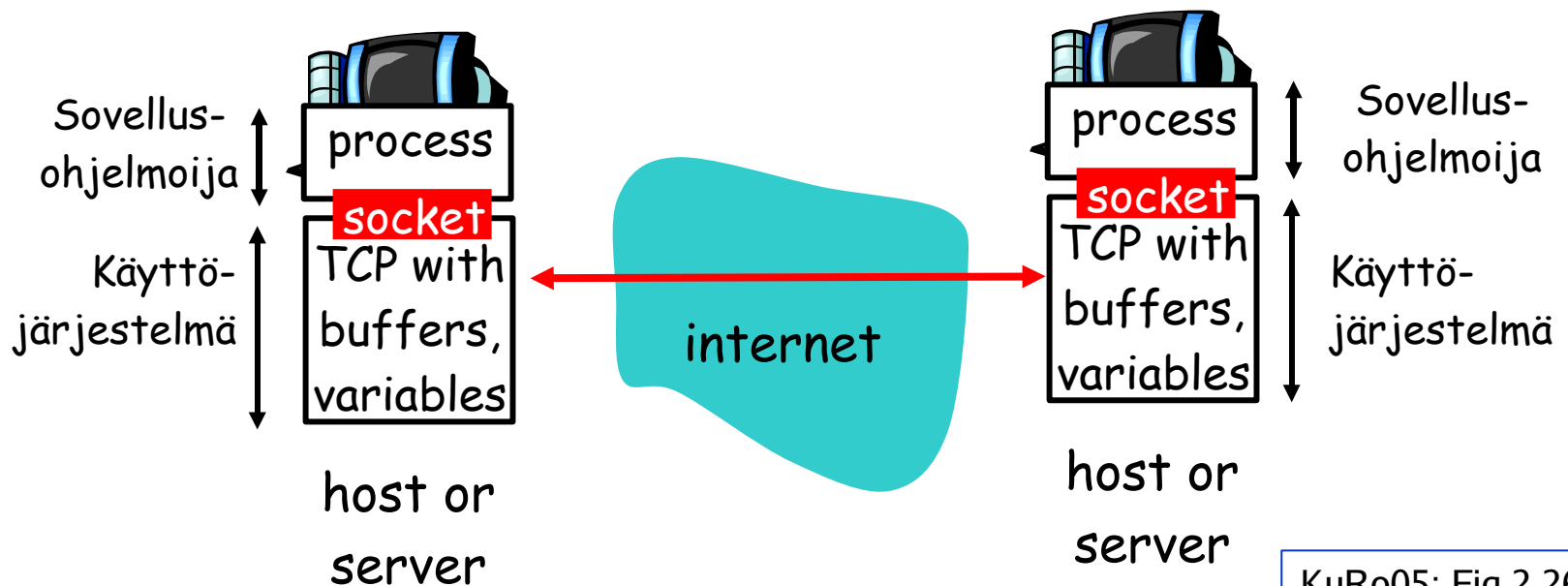
Pistoke (socket)

- Kuljetuspalvelun ja sitä käyttävän sovelluksen rajapinta isäntäkoneessa

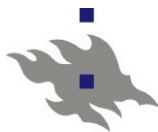
Sovelluksen tietoliikenne = KJ:n palvelupyyntöjä

Pistoke on “palveluluukku”

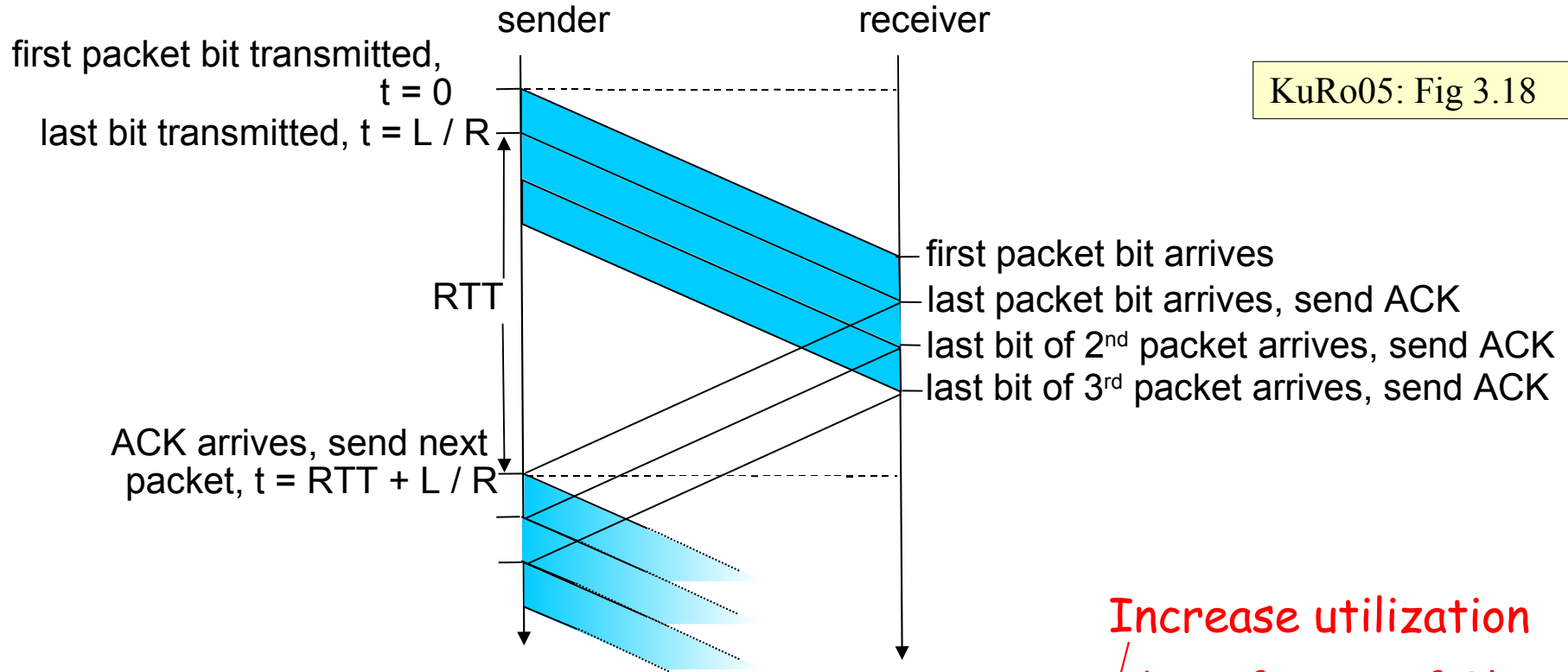
- Alunperin Berkeley UNIXin (BSD) mukana



