Tietoliikenteen perusteet

Kertausta

Kurssin sisältö

1. **Tietokoneverkot ja Internet**

   Internetin rakenne, terminologiaa

2. **Verkkosovelluksia ja sovellusprotokollia**

   Web, sähköposti, nimipalvelu, tiedostopalvelu, pistokerajapinta

3. **Kuljetuskerros: TCP, UDP**

   yhteydellinen / yhteydetön, ruuhkanhallinta

4. **Verkkokerros: IP**

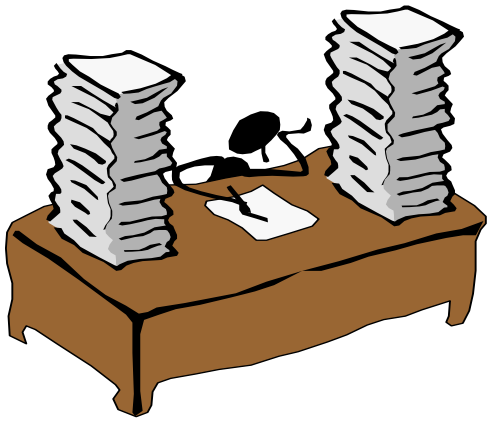
   reitittimet ja reititys

5. **Linkkikerros, lähiverkot**

   Ethernet, kytkimet

6. **Tietoturvasta**

   Uhkat, palomuuri



Sisältö kirjassa

1. Chapter 1: Computer Networks and the Internet

   What is the Internet, The Network Core, The Network Edge, Access Networks and Physical Media, Delay and Loss in Packet-Switched Networks, Protocol Layers and Their Service Models

2. Chapter 2: Application Layer

   Principles of Network Applications, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, DNS - The Internet's Directory Service, P2P File Sharing, Socket Programming

3. Chapter 3 (not included: 3.6, 3.7.2): Transport Layer

   Introduction and Transport-Layer Services, Multiplexing and Demultiplexing, Connectionless transport: UDP, Principles of Reliable Data Transfer, Connection-Oriented Transport: TCP, TCP Congestion Control

4. Chapter 4.1-4.5 (not included: 4.2, 4.4.4): Network Layer

   Introduction, What's Inside a Router?, The Internet Protocol (IP): Forwarding and Addressing in the Internet, Routing Algorithms (Link State Routing, Distance Vector Routing, Hierarchical Routing)

5. Chapter 5.1-5.6: Link Layer and Local Area Networks

   Link Layer: Introduction and Services, Error-Detection and -Correction Techniques, Multiple Access Protocols, Link-layer Addressing (MAC, ARP, DHCP), Ethernet (Frame structure, CSMA/CD), Interconnections: Hubs and Switches)

6. Chapter 8.1, 8.6-8.7: Security in Computer Networks

   What is Network Security, Attacks and Countermeasures, Access Control: Firewalls

Viikko 1

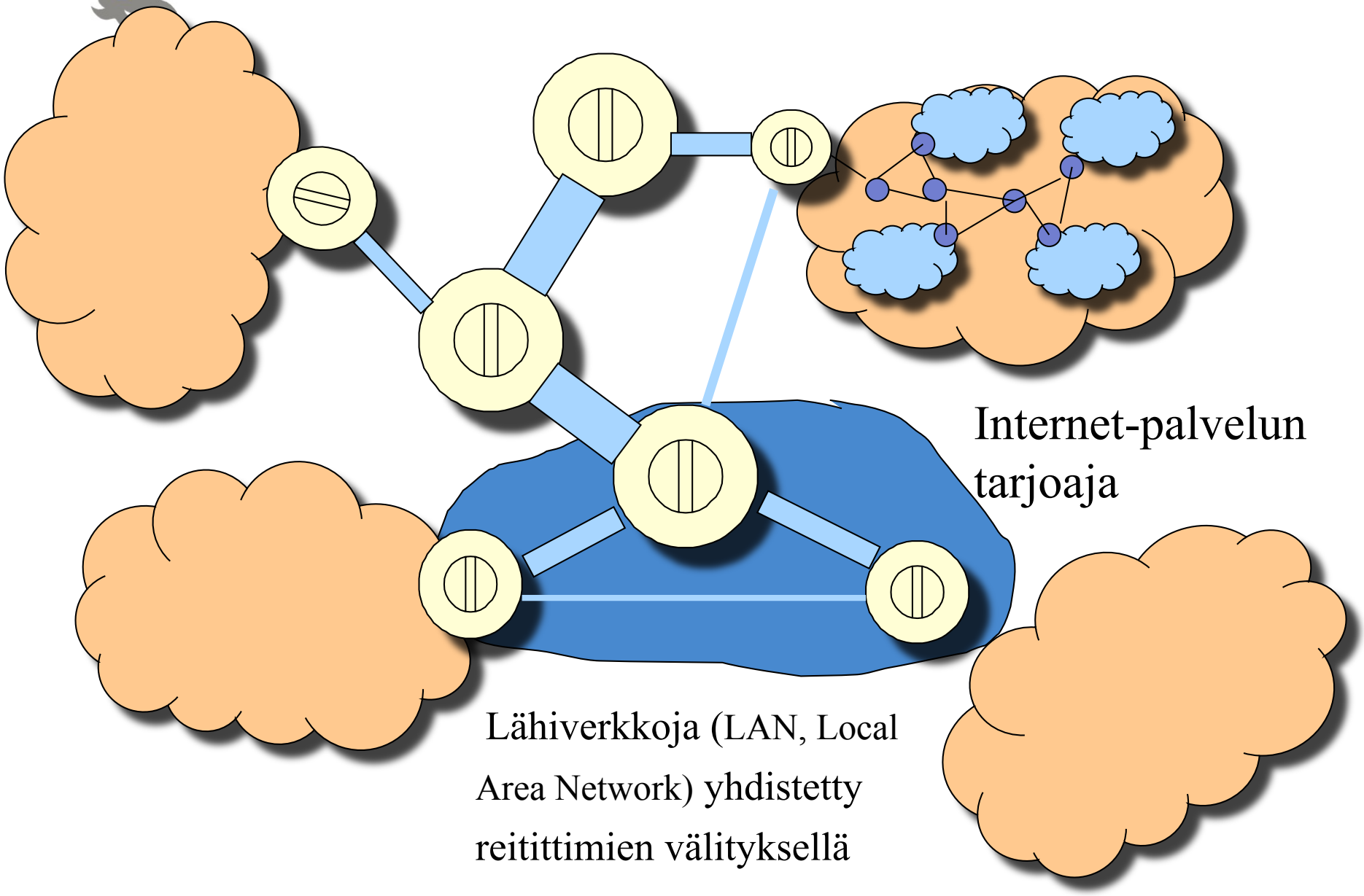
Oppimistavoitteet:
-Perusterminologiaa tutuksi
-Yleiskuva Internetistä
- rakenne
- toiminnallisuus
-Internetin protokollapino ja sen eri kerrosten tehtävät

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

Internetin rakenneosat

- **Miljoonia koneita**
  - **isäntäkoneita** (host, end system)
    - työasemia (workstation), palvelinkoneita (server)
    - mobiililaitteita, erilaisia tunnistimia, kameroita, autoja, ….
    - Suorittavat hajautettuja sovelluksia
  - **Pakettikytkimiä: siirtävät dataa paketteina eli pieninä lohkoina** (**reititin** (router), linkkitason **kytkin** (link-level switch))
    - Välittävät sovellusten sanomia koneiden välillä
- **Tietoliikennelinkkejä**
  - **erilaisia siirtomedioita**
    - Optinen kuitu, kuparijohto, koaksiaalikaapeli, elektromagneettiset aallot (radio, infrapuna, satelliitti)
    - Siirtonopeus (transmission rate) bittiä sekunnissa (bps)

Internet = verkkojen verkko (löyhää hierarkiaa)



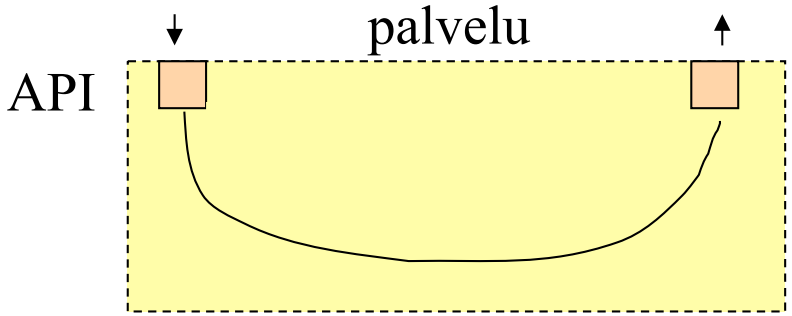
Internet

- Julkinen Internet vs. rajattu **intranet** ja **extranet**
- Päästä-päähän suunnittelumalli: tila ja toiminnot reunoilla
- Sovellukset voivat lähettää sanomia verkon välityksellä toisilleen
  - **yhteydellinen** (connection-oriented) **palvelu** / **yhteydetön** (connectionless) **palvelu**
    - Yhteydellinen: Yhteyden muodostus – yhteyden käyttö – yhteyden purku (~puhelu)
    - Yhteydetön: yhteyden käyttö (~posti)
  - **luotettava** (reliable) (= pyrkii estämään, havaitsemaan ja paikkaamaan virheet) / **epäluotettava** (unreliable) (= ’hälläväliä’)

- **Internetissä**: yhteydellinen = luotettava, yhteydetön = epäluotettava
  - **TCP**-protokolla => yhteydellinen ja luotettava
  - **UDP**-protokolla =>yhteydetön ja epäluotettava

Palvelu vs. protokolla

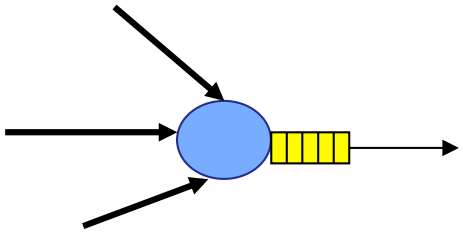
- **Palvelu**: joukko toimintoja, jotka ovat käytettävissä
  - Internetin kuljetuspalvelu, API = miten ohjelma pääsee käyttämään Internetin infrastruktuurin palveluja
  - ~ postin kuljetuspalvelu: kirje postilaatikkoon
- **Protokolla**: säännöt, jotka määräävät, miten sanomia vaihdetaan palvelun toteuttamiseksi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



Pakettikytkentä (packet switching)

- Jaa data paketeiksi ja lähetä paketti kerrallaan verkkoon
- Ei varata resursseja eikä siis reittiä etukäteen,
  - Varaus tarvittaessa (on-demand)
  - Tilastollinen kanavointi (Statistical multiplexing)

  vaan jokainen paketti reititetään erikseen => paketit voivat kulkea eri reittejä lähettäjältä vastaanottajalle
- **Etappivälitys** (store and forward) = paketti vastaanotetaan kokonaan ja vasta sitten lähetetään eteenpäin
- Koko linkin kapasiteetti siirrettävälle paketille
- Yhteenlaskettu siirtotarve voi ylittää lähtevän linjan siirtonopeuden
  - Paketti joutuu odottamaan vuoroaan reitittimen muistissa
  - **Ruuhka** (congestion) => jopa paketin häviäminen



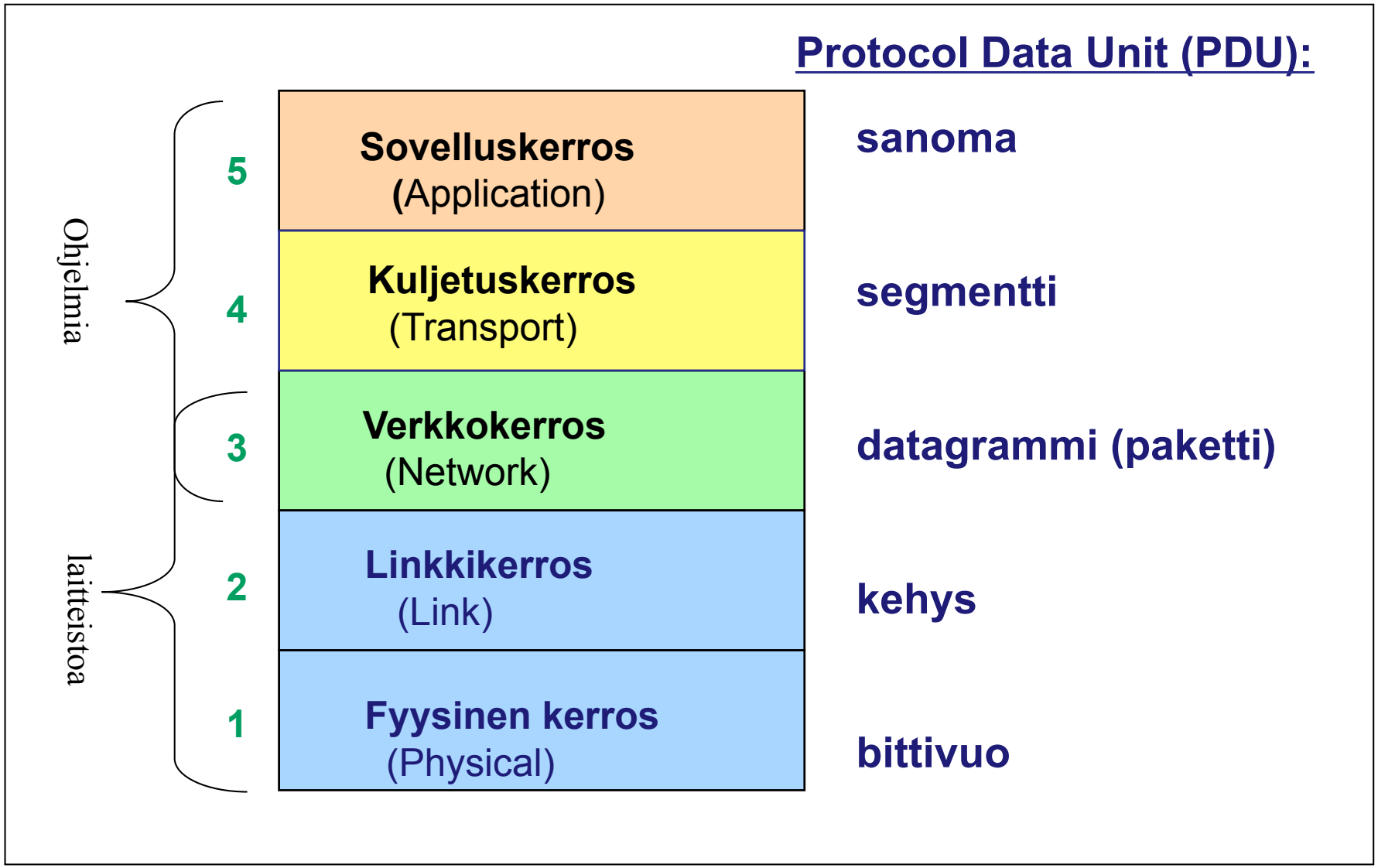
Protokollien kerrostaminen

- **Protokolla = yhteyskäytäntö**
  - Mitä sanomia, missä tilanteessa ja missä järjestyksessä lähetetään
  - Miten saatuihin sanomiin reagoidaan
  - Sanomien syntaksi ja semantiikka
- **Protokollapino = protokollien kerrosrakenne**
  - Toiminnot on jaettu kerroksiin
    - Järkevä kerrosjako
  - Alemman kerroksen toiminnot ovat ylemmän käytössä
    - Palvelu ja sen toteutus erotettu
  - Kukin protokolla toimii yhdellä kerroksella ja toteuttaa tämän kerroksen jonkin palvelun.
    - HTTP, SMTP
    - TCP, UDP
    - IP

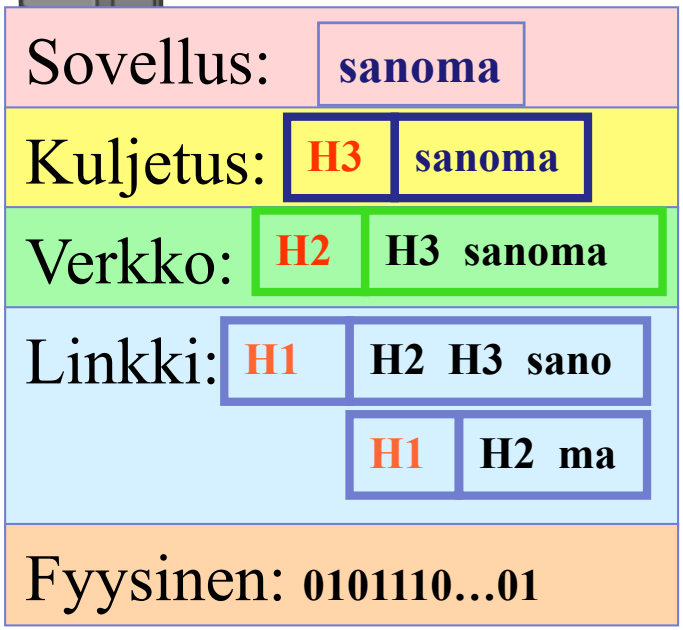
Miksi kerrosrakenne?

