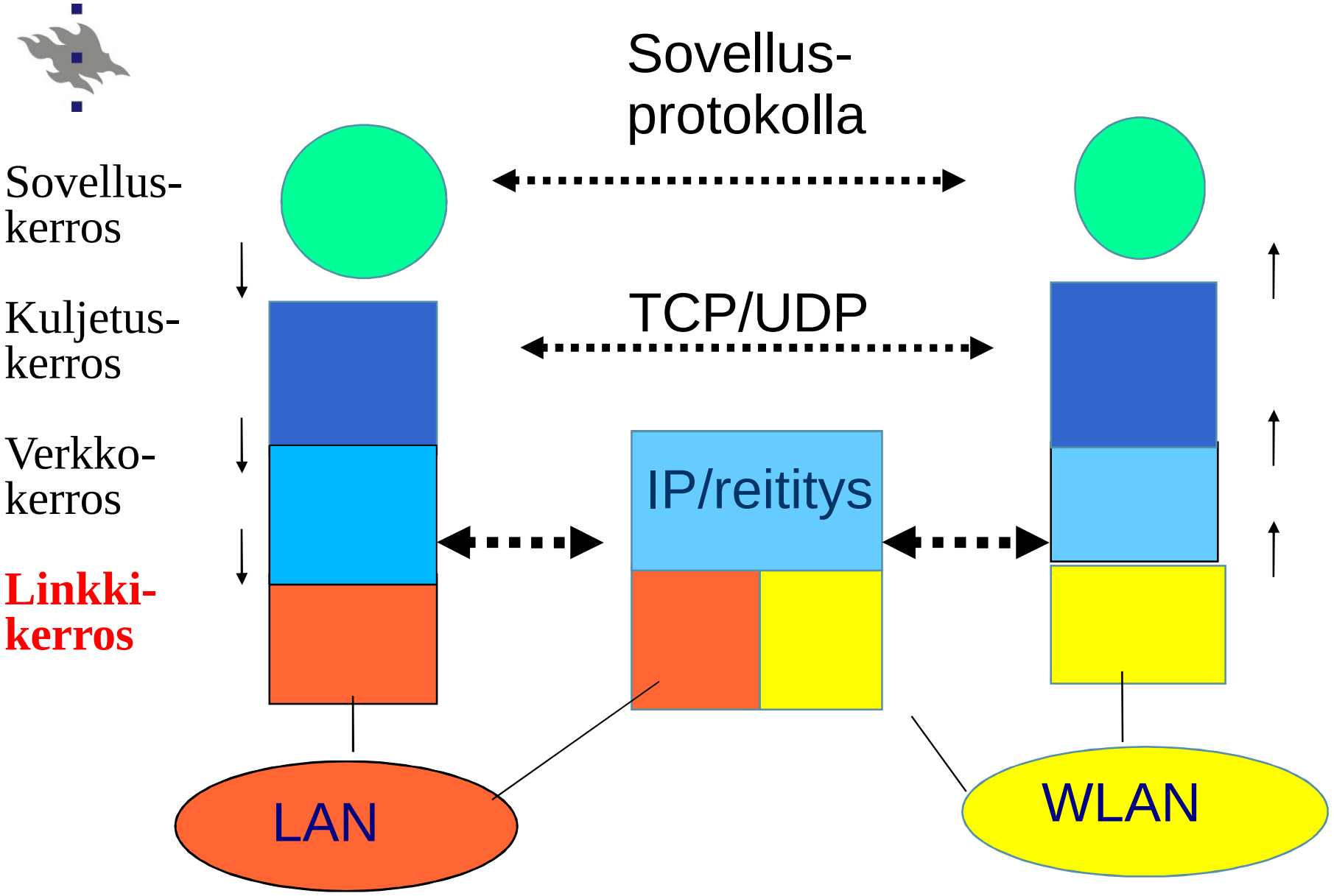
Tietoliikenteen perusteet

Linkkikerros

Kurose, Ross: Ch 5.1- 5.6


Sovellus-
protokolla
Sovellus-
kerros
Kuljetus-
kerros
TCP/UDP
Verkko-
kerros
IP/reititys
Linkki-
kerros
LAN
WLAN

Sisältö

- 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

Linkkikerroksen tehtävät

Ch 5.1

Linkkikerros

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

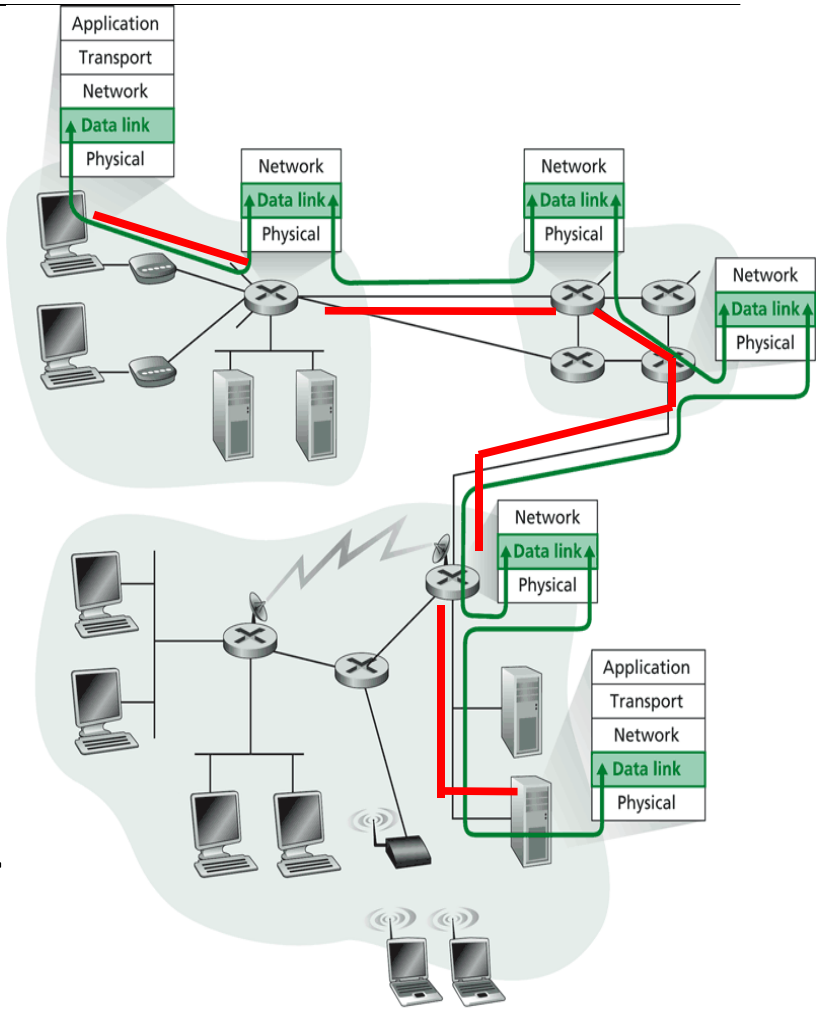


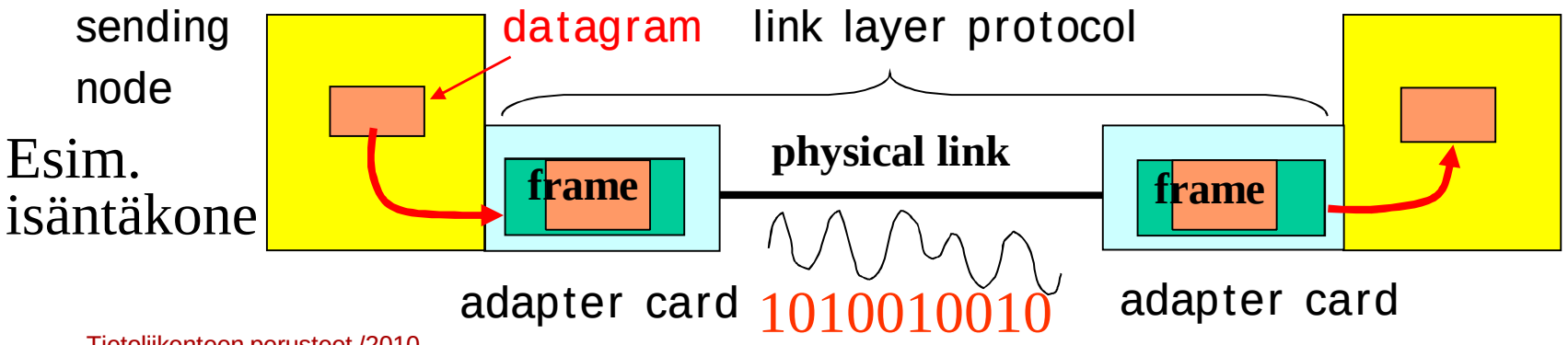
Figure 5.1 ♦ The link layer

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

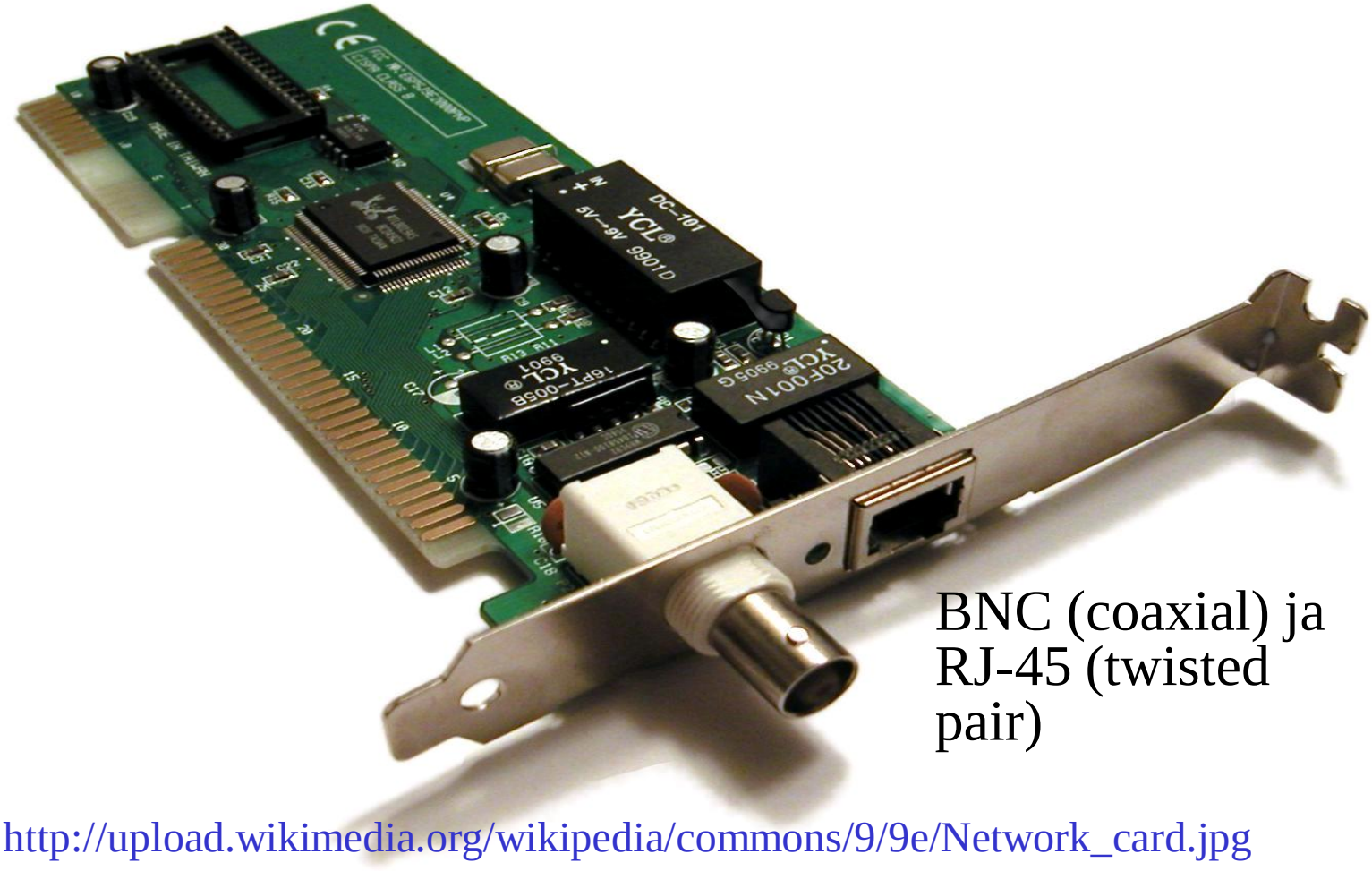


Linkkikerroksen tehtäviä

- **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 ..