KuRo05: Fig 2.26

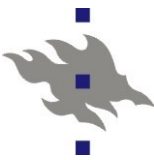


Liukuhihnoitus: käyttöasteen kasvattaminen

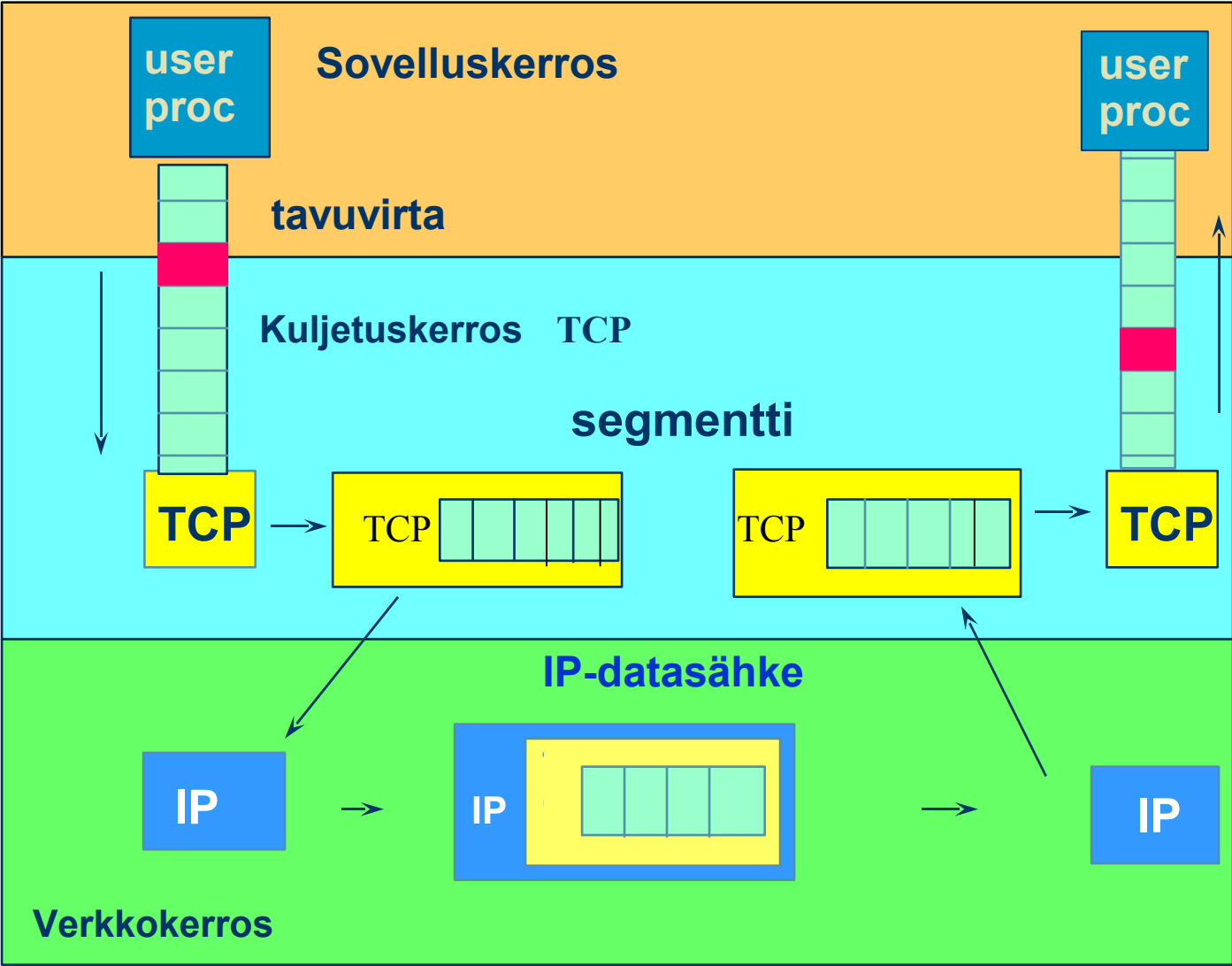


KuRo05: Fig 3.18

$$U_{\text{sender}} = \frac{3 * L / R}{RTT + L / R} = \frac{.024}{30.008} = 0.0008$$



TCP: prosessilta prosessille -tavuvirta



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
- Ikkuna kaksinkertaistuu yhden RTT:n aikana

hidas aloitus

Jos uudelleenlähetys, puolita ruuhkaikkunan koko

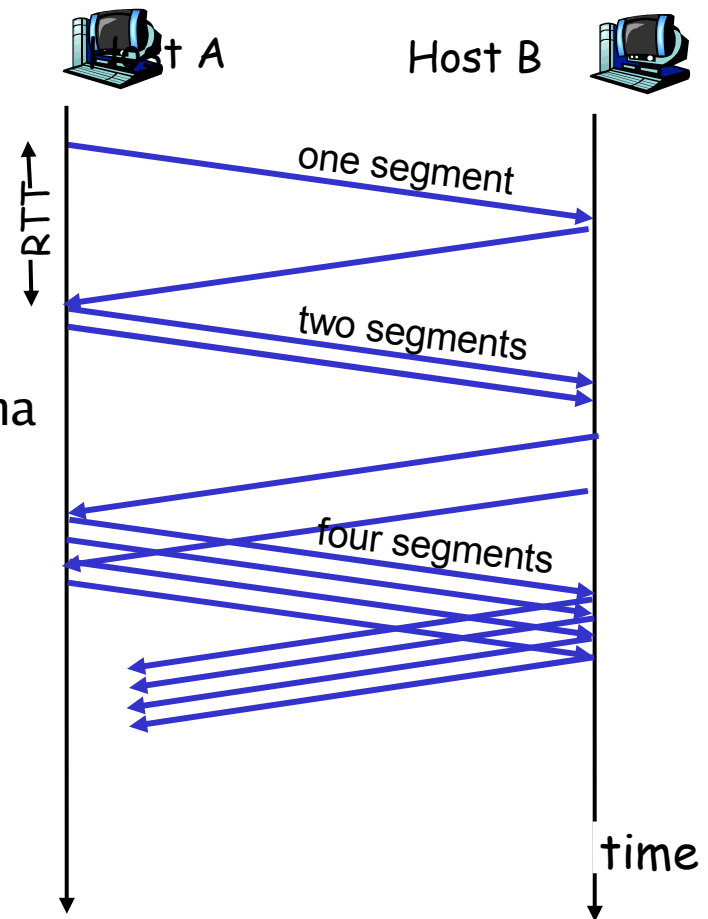
- Multiplicative decrease

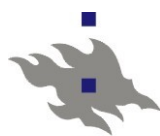
Sen jälkeen kasvata ikkunaa yksi segmentti/RTT