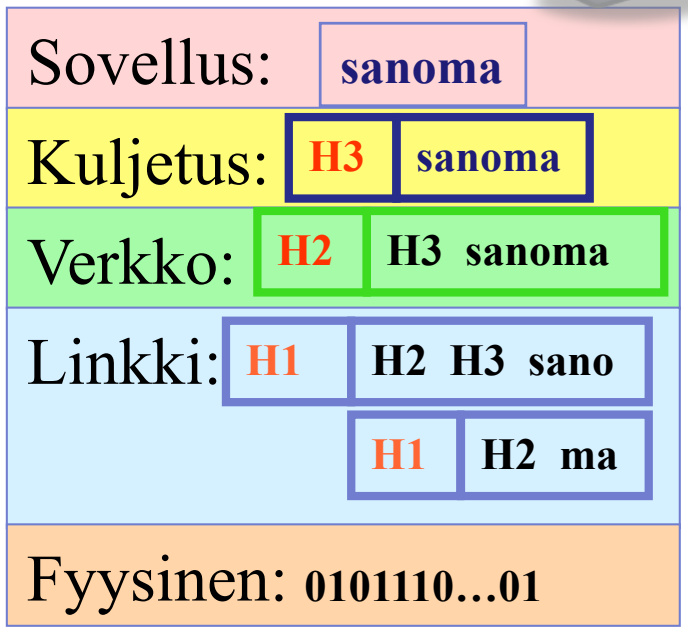
- **Monimutkaisuuden hallinta**
  - Kerroksittainen **viitemalli** (reference model) helpottaa asiakokonaisuuksiin viittaamista
- **Kullakin kerroksella omat selkeät tehtävänsä**
  - Kerroksissa toteutuu omat ’lisä’toiminnot
  - Voi käyttää olemassaolevia alemman kerroksen toimintoja
  - Kerrosten rajapinnat (interface) hyvin määriteltyjä
  - Kaksisuuntainen ’palveluluukku’: mitä tekee, kuinka on käytettävissä
- **Joustavuus**
  - Pino koottavissa erilaisista protokollista
  - Kerroksen toteutusta voi muuttaa, kunhan rajapinnat ennalllaan
- **Jos kerroksia on paljon, se voi vaikuttaa suorituskykyyn**
  - Sama työ toistamiseen, esim. virhetarkistus
  - Kutsumekanismi: kopiointia paikasta toiseen, ..

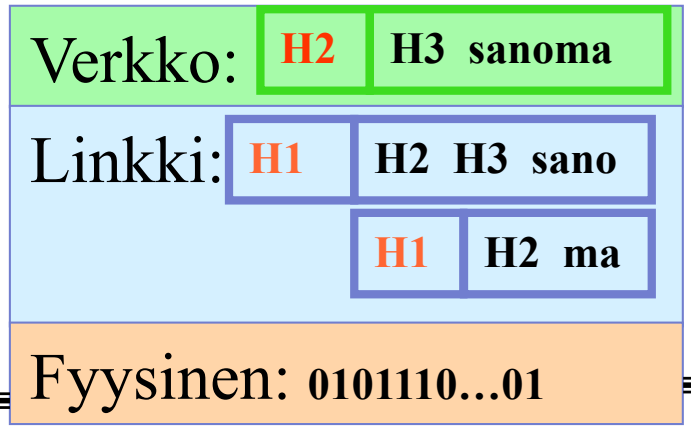
Internet-protokollapino (2)

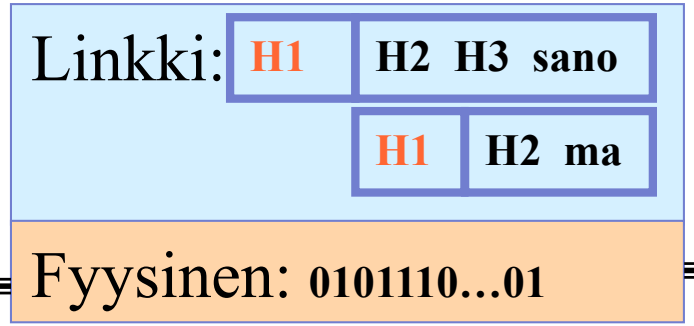


Kapselointi









Reititin

Linkkitason kytkin

Kertauskysymyksiä

- Isäntäkone vs. reititin?
- Protokolla vs. palvelu?
- Vertaisverkkomalli vs. asiakas-palvelin malli?
- Fyysinen siirtomedia?
- Piiri- ja pakettikytkentä? Hyödyt ja haitat?
- Viipeet ja pakettien katoamiset
- Internet-protokollakerrokset ja niiden tehtävät?
- Miksi kerrosrakenne?
- Mitä protokollakerroksia eri laitteissa tarvitaan?

Ks . myös kurssikirja ss. 67-69.

Tietoliikenteen perusteet

SOVELLUSKERROS

(Application layer)

Kurose, Ross: Ch 2

Viikko 2

- **Verkkosovellusten periaatteet**
- **World Wide Web ja HTTP**
- **Tiedostonsiirto ja FTP**
- **Sähköposti ja SMTP, IMAP, POP3**
- **Nimipalvelu ja DNS**
- **Vertaistoimijat (peer-to-peer)**
- **Pistoke ja sen käyttö**

Oppimistavoitteet:

- Osata selittää asiakas-palvelija –malliin perustuvien verkkosovellusten toimintaperiaatteet

-Tuntea sovellusprotokollien syntaksia ja semantiikkaa

- Osata selittää nimipalvelun, www:n ja sähköpostin toimintaideat

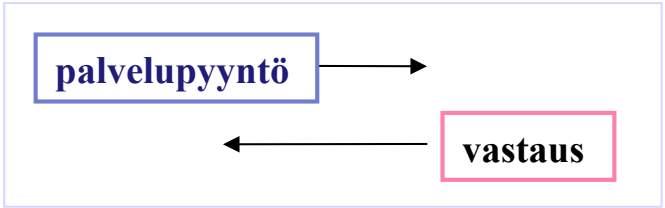
-Tunnistaa pistokkeiden käytön periaatteet

Sovellusarkkitehtuuri

Asiakas-palvelija-malli (esim. selain ja www-palve

- Aina toiminnassa oleva palvelinohjelma, jolla kiinteä, tunnettu IP-osoite
- Asiakasohjelmat ottavat yhteyttä palvelimeen ja pyytävät siltä palvelua

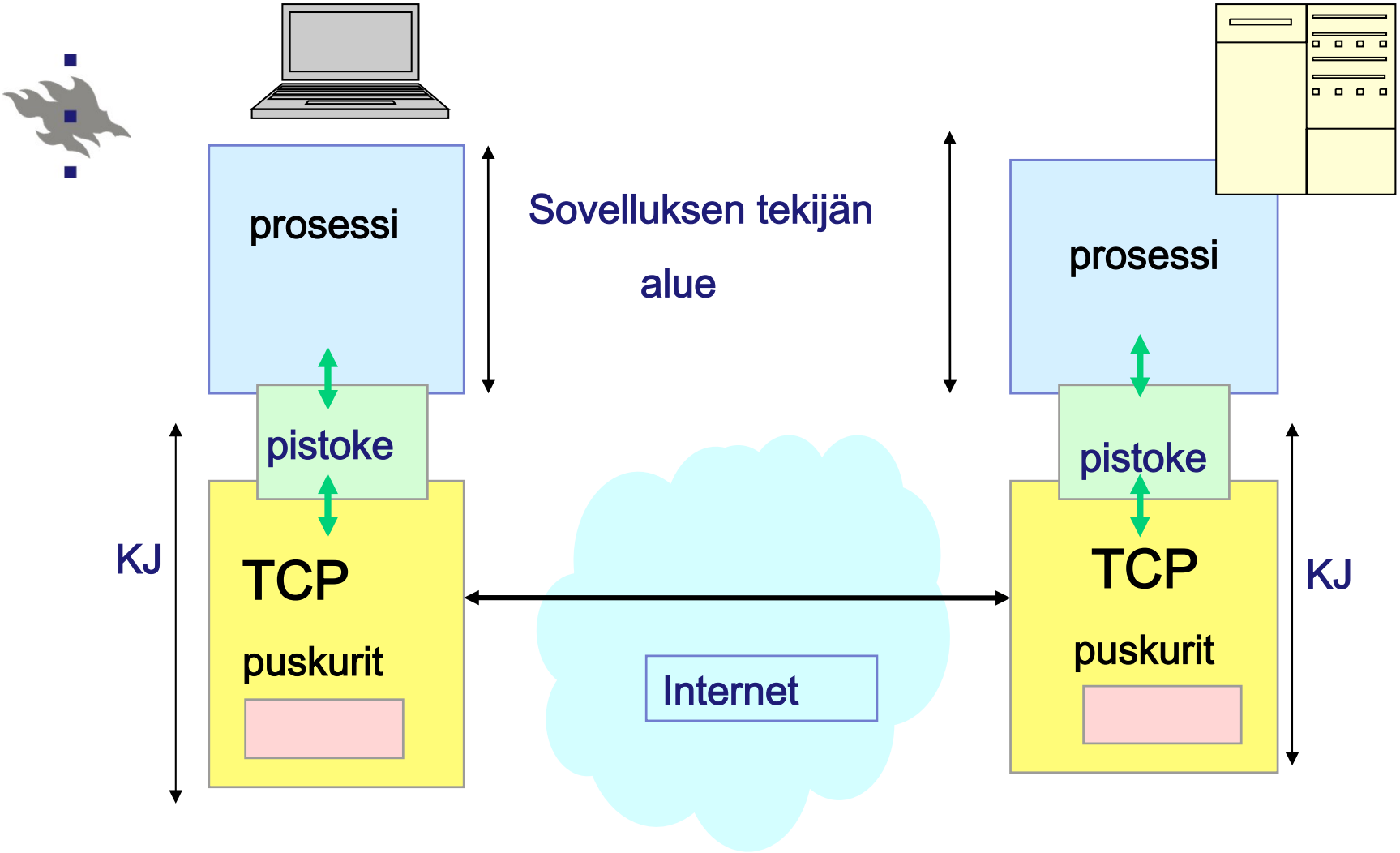
Google, e-Bay, Facebook, YouTube, Amazon, ..



Vertaistoimijamalli (esim. BitTorrent, eMule, Skype)

- Vertaisisännät kommunikoivat suoraan keskenään
- Ei tarvitse olla aina toiminnassa, IP-osoite voi muuttua
- Jokainen toimii sekä palvelijana että asiakkaana

Hybridimalli (esim. Napster, pikaviestimet)



KJ = käyttöjärjestelmä

Prosessien kommunikointi TCP-pistokkeita käyttäen

Osoittaminen

- **Sanomissa oltava lähettäjän ja vastaanottajan IP-osoite ja porttinumero**

- **IP-osoite → oikea kone**

  www.iana.org

  koneen (verkkokortin) yksilöivä 32-bittinen tunniste
  osoitteen verkko-osa yksilöi verkon
  osoitteen koneosa yksilöi koneen verkossa

- **porttinumero → oikea prosessi**

  Yleisillä palveluilla standardoidut tunnetut porttinumerot:
  - www-palvelin kuuntelee porttia 80,
  - Postipalvelin kuuntelee porttia 25

  KJ osaa liittää porttinumeron prosessiin

DNS (Domain Name System)

- **Hakemistopalvelu ja sovelluskerroksen protokolla**
  - **Isännät ja nimipalvelimet käyttävät**
  - **Käyttää itse UDP-kuljetuspalvelua DNS-sanomien kuljettamiseen**
- **Nimien muuttaminen IP-osoitteiksi (ja päinvastoin)**
  - **Posix: gethostbyname(hydra.cs.helsinki.fi) 218.214.4.29**
  - **Kone = hydra =29, verkko= cs.helsinki.fi = 218.214.4.0**
- **Sallii aliasnimet, palvelijan replikoinnin**
  - **Esim. WWW.cs.helsinki.fi ja cs.helsinki.fi ovat aliasnimiä**
  - **Esim. www-palvelijaan voi liittyä useita IP-osoitteita, rotaatio**
- **Hajautettu, hierarkinen tietokanta (hakemisto)**
  - **Toteutettu useiden replikoitujen nimipalvelimien yhteistyönä**
  - **skaalautuvuus, kuormantasaus, ylläpito, vikasietoisuus, ..**
  - **Jos oma nimipalvelija ei tunne, se kysyy muilta.**

IP-nimen selvittäminen

- sovellusohjelma kutsuu kirjastorutiinia parametrina nimi merkkijonona
  - esim Unix:ssa **gethostbyname()**
- kirjastorutiini lähettää UDP-datasähkeen paikalliselle DNS-palvelimelle, joka etsii nimeä vastaavan IP-osoitteen ja palauttaa sen kirjastorutiinille
  - etsinnässä tarvitaan usein monien palvelimien yhteistyötä
  - Iteratiivinen kysely / rekursiivinen kysely
  - Välimuistin käyttö

Kertauskysymyksiä

- Asiakas-palvelija-malli? Vertaisverkkomalli?
- Kuinka asiakas löytää palvelimen?
- Miten KJ osaa antaa bitit oikealle sovellukselle?
- Miten koneen nimestä saadaan selville sen IP-osoite?
- Miten HTTP-protokolla toimii?
- Miksi SMTP ei riitä, vaan tarvitaan POP3 tai IMAP?
- Mitä hyötyä on proxy-palvelimesta?
- Miksi käytetään evästeitä?
- Mikä on pistoke ja missä sitä käytetään?

