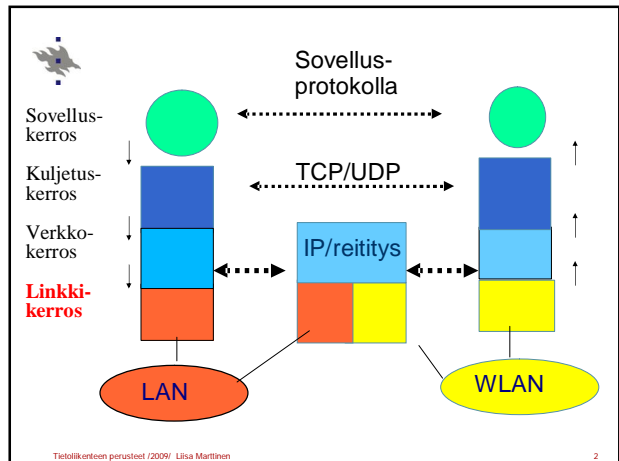


# Tietoliikenteen perusteet

## Linkkikerros

Kurose, Ross: Ch 5.1- 5.6



# Sisältö

- Linkkikerroksen tehtävät
- Virheiden havaitseminen ja korjaaminen
- Yhteiskäyttöisen kanavan varaus
- Osottaminen 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

## 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

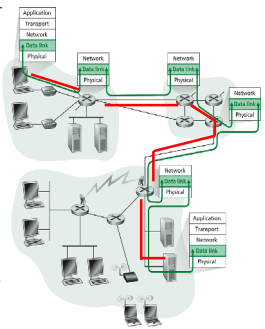
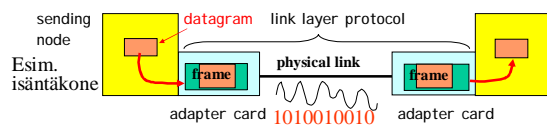


Figure 5.1 • The link layer

# Linkkikerroksen tehtäviä (2)

**NIC (Network Interface Card) linkki- ja fyysinen kerros**

- **Vuonvalvonta, puskuointi**  
Kytkimessä on useita erinopeuksia linkkejä
- **Virhevalvonta**  
signaali vaimenee, taustakohina häiritsee, ...  
Kehyksessä on tarkistusbittejä (error detection and correction bits)  
Vastaanottava solmu korjaa, jos pystyy  
Jos ei pysty, pyytää uudelleen tai häviää
- **Yksisuuntainen /kaksisuuntainen liikenne**  
Yksisuuntainen: lähetysvuorojen hallinta



## 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 liitettyä useita laitteita  
Käytössä laitetason MAC-osoite (Medium access control)

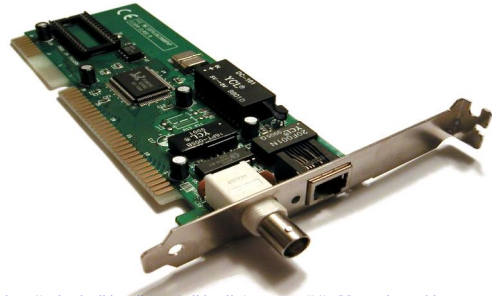
### Yhteisen linkin varaus ja käyttö (link access)

Esim. langaton linkki, keskitimiin yhdistetyt linkit

### Luotettava silto

Langattomilla linkeillä suuri virhetodennäköisyys  
Linkkitaso huolehtii oikeellisuudesta  
Miksi tästä täytyy huolehtia vielä kuljetuserroksella?  
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](http://upload.wikimedia.org/wikipedia/commons/9/9e/Network_card.jpg)

## Linkkikerros

# Virheiden havaitseminen ja korjaaminen

Ch 5.2

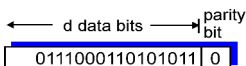
## Bittitason virheet

- 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, satelliityhteydet, digitelevisio, ... (Reed-Salomon-koodi)
- Tietoliikenne yleensä tyytyväinen vain havaitsemaan virheet
  - Virheelliset hylätään ja korjauksena on uudelleenlähetyks

## Pariteettitarkistus

### Pariteettibitti

Parillinen vs. pariton pariteetti  
Virheryöppyssä jopa 50% voi jäädä huomaamatta

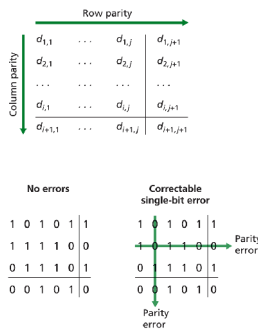


### Kaksiulotteinen pariteetti

Eriksen horisontaalinen (parillinen) ja vertikaalinen (pariton) pariteetti  
Pystyy korjaamaan yhden bitin virheen.