ruuhkanvälttely

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

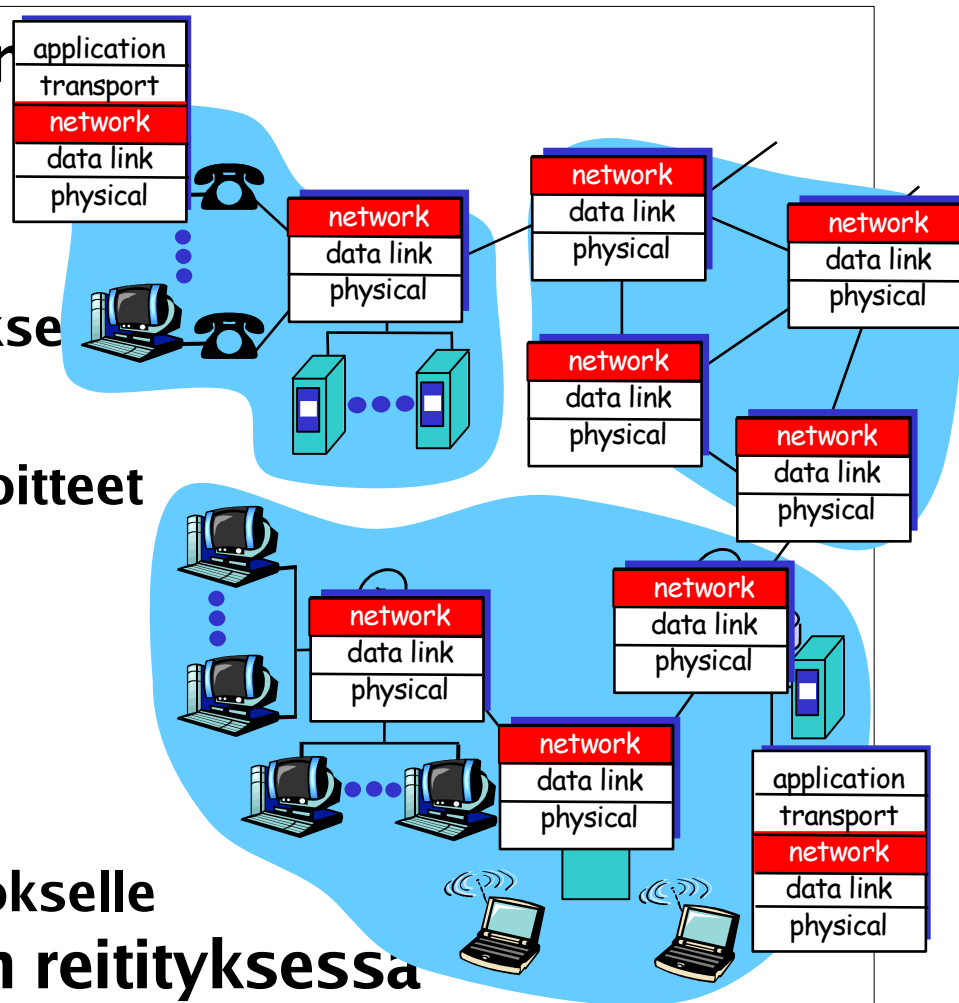
Siirtonopeus = $CognWin / RTT$ tavua/sek



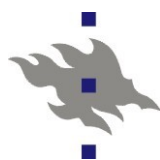


Verkkokerros

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



KuRo05: Fig 4.1



IP-pakettien paloittelu (fragmentointi)

Maximum transfer Unit (MTU)

suurin mahdollinen IP-paketti

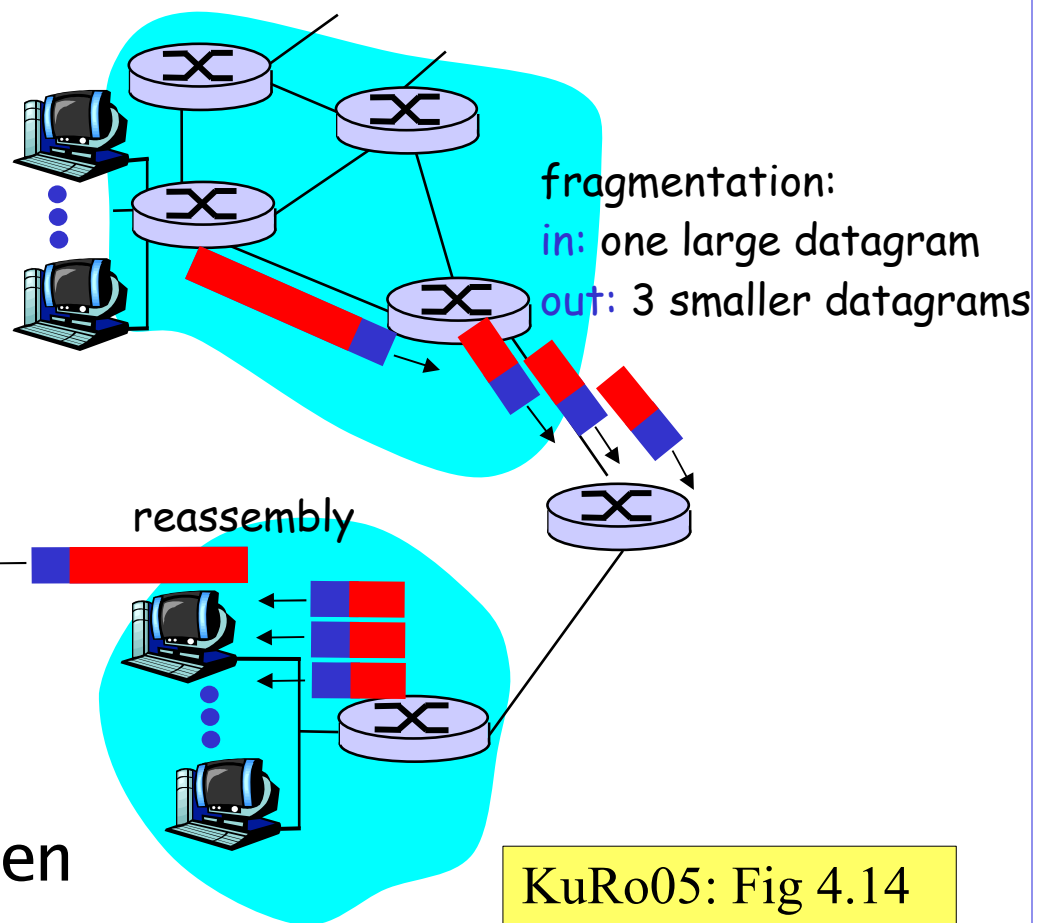
eri linkeillä eri koko

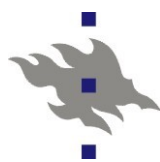
Esim. Ethernet 1500 B

Liian iso paketti pilkottava
reitittimessä pienemmiksi
paketeiksi (fragmenteiksi),
jotka kohdekone kokoaa

voivat kukin kulkea eri reittiä

IP-otsakkeessa kentät
yhteenkuuluvien fragmenttien
tunnistamiseksi





CIDR: Classless InterDomain Routing

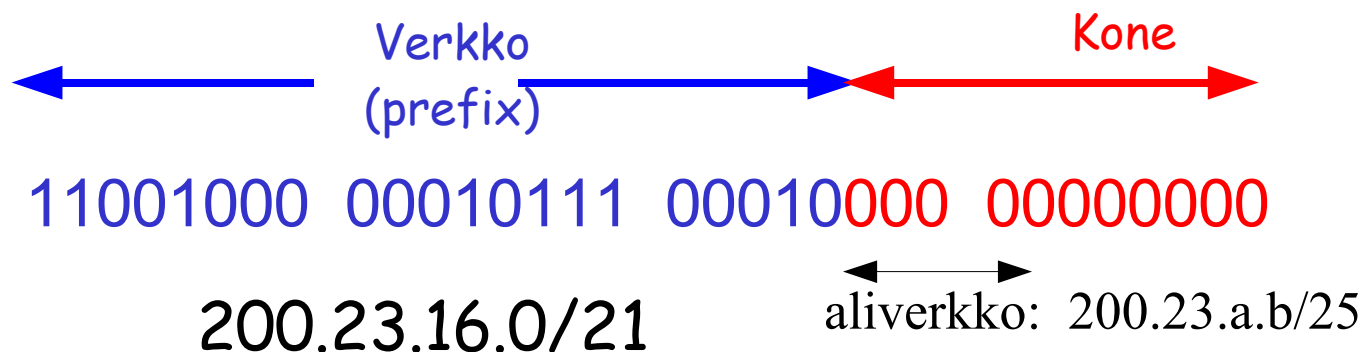
■ Verkko-osa voi olla minkä tahansa kokoinen

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 bittien lukumäärän (**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.