Ks. myös kurssikirja s.170.♪

Tietoliikenteen perusteet

Kuljetuskerros

Kurose, Ross: Ch 3

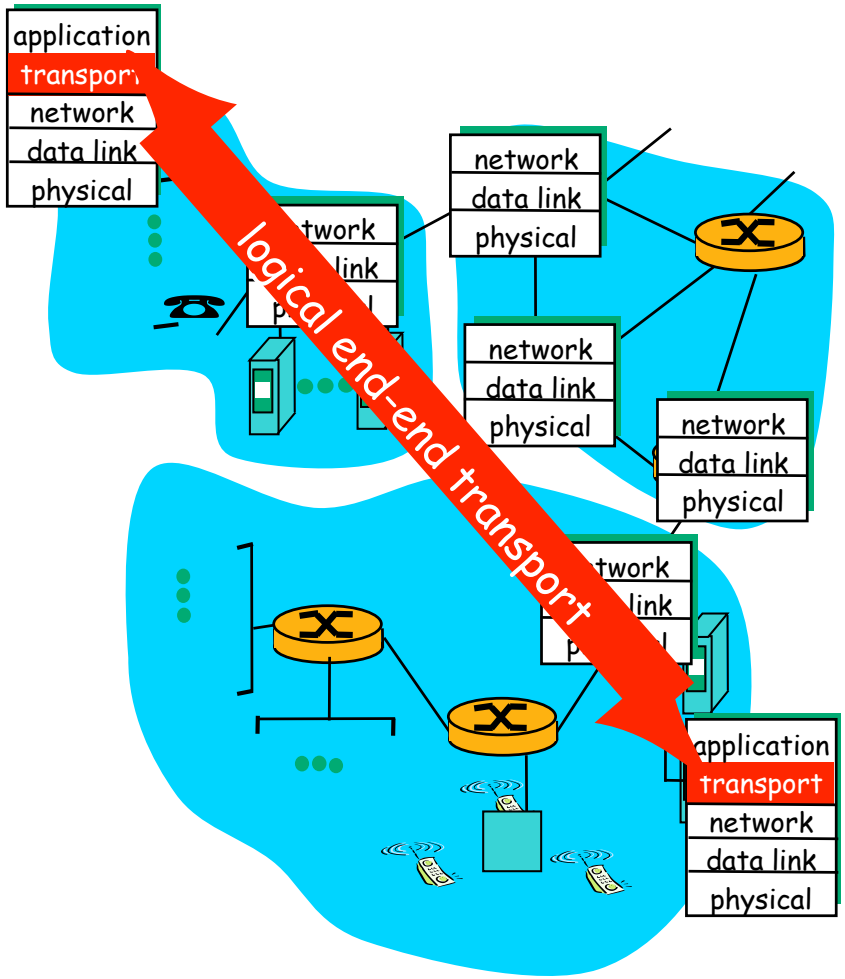
Viikko 3

- **Kuljetuspalvelut**
- **Yhteydetön kuljetuspalvelu, UDP**
- **Luotettavan kuljetuspalvelun periaatteet**
- **Yhteydellinen kuljetuspalvelu, TCP**
- **Ruuhkanhallinta TCP:ssä**

Oppimistavoitteet:
- Tuntea Internetin kuljetusprotokollien (UDP/TCP) toiminnallisuus ja periaatteet
-Osata luotettavan kuljetuspalvelun ja vuonvalvonnan periaatteet ja toteutukset
-Osata TCP-ruuhkanhallinnan

Kuljetuskerros

- **Tarjoaa kuljetuspalvelun prosessien välille**
- **Vain isäntäkoneissa**
  - **Lähetys**: Pilko sovelluskerroksen sanoma pienemmiksi segmenteiksi, jotka verkkokerros toimittaa perille.
  - **Vastaanotto**: Kokoa segmentit sanomaksi, jonka sovellus lukee.
- **Verkkokerros reitittää koneesta koneelle**
- **Segmentin koko s.e. verkkokerros pystyisi välittämään sellaisena**



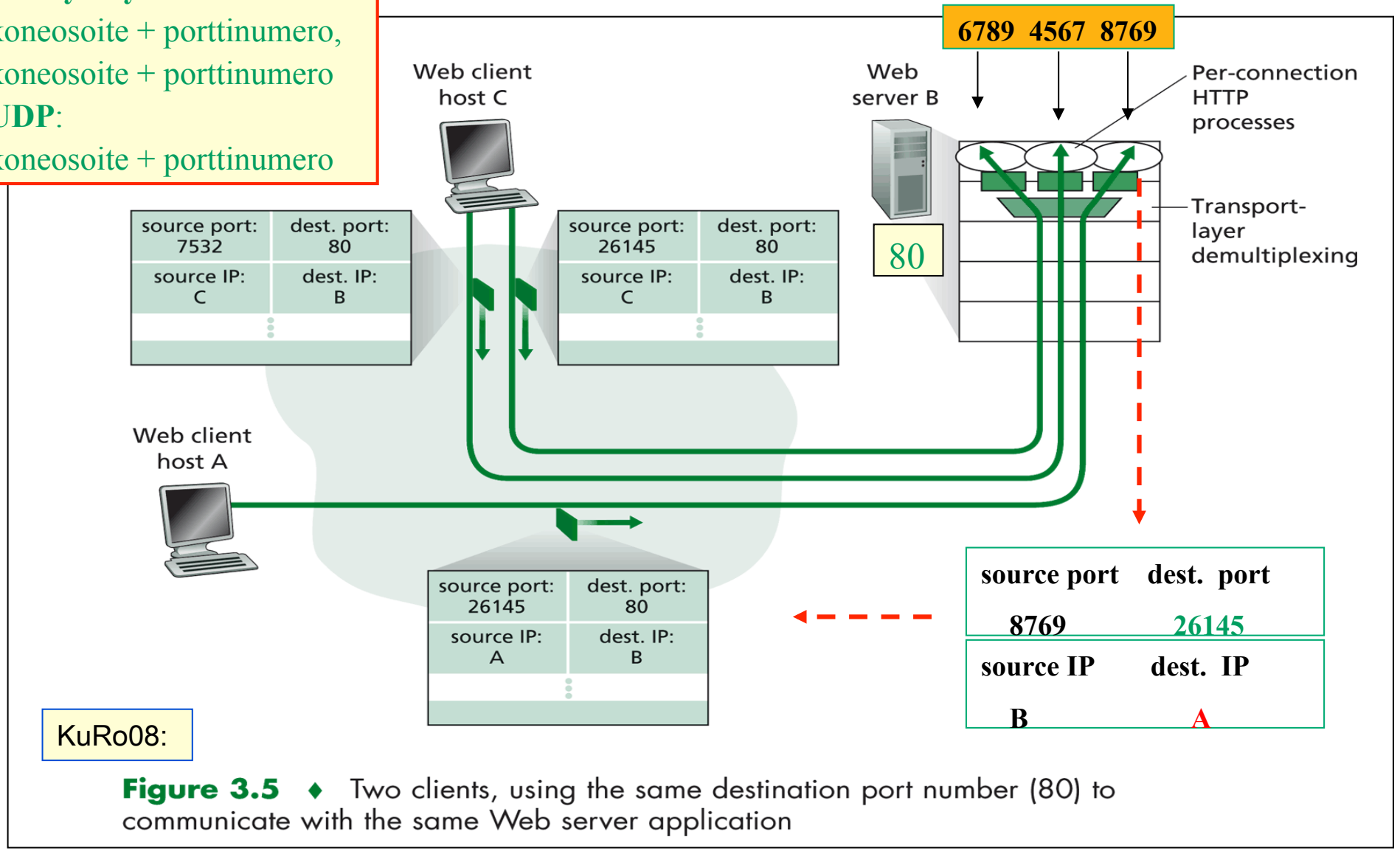
KuRo08: Fig 3.1

Internetin kuljetusprotokollat

- **TCP: luotettava, järjestyksen säilyttävä tavujen kuljetuspalvelu**
  - **Virheenvalvonta** (error control): Huomaa ja korjaa virheet, hylkää kaksoiskappaleet
  - **Vuonvalvonta** (flow control): Älä ylikuormita vastaanottajaa
  - **Ruuhkanhallinta** (congestion control): Älä ylikuormita verkkoa
  - **Yhteyden muodostaminen ja purku**
- UDP: Ei-luotettava, ei-järjestyksen säilyttävä sanomien kuljetuspalvelu
  - Välittää vain sanomia, ei pyri mitenkään parantamaan verkkokerroksen tarjoamaa palvelun laatua
  - Luotettavuus jää sovelluskerroksen hoidettavaksi
- **Kumpikaan kuljetuspalvelu ei anna takuita viiveelle tai siirtonopeudelle** (“best effort”)

Kaksi www-asiakasta ja palvelija

TCP-yhteys:
koneosoite + porttinumero,
koneosoite + porttinumero
UDP:
koneosoite + porttinumero



KuRo08:

Figure 3.5 ♦ Two clients, using the same destination port number (80) to communicate with the same Web server application

Kuljetusprotokolla TCP

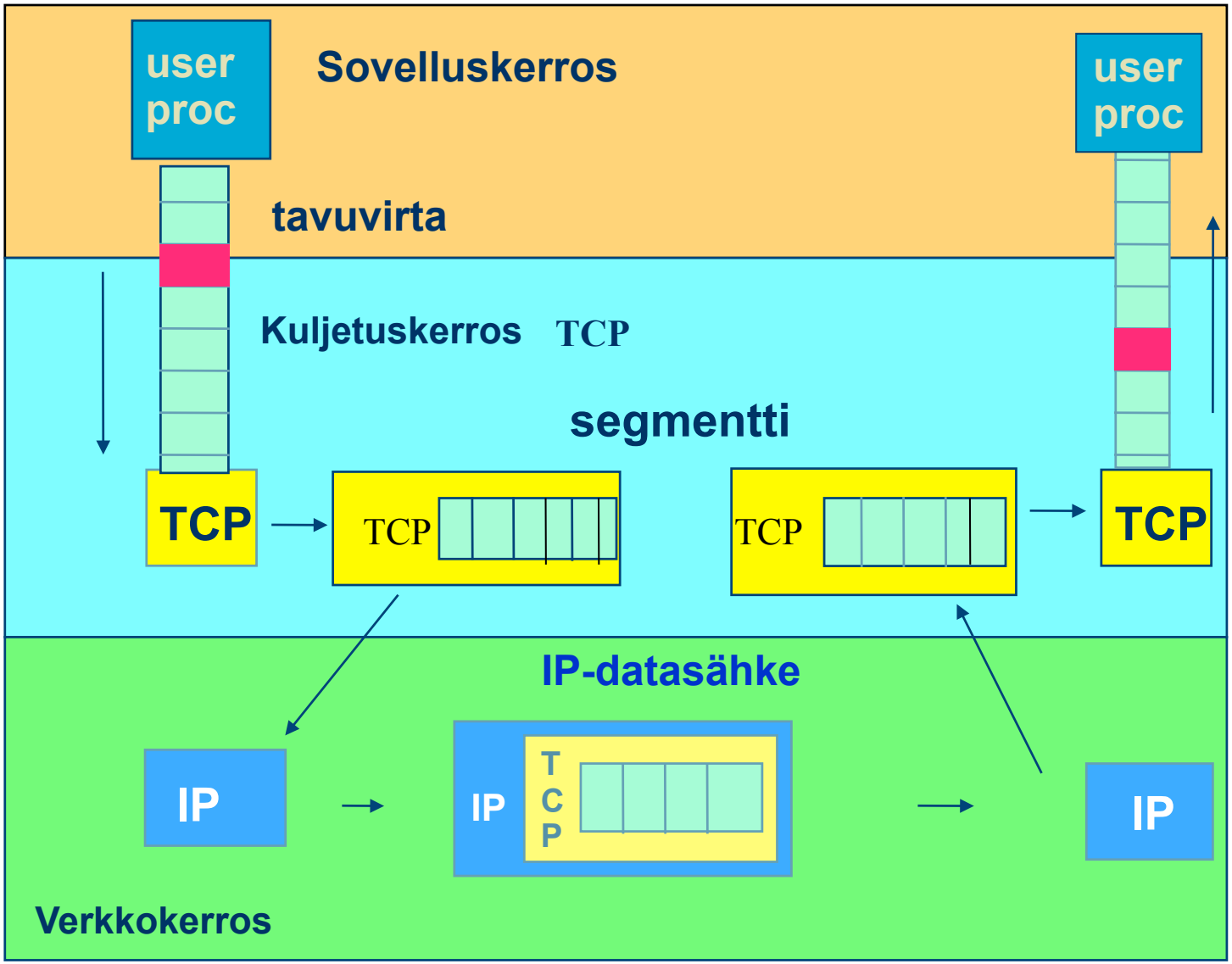
- **TCP** (Transmission Control Protocol) [RFC 793]
  - **Yhteydellinen palvelu** (connection-oriented)
    - Yhteyden muodostus ennen datan siirtoa (handshaking)
    - Kaksisuuntainen TCP-yhteys (full-duplex)
    - Yhteyden purku (shutdown)
  - **Luotettava kuljetuspalvelu**
    - Järjestyksen säilyttävä tavuvirta sovellukselle
    - segmenttinumerot, kuittaukset, uudelleenlähetykset
  - **Vuonvalvonta** (flow control)
    - Lähettäjä hiljentää vauhtia, jos **vastaanottaja** ei ehdi käsitellä
  - **Ruuhkanvalvonta** (congestion control)
    - Lähettäjä hiljentää vauhtia, jos **reititttimet** eivät ehdi käsitellä

Yhteenveto menetelmistä

- Ks. KuRo08 Table 3.1

- Tarkistussumma
- Ajastin
- Järjestysnumero
  - Uudelleenlähetys vai uusi paketti
- Kuittaukset
  - Positiiviset ACK, tuplakuittaukset
  - Negatiiviset NAK
- Ikkunat, liukuhihnoitus

TCP: prosessilta prosessille -tavuvirta



TCP-protokolla

- **Päästä-päähän kuljetuspalvelu**
  - Yksi lähettäjä, yksi vastaanottaja
  - Reitittimet eivät ole kiinnostuneita kuljetustason protokollasta
- **Yhteydellinen** (connection-oriented)
  - Yhteydenmuodostus
  - Isäntäkoneissa: puskuritila, ikkunakoko, tavunumerointi
  - Yhteyden purku
- **Kaksisuuntainen** (full duplex)
  - Yksi yhteys, jossa yhtä aikaa liikennettä molempiin suuntiin
- **Luotettava, järjestyksen säilyttävä tavuvirta**
  - Ei sanomarajoja
  - Tavunumerointi
  - Kumulatiiviset kuittaukset

Yhteyden muodostus

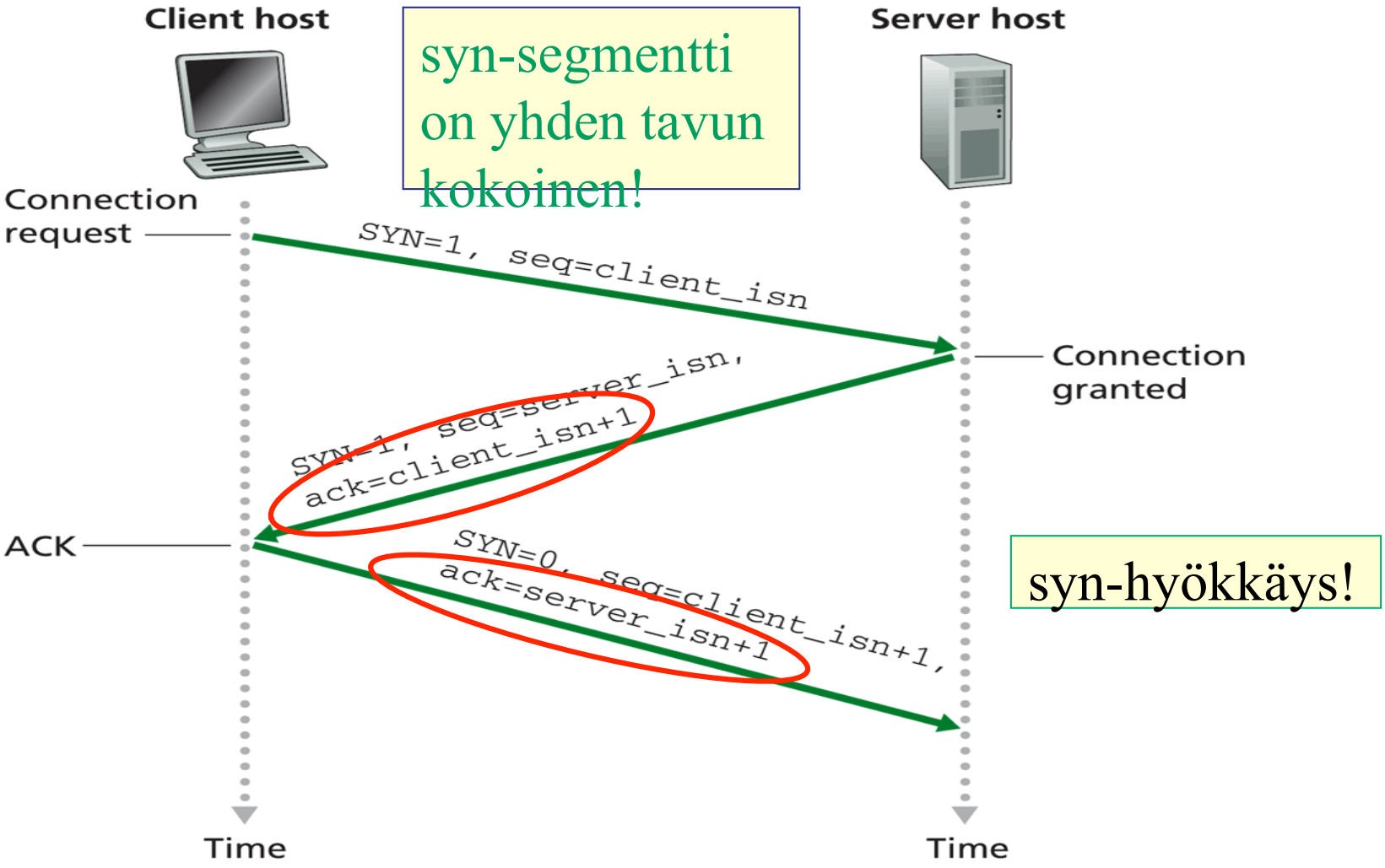


Figure 3.39 ◆ TCP three-way handshake: segment exchange

Ruuhkaikkuna (congestion window)