A 1990s Ethernet network interface card.



http://upload.wikimedia.org/wikipedia/commons/9/9e/Network_card.jpg

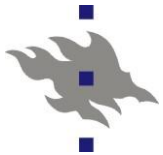
Linkkikerros

Virheiden havaitseminen ja korjaaminen

Ch 5.2

Bittitason virheet

- Yhden bitin virheitä siellä täällä tai peräkkäisten bittien virheryöppyjä (burst)
- Virheiden esiintymistiheys BER (bit error rate)
  - Mitä suurempi BER, sitä lyhyempiä kehyksiä kannattaa käyttää
- Havaitsemiseksi lisäbittejä
  - feedback/backward error control
- Korjaamiseksi enemmän lisäbittejä
  - Forward error correction (FEC) (esim. Hamming-koodi)
  - Esim. CD, DVD, Blu-Ray, viivakoodit, satelliitiyhteydet, digitelevisio, ... (Reed-Salomon-koodi)
- Tietoliikenne yleensä tyytyy vain havaitsemaan virheet
  - Virheelliset hylätään ja korjauksena on uudelleenlähetys

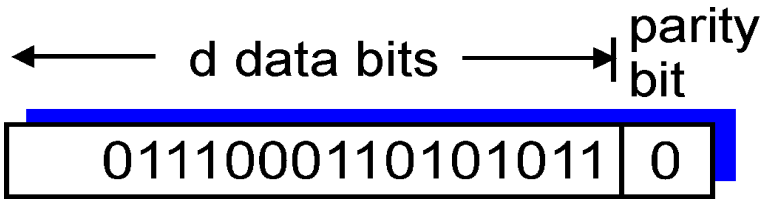


Virheenkorjaus

- Kaksi keskeistä tyyppiä
  - Error Detection Codes (CRC, Parity, Checksums)
  - Error Correction Codes (e.g. Hamming, Reed Solomon)
- Yleinen periaate
  - Lisätään pakettiin ylimääräisiä bittejä tarkistusta varten
- Forward Error Correction (FEC)
  - Redundanssin avulla voidaan korjata bittivirheitä
  - Näin säästetään signaloinnissa (NAK ja RTT)
- Pariteettibitti
  - Yhden virheen (tai parittoman lukumäärän) tunnistaminen
  - Yksi bitti jonon päässä

Pariteettitarkistus

- **Pariteettibitti**
  - Parillinen vs. pariton pariteetti
  - Virheryöpyssä jopa 50% voi jäädä huomaamatta



- **Kaksiulotteinen pariteetti**
  - Erikseen horisontaalinen (parillinen) ja vertikaalinen (pariton) pariteetti
  - Pystyy korjaamaan yhden bitin virheen.
  - Ei tunnista parillistä määrää virheitä.
- **Hamming-koodi**
  - Korjaa yhden bitin virheen
  - m pariteettibittiä, 2^m-m-1 databittiä

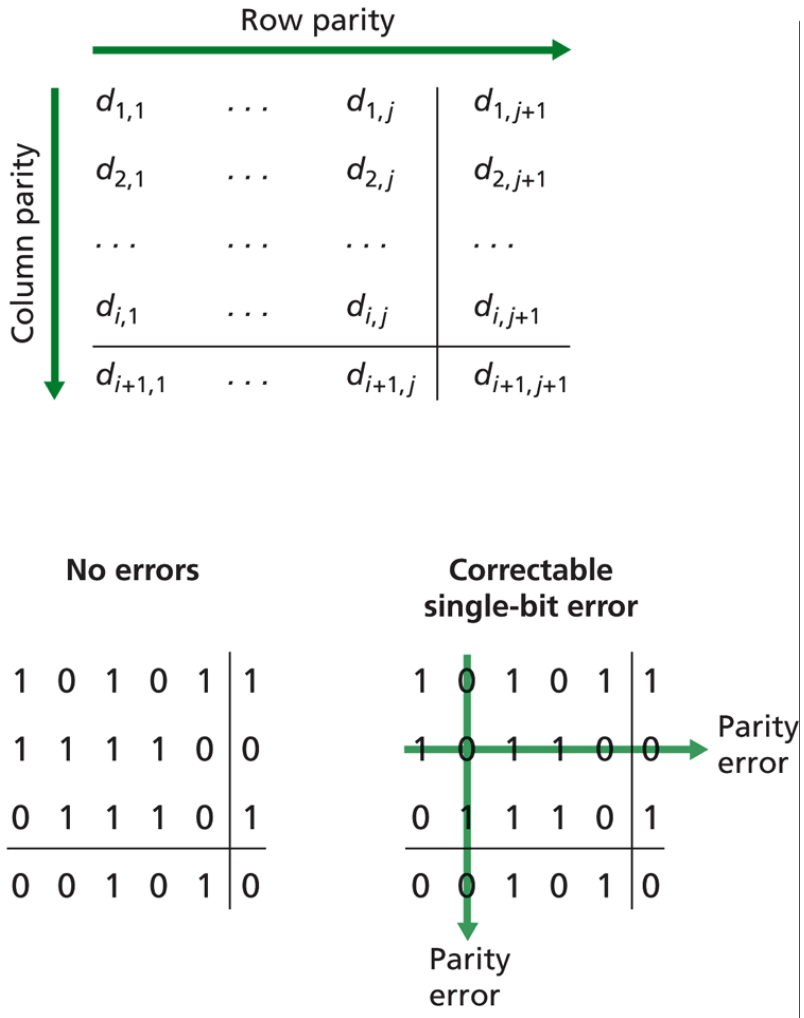


Figure 5.6 ♦ Two-dimensional even parity

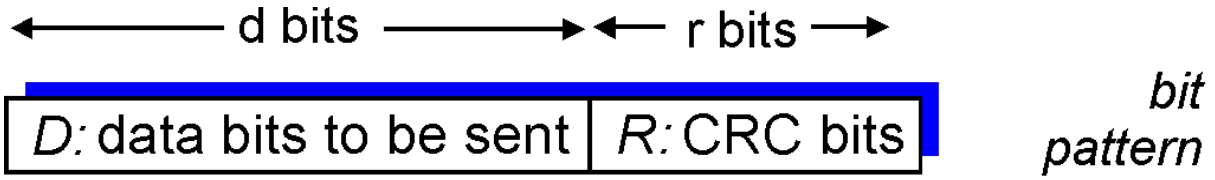
Tarkistussumma

- Internet-checksum
  - Yhteenlasketaan 16 bitin kokonaisuuksia, yhden komplementti
  - Kuljetuskerros laskee ja tarkastaa UDP- ja TCP-protokollissa
  - 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.

CRC

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



bit pattern

$D * 2^r$ XOR R

mathematical formula

CRC-esimerkki

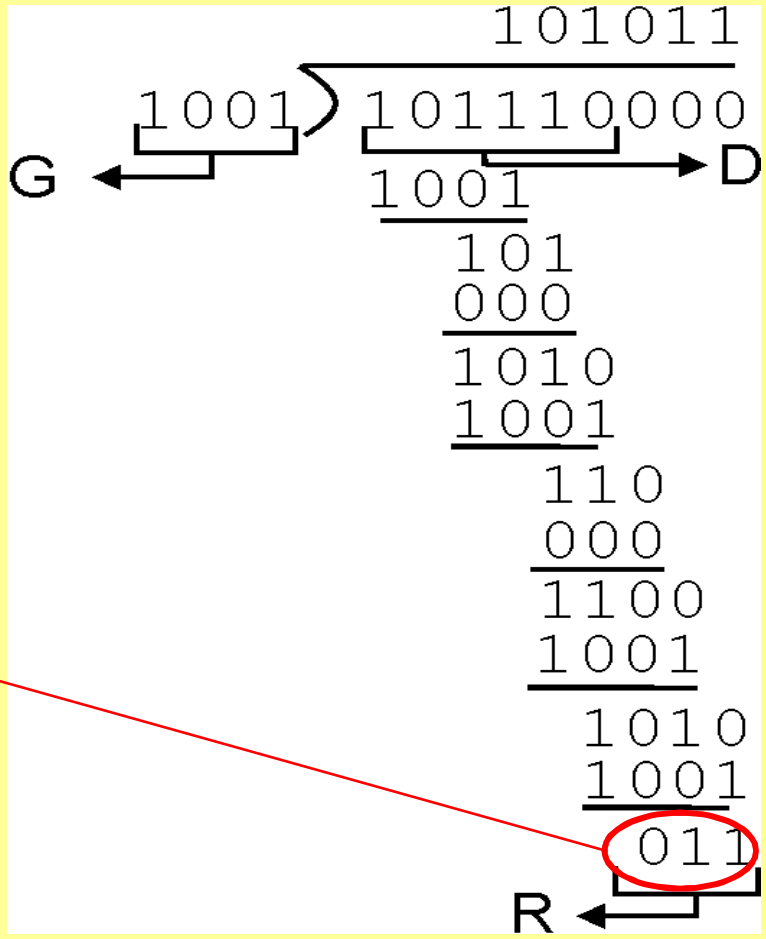
Data: 101110

G: 1001, polynomina

$1*x^3 + 0*x^2 + 0*x^1 + 1*x^0$

<D,R>: 101110???

Lähetä: 101110011



Modulo 2-aritmetiikka

vähennyslasku yhteenlaskuna

ei lainaamista, ei muistinumeroita

= bittitason XOR

1+1 =0, 1+0 = 0+1 = 1, 0+0=0

KuRo08:Fig 5.8

Standardoituja virittäjäpolynomeja

- $G_{CRC-12} = x^{12} + x^{11} + x^3 + x^2 + x + 1$
- $G_{CRC-16} = x^{16} + x^{15} + x^2 + 1$
- $G_{CRC-32} = x^{32} + x^{26} + x^{23} + \ldots + x^4 + x^2 + x + 1$