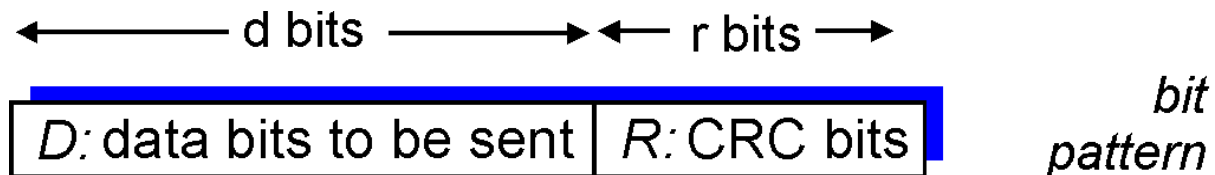


Reititysalgoritmi

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

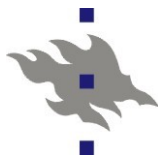
CRC

- Käsittelee databittejä yhtenä kokonaislukuna
- Sovittu **virittäjäpolynomi** G
bittejä yksi enemmän kuin lisättäviä tarkistusbittejä ($=r$) eli $r+1$
- Lähettäjä
Asettaa tarkistusbitit R s.e. datan bitit + niiden perään liitetyt tarkistusbitit ovat jaollisia virittäjällä D (**modulo 2-aritmetiikka**)
- Vastaanottaja
Jakaa samoin saamansa bittijonon $(D+R)$ virittäjällä D
Jos **jakoäännös $\neq 0$, niin on virhe.**



$$D * 2^r \text{ XOR } R$$

mathematical formula



CSMA/CD (with Collision Detection)

■ Asema kuuntelee myös lähettämisen jälkeen

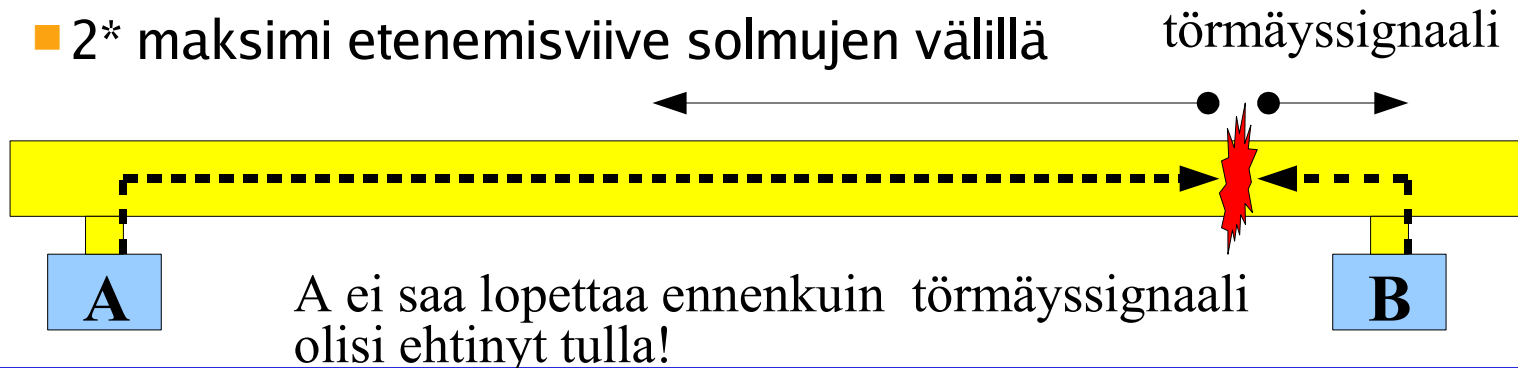
- Langallinen LAN: signaalin voimakkuus muuttuu
 - Esim. Ethernet
- Langaton LAN: hankalaa

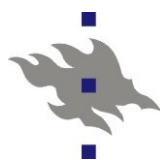
■ Jos törmäys

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

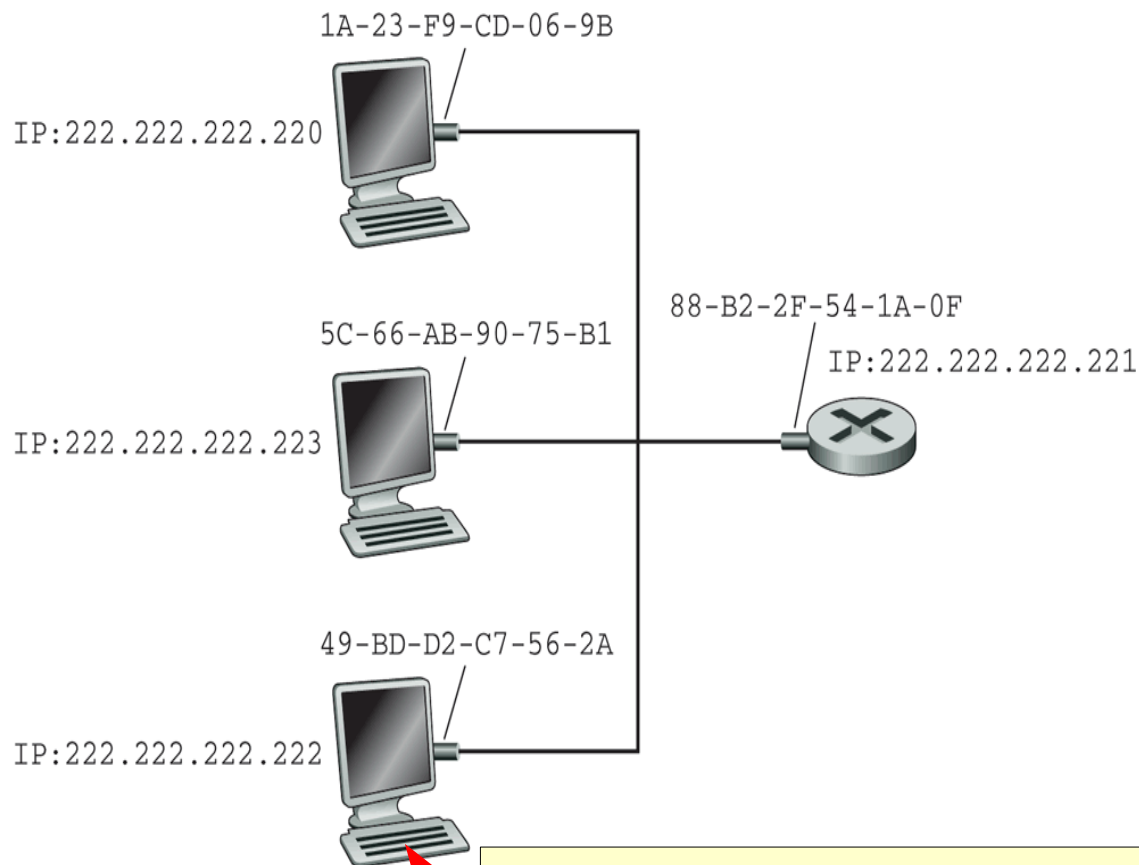
■ Kauanko kuunneltava?