- Internetin hetkellinen kuormitus vaihtelee
- Ruuhkaikkuna
  - Paljonko lähettäjä saa tietyllä hetkellä kuormittaa verkkoa
  - Paljonko lähettäjällä saa olla kuittaamattomia segmenttejä
- Lähettäjän pääteltävä itse sopiva ikkunankoko
  - **Jos uudelleenlähetysajastin laukeaa, on ruuhkaa**
  - **Jos kuittaukset tulevat tasaisesti, ei ole ruuhkaa**
- Dynaaminen ruuhkaikkunan koko
  - Kasvata ikkunaa ensin nopeasti, kunnes törmätään ruuhkaan
  - Pienennä sitten ikkunaa reilusti ja kasvata varovasti
- Lähetysikkunan raja voi tulla vastaan ensin
  - Kuittamatta saa olla **min(lähetysikkuna, ruuhkaikkuna)**

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

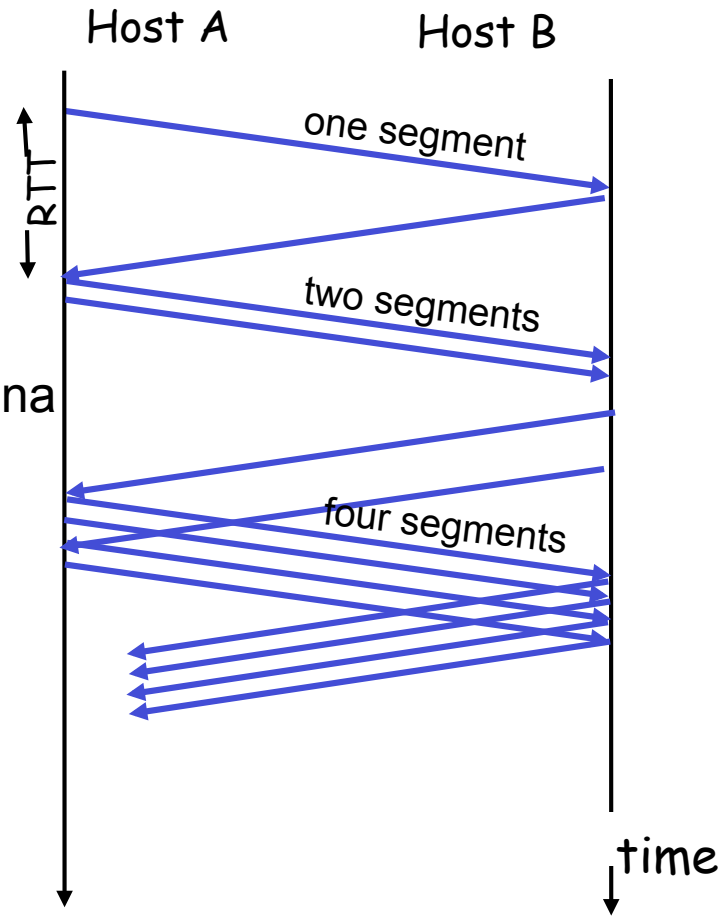
- **Aluksi ruuhkaikkuna = yksi segmentti**
  - Alussa hidas siirtonopeus= MSS/RTT
- **Kukin kuittaus kasvattaa yhdellä ruuhkaikkunan kokoa**
  - Eksponentiaalinen kasvu
  - Ikkuna kaksinkertaistuu yhden RTT:n aikana

hidas aloitus

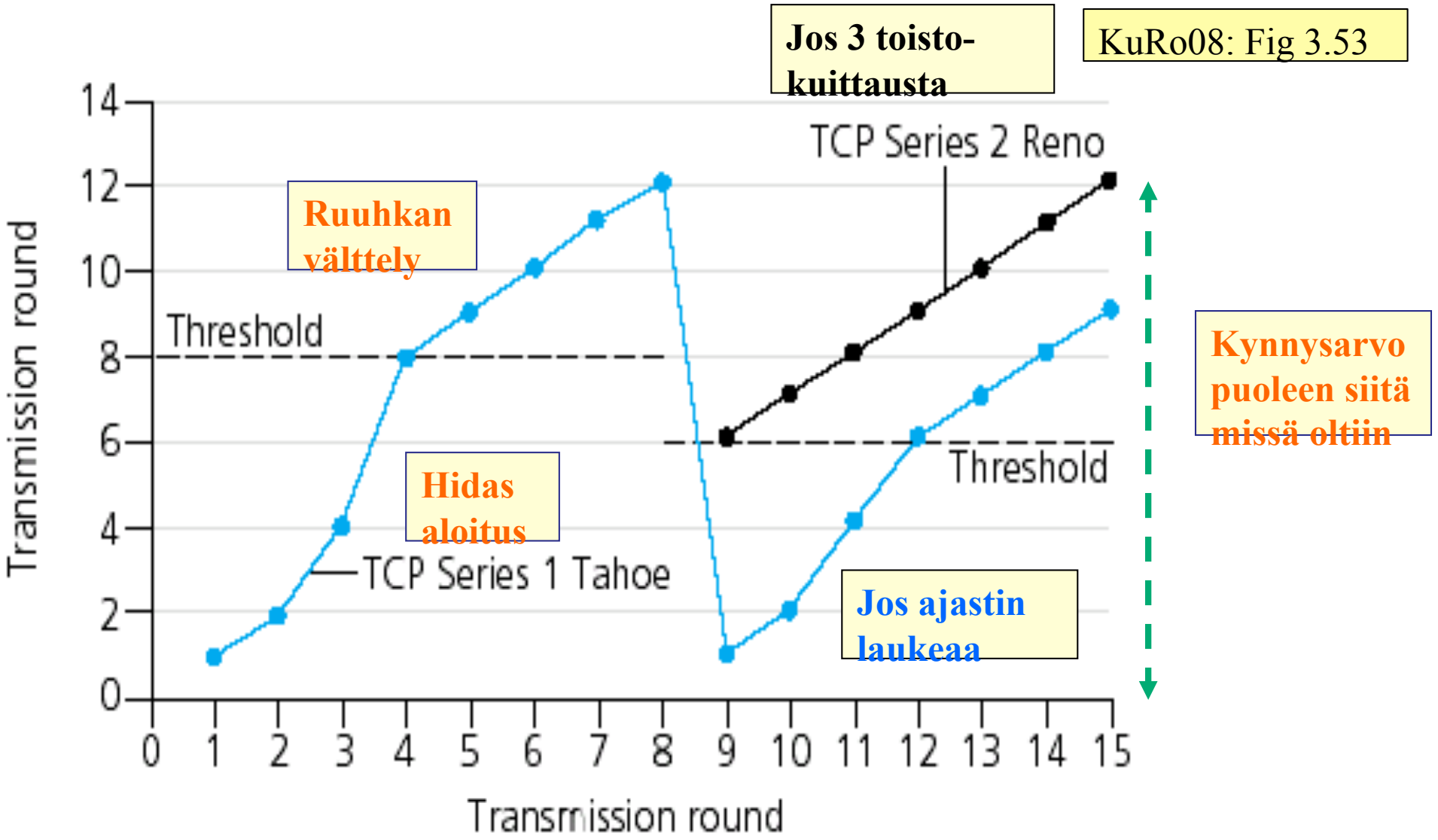
- **Jos uudelleenlähetys, ruuhkaikkunan kooksi 1 segmentti**
  - Multiplicative decrease
- **Sen jälkeen kasvata ikkunaa yksi segmentti/RTT**
  - Lineaarinen kasvu (Additive increase)
  - Ruuhkan välttely (congestion avoidance)

ruuhkanvälttely

- Siirtonopeus = CognWin / RTT tavua/sek



TCP Tahoe vs. TCP Reno



Kertauskysymyksiä

- TCP vs. UDP?
- Miten tehdä kuljetuspalvelusta luotettava?
- Keskeisimmät TCP:n otsakkeessa olevat tiedot?
- Mitä tapahtuu TCP:n yhteydenmuodostuksessa?
- Vuonvalvonta?
- Ruuhkanhallinta?

ks. Kurssikirja s. 293

Tietoliikenteen perusteet

Verkkokerros

Kurose, Ross: Ch 4.1- 4.42 ja 4.5

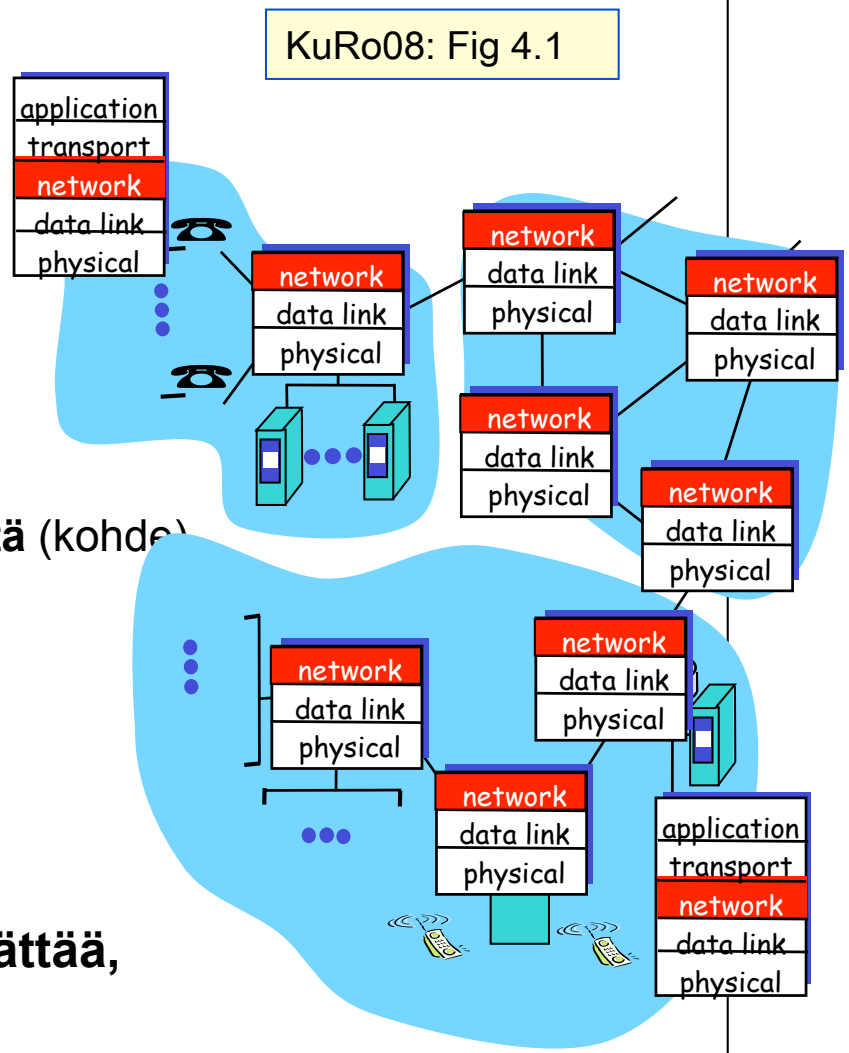
Viikko 4

- **Verkkokerros**
- **Reititin**
- **IP-protokolla**
- **Reititysalgoritmit**

Oppimistavoitteet:
- Osata selittää, kuinka IP-paketteja välitetään verkossa
- Tietää, mitä tietoja sisältyy IP-pakettiin (ja miksi)
- Osata selittää reitittimien rakenne ja toiminta
- Osata kuvailla, kuinka reitittimet kokoavat reititystietonsa
= linkkitila- ja etäisyysvektorialgoritmien toimintaideat

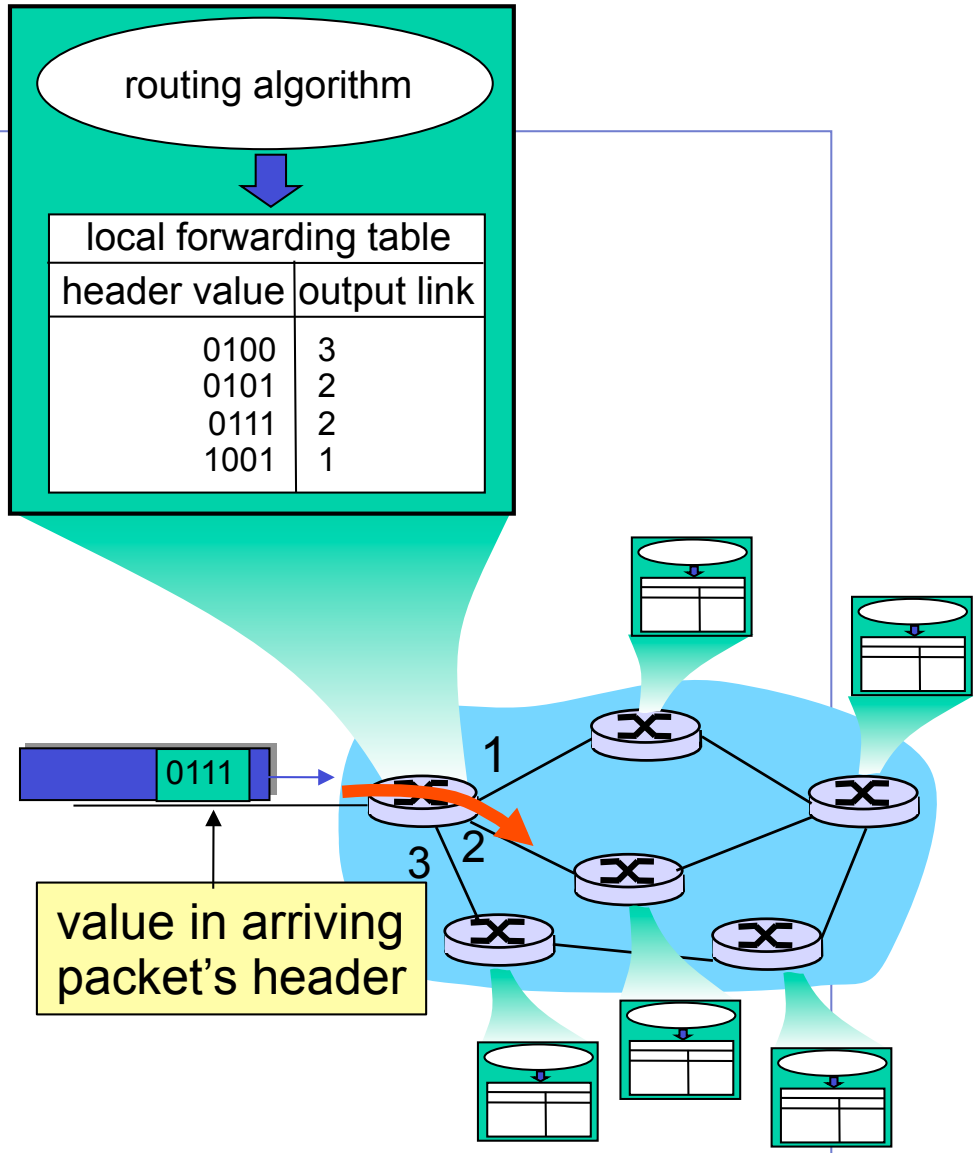
Verkkokerros

- **Toimittaa kuljetuskerroksen segmentit vastaanottajalle**
  - **Lähetys**
    - **Luo segmenteistä verkkokerroksen IP-paketteja**
    - **Lisää otsaketietoja: mm. IP-osoitteet**
  - **Pakettien kulku verkossa**
    - **Isäntä** (lähde) **– reititin - ...- reititin – isäntä** (kohde)
  - **Vastaanotto**
    - **Poista otsake**
    - **Anna segmentti kuljetuskerrokselle**
- **Toimii etenkin reitityksessä**
  - **Reititin tutkii IP-paketin otsakkeen ja päättää, linkkiin se lähetetään seuraavaksi**