  $= 1\ 0000\ 0100\ 1100\ 0001\ 0001\ 1101\ 1011\ 0111$

Virittäjäpolynomin valinta keskeistä

Virittäjäpolynomin merkitsevin bitti =1

Havaitsee

kaikki virheryöpyt, joiden pituus < tai = kuin virittäjän pituus

lähes kaikki virheryöpyt, joiden pituus on suurempi

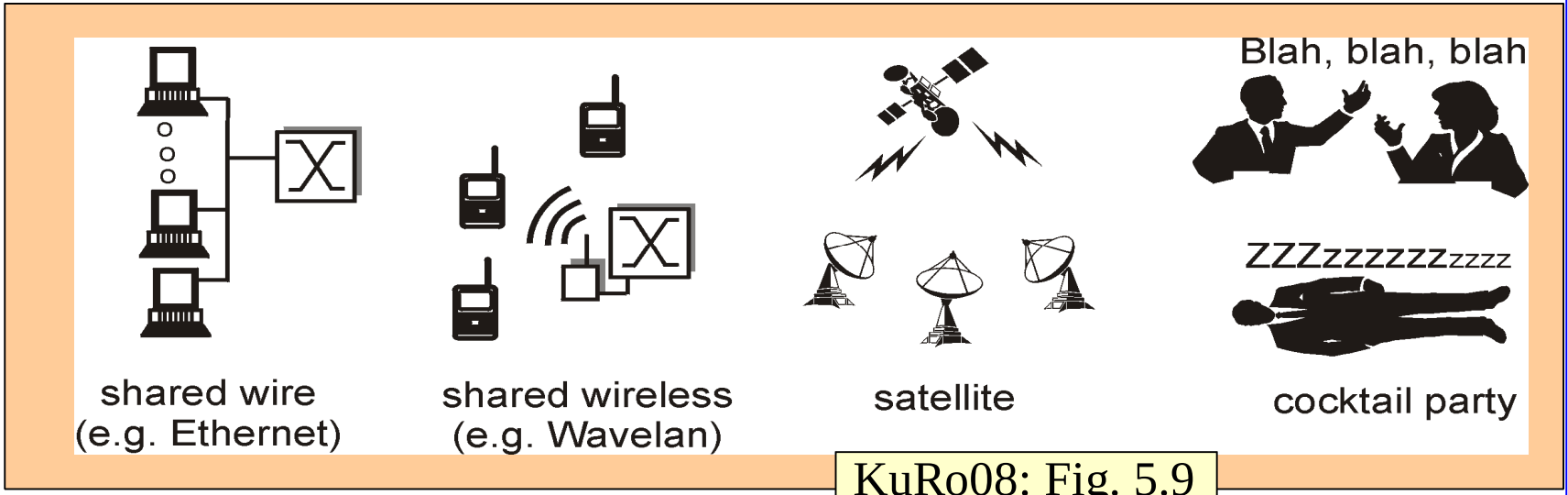
Linkkikerros

Yhteiskäyttöinen kanava

Ch 5.3

Yksi kanava

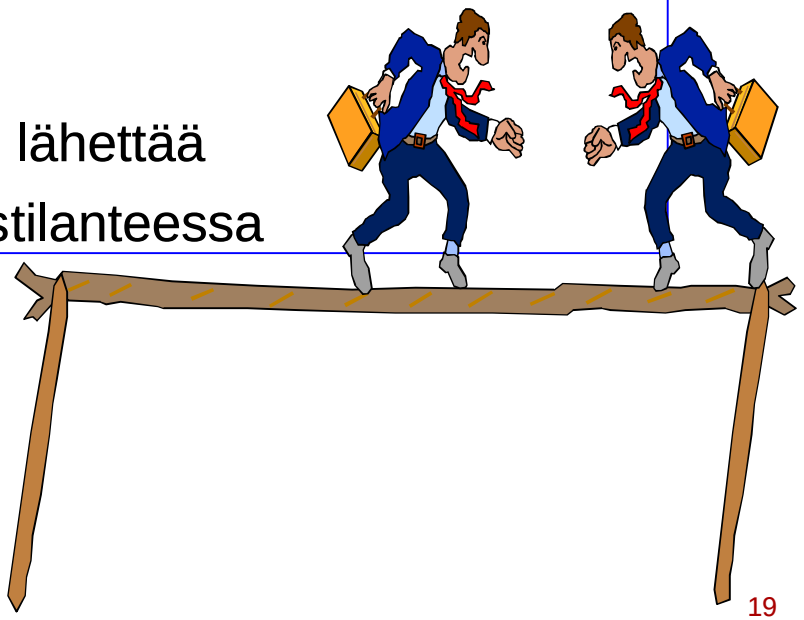
- **Kaksipisteyhteys** (point-to-point)
  - PPP-protokolla, puhelinyhteys (dial-up access)
  - Ethernet-piuha kytkimen ja isäntäkoneen välissä
- **Yleislähetysyhteys** (broadcast)
  - Alkuperäinen Ethernet, Ethernet keskittimen ja isäntäkoneen välissä, kaapelimodeemiyhteys (upstream), WLAN, satelliitti,



KuRo08: Fig. 5.9

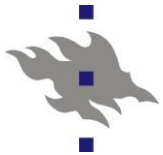
Lähetysvuorojen jakelu

- Yksi yhteinen kanava lähettäjille
  - Lähetys onnistuu vain, jos yksi kerrallaan lähettää
- Jos useampi lähettää yhtäaikaa, syntyy yhteentörmäys
  - Kaikki solmut saavat useita signaaleja, “bittimössöä”
  - Törmänneet sanomat tuhoutuvat ja ne on lähetettävä uudelleen
- Mutiple Access Protocol
  - Tapa, jolla solmu päättelee, voiko se lähettää
  - Kuinka solmun on toimittava törmäystilanteessa



Tätä tavoitellaan

- Pieni yleisrasite
  - Kun vain yksi lähettää, se pystyy hyödyntämään koko kanavan siirtonopeuden R bps
- Tasapuolisuus
  - Kun M lähettäjää, kukin saa keskimäärin saman osuuden linjan siirtonopeudesta (R/M bps)
- Toimintavarmuus
  - Yksikään solmu ei ole erikoisasemassa, koordinaattorina
  - Ei kellojen sykronointia tms
  - Hajautettu vuoroista sopiminen
- Kustannustehokkuus
  - Yksinkertainen ja halpa toteuttaa



Lähetysvuorojen jakelu

1) Kanavanjakoprotokollat (channel partitioning protocol)
   Jaa kanavan käyttö 'viipaleisiin' (time slots, frequency, code)
   Kukin solmu saa oman viipaleensa
   TDMA, FDMA, CDMA
   “käytä sinä tätä puolta, minä tätä toista”

2) Kilpailuprotokollat (random access protocols)
   “Se ottaa, joka ehtii.”
   Jos sattuu törmäys, yritä myöhemmin uudelleen.
   Aloha,CSMA, CSMA/CD

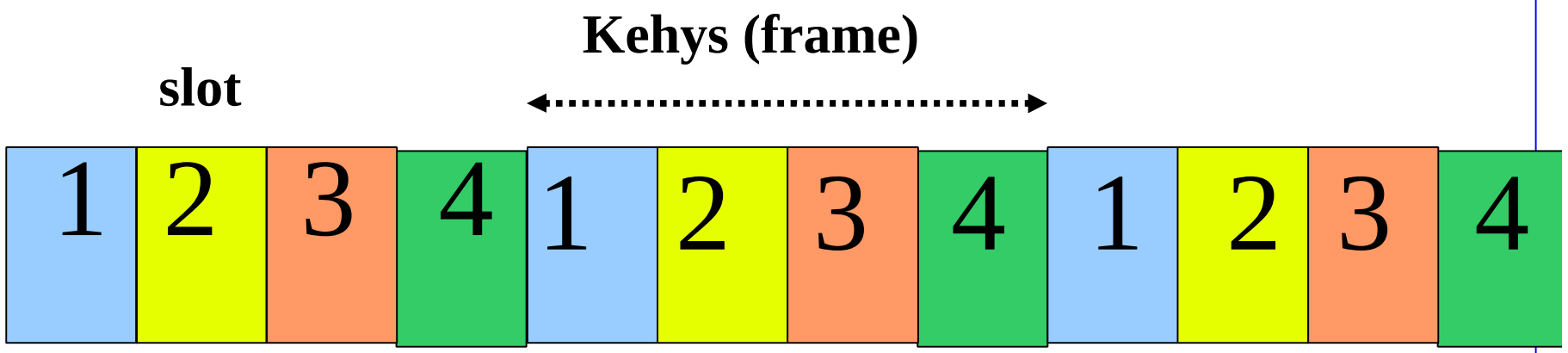
3) Vuoronantoprotokollat (taking-turns protocols)
   Jaa käyttövuorot jollakin sovitulla tavalla:
   pollaus, vuoromerkki, ...
   “Minä ensin, sinä sitten.”

1) Kanavanjako: TDMA

TDMA: Time Division Multiple Access

- Anna aikaviipale kullekin kanavaan kytketylle vuorotellen
  - Koko kanava on hetken yksityiskäytössä => R/M bps
  - Ehtii lähettää yhden kehyksellisen (data frame)
  - Vaikka lähetettävää ei olisikaan, aikaviipale on silti varattuna

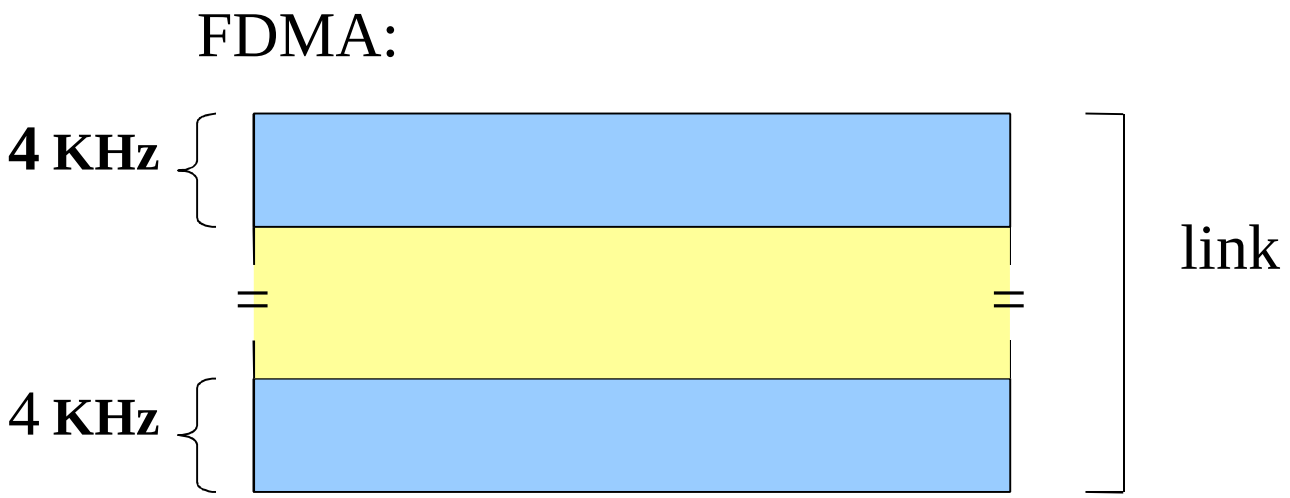
TDMA:



Kanavanjako: FDMA

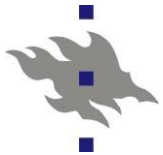
- FDMA (Frequency Division Multiple Access)

  Jaa kanavan taajuusalueet kanavan käyttäjien (varaajien) kesken
  - **Vain osa kanavasta yksityisessä käytössä => R/M bps**
  - **Varattuna, vaikka ei olisi lähetettävää**



Kanavanjako: CDMA

- CDMA (Code Division Multiple Access)
  Radiolinjoilla käytettävä koodinjakoon perustuva protokolla
  - Matkapuhelimet, ..
  - Kullakin asemalla oma tapansa koodata bitit 1 ja 0 (oma sirukoodi)
- Asemat voivat lähettää yhtäaikaa koko kanavan taajuudella
  - Kaikkien signaalit saavat yhdistyä linkillä
  - Asemat pystyvät erottelemaan yhteissignaalista itselleen kuuluvat bitit (oma sirukoodi)
- Tarkat ajoitukset

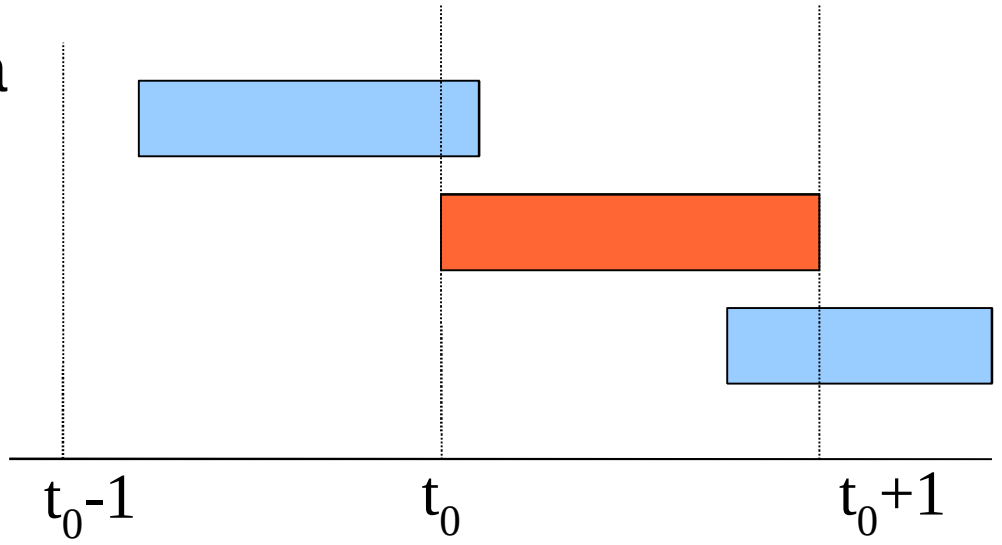


2) Kilpailuprotokollat

- Kun asema haluaa lähettää
  - Se kuuntelee ensin, onko joku muu asema jo lähettämässä
  - Jos ei, lähettää heti täydellä nopeudella
- Jos kaksi aloittaa yhtäaikaa => törmäys
  - Odota satunnainen aika ja yritä uudestaan (random access)
- Protokolla määrittää
  - Miten törmäys huomataan
  - Miten törmäyksestä toivutaan
- Esim.
  - ALOHA, viipale ALOHA (slotted ALOHA)
  - CSMA (carrier sense multiple access)
  - CSMA/CD (with collision detection, Ethernet 802.3)
  - CSMA/CA (collision avoidance, WLAN 802.11)