- 2^* maksimi etenemisviive solmujen välillä



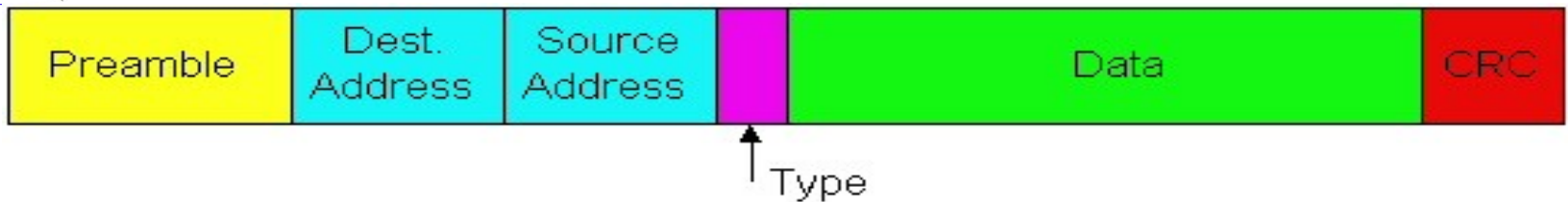


MAC-osoitteet ja ARP-taulu



IP-osoite	MAC-osoite	TTL
222.222.222.220	1A-23-F9-CD-06-9B	13:24:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

Ethernet kehys



Tahdistuskuvio (preamble) (8 B)

7 tavussa 10101010 kellojen tahdistusta varten

8. tavu 10101011 kertoo varsinaisen kehyksen alkavan

Kohteen ja lähteen MAC-osoitteet (6 + 6 B)

Type (2 B)

verkkoprotokolla, jolle vastaanottaja luovuttaa kehyksen datan

IP, ARP, jokin muu esim, Apple Talk, Novell IPX, ..

Data (46 ... 1500 B)

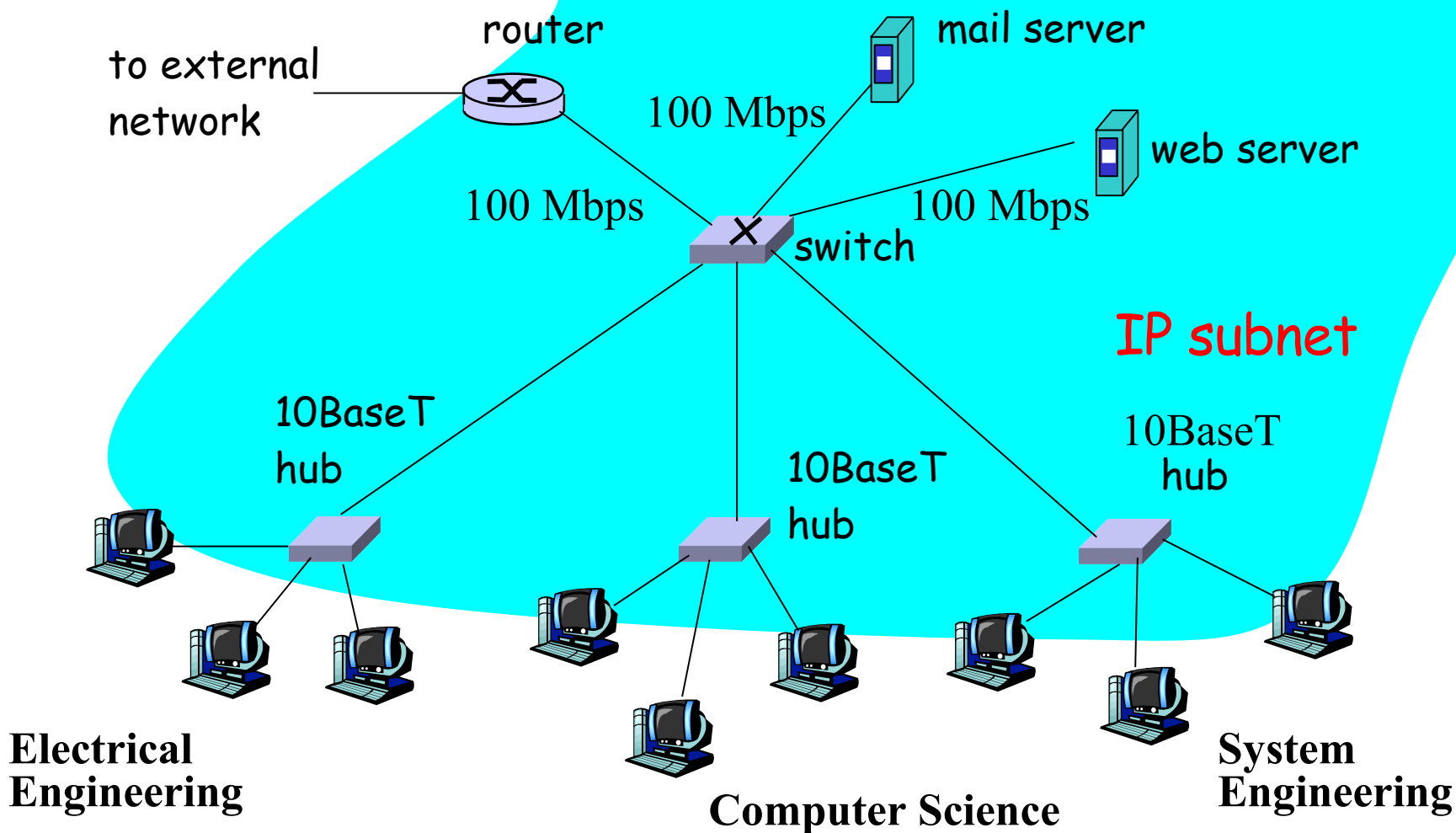
Ethernet MTU = 1500 B

CRC (4 B)

tarkistusbitit, tahdistuskuvio mukana laskennassa

LAN, verkkosegmentit

KuRo05: Fig 5.29



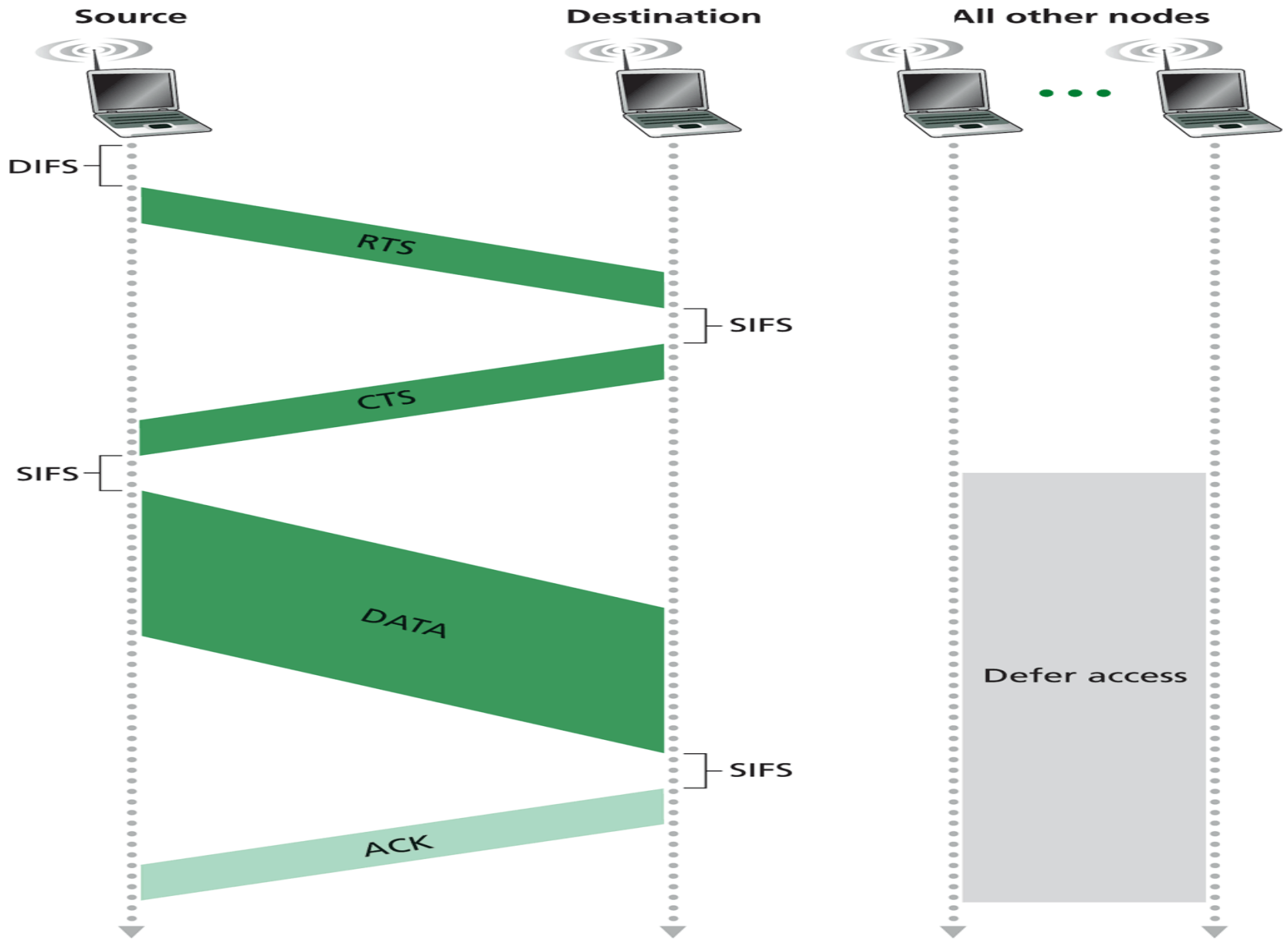
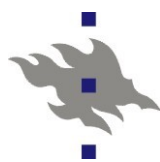