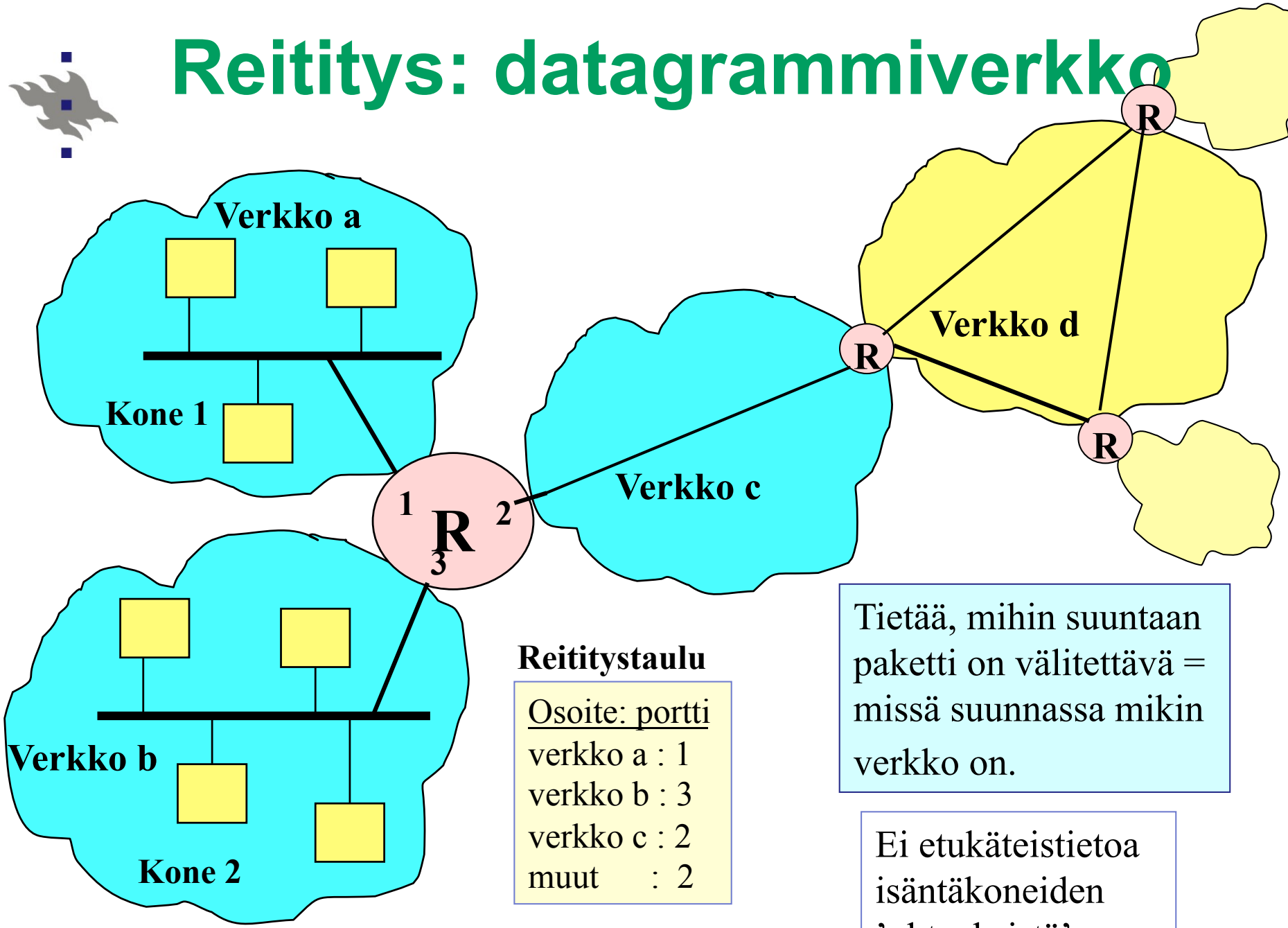
Reititin

- Ohjaa paketin reitittimen sisääntulolinkistä johonkin ulosmenolinkkiin (forwarding)
  - Katsoo reititystaulusta minne
- Kertoo muille reitittimille reitittämiseen liittyviä tietoja (routing)
  - Reittien selvittäminen
  - Reititystaulun ylläpito
  - Oma protokolla tätä varten (reititysalgoritmi, reititysprotokolla)
  - Käsin konfigurointi on hankalaa
- Piirikytkentäisessä verkossa myös yhteydenmuodostus ja purku



KuRo05: Fig 4.2

Reititys: datagrammiverkko



Reititystaulu

Osoite: portti	
verkko a	: 1
verkko b	: 3
verkko c	: 2
muut	: 2

Tietää, mihin suuntaan paketti on välitettävä = missä suunnassa mikin verkko on.

Ei etukäteistietoa isäntäkoneiden ’yhteyksistä’

CIDR: Classless InterDomain Routing

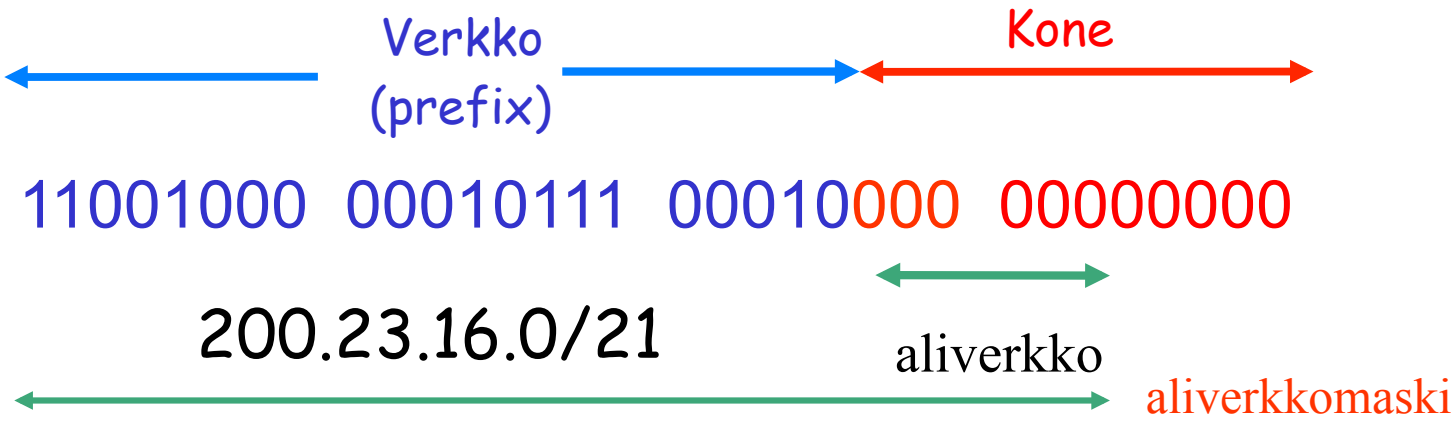
- Verkko-osa voi olla minkä tahansa kokoinen

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

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

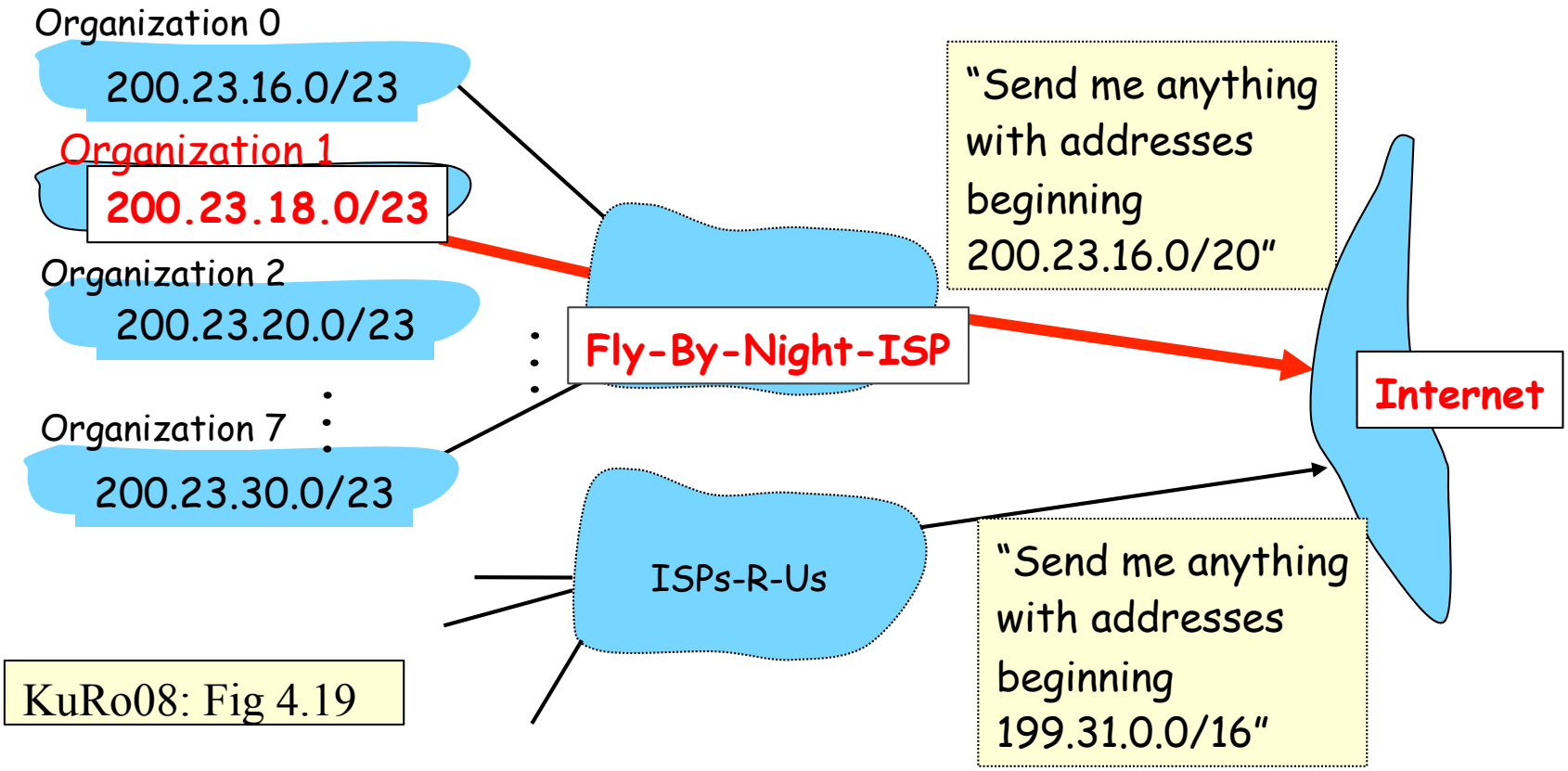
  x ilmoittaa verkko-osan bittien lukumäärän (prefix)

  Esim. Organisaatio, jolla 2000 konetta varaa 2048 = 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



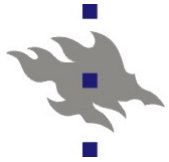
KuRo08: Fig 4.19

Koneen IP-osoite

- Koneen IP-osoite konfiguroidaan (usein) käsin koneelle
- Tai yhä useammin saadaan automaattiseti käyttäen **DHCP**:tä (Dynamic Host Configuration Protocol)
  - Eri osoite eri kerroilla tai pysyvämpi osoite
  - DHCP-palvelija vastaa
    - antaa koneen käyttöön IP-osoitteen (rajallinen elinaika)
    - antaa DNS-tiedot
    - yms
  - Palvelun tarjoaja: pienempi numeromäärä riittää
  - WLAN
    - “wash-and-go”, “plug-and-play”

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)



Reititystiedon kerääminen

- Reitin saa tietoja seuraavasti:
  - Linkkikerros tarjoaa yhteyden naapureihin
    - MAC-osoite → IP-osoite
  - Naapurit havaitaan saapuvista kehyksistä ja paketeista (broadcast, unicast)
- Naapurit kertovat omasta reititystaulustaan
  - Linkkitila: kaikki tiedot
  - Etäisyysvektori: lyhimmät etäisyydet kohteisiin naapurien kautta
- Tietojen päivitys: reaktiivinen tai ajastettu

1) Linkkitila: Dijkstran algoritmi

- **Aluksi kaikilla reitittimillä on tiedossa verkon rakenne ja kaikkien linkkien kustannukset**
  - Kaikki reitittimet lähettävät tietonsa naapureistaan ja linkkikustannuksista naapureihin (mitatut /havaitut) joko kaikille muille tai jollekin keskussolmulle, joka välittää tiedon muille
- **Reititin laskee Dijkstran algoritmilla edullisimman kustannuksen kaikkiin muihin kohteisiin**
  - Kokoaa näistä oman reititystaulunsa
- **Merkinnät**

  C(x,y) linkin x,y kustannus; jos eivät naapureita = ∞

  D(v) toistaiseksi edullisin kustannus solmuun v

  p(v) solmun v edeltäjä reitillä

  N = solmujen joukko, N' = jo käsiteltyjen solmujen joukko

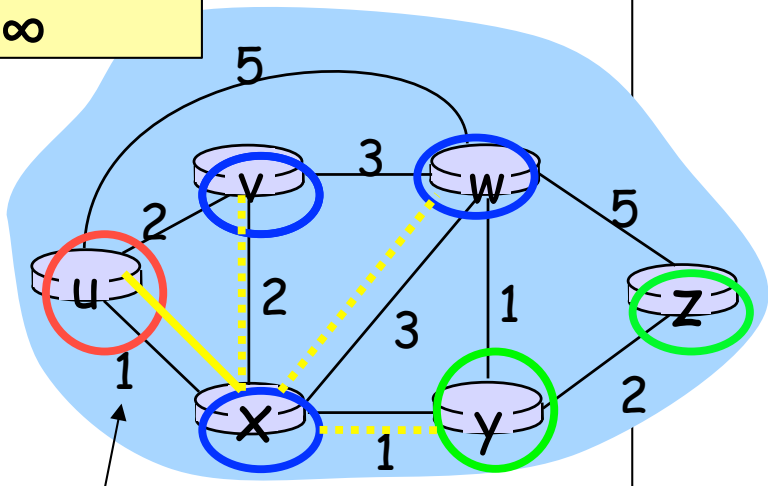
Dijkstran algoritmi

D(v)=2, D(w) = 5, **D(x)=1**

D(y) = ∞ , D(z)= ∞

```
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) = ∞
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'
```