### Hamming-koodi

Korjaa yhden bitin virheen



## Tarkistussumma

### Internet-checksum

Yhteenlasketaan 16 bitin kokonaisuuksia, yhden komplementti  
Kuljetuserros laskee ja tarkastaa UDP- ja TCP-protokollissa  
Ei ole kovin tehokas; linkkikerros ei käytä

### CRC (cyclic redundancy check)

Yleisesti linkkikerroksella käytetty virheenpaljastusmenetelmä, helppo toteuttaa laitteistotasolla, luotettava  
Perustuu polynomien aritmetiikkaan  
Tunnetaan myös nimellä polynomikoodi  
Useita tarkistusbittejä; havaitsee usean bittivirheen ryöpyin.



## 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

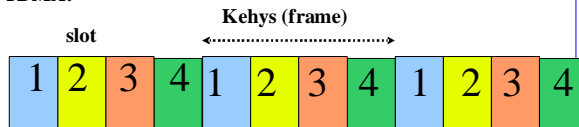
- 1) Kanavanjakoprotokollat (channel partitioning protocol)  
Jaa kanavan käyttö "viipaleisiin" (time slots, frequency, code)  
Kukin solmu saa oman viipaleensa  
TDMA, FDMA, CDMA  
"käytä sinä tätä puolta, minä tätä toista"
- 2) Kilpailuprotokollat (random access protocols)  
"Se ottaa, joka ehtii."  
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:

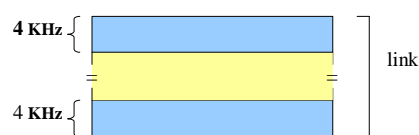


## Kanavanjako: FDMA

### 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**

#### FDMA:



## Kanavanjako: CDMA

### CDMA (Code Division Multiple Access)

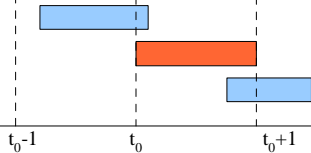
- Radiolinjoilla käytettävä koodinjakoon perustuva protokolla
  - Matkapuhelimet, ...
  - Kullakin asemalla oma tapansa koodata bitit 1 ja 0 (oma sirukoodi)
- Asemat voivat lähettää yhtäaikaan koko kanavan taajuudella
  - Kaikkien signaalit saavat yhdistyä linkillä
  - Asemat pystyvät erottelemaan yhteissignaalista 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 yhtäaikaan => 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)
  - CSMA/CA (collision avoidance)

## Aloha

- Hawajilla, 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%



Tietoliikenteen perusteet / 2009/ Liisa Marttinen

25

## Viipaloitu Aloha (slotted Aloha)

- Lähetysaika jaettu aikaviipaleiksi (slot)
  - Kaikki siirtokehukset samankokoisia => siirtoaika aina vakioittainen
- Lähetys voi alkaa vain aikaviipaleen alussa
  - Törmäykset täydellisiä => törmäysaika = yhden aikaviipaleen mittainen
- Solmut synkronoitava: aikaviipaleen alku
- Jos törmäys, niin kaikki solmut huomaavat
  - Uudelleenyritys seuraavalla 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

Tietoliikenteen perusteet / 2009/ Liisa Marttinen

26

## Lähetyskanavan kuuntelu

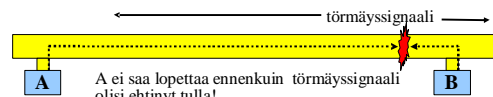
- Kuuntele ennenkuin lähetät
  - Asema tutkii, onko kanava jo käytössä (carrier sense)
  - Jos siirtotie on vapaa, saa lähettää
  - Jos siirtotie on varattu, odota satunnainen aika ja yritä uudelleen
- Ei aina paljasta jo alkanutta lähetystä
  - Etenemiskiiveen takia ei huomata toisen signaalia ajoissa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

Tietoliikenteen perusteet / 2009/ Liisa Marttinen

27

## 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
- Kuunko kuunneltava?
  - 2\* maksimi etenemiskiive solmujen välillä