Aloha

- Hawaijilla, 70-luvulla radiotietä varten
- Lähetä heti, kun on lähetettävää
  - Ei mitään kuuntelua ennen lähetystä
- Kuuntele sitten, onnistuiko lähetys
  - Lähiverkossa törmäys havaitaan 'heti', sillä siirtoviive on pieni (toisin kuin satelliitilla)
- Jos törmäys, niin odota satunnainen aika ja yritä uudelleen
- Yksinkertainen
- Törmäyksen td. suuri
  - Max tehokkuus ~ 18%



Viipaloitu Aloha (slotted Aloha)

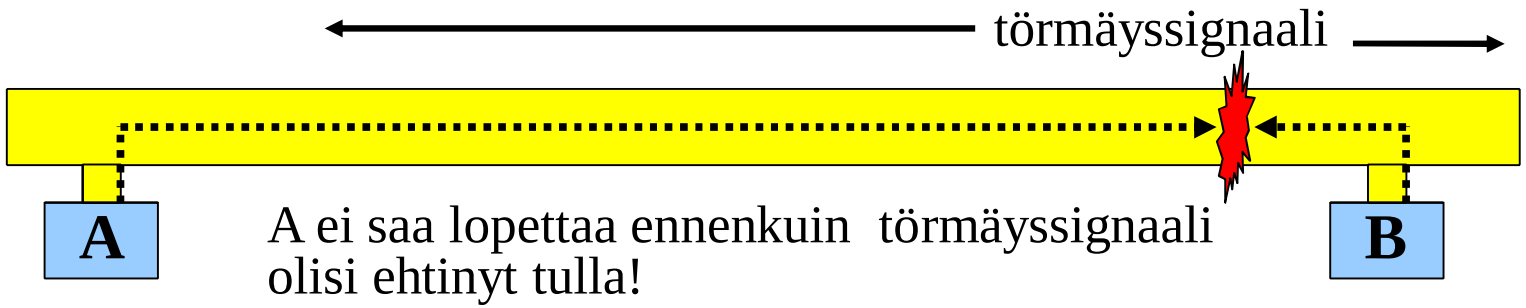
- Lähetysaika jaettu aikaviipaleiksi (slot)
  - Kaikki siirtokehykset samankokoisia => siirtoaika aina vakiomittainen
- Lähetys voi alkaa vain aikaviipaleen alussa
  - Törmäykset täydellisiä => törmäysaika = yhden aikaviipaleen mittainen
- Solmut synkronoitava: aikaviipaleen alku
- Jos törmäys, niin kaikki solmut huomaavat
  - Uudelleenyritys seuravalla viipaleella todennäköisyydellä p (ts. jättää yrittämättä seuraavalla viipaleella tn:llä 1-p)
  - Yrittää, kunnes onnistuu
- Suorituskyky kaksinkertaistuu (Alohaan verrattuna)
  - Jos paljon lähettäjiä max., ~37 % tehokkuus
  - Siis 37% tyhjiä,37% onnistumisia, 26% törmäyksiä

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

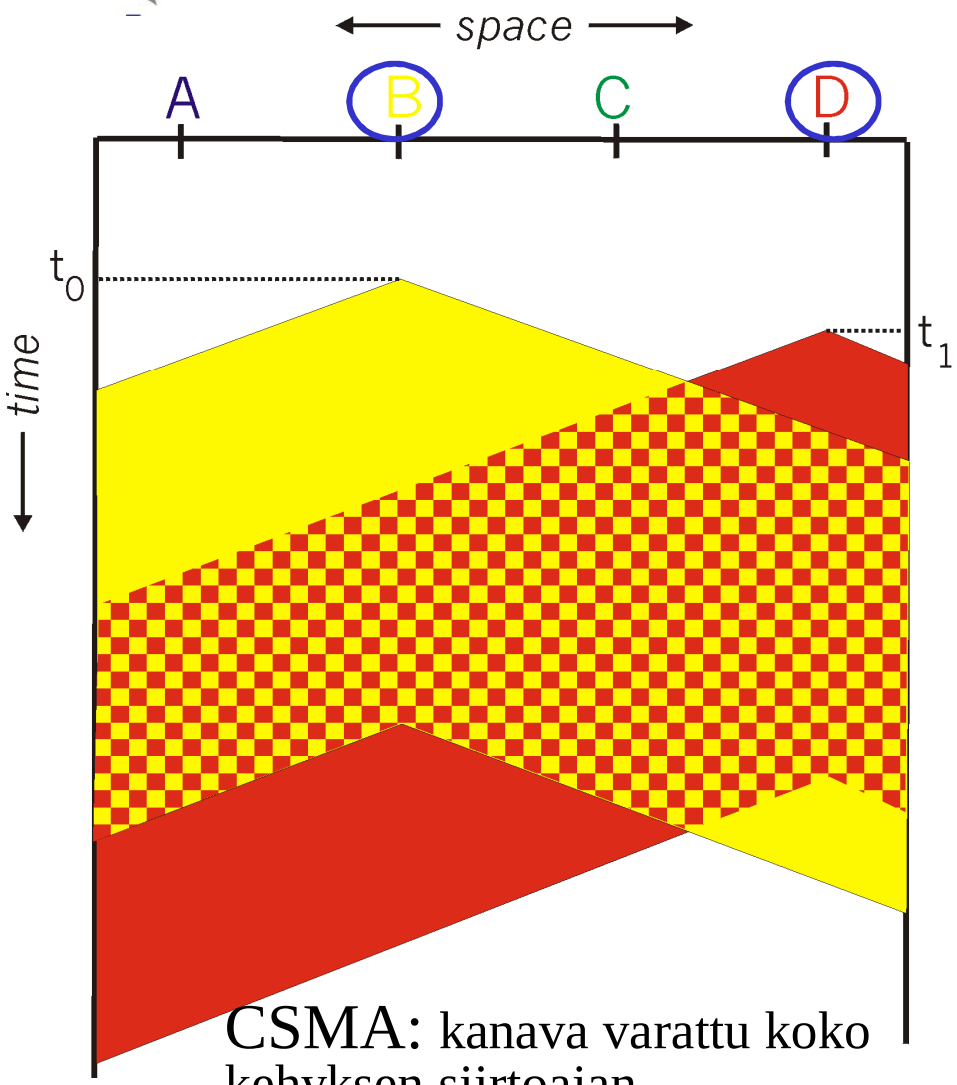
CSMA/CD (with Collision Detection)

- Asema kuuntelee myös lähettämisen jälkeen
  - Langallinen LAN: törmäys => 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

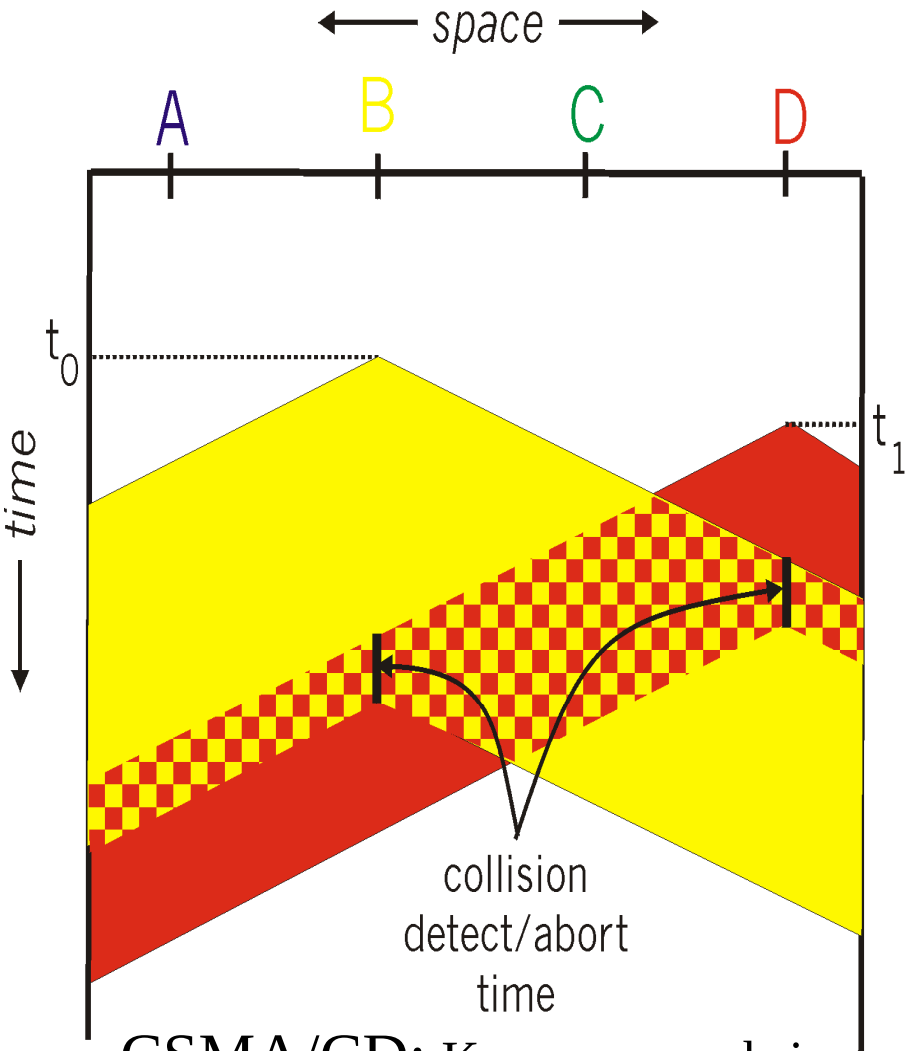


CSMA ja törmäys

KuRo08: Fig. 5.13 ja 5.14



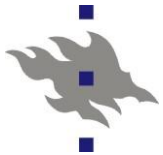
CSMA: kanava varattu koko kehyksen siirtoajan



CSMA/CD: Kanava vapaaksi heti, kun törmäys on huomattu

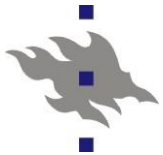
CSMA/CA

- Soveltuu langattomille verkoille (kun CSMA/CD:tä ei voida soveltaa)
  - Hidden terminal problem
  - RTS/CTS (Request to Send, Clear to Send)
- Vähemmän ahne kanavan käytön suhteen kuin CSMA
  - Kun kanava on vapaa, odota satunnainen aika ja kokeile uudestaan, sitten lähetä
  - Kun kanava on vapaa, lähetä signaali että lähetys alkaa ja kerro lähetyksen pituus



3) Vuoronantoprotokollat

- Yhdistä edellisten parhaita puolia
  - Älä pidä kapasiteettia turhaan varattuna
  - Älä aiheuta törmäystä
- Pollaus
  - Isäntäasema kyselee vuorotellen jokaiselta asemalta, onko sillä lähetettävää (vuorokysely, polling)
  - Isäntä kuuntelee signaalia, osaa päätellä, milloin lähetys loppuu
- Vuoromerkki
  - Se, jolla on vuoromerkki, saa lähettää
  - Jos ei ole lähetettävää, niin vuoromerkki siirtyy seuraavalle
- Kummastakin useita versioita
  - Ongelmia: lisäviive, 'single point of failure', ..
  - Montako kehystä yhdessä vuorossa saa lähettää



Linkkikerros

Linkkikerroksen osoitteet

Ch 5.4

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:

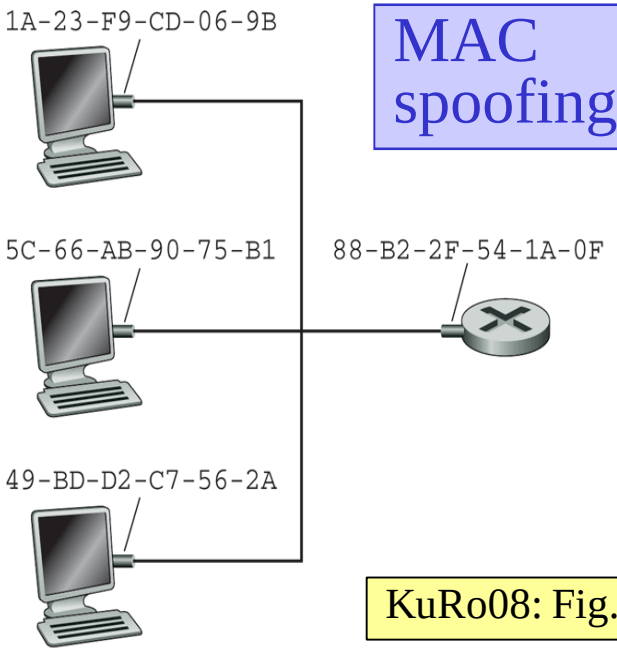
IP-osoite ~ katuosoite MAC-osoite ~ henkilötunnus

MAC-osoite

Lähes 300 biljoonaa erilaista osoitetta.