Figure 6.10 ♦ Collision avoidance using the RTS and CTS frames



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 /kellolyömää
- Kaapatut koneet aloittavat samaan aikaan hyökkäyksen uhrin kimppuun
 - Hajautetusti
 - IP-osoitteet peukaloituina

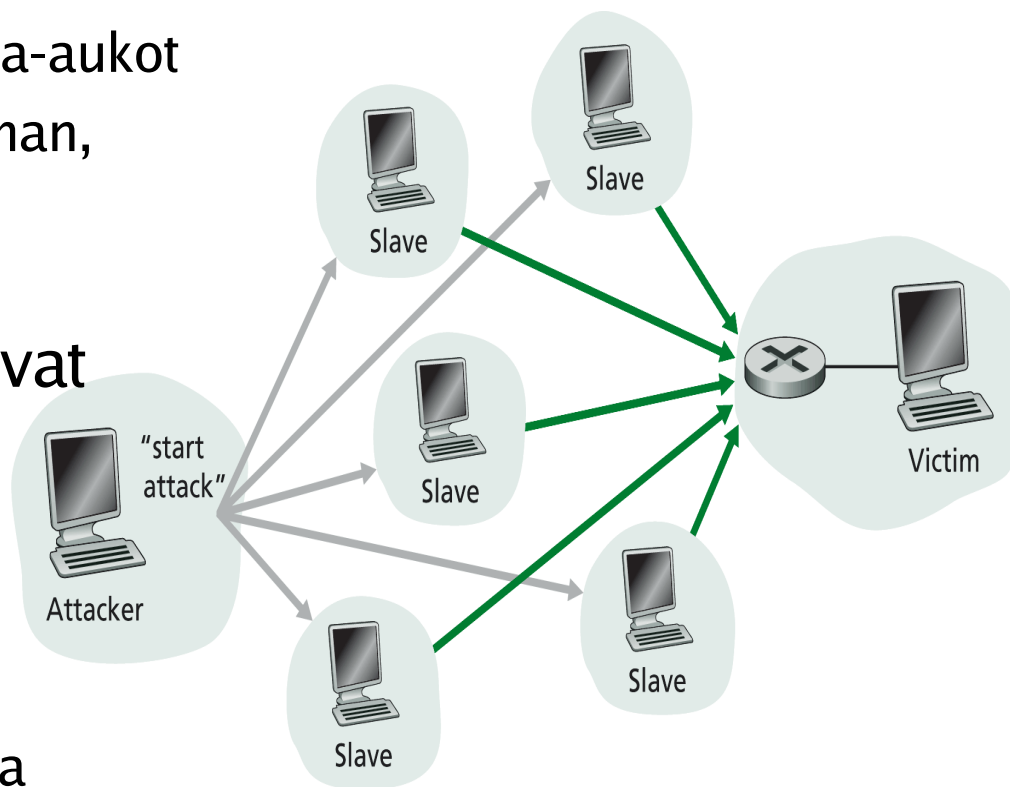
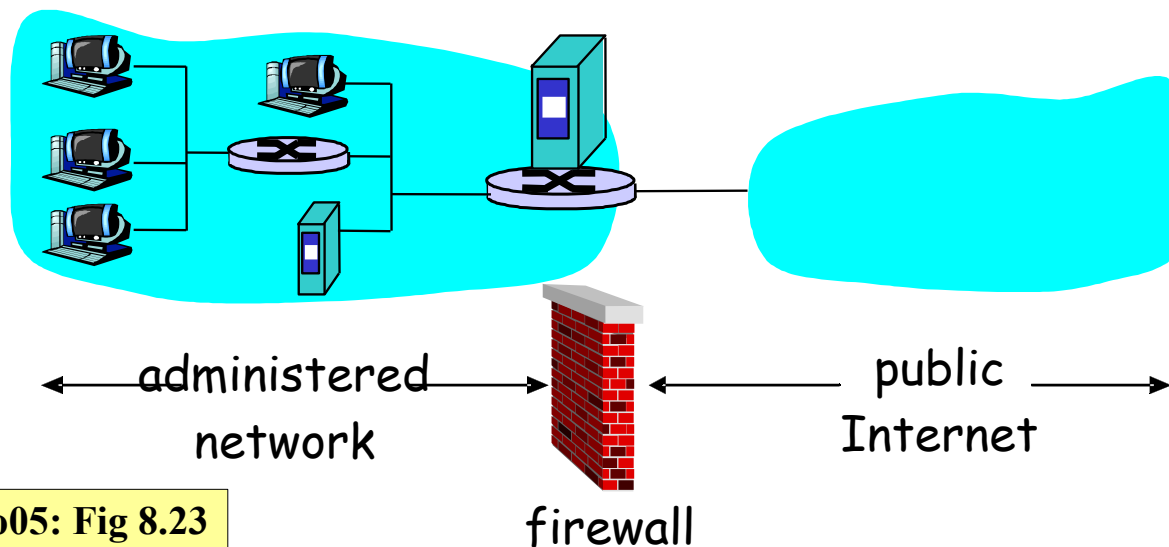