1. Eli jos u:n vieressä



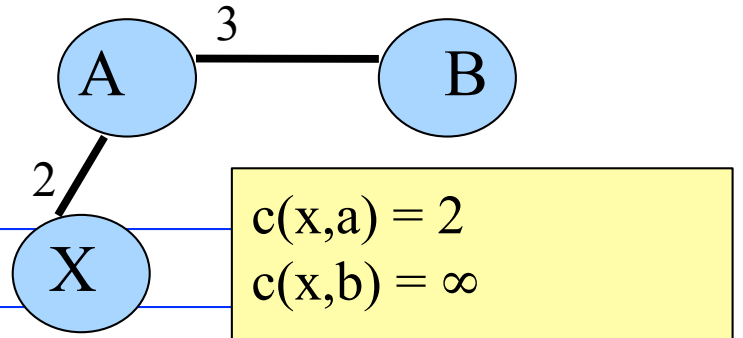
2. Aina valitaan käsittelemätön jonka etäisyys u:sta on pienin

3. Päivitetään etäisyys w:n naapureille joita ei vielä ole käsitelty

2) Etäisyysvektorireititys (distance vector)

- **Arpanet-verkon alkuperäinen reititysalgoritmi**
  - Käytössä useissa Internetin reititysprotokollissa RIP, BGP, Novell IPX, ISO IDRP
- **Interaktiivinen, hajautettu ja asykroninen**
- **Tiedot tarkentuvat asteittain, iteratiivisesti**
  - Tietyin väliajoin, linkin tilan vaihtuessa, naapurin tietojen muuttuessa, ..
- **Kukin solmu laskee itsenäisesti, mutta saa tietoa naapureiltaan**
  - Tietää / arvioi kustannuksen omiin naapureihinsa
  - Kuulee naapureiden kustannukset muihin kohdesolmuihin, jotka nämä puolestaan ovat kuulleet omilta naapureiltaan
  - Valitsee kullekin kohdesolmulle kuulemansa edullisimman reitin

Etäisyysvektorireititys



Merkinnät

c(x,v) kustannus solmusta x naapuriin v,
jos v ei ole x:n naapuri, c(x,v) = ∞

$D_x(y)$ edullisimman x:stä y:hyn johtavan reitin kustannus

- **Kukin solmu ylläpitää omaa etäisyysvektoria kaikkiin tuntemiinsa kohteisiin $D_x = [D_x(y): y \epsilon N]$**
  - **edullisin tiedetty kustannus solmusta x kuhunkin solmuun y**
- **Sekä saa naapureiltaan niiden etäisyysvektorit**

  $D_v(y) = [D_v(y): y \epsilon N]$ = Naapurin v tiedot edullisimmista kustannuksista kuhunkin solmuun y
- **$D_x(y) = \min \{c(x,v) + D_v(y)\}$** (Bellman-Ford)
  - **Kustannus solmusta x naapurisolmuun v ja sieltä solmuun y**
  - **Reittejä useita (eri naapureiden kautta); valitaan edullisin eli pienin kustannus**

3) Hierarkkinen reititys

- Reitityksen skaalautuus?
  - Isossa verkossa runsaasti reitittimiä
    - Kaikki eivät voi tuntea kaikkia muita
    - Reititystaulut suuria, reittien laskeminen raskasta
    - Reititystietojen vaihtaminen kuluttaa linjakapasiteettiä
- Autonomiset järjestelmät AS (Autonomous Systems)
  - Internet ~ verkkojen verkko
- Intra-AS routing
  - Kukin verkko päättää itse sisäisestä reitityksestään
  - RIP, OSPF
- Inter-As routing
  - AS:t ilmoittelevat toisilleen, mihin muihin AS:iin niistä pääsee
  - BGP (Border Gateway Protocol)

Kertauskysymyksiä

- Keskeisimmät IP-osakkeen tiedot?
- Paketin paloittelu
- Millainen on IP-osoite?
- Reitittimen arkkitehtuuri?
- Longest prefix match?
- CIDR
- NAT:n toiminta
- Miten reititin saa reititystiedot?
- Linkkitila-algoritmi, Dijkstran algoritmi
- Etäisyysvektorialgoritmi, count-to-infinity-ongelma

ks. kurssikirja s. 417

Tietoliikenteen perusteet

Linkkikerros

Kurose, Ross: Ch 5.1- 5.6

Viikko 5

- **Linkkikerroksen tehtävät**
- **Virheiden havaitseminen ja korjaaminen**
- **Yhteiskäyttöisen kanavan varaus**
- **Osoittaminen linkkikerroksella**
- **Ethernet**
- **Keskitin ja kytkin**

Oppimistavoitteet:

- Osata selittää linkkikerroksen toiminnallisuus (MAC-osoitteet, bittivirheiden havaitseminen) ja ARP-protokollan käyttö.
- Osata selittää yhteiskäyttöisen siirtokanavan varaus ja käyttö
- Osata selittää, kuinka koneita voi yhdistellä lähiverkoiksi
- Osata selittää reitittimen, kytkimen ja keskittimen erot

Linkkikerros

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

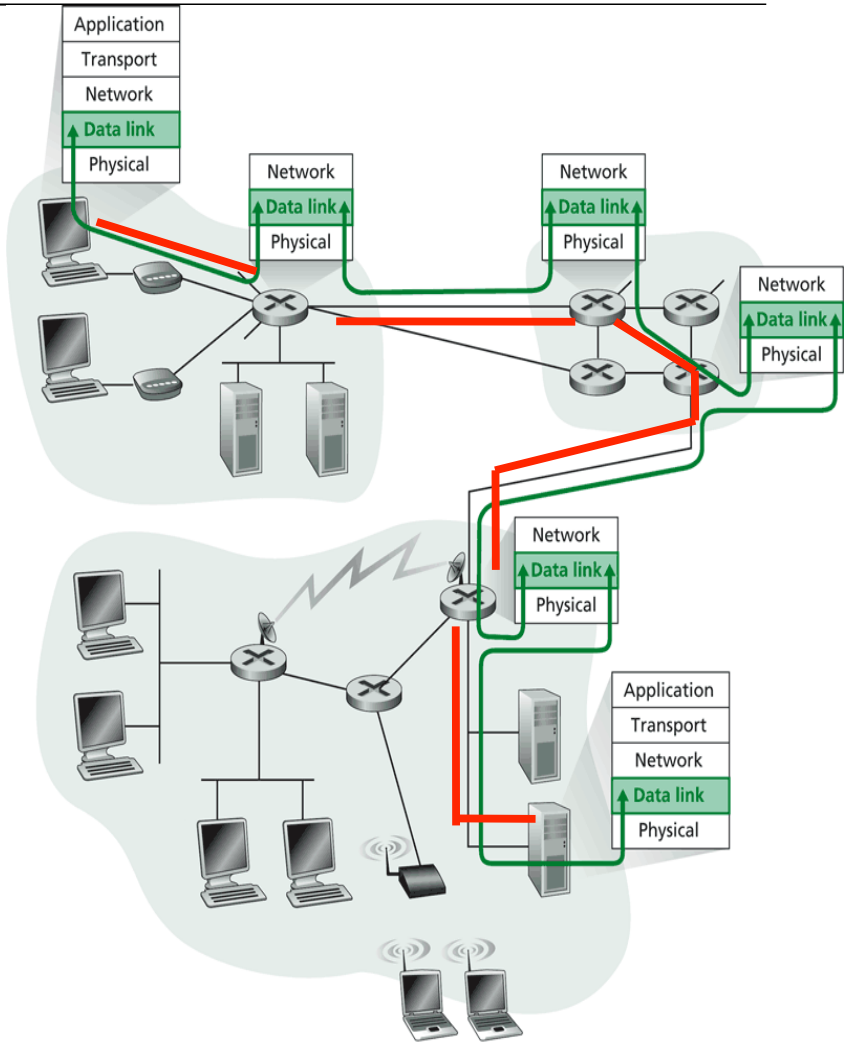


Figure 5.1 ♦ The link layer

Linkkikerroksen tehtäviä (2)

NIC (Network Interface Card)
linkki- ja fyysinen kerros

- **Vuonvalvonta, puskurointi**

  Kytkimessä on useita erinopeuksisia linkkejä

- **Virhevalvonta**

  signaali vaimenee, taustakohina häiritsee, ...

  Kehyksessä on tarkistustietoa (error detection and correction bits)

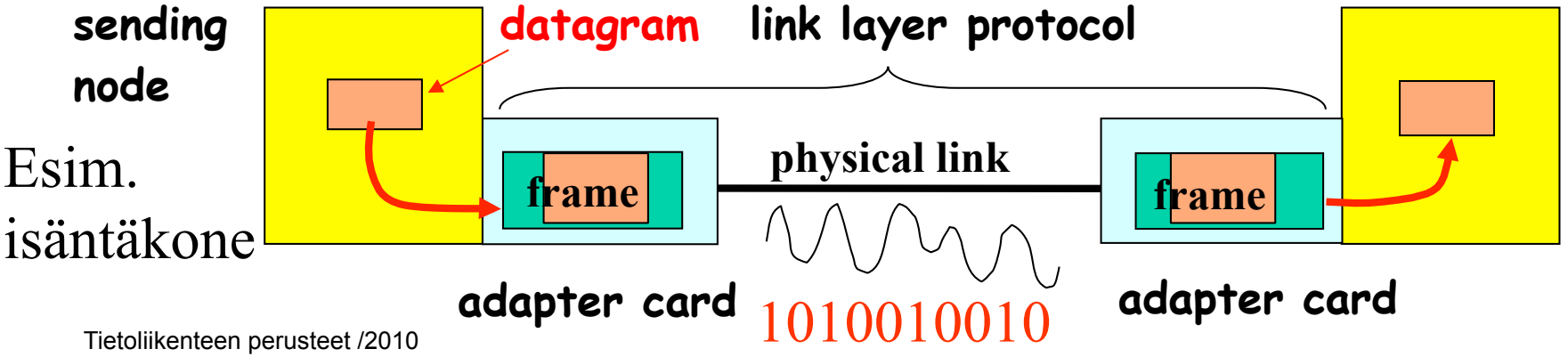
  Vastaanottava solmu korjaa, jos pystyy

  Jos ei pysty, pyytää uudelleen tai hävittää

- **Yksisuuntainen /kaksisuuntainen liikenne**

  Yksisuuntainen: lähetysvuorojen hallinta

Esim. reititin



Esim. isäntäkone

Linkkikerroksen tehtäviä

- **Kehystys** (framing)
  - Kehyksen rakenne ja koko riippuu siitä, millainen linkki on kyseessä
  - Otsake, data, lopuke

- **Kohteen ja lähteen osoittaminen**
  - Yhteiseen linkkiin voi olla liitettynä useita laitteita
  - Käytössä laitetason MAC-osoite (Medium access control)
- **Yhteisen linkin varaus ja käyttö** (link access)
  - Esim. langaton linkki, keskittimiin yhdistetyt linkit
- **Luotettava siirto**
  - Langattomilla linkeillä suuri virhetodennäköisyys
    - Linkkitaso huolehtii oikeellisuudesta
  - Miksi tästä täytyy huolehtia vielä kuljetuskerroksella?
    - Jotkut linkkityypit eivät huolehdi lainkaan!
    - Jos kehys hävitettävä ..

Tarkistussumma

- Internet-checksum
  - Yhteenlasketaan 16 bitin kokonaisuuksia, yhden komplementti
  - Kuljetuskerros laskee ja tarkastaa UDP- ja TCP-protokollissa
  - Huom. IP sekä UDP/TCP ja UDP optionaalinen
  - Ei ole kovin tehokas; linkkikerros ei käytä

- CRC (cyclic redundance check)
  - Yleisesti linkkikerroksella käytetty virheenpaljastusmenetelmä,
    - helppo toteuttaa laitteistotasolla, luotettava
  - Perustuu polynomien aritmetiikkaan
    - tunnetaan myös nimellä polynomikoodi
  - Useita tarkistusbittejä; havaitsee usean bittivirheen ryöpyn.

Lähetyskanavan kuuntelu

- Kuuntele ennenkuin lähetät
  - Asema tutkii, onko kanava jo käytössä (carrier sense)
  - Jos siirtotie on vapaa, saa lähettää
  - Jos siirtotie on varattu, odota satunnainen aika ja yritä uudelleen
- Ei aina paljasta jo alkanutta lähetystä
  - Etenemisviiveen takia ei huomata toisen signaalia ajoissa
    - Seurauksena on törmäys
- Aina huomaaminen ei ole edes mahdollista
  - Esim. satelliittikanavan kuuntelu ei paljasta, onko jokin muu maa-asema jo aloittanut lähetyksen
  - Langattomassa lähiverkossa lähettäjän ympäristön kuuntelu ei kerro, onko vastaanottaja saamassa sanomia muilta
- **CSMA** (Carrier Sense Multiple Access)
  - Useita variaatioita

Linkkikerroksen fyysinen osoite

- 32 bitin IP-osoite verkkokerroksella
  - Reitityksen tapa viitata koneeseen
- Erilaisilla linkkikerroksilla omat tapansa osoittaa oikea linkki (~ verkkokortti)
  - Siirtokehys on kuljetettava fyysisen linkin yli jollekin toiselle samaan verkkoon (LAN) kytketyistä laitteista
- **MAC-osoite** (Media Access Control Address)
  - Käytetään myös nimiä LAN-osoite, fyysinen osoite, laiteosoite, Ethernet-osoite, ...
  - Liitetty valmistusvaiheessa kiinteästi laitteeseen

Analogia:

IP-osoite ~ katuosoite MAC-osoite ~ henkilötunnus

ARP-protokolla (Address Resolution Protocol)

- Ratkaisuna ARP-protokolla ja ARP-taulu
  - **ARP-protokolla** lähettää **yleislähetysosoitteella** kyselyn, jonka kaikki vastaanottavat.
    Oman osoitteensa tunnistava laite **vastaa kyselijän MAC-osoitteeseen** ja kertoo oman MAC-osoitteensa

“aa-bb-cc-dd-ee-ff”, “FF-FF-FF-FF-FF-FF”
“Kenen IP-osoite on “xx:yy:zz:vv”?

MAC-yleislähetysosoite: FF-FF-FF-FF-FF-FF

“kk-ll-mm-nn-oo-pp”, “aa-bb-cc-dd-ee-ff”

  - **ARP-taulu** pitää tallessa kyselyjen vastauksia: IP-osoite, MAC-osoite, TTL)
    Kussakin koneessa (myös reitittimessä) jokaiselle aliverkolle oma taulunsa
    Tiedot vanhenevat n. 20 minuutissa (time-to-live)
- IPV6:ssa Neighbour Discovery Protocol (NDP)

Ethernet

- Yleisin lähiverkkoteknologia
  - Yksinkertainen, edullinen, helppo laajentaa
  - Lähiverkko syntyy kytkemällä koneet keskittimeen tai kytkimeen
- IEEE:n standardoima LAN-verkko
  - Klassinen Ethernet (10 Mbps): CSMA/CD (kuulosteluväylä)
  - Fast Ethernet (FE, 100 Mbps), Gigabit Ethernet (GE), 10 Gigabit Ethernet, 100 GB Ethernet (pian??), 1 TB Ethernet (joskus??!)
    - Yleensä kytkentäisiä kaksipisteyhteyksiä
- Muita lähiverkkostandardeja
  - Token Ring (vuororengas)
  - FDDI (Fiber Distributed Data Interface)
  - **WLAN** (langaton lähiverkko)

Ethernet Timeline (2003)

* 10 Megabit Ethernet 1990
* 100 Megabit Ethernet 1995
* 1 Gigabit Ethernet 1998
* 10 Gigabit Ethernet 2002
* 100 Gigabit Ethernet 2006**
* 1 Terabit Ethernet 2008**
* 10 Terabit Ethernet 2010**

April 24, 2008
Terabit Ethernet around 2015

Bob Metcalfe (ethernet coinventor)
gave a keynote speech,
"Toward Terabit Ethernet.