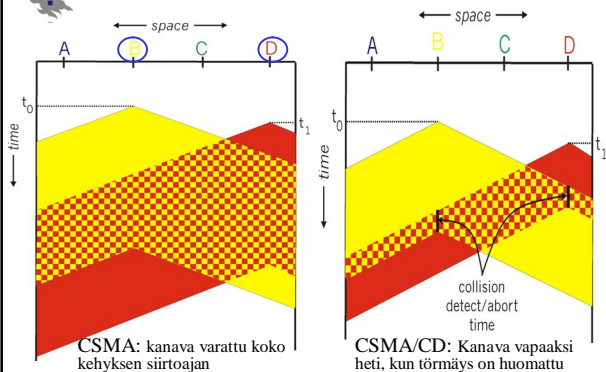


Tietoliikenteen perusteet / 2009/ Liisa Marttinen

28

## CSMA ja törmäys

KuRo08: Fig. 5.13 ja 5.14



Tietoliikenteen perusteet / 2009/ Liisa Marttinen

29

## 3) Vuoronantoprotokollat

- Yhdistä edellisten parhaita puolia
  - Älä pidä kapasiteettia turhaan varattuna
  - Älä aiheuta törmäystä
- Pollaus
  - Isäntäasema kysyy 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

Tietoliikenteen perusteet / 2009/ Liisa Marttinen

30

## 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

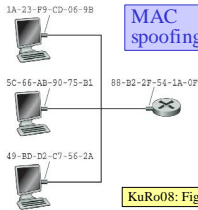
Analogia:

IP-osoite ~ katuosoite    MAC-osoite ~ henkilötunnus

## MAC-osoite

Lähes 300 miljoonaa 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



KuRo08: Fig. 5.16

mm. Ethernet, Bluetooth, IEEE 802.11 langattomat verkot käyttävät

## Koneen MAC-osoitteen selvittäminen

- Reiitys: 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?

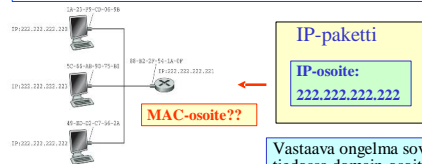


Figure 5.17 Each node on a LAN has an IP address, and each node's adapter has a MAC address.

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 kyselyllään 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"?"

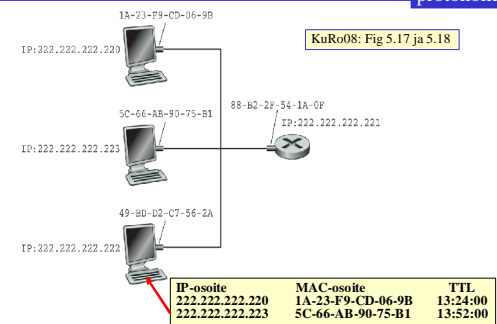
MAC-yleislähetysosoite: FF-FF-FF-FF-FF-FF

"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

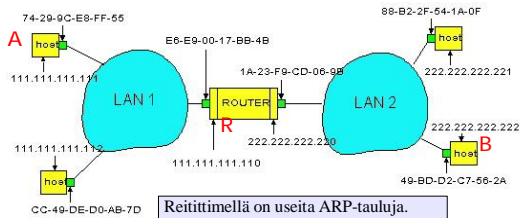
Minkä kerroksen protokolla?



KuRo08: Fig 5.17 ja 5.18

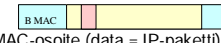
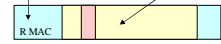
## Lähtettäminen toiseen verkkoon (1)

- Ensimmäiselle 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



## Lähtettäminen toiseen verkkoon (2)

- Lähtettäjä 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 siirtokehksen.
- Reititin R**
  - Verkkokortti ottaa siirtokehksen 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-paketin verkkokerrokselle.



## Linkkikerros

# Ethernet

Ch 5.5

## 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 colinventor) gave a keynote speech, "Toward Terabit Ethernet."

## 10BaseT ja 100BaseT

10 Mbps tai 100Mbps (Fast Ethernet, FE)  
T = Twisted Pair eli kierretty pariakaapeli  
Maks. etäisyys keskittimeen 100 m

**Keskitin (hub)** toistaa bitit heti sellaisenaan muille

Fyysisen tason toistin (repeater); yleislähetys  
Signaalin vahvistus

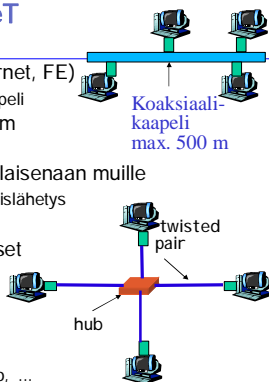
Verkkokortit käsittelevät törmäykset

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

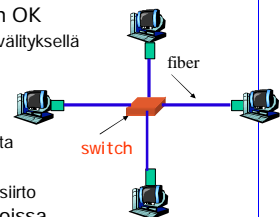
pitkät välimatkat mahdollisia

kaksisuuntainen täysivauhtinen siirto

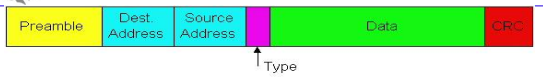
Käytetään yleisesti runkoverkoissa

verkkojen yhdistely (reititin -> reititin)