Lähes 17 miljoonaa valmistajanumeroa, kuhunkin mahdollista lähes 17 miljoonaa osoitetta.

- 48 bittinen (6 tavua)
  - 24 b kertoo valmistajan ja 24 b identifioi ohjainkortin (adapter)
  - IEEE jakaa valmistajanumerot
- Kiinteä
  - Liitetty mukaan valmistuksessa
  - Säilyy, vaikka laite toiseen verkkoon (toisin kuin IP-osoite)
- Ohjain
  - Kuulee kaikki kanavalla kulkevat kehykset
  - Välittää omalle koneelle vain sen MAC-osoitteella tai yleislähetysosoitteella FF-FF-FF-FF-FF-FF merkityt lähetykset



MAC spoofing

KuRo08: Fig. 5.16

mm. Ethernet, Bluetooth, IEEE 802.11 langattomat verkot käyttävät

Koneen MAC-osoitteen selvittäminen

- Reititys: Paketissa on IP-osoite
  - IP-osoitteen verkko-osa reitityksen perusteena
  - Paketti saapuu kohdeverkon reitittimelle
- Miten selvitetään IP-osoitetta vastaava MAC-osoite?
  - Oikea verkko saavutettu, mutta mille koneelle se pitäisi toimittaa?

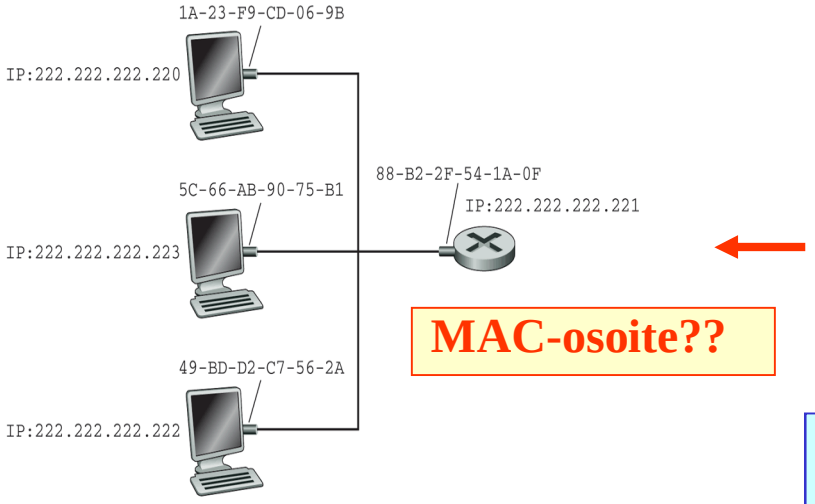
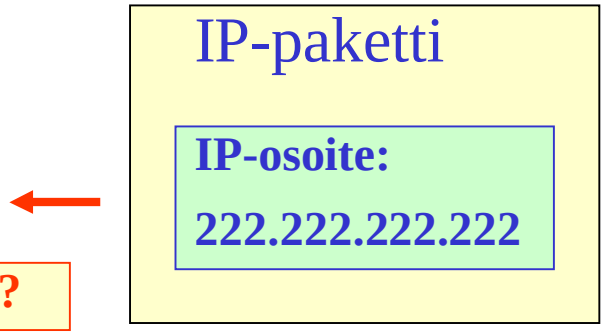


Figure 5.17 ♦ Each node on a LAN has an IP address, and each node's adapter has a MAC address.



Vastaava ongelma sovelluskerroksella: tiedossa domain-osoite (esim. URL), mutta tarvitaan IP-osoite!

Ratkaisu sovelluskerroksella on **DNS**!

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”?

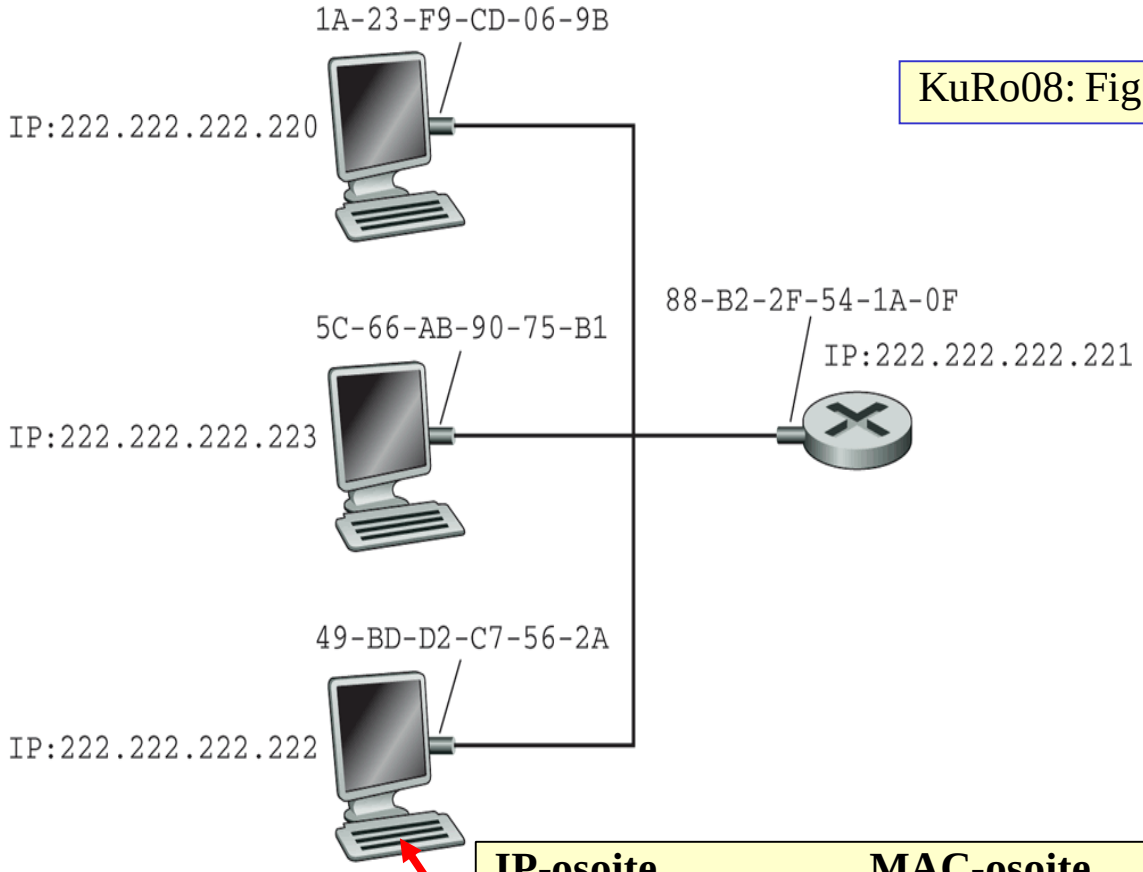
MAC-yleislähetysosoite: FF-FF-FF-FF-FF-FF

“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)

MAC-osoitteet ja ARP-taulu

Minkä kerroksen protokolla?

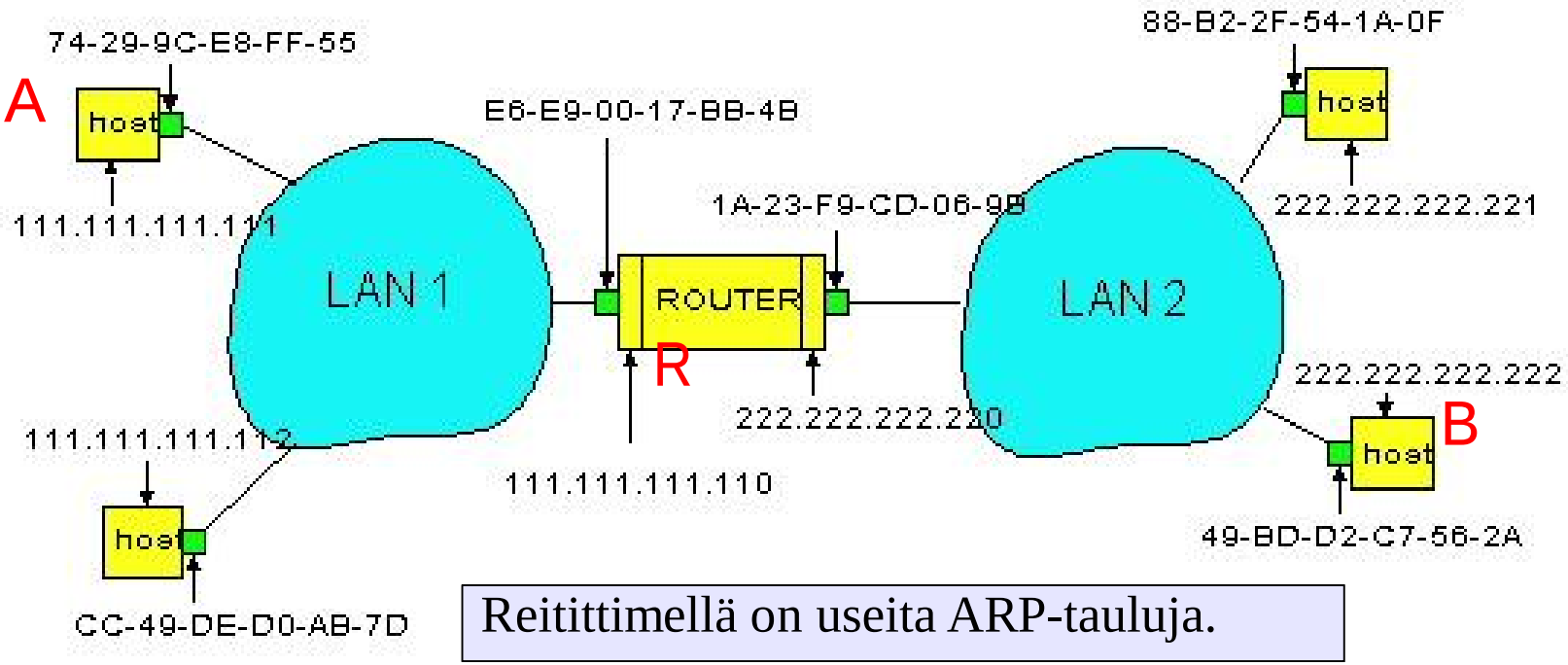


KuRo08: Fig 5.17 ja 5.18

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

Lähettäminen toiseen verkkoon (1)

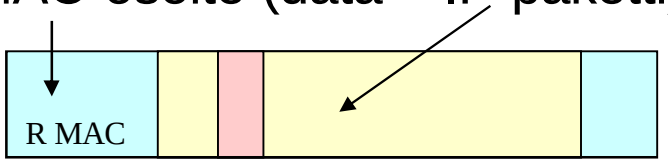
- Ensin omalle reitittimelle sen MAC-osoitteella ja reititin ohjaa eteenpäin
  - **Reititystaulussa on verkko-osoite, jonne paketti seuraavaksi ohjattava**
  - **Katso kohdeverkon ARP-taulusta kohteen MAC-osoite**
  - **Jos ei ole taulussa, tee ARP-kysely kohdeverkon koneille**



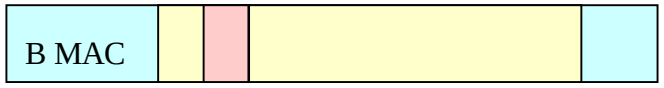
Reitittimellä on useita ARP-tauluja.

Lähettäminen toiseen verkkoon (2)

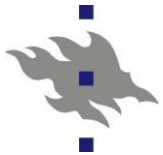
- **Lähettäjä A**
  - Muodosta IP-paketti, jossa Source IP = A, Dest. IP = B
  - Etsi ARP-taulusta **reitittimen** IP-osoitetta vastaava MAC-osoite
  - Luo siirtokehys, osoitteena reitittimen MAC-osoite (data = IP-paketti).
  - Verkkokortti lähettää siirtokehyksen.



- **Reititin R**
  - Verkkokortti ottaa siirtokehyksen vastaan.
  - Ota IP-paketti kehyksestä ja tutki otsakkeesta kohteen IP-osoite (B)
  - Katso reititystaulusta, mihin verkkoon seuraavaksi (mille reitittimelle)
  - Koska omassa verkossa, etsi kohdeverkon ARP-taulusta kohteen MAC-osoite



  - Muodosta siirtokehys, osoitteena B:n MAC-osoite (data = IP-paketti)
- **Vastaanottaja B**
  - Verkkokortti ottaa kehyksen vastaan; ohjaa IP-paketinverkkokerrokselle.