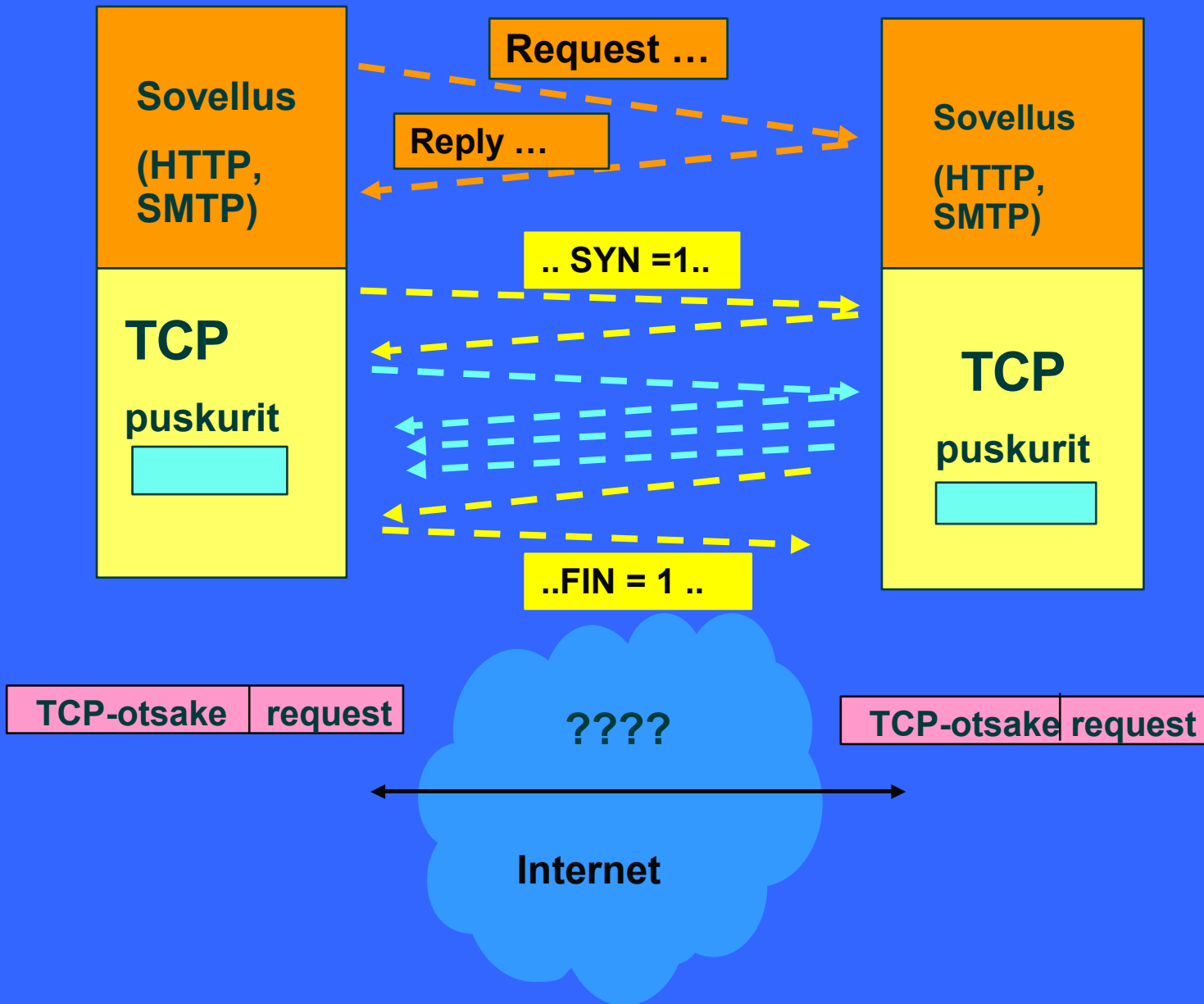
Figure 8.26 ♦ A DDoS attack

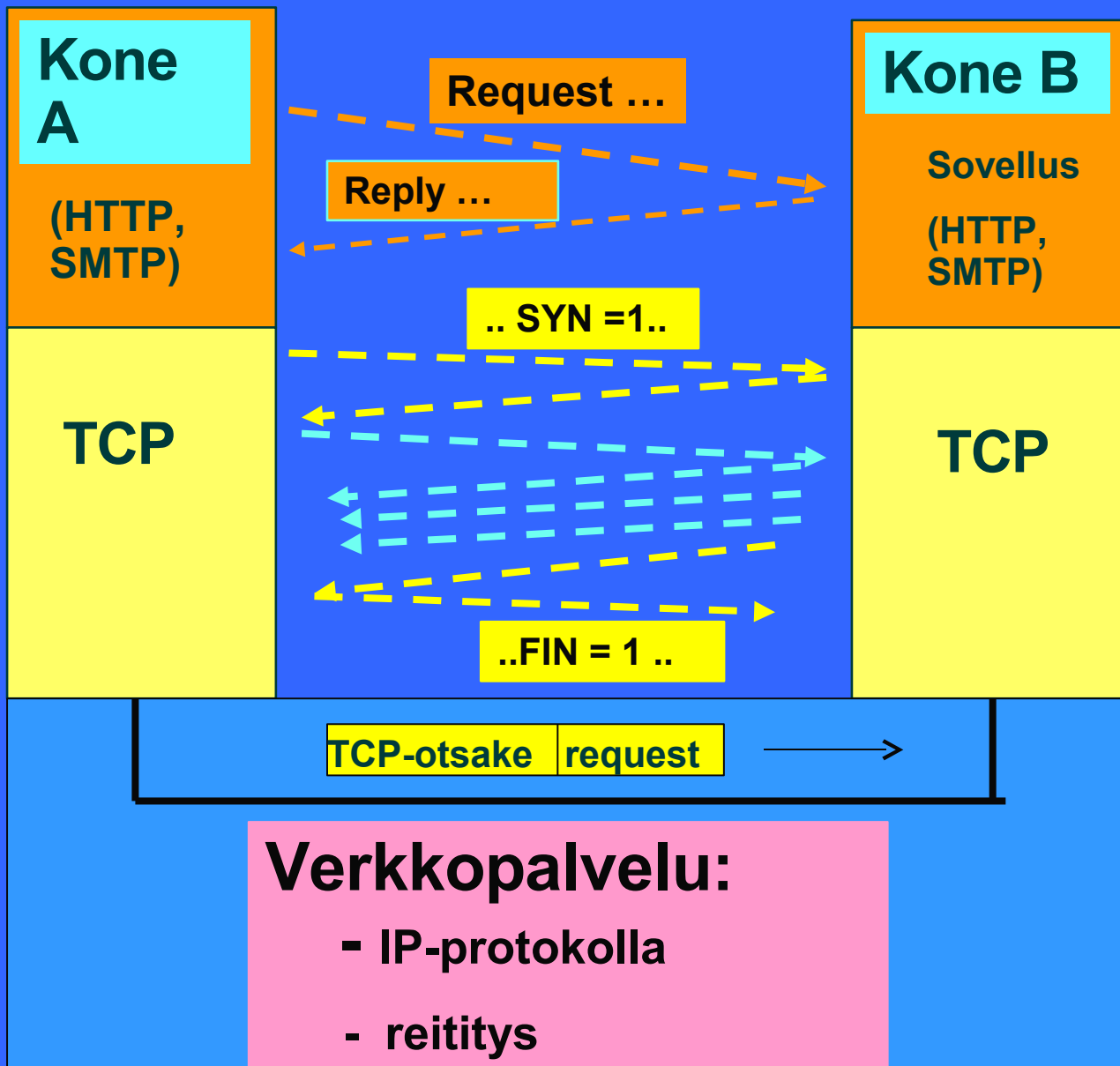
Palomuri (firewall)