valokaapeli, myös cat5/cat6 pariakaapeli



## 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 eli 32 bittinä)

tarkistusbitit, tahdistuskuvio mukana laskennassa

## Kehyksen minimipituus

n Data-osan pituus min 46 B

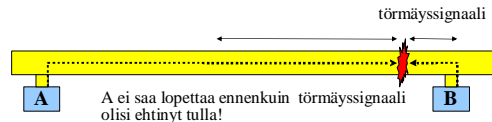
n Tarvittaessa täytetäviä (pad), jotka vastaanotto poistaa

n Lähettäjän **ehdittävä huomata mahdollinen törmäys**

n Kehyksen lähetyksen ei saa päättyä ennenkuin alku on perillä ja mahdollinen törmäysääni kuuluu

- Alku perillä -> loppukin onnistuu

n Lähetyksen minimikesto =  $2 \times$  etenemisvive



## Epäluotettava siirto

n Ethernet ei kätele, ei kuittaile

n Uudelleenlähetyksen vain, jos törmäys

n Mutta tarkistussumma

n Hylkää kehyksen, jos siirrossa virheitä

Verkkokerros

saa vain kelpoisia 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)

n Carrier Sense

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

n Jos vapaa, lähetä heti

n Muuten odota ja lähetä, kun linja vapautuu

n Collision Detection

n Kun lähetetty, kuuntele onnistuiko

n Törmäys?

n Huomaa signaalin voimakkuudesta

n Lopeta kehyksen lähetyksen heti

n Lähetä 48 bitin soikussignaali (jam) => muutkin huomaavat varmasti

n Random Access

n Odota törmäyksen jälkeen satunnainen aika

## Törmäys

n Binary Exponential Backoff

n Kun kuorma kasvaa eli törmäykset lisääntyvät, uudelleenyritysten väli kasvaa

n Odota törmäyksen jälkeen

n  $N \times (512 \text{ bitin} = 64 \text{ tavun siirtoon kuluva aika} = \text{minimikehyksen})$

n 1. törmäys:  $N = 0$  tai  $1$

n 2. törmäys:  $N = 0, 1, 2$  tai  $3$

n k:s törmäys:  $N = 0, \dots$  tai  $2^k - 1$

n 10. törmäyksen jälkeen ei enää kasvata väliä [0-1023]

n 16 törmäyksen jälkeen luopuu ja ilmoittaa 'asiakkaalle' (eli verkkokerrokselle) epäonnistumisesta

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

## Signaalin koodaus

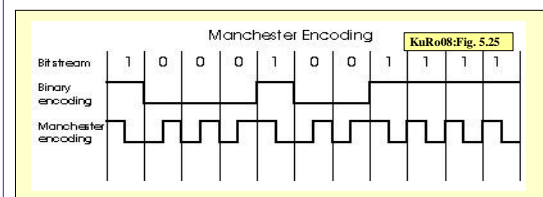
n Lähettäjän ja vastaanottajan kellopulssit on tahdistettava

n Manchester-koodaus (10BaseT)

n Ethernetissä ei ole kellopulssia, tahdistus osana bittijonoa

n Jännitemuutos aina keskellä bittinä

- 1-bitti: ylhäältä alas, 0-bitti: alhaalta ylös





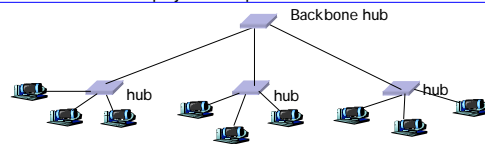
## Linkkikerros

# Keskitin, kytkin

Ch 5.6

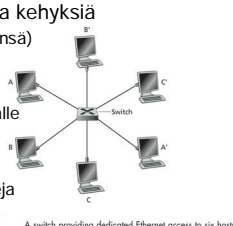
## Keskitin (hub)

- Toimii fyysisellä kerroksella (layer-1)
  - Käsittelee bittejä
- Toistaa saamansa bitit heti kaikille muille linkeille
  - Signaalin vahvistus
- Yhteinen törmäysalue
  - Sopii vain pieniin verkkoihin
- Yhdistää vain samanlaisia laitteita
  - Ei esim. 10 Mbps ja 100 Mbps samaan keskittimeen