Linkkikerros

Ethernet

Ch 5.5

Ethernet

Ethernet Timeline (2003)

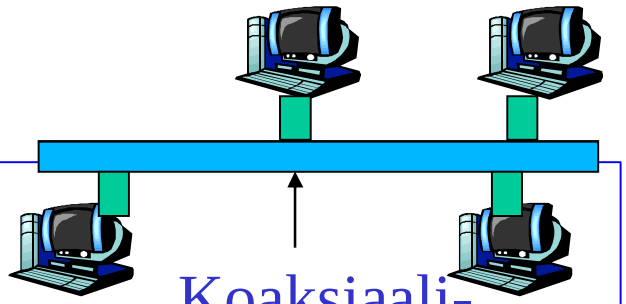
* 10 Megabit Ethernet 1990
* 100 Megabit Ethernet 1995
* 1 Gigabit Ethernet 1998
* 10 Gigabit Ethernet 2002
* 100 Gigabit Ethernet 2006**
* 1 Terabit Ethernet 2008**
* 10 Terabit Ethernet 2010**

- 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)

April 24, 2008
Terabit Ethernet around 2015

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

10BaseT ja 100BaseT



10 Mbps tai 100Mbps (Fast Ethernet, FE)

- T = Twisted Pair eli kierretty parikaapeli

Maks. etäisyys keskittimeen 100 m

Keskitin (hub) toistaa bitit heti sellaisenaan muille

- Fyysisen tason toistin (repeater); yleislähetys
- Signaalin vahvistus

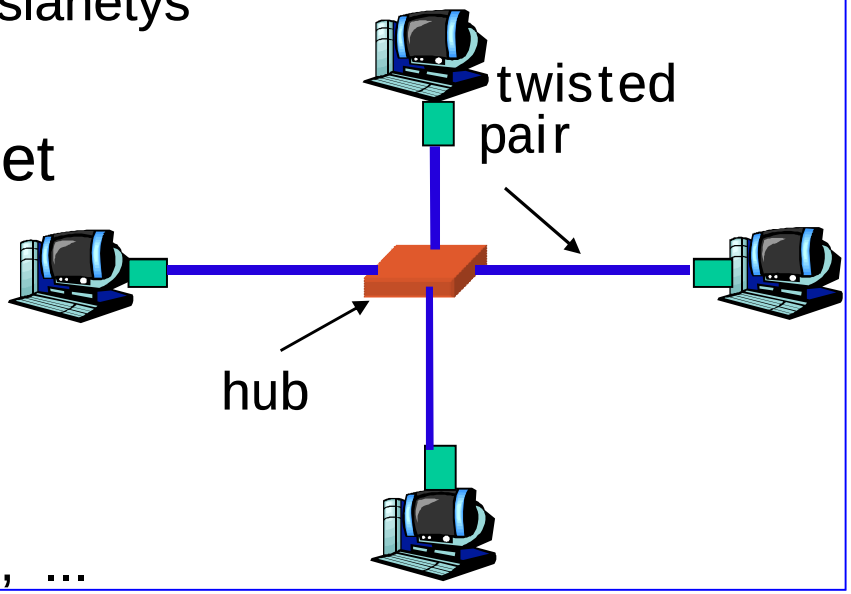
Verkkokortit käsittelevät törmäykset

- Maks. 30 konetta / keskitin

Keskitin osaa jättää huomiotta vikaantuneen kortin

Kerää myös tietoa liikenteestä

- Törmäysten lkm, keskim. kehyskoko, ...



Gigabitin Ethernet (GE)

1 Gbps tai 10 Gbps

Edelleen sama kehysformaatti

- Taaksepäin yhteensopiva

Yhteiskäyttöiset linkit edelleen OK

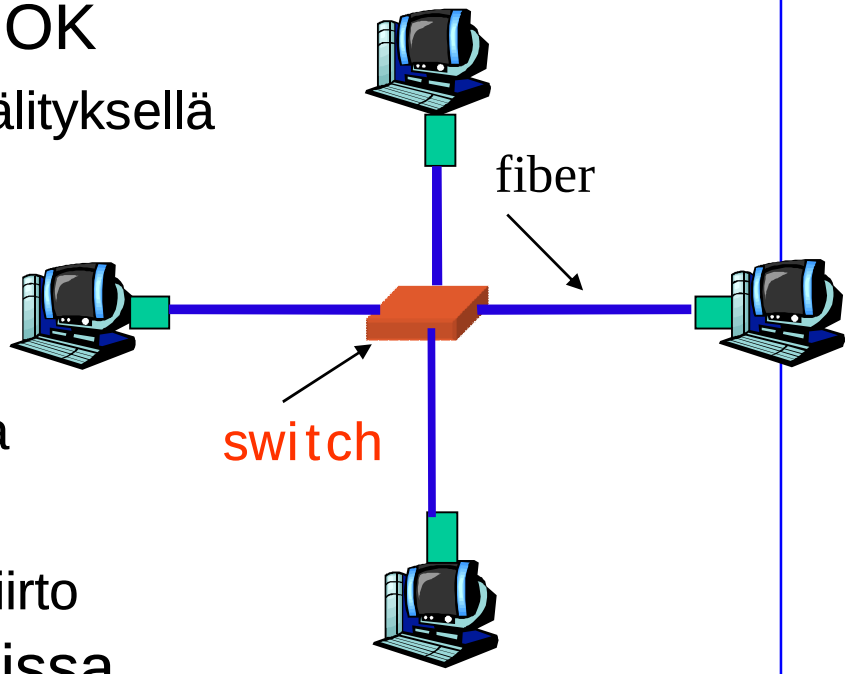
- Koneiden yhdistely keskittimen välityksellä
- CSMA/CD

Kaksipisteyhteydet

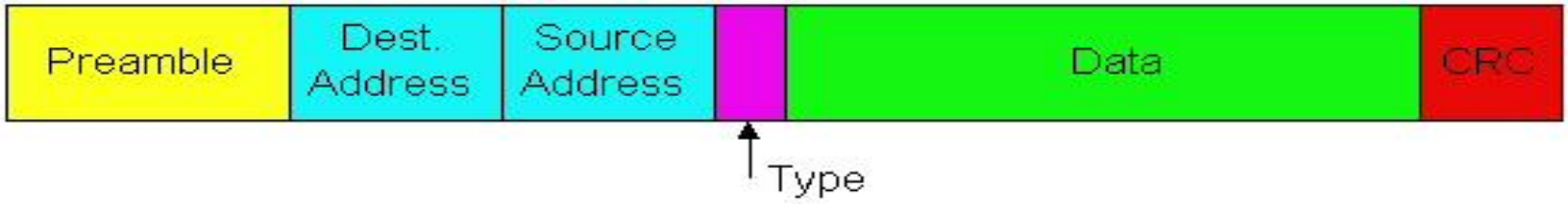
- ei törmäyksiä
- koneet yhdistetty **kytkimien** kautta
- pitkät välimatkat mahdollisia
- kaksisuuntainen täysivauhtinen siirto

Käytetään yleisesti runkoverkoissa

- verkkojen yhdistely (reititin -> reititin)
- valokaapeli, myös cat5/cat6 parikaapeli



Ethernet-kehys



Tahdistuskuvio (preample) (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)

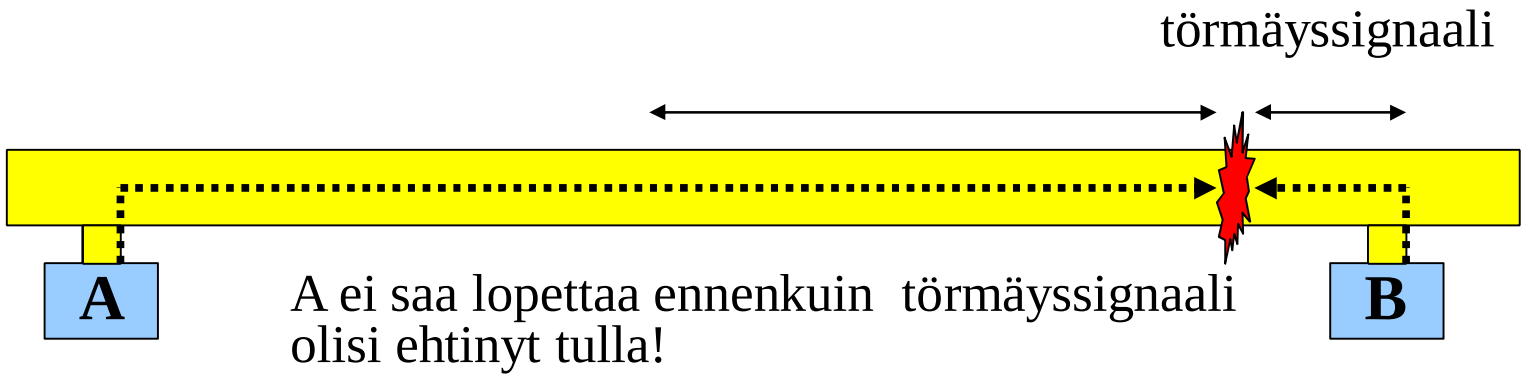
- Ethernet MTU = 1500 B

CRC (4 B eli 32 bittiä)

- tarkistusbitit, tahdistuskuvio mukana laskennassa

Kehyksen minimipituus

- Data-osan pituus min 46 B
  - Tarvittaessa täytetavuja (pad), jotka vastaanotto poistaa
- Lähettäjän **ehdittävä huomata mahdollinen törmäys**
  - Kehyksen lähetys ei saa päättyä ennenkuin alku on perillä ja mahdollinen törmäysääni kuuluu
    - Alku perillä -> loppukin onnistuu
- Lähetyksen minimikesto = 2* etenemisvive



Epäluotettava siirto

- Ethernet ei kättele, ei kuittaile
  - Uudelleenlähetys vain, jos törmäys
- Mutta tarkistussumma
  - Hylkää kehyksen, jos siirrossa virheitä

Verkkokerros

saa vain kelvollisia paketteja, antaa kuljetuskerrokselle

Kuljetuskerros

TCP: huolehtii luotettavuudesta

UDP: välistä voi puuttua segmenttejä

Sovelluskerros

Voi huolehtia halutessaan luotettavuudesta (esim. käytettäessä UDP-protokollaa)

Ethernet varaus: CSMA/CD

(klassinen Ethernet-verkko on yleislähetysverkko!)

- Carrier Sense
  - Kuuntele, onko väylä vapaa (96 b:n ajan)
  - 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

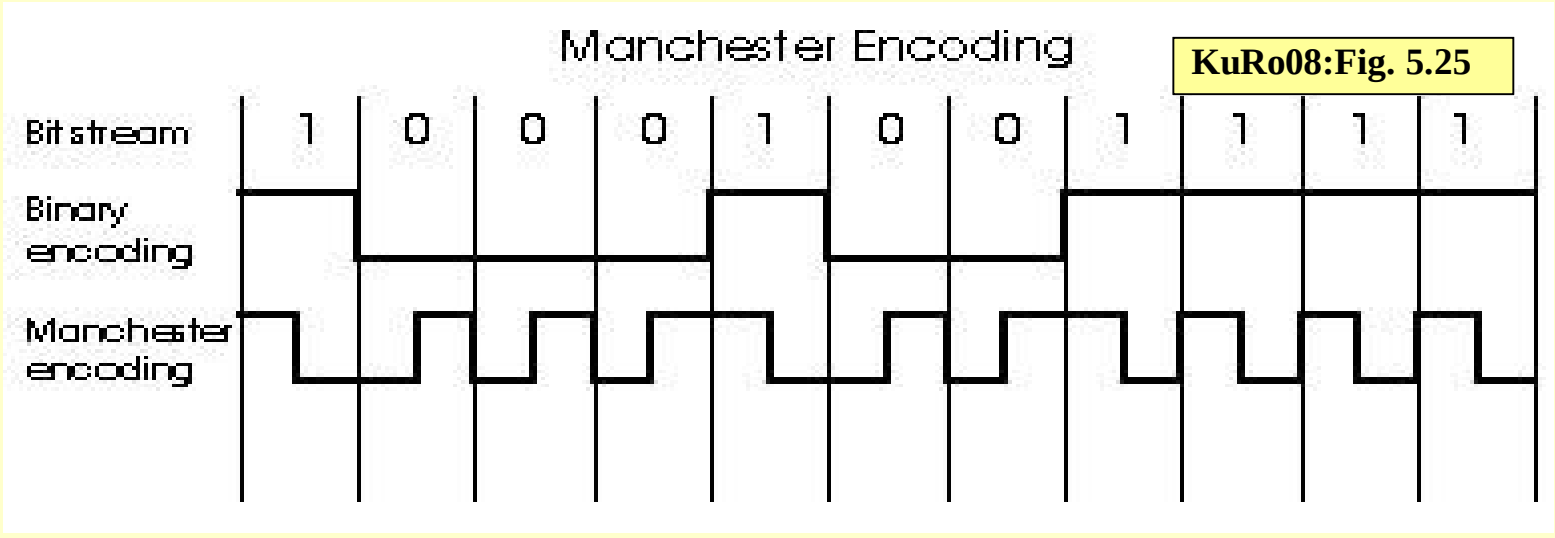
Törmäys

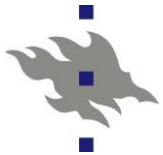
- **Binary Exponential Backoff**
  - Kun kuorma kasvaa eli törmäykset lisääntyvät, uudelleenyritysten väli kasvaa
- Odota törmäyksen jälkeen
  - N*(512 bitin = 64 tavun siirtoon kuluva aika = minimikehys)
  - 1. törmäys: N = 0 tai 1
  - 2. törmäys: N= 0,1,2 tai 3
  - k:s törmäys: N = 0, ... tai 2^k -1
  - 10. törmäyksen jälkeen ei enää kasvata väliä [0-1023]
  - 16 törmäyksen jälkeen luopuu ja ilmoittaa 'asiakkaalle' (eli verkkokerrokselle) epäonnistumisesta