Ethernet varaus: CSMA/CD
(klassinen Ethernet-verkko on yleislähetysverkko!)

- Carrier Sense
  - Kuuntele, onko väylä vapaa (96 bitin ajan, ns. interframe gap)
  - Jos vapaa, lähetä heti
  - Muuten odota ja lähetä, kun linja vapautuu
- Collision Detection
  - Kun lähetetty, kuuntele onnistuiko
- Törmäys?
  - Huomaa signaalin voimakkuudesta
  - Lopeta kehyksen lähetys heti
  - Lähetä 48 bitin sotkusignaali (jam): muutkin huomaavat varmasti
- Random Access
  - Odota törmäyksen jälkeen satunnainen aika

Vertailua

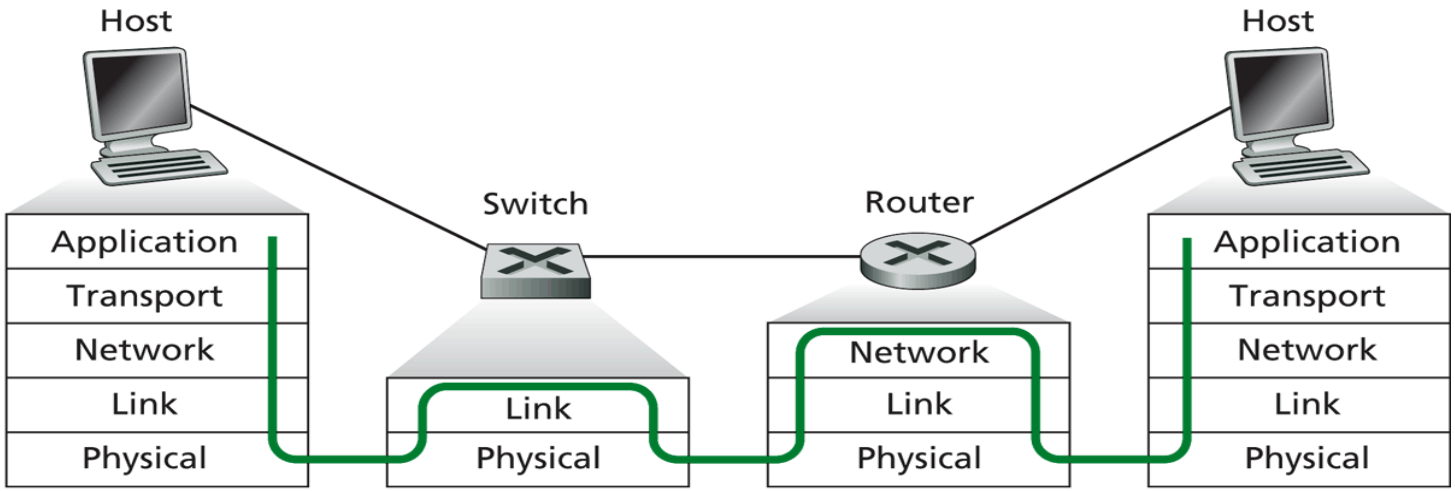


Figure 5.33 ♦ Packet processing in switches, routers, and hosts

	Keskitin (hub)	Kytkin (switch)	Reititin (router)
Traffic isolation	no	yes	yes
Plug and play	yes	yes	no
Optimal routing	no	no	yes
Cut through	yes	yes	no

KuRo08: Table 5.1

Kertauskysymyksiä

- Miten lähiverkko rakennetaan?
- Reititin vs. kytkin vs. keskitin?
- IP-osoite vs. MAC-osoite?
- ARP-protokolla ja ARP-taulu?
- Takaperinoppiminen ja kytkentätaulu?
- Bittivirheiden havaitseminen?
- CRC?
- Lähetyskanavanjako?
- CSMA/CD?

ks. kurssikirja s. 501

Tietoliikenteen perusteet

Langaton linkki

Kurose, Ross: Ch 6.1, 6.2, 6.3

(ei: 6.2.1, 6.3.4 ja 6.3.5)

Viikko 6

- Langattoman linkin ominaisuudet
- Langattoman lähiverkon arkkitehtuuri
- Yhteiskäyttöisen kanavan varaus langattomassa verkossa
- IEEE 802.11 -kehys ja osoittaminen

Oppimistavoitteet:
- Osata selittää yhteiskäyttössä olevan linkin käyttö (WLAN: CSMA/CA)

Langattoman verkon komponentit

- Tukiasema
  - LAN-yhteys
  - pääsy Internetiin
- Langattomat linkit
  - koneesta tukiasemaan
  - koneesta koneeseen
  - Rajattu kuuluvuusalue
- Isäntäkoneet
  - Laptop, PDA, IP-puhelin
  - Suorittaa sovelluksia
  - kiinteä tai liikkuva
- Haasteet
  - virhealtis linkki
  - liikkuva työasema

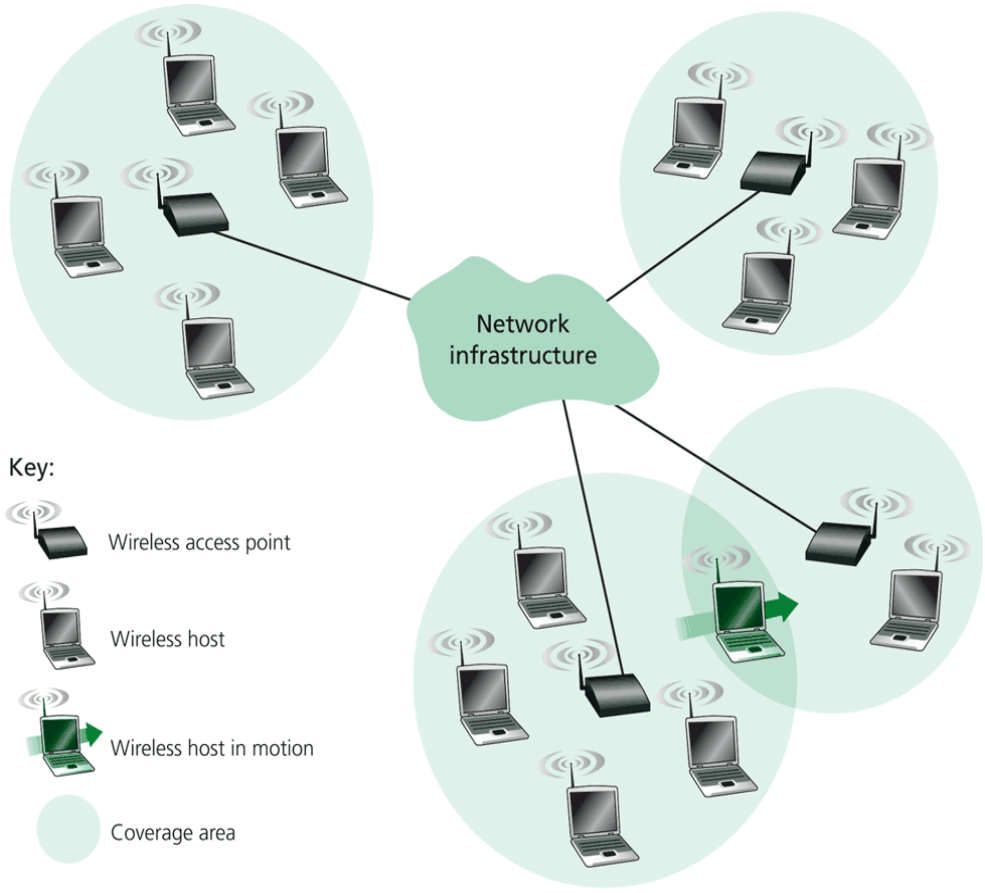
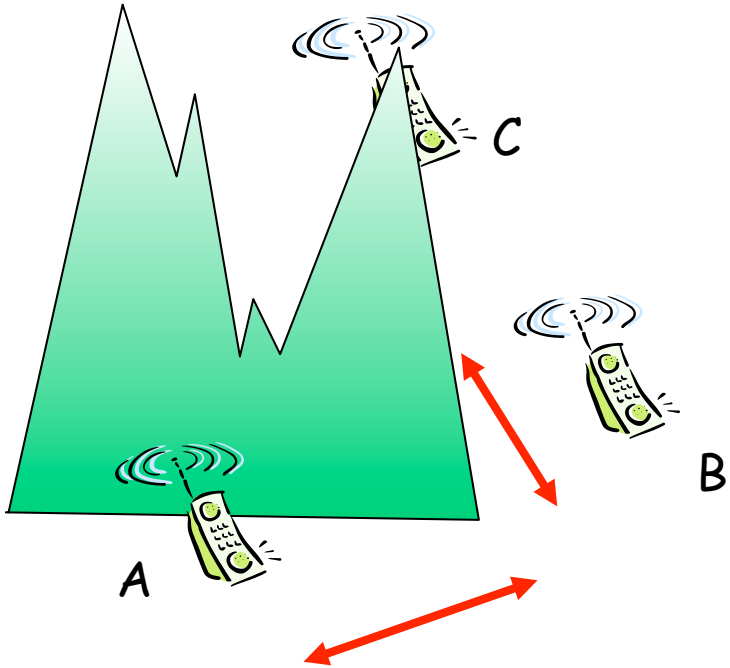


Figure 6.1 ♦ Elements of a wireless network

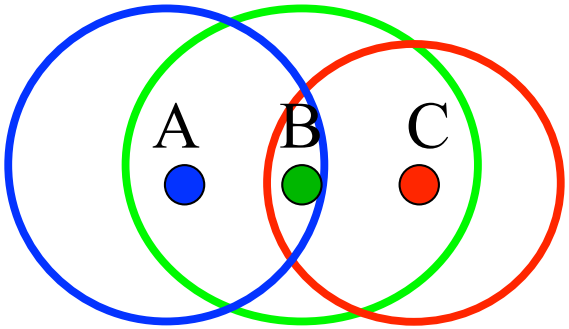
Kätketyn aseman ongelma (Hidden terminal)



KuRo08:Fig. 6.4

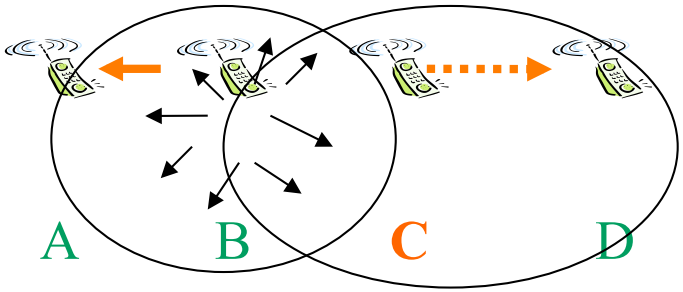
Asemat A ja C eivät kuule toisiaan eivätkä huomaa, milloin toinen lähettää samaan aikaaa ja syntyy törmäys.

Miten asema voi tietää, menikö sen lähetys perille?



Exposed terminal

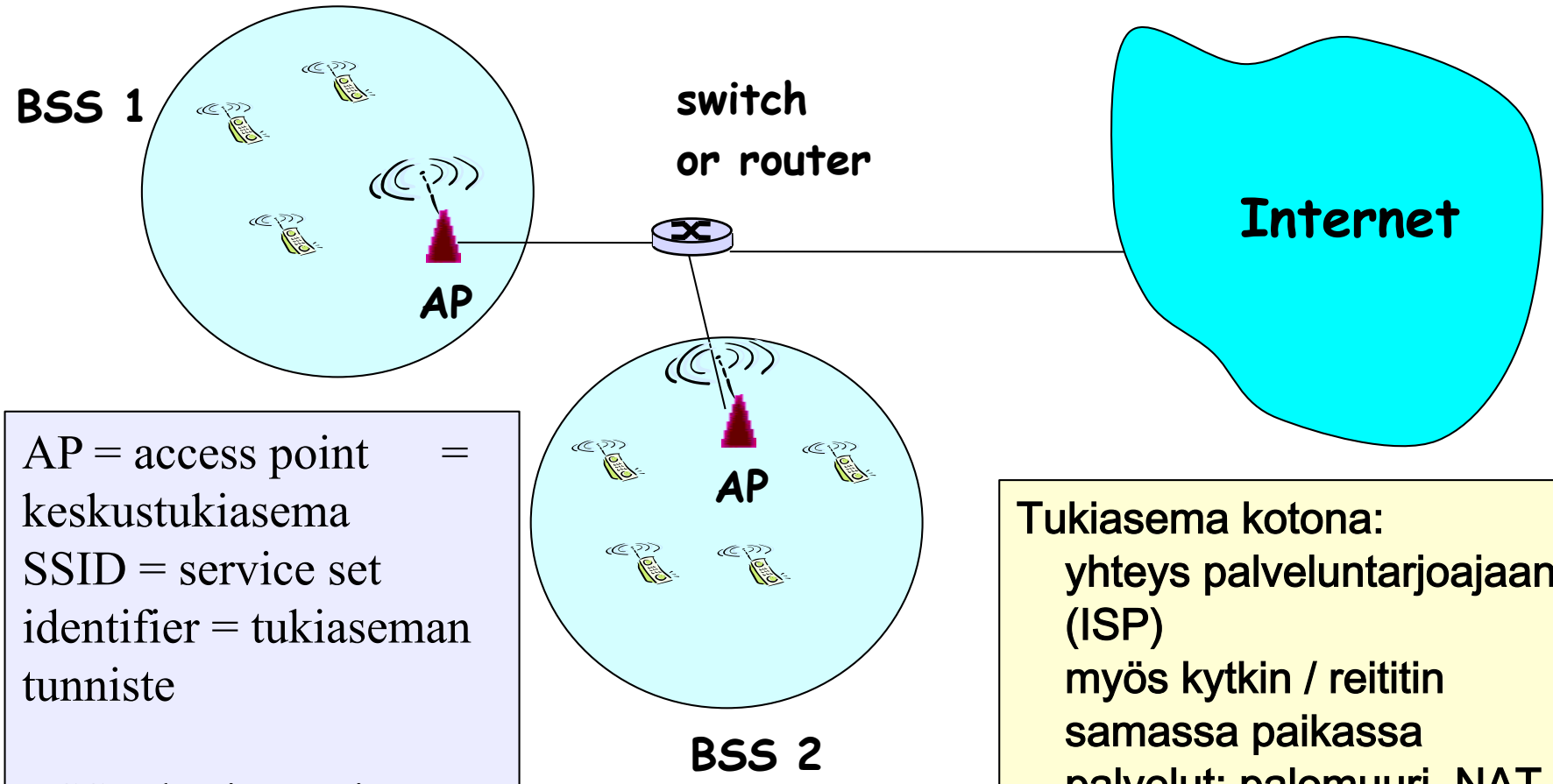
- C ei voi lähettää D:lle, koska kuulee itse B:n lähetyksen eli joku on lähettämässä
- Vaikka tämä lähetys ei lainkaan häiritsisi C:n lähettämistä D:lle eikä B:n lähettämistä A:lle



IEEE 802.11 -lähiverkko

(infrastructure wireless LAN, Wi-Fi)

KuRo08:Fig 6.7



AP = access point = keskustukiasema
SSID = service set identifier = tukiaseman tunniste

BSS = basic service set = tukiaseman palvelemat koneet

Tukiasema kotona:
- yhteys palveluntarjoajaan (ISP)
- myös kytkin / reititin samassa paikassa
- palvelut: palomuuri, NAT, DHCP