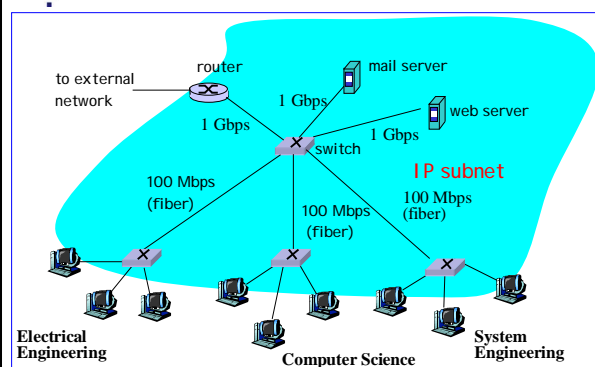
## Kytkin (switch)

- Toimii linkkikerroksella (layer-2)
  - Käsittelee siirtokehyksiä, useita yhtäaikaista 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



KuRo08: Fig. 5.24

## LAN, verkkosegmentit



## Kytkin ja kehyksen välitys

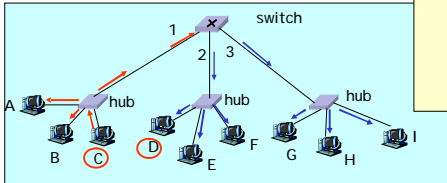
- Miten kytkin osaa välittää kehyksen juuri oikeaan piuhaan?
- Se kerää itse ('oppii') tarvittavat tiedot
  - **takaperinoppimista** (backward learning): saapuva kehys kertoo, mistä linkistä **lähettäjä** saavutetaan
- Ylläpitää kytkentätaulukkoa (MAC-osoite, linkki, TTL)
  - TTL-aikaleima: poista ne, 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 = **tuulitus** (flooding)
  - Opetaan myöhemmin Y:n oikea portti jostain sen lähettämästä kehyksestä
- Lähde X ja kohde Y ovat jo taulussa
  - X ja Y samassa portissa => hylkää kehys (**on jo oikeassa aliverkossa**)
  - X ja Y eri portteissa => lähetä kehys Y:n porttiin

## Esimerkki

C lähettää kehyksen D:lle

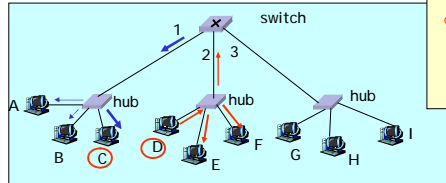


address	interface
A	1
B	1
E	2
G	3
C	1

- r Kytin vastaanottaa kehyksen (A ja B kuulevat myös)
  - m Merkitsee tauluun C:n MAC-osoitteen ja portin 1
  - m Koska D ei ole taulussa, tulvittaa linkeilla 2 ja 3.
- r 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

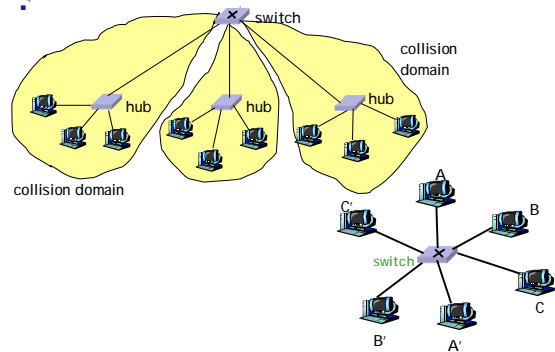
- r Kytin vastaanottaa kehyksen (E ja F kuulevat myös)
  - m Merkitsee tauluun D:n MAC-osoitteen ja portin 2
  - m C:n osoite on taulussa, joten lähettää kehyksen linkkiin 1
- r C vastaanottaa kehyksen (A ja B kuulevat myös)

## Tulvitus (flooding)

Tulvitus voi olla ongelma

- n Kehykset voivat jäädä kiertämään silmukoissa
- n Koko verkko tukkeutuu
- n Siis silmukoita ei saa muodostua!
- n Verkon loogisen rakenteen pitää olla puu.
  - n Virittävä puu (Spanning tree)
  - n Lyhyimmän polun virittävä puu Dijkstran algoritmilla

## Rajoitetut törmäysalueet / ei törmäyksiä



## Suorakytkentä (cut-through switching)

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

## Vertailua

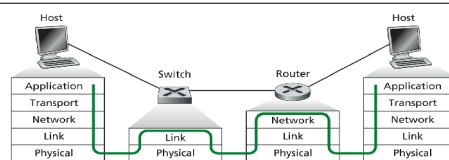


Figure 5.33 ♦ Packet processing in switches, routers, and hosts

	Keskitin (hub)	Kytin (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

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

ks. kurssikirja s. 501