10 Mbps:n linkillä 512 bitin siirtoon kuluu 51,2 mikrosekuntia

Signaalin koodaus

- Lähettäjän ja vastaanottajan kellopulssit on tahdistettava
- Manchester-koodaus (10BaseT)
  - Ethernetissä ei ole kellopulssia, tahdistus osana bittijonoa
  - Jännitemuutos aina keskellä bittiä
    - 1-bitti: ylhäältä alas, 0-bitti: alhaalta ylös





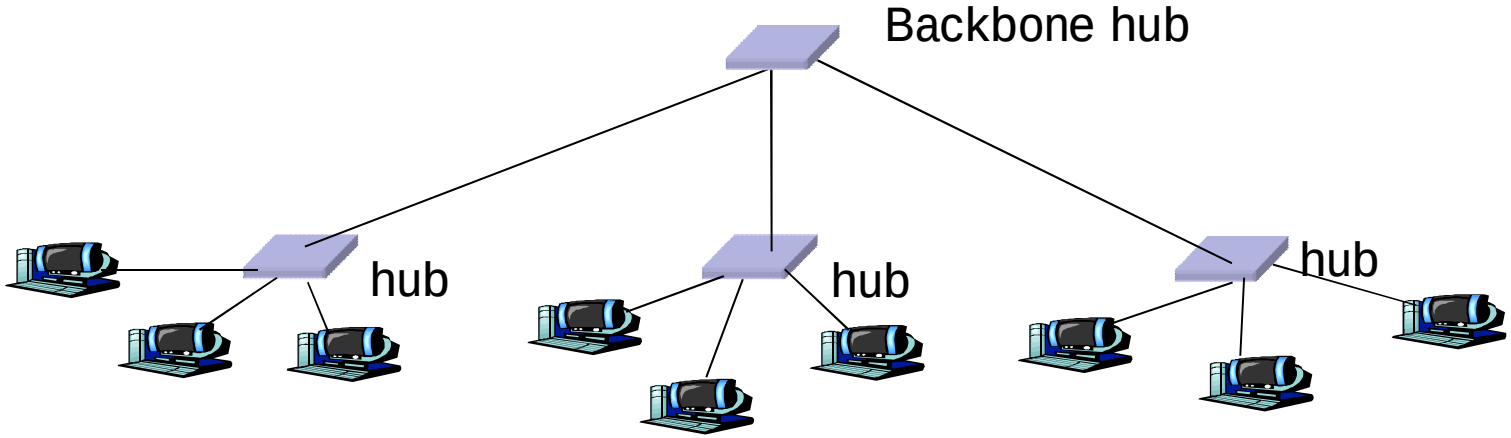
Linkkikerros

Keskitin, kytkin

Ch 5.6

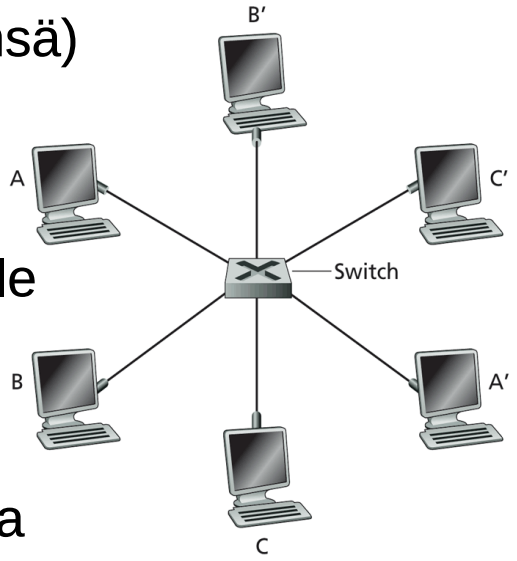
Keskitin (hub)

- Toimii fyysisellä kerroksella (layer-1)
  - Käsittelee bittejä
- Toistaa saamansa bitit heti kaikille muille linkeille
  - Signaalin vahvistus
- Yhteinen törmäysalue
  - Sopii vain pieniin verkkoihin
- Laitteet samanlaisia
  - Ei esim. 10 Mbps ja100 Mbps samaan keskittimeen



Kytkin (switch)

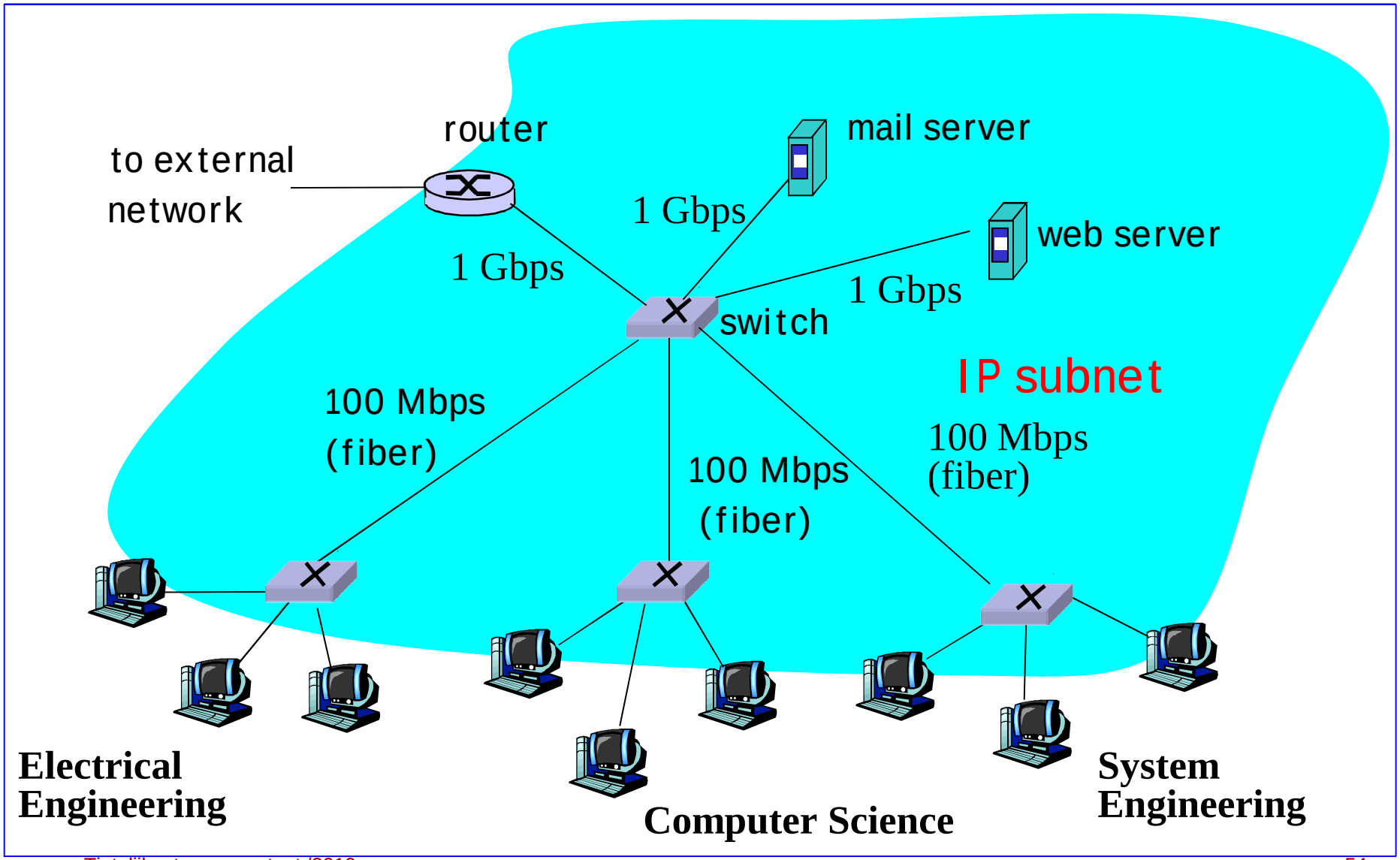
- Toimii linkkikerroksella (layer-2)
  - Käsittelee siirtokehyksiä, useita yhtäaikaisia yhteyksiä
- Vastaanottaa ja lähettää kokonaisia kehyksiä
  - Etappivälitys (store and forward) (yleensä)
- Ei törmäyksiä
  - Suora piuha koneelta kytkimeen
  - Kytkin lähettää ulos vain yhdelle piuhalle
- Voi yhdistää erilaisia verkkosegmenttejä
  - Kytkimessä esim. 10/100 Mbps portteja
  - Puskurointia
- Tuntumaton (transparent)
  - Sopeutuu itse verkon muutoksiin
  - 'plug-and-play, self-learning

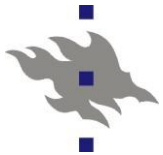


A switch providing dedicated Ethernet access to six hosts

KuRo08: Fig. 5.24

LAN, verkkosegmentit





Kytkin ja kehyksen välitys

- Miten kytkin osaa välittää kehyksen juuri oikeaan piuhaan?

- Se kerää itse ('oppii') tarvittavat tiedot

  takaperinoppimista (backward learning): saapuva kehys kertoo, mistä linkistä **lähettäjä s**aavutetaan

- Ylläpitää kytkentätaulukkoa
  (MAC-osoite, linkki, TTL)

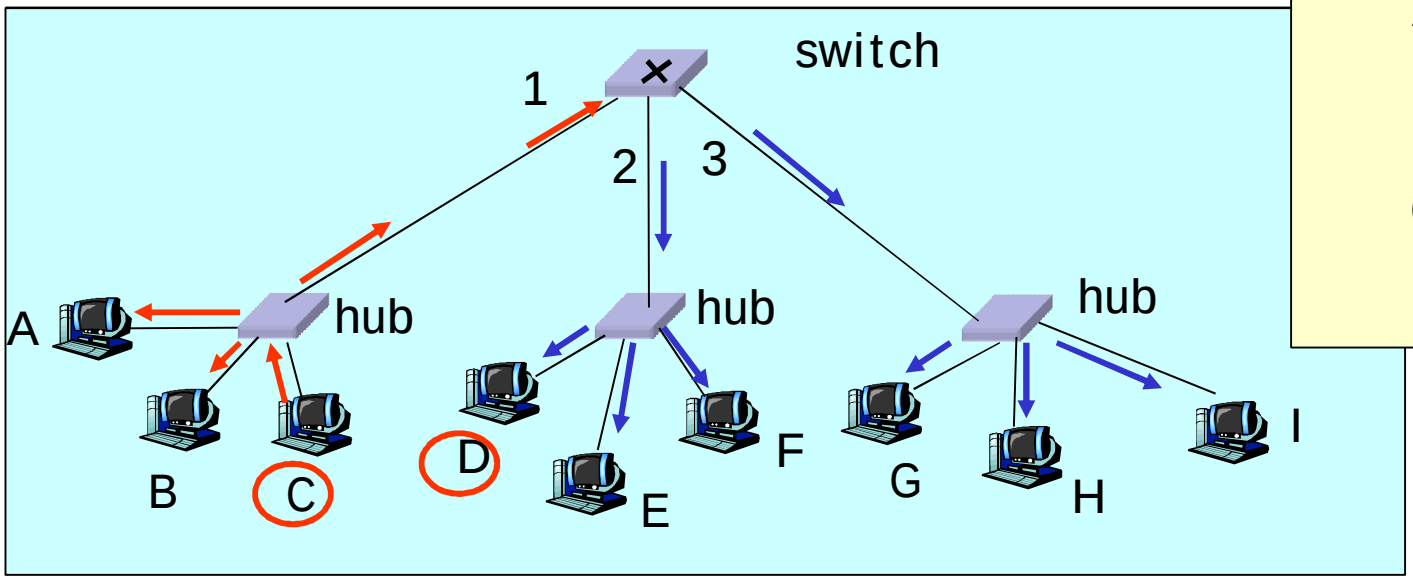
  TTL-aikaleima: poista ne, joita ei ole käytetty esim. 60 minuutin aikana