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

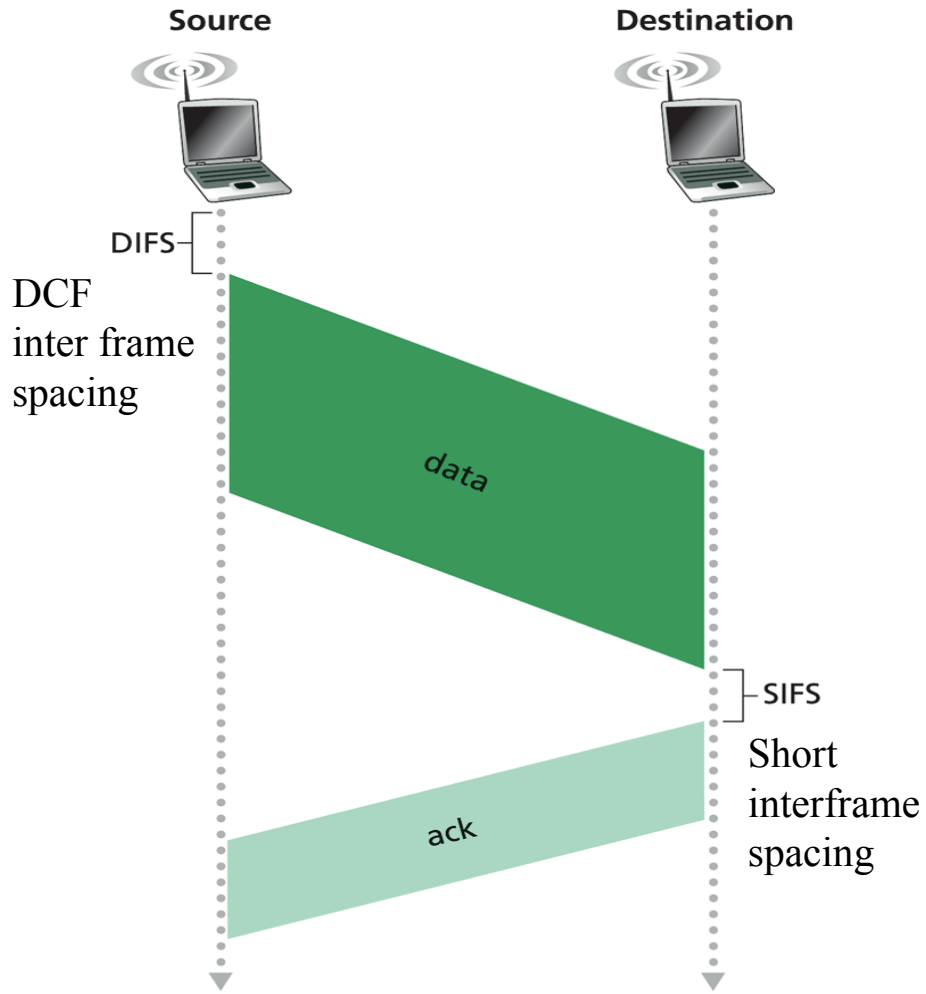


Figure 6.8 ♦ 802.11 uses link-layer acknowledgments

Kertauskysymyksiä

- Miksi WLAN:ssa ei hyödytä käyttää törmäysten havaitsemista?
- Miten sitten tiedetään, onko törmäystä tapahtunut?
- Miten WLAN:ssa hoidetaan linkin yhteiskäyttö?
- Miksi WLAN-kehyksessä kaksi osoitetta ei oikein riitä?
- Onko törmäys lainkaan mahdollinen, jos käytetään RTS/CTS-varausmenetelmää?

Ks. myös kurssikirja s. 579-580

Tietoturvasta

Kurose, Ross: Ch 1.6, Ch 8.1, Ch 8.9.1

Viikko 6

Tietoturva-kurssit:
kryptografian perusteet
IPSec

- **Turvavaatimukset**
- **Uhkia**
- **Palomuuri**

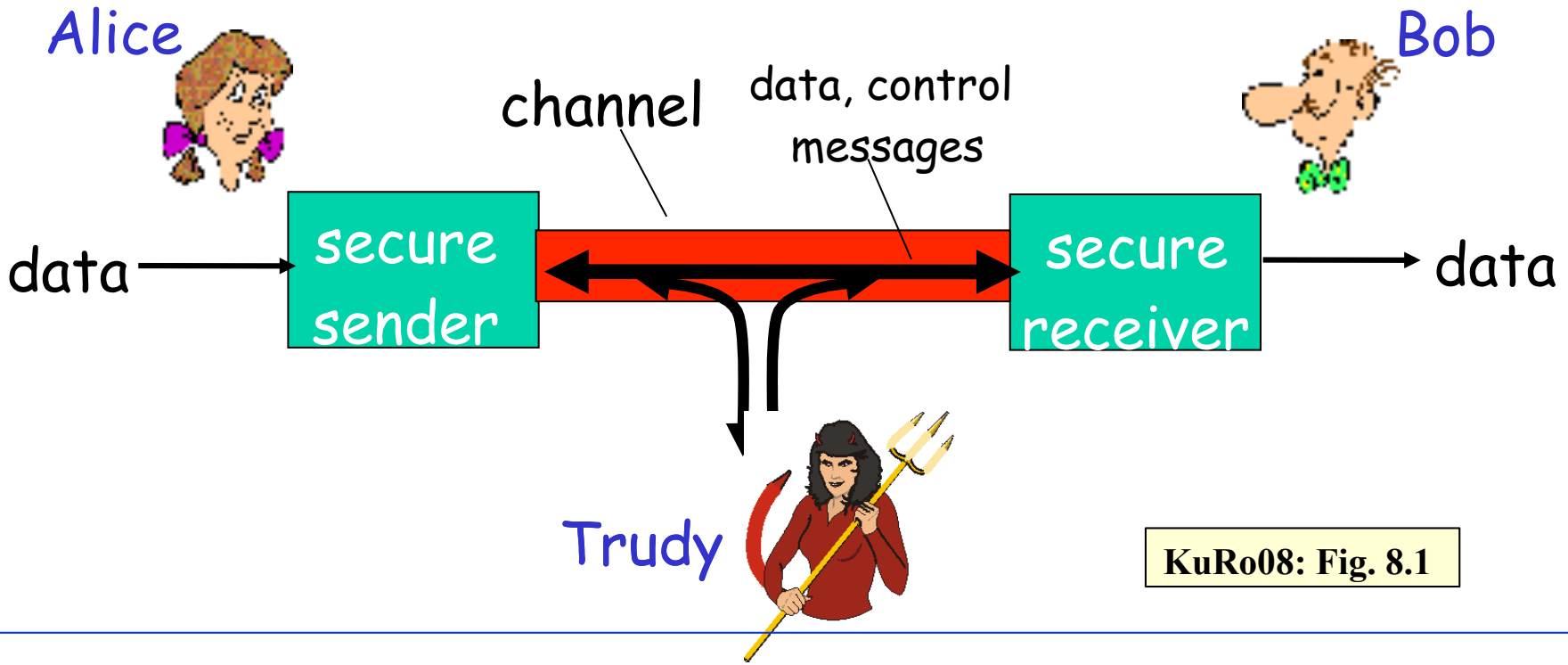
Oppimistavoitteet:
- Osata kuvailla tietoliikenteeseen kohdistuvat riskitekijät ja turvallisuusuhat
- Osata selittää, kuinka palomuuri toimii
- Ymmärtää tietoturvasta sen verran, että osaa huolehtia oman koneen turvallisuudesta

Turvavaatimukset

- Luottamuksellisuus (confidential, secrecy)
  - Vain lähettäjä ja vastaanottaja 'ymmärtävät' sanoman sisällön
  - Muu eivät saa välttämättä tietoa edes sen olemassaolosta
    - Salakirjoitus
- Autentikointi (authentication)
  - Lähettäjä ja vastaanottaja varmistuvat toistensa identiteeteistä
    - Oikeaksi todentaminen, salakirjoitus
- Eheys, koskemattomuus (message integrity)
  - Lähettäjä ja vastaanorraja varmoja siitä, ettei sanomaa ole muutettu (siirron aikana ta myöhemmin)
    - Digitaalinen allekirjoitus
- Palveluiden saatavuus ja suojaus
  - Palvelut ovat saatavilla käyttötarkoituksen mukaisesti
  - Vain niillä pääsy, joilla lupa käyttää käyttöoikeuksien mukaisesti
    - Käyttäjätunnus ja salasana, tiedostojen / objektien käyttöoikeudet, …
  - Suojautuminen ‘ulkoa’ tulevia hyökkäyksiä vastaan (haittaohjelmat, palvelunestohyökkäys) vastaan
    - palomuuri, havaitsemis- ja puhdistusohjelmat

Ystävä ja tunkeutuja

- Tuttu asetelma reaalimaailmastakin
  - Bob ja Alice kommunikoivat keskenään (salassa muilta?)
  - Trudy (intruder) voi siepata sanomia: nuuskia, kerätä tietoa
  - Trudy voi muunnella, tuhota ja lisätä sanomia



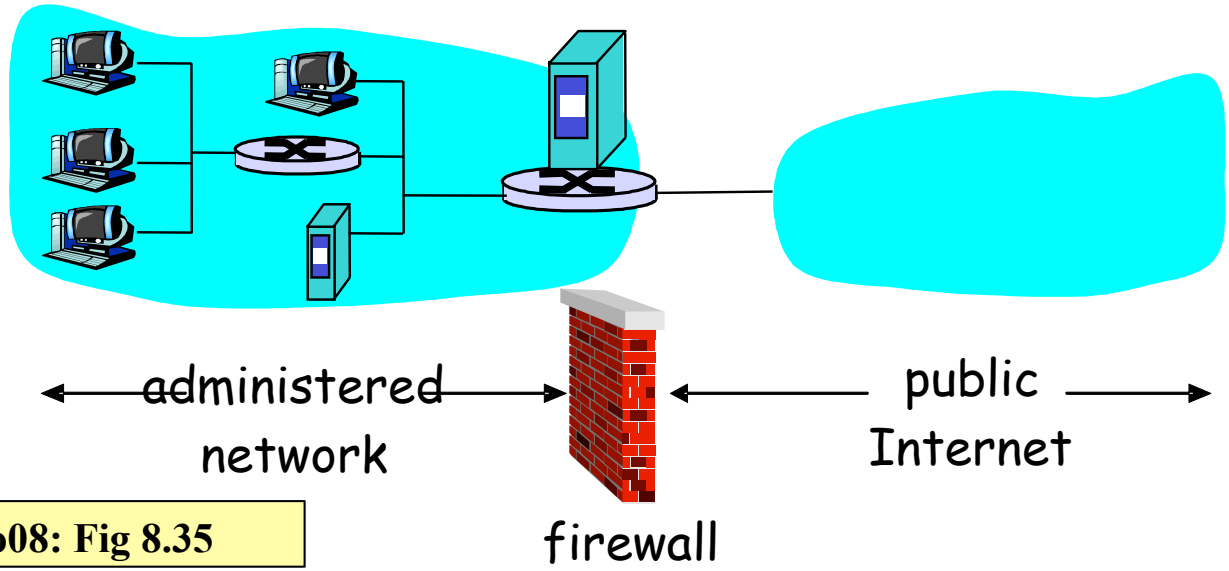
KuRo08: Fig. 8.1

Mitä Trudy puuhii?

- Koputtelee koneen portteja (mapping)
  - Turva-aukkojen löytämiseksi ja koneen valtaamiseksi
- Salakuuntelee (eavesdropping, sniffing)
  - Sieppaa sanoman matkalla ja tutkii sisällön
- Väärentää, “peukaloi”(impersonation, spoofing)
  - Vaihtaa paketin tietoja, esim. IP-osoitteen
- Tehtailee sanomia, “satuilee” (fabrication)
  - Tekee ja lisää liikenteeseen ylimääräisiä sanomia
- Kaappaa yhteyden (hijacking)
  - Vaihtaa oman IP-osoitteen lähettäjän / vastaanottajan tilalle
- Estää palvelun (DoS, Denial of Service)
  - Kuormittaa palvelinta, jotta se ei ehdi palvella oikeita käyttäjiä

Palomuuri (firewall)

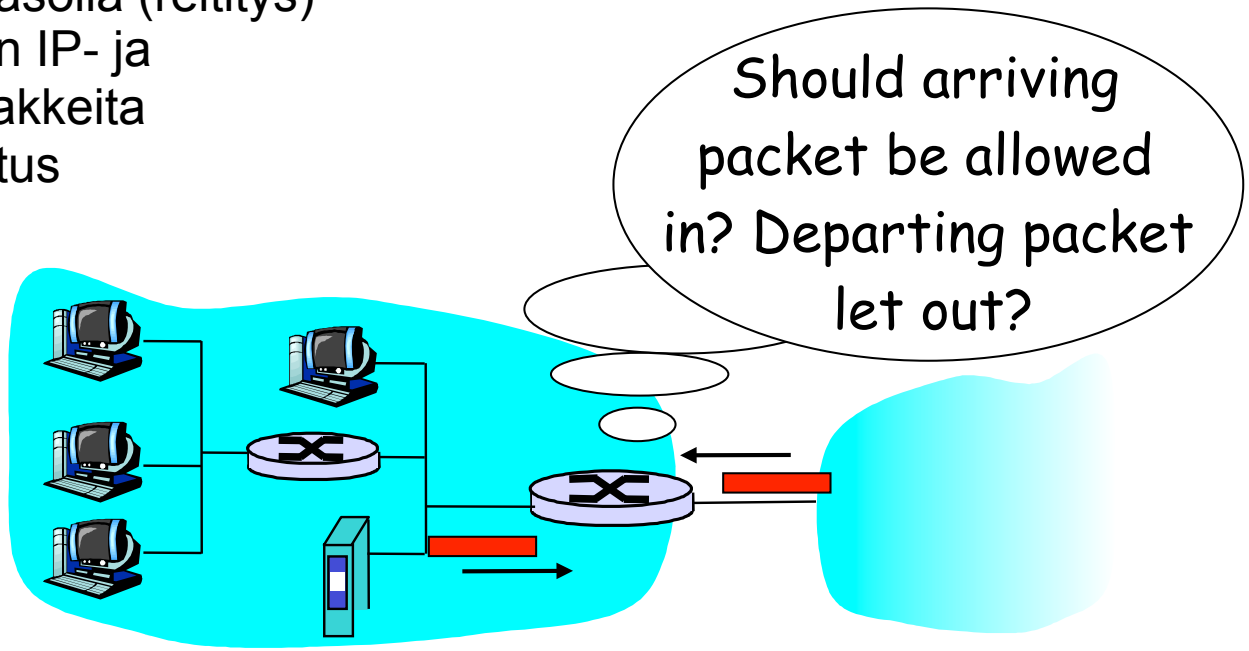
- Ohjelmisto + laitteisto
- Suodattaa (filteroi) liikennettä organisaation oman verkon (intranet) ja julkisen Internetin välillä
  - Osa IP-paketeista pääsee palomuurin läpi, osa ei



KuRo08: Fig 8.35

Kaksi erilaista palomuuria

- Paketteja suodattava palomuuri (packet filtering firewall)
  - Toimii verkkotasolla (reititys)
  - Tutkii pakettien IP- ja TCP/UDP-otsakkeita
  - Karkea suodatus



- Sovellustason yhdyskäytävä (application-level gateway)
  - Toimii sovelluskerroksella välittäjänä (relay)
  - Tutkii sovellusdataa
  - Hienojakoisempi suodatus

Kertauskysymyksiä

- Mitä ominaisuuksia halutaan turvalliselta yhteydeltä?
- Millaisia uhkia verkkoihin (koneisiin, tietoliikenteeseen ja palveluihin) kohdistuu?
- Miten eri uhkiin pyritään varautumaan?
- Mitä ovat haittaohjelmat?
- Mikä on DoS? Entä DDoS?
- Miten palomuuri toimii? Mihin sitä käytetään?