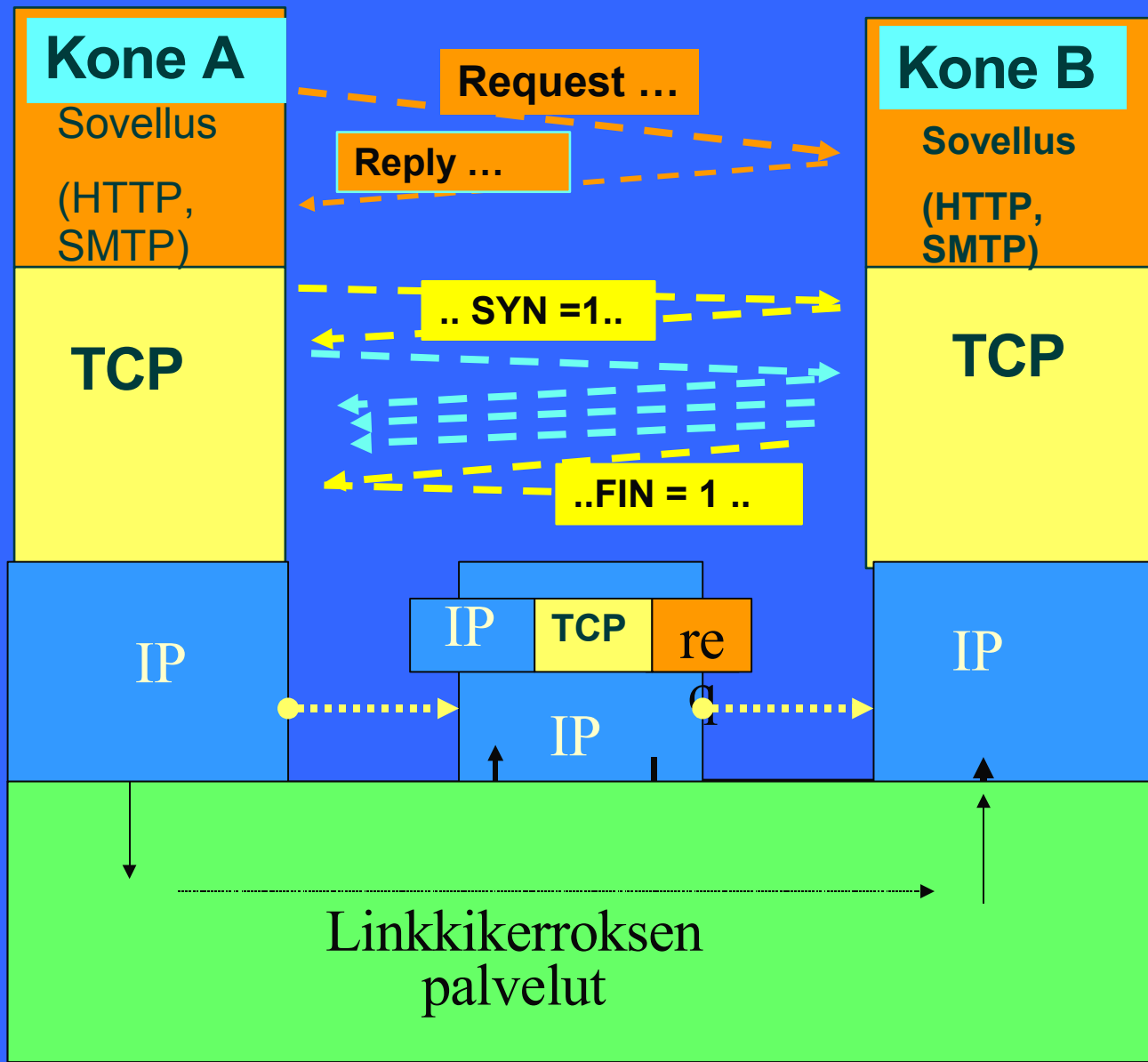
- Ohjelmisto + laitteisto
- Suodattaa (filteroi) liikennettä organisaation oman verkon (intranet) ja julkisen Internetin välillä
 - Osa IP-paketeista pääsee palomuurin läpi, osa ei

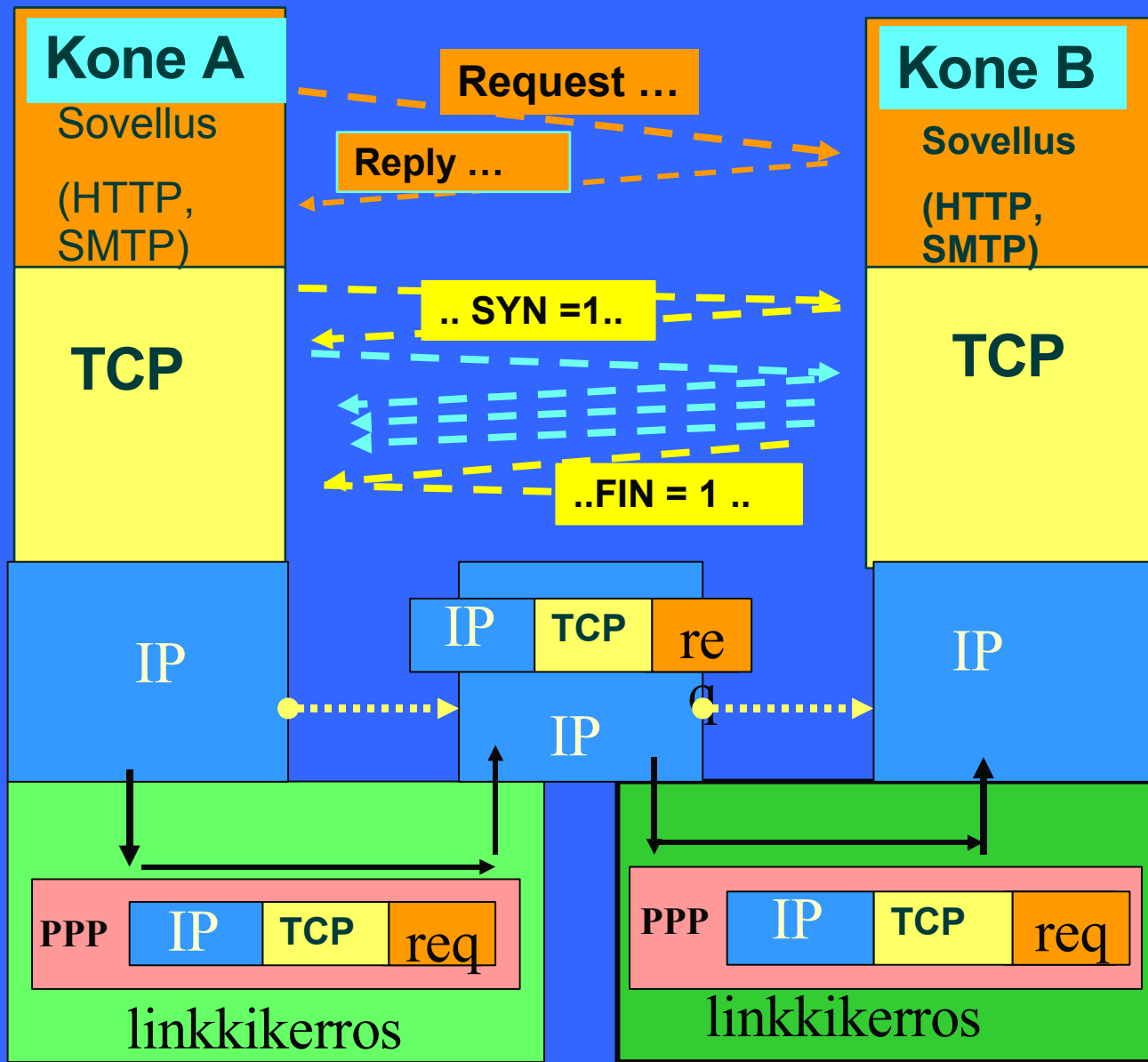


KuRo05: Fig 8.23









• Tietoliikenteen perusteet



Siinäpä se!