Kytkentätaulu (switching table)

- Aluksi taulu on tyhjä
- Saapuva kehys
  - **Lähteen MAC-osoite** x, kohteen MAC-osoite y, tuloportti p, yms
- Lähde X ei ole taulussa
  - Lisää (X, p ,TTL) tauluun eli **kytkin oppii, että osoite X on saavutettavissa portin p kautta**
- Lähde X on taulussa => päivitä TTL
- Kohde Y ei ole taulussa
  - Lähetetään kehys kaikkiin muihin portteihin = **tulvitus** (flooding)
  - Opitaan myöhemmin Y:n oikea portti jostain sen lähettämästä kehyksestä
- Lähde X ja kohde Y ovat jo taulussa
  - X ja Y samassa portisssa => hylkää kehys **(on jo oikeassa aliverkossa)**
  - X ja Y eri porteissa => lähetä kehys Y:n porttiin

Esimerkki

C lähettää kehyksen D:lle

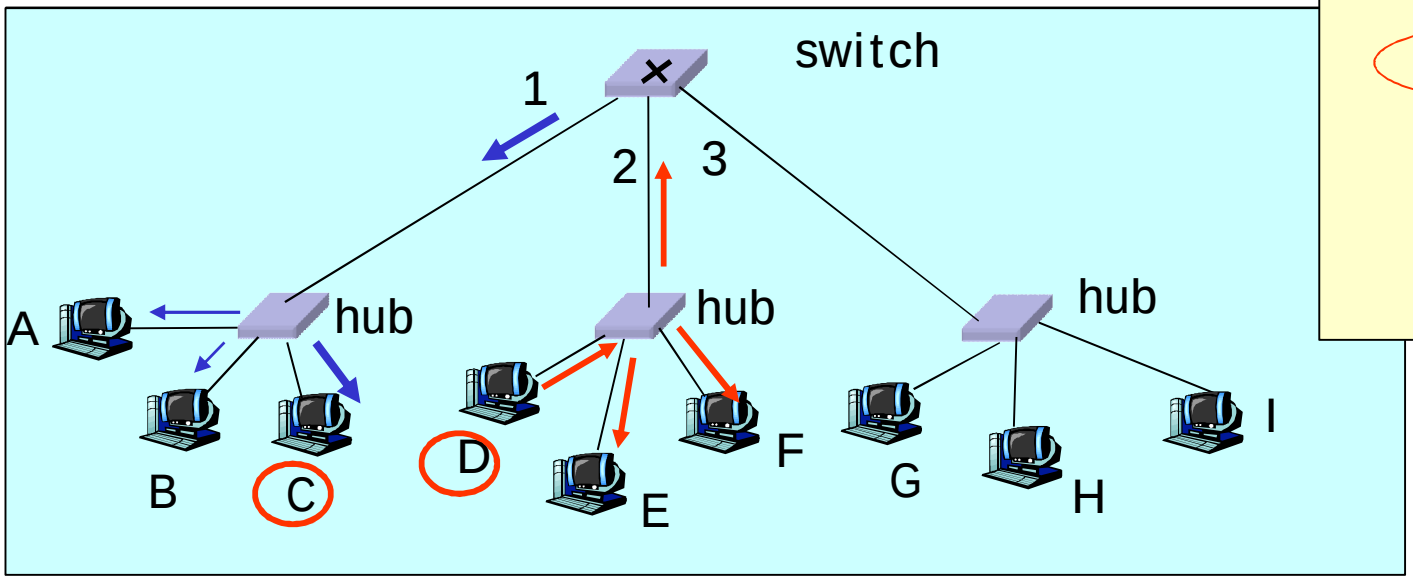


address	interface
A	1
B	1
E	2
G	3
C	1

- Kytkin vastaanottaa kehyksen (A ja B kuulevat myös)
  - Merkitsee tauluun C:n MAC-osoitteen ja portin 1
  - Koska D ei ole taulussa, tulvittaa linkeilla 2 ja 3.
- D vastaanottaa kehyksen (E, F, G, H, I kuulevat myös)

Esimerkki jatkuu

D lähettää kehyksen C:lle



address	interface
A	1
B	1
C	1
E	2
G	3
D	2

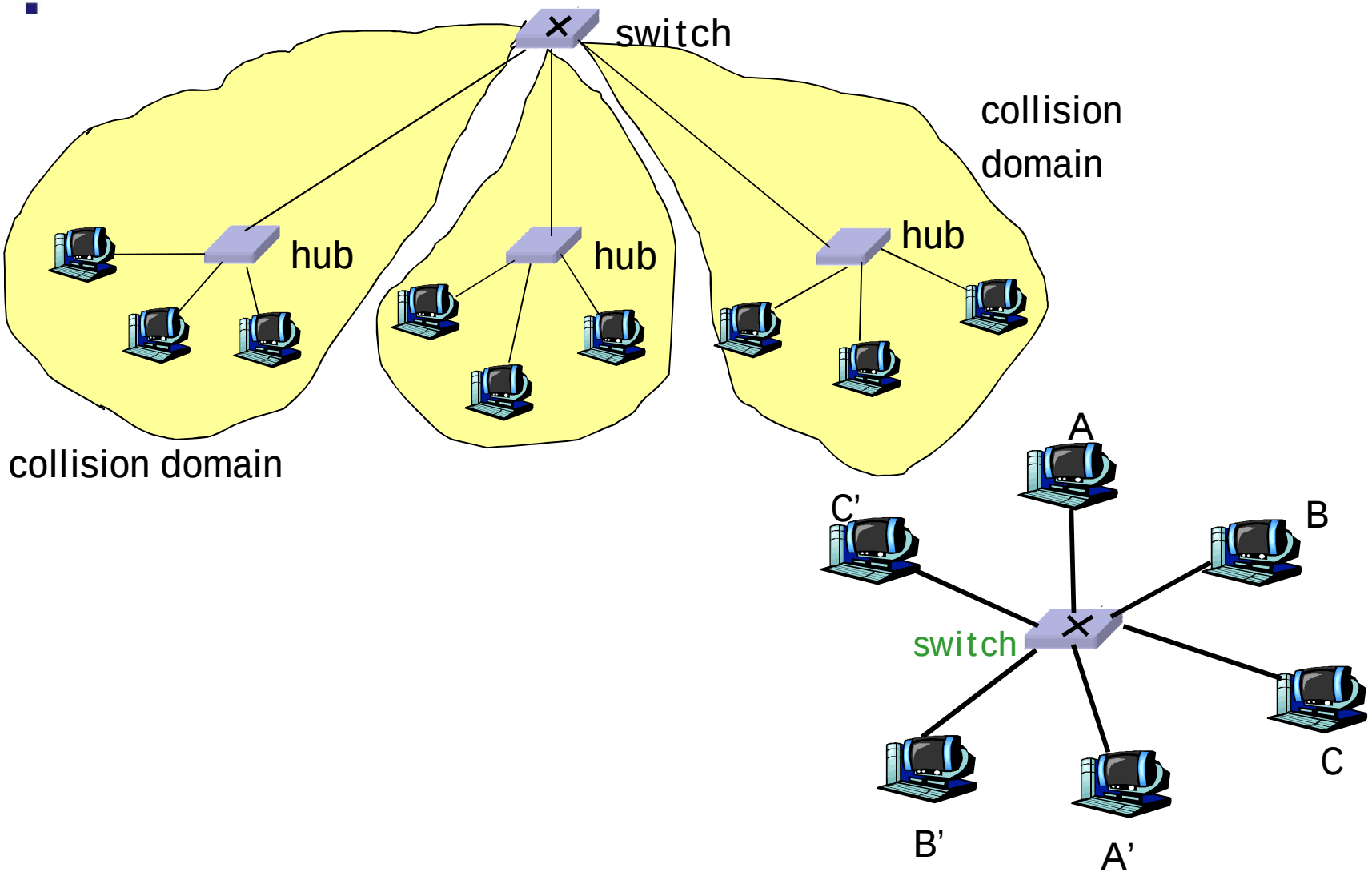
- Kytkin vastaanottaa kehyksen (E ja F kuulevat myös)
  - Merkitsee tauluun D:n MAC-osoitteen ja portin 2
  - C:n osoite on taulussa, joten lähettää kehyksen linkkiin 1
- C vastaanottaa kehyksen (A ja B kuulevat myös)

Tulvitus (flooding)

- Tulvitus voi olla ongelma
  - Kehykset voivat jäädä kiertämään silmukoissa
  - Koko verkko tukkeutuu
- Siis silmukoita ei saa muodostua!

- Verkon loogisen rakenteen pitää olla puu.
  - Virittävä puu (Spanning tree)
  - Lyhyimmin poluin virittävä puu Dijkstran algoritmilla

Rajoitetut törmäysalueet / ei törmäyksiä



Suorakytkentä (cut-through switching)

- Jotkut kytkimet voivat välittää kehyksen bitit ulos sitä mukaa kuin itse ne saavat
  - Välityspäätöksen tekoon riittää tutkia otsakkeesta kohdeosoite
  - Ei siis enää etappivälitteistä (store-and-forward)
- Pienentää latenssiaikaa
  - Ei kuitenkaan mahdottomasti ...
  - 100 Mbps:n linjalla odotusta maksimissaan noin 0.12 ms

Vertailua

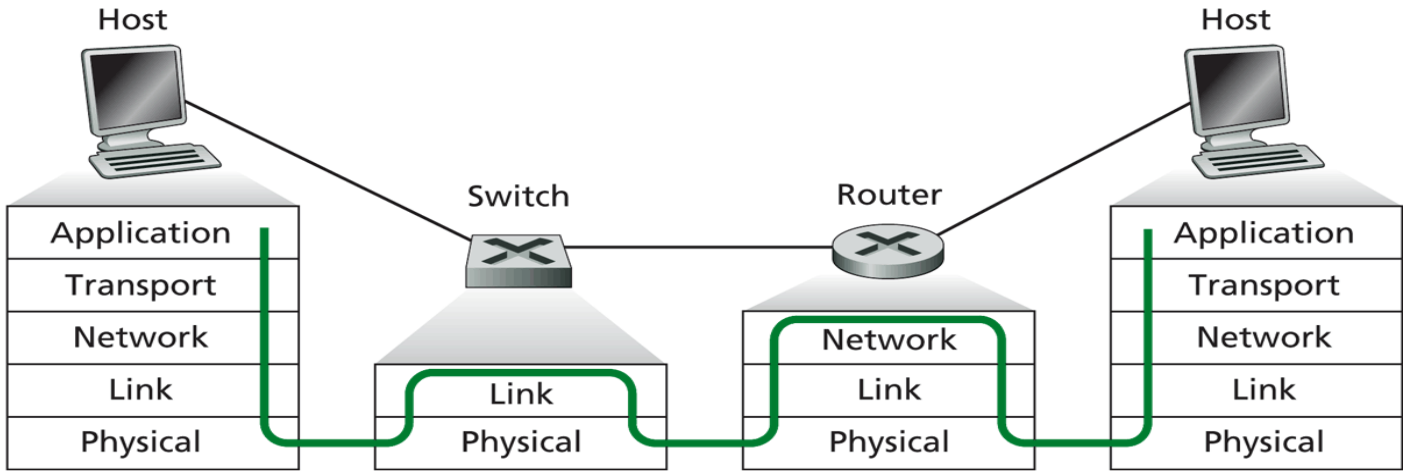


Figure 5.33 ♦ Packet processing in switches, routers, and hosts

	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