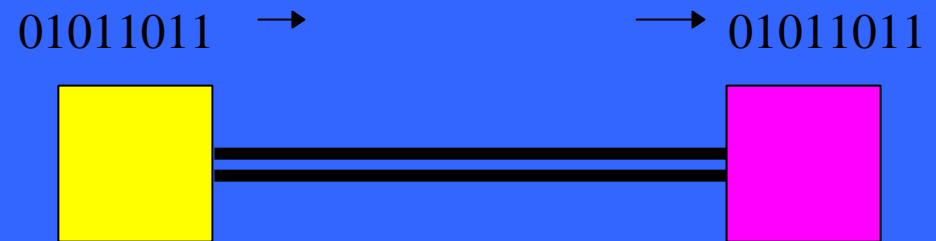


5. Siirtoyhteyskerros

linkkikerros (Data Link Layer)

- yhtenäinen linkki solmusta solmuun

– bitit sisään => bitit ulos



- ongelmia:

– siirtovirheet

- havaitseminen
- korjaaminen

– solmun kapasiteetti

- vuonvalvonta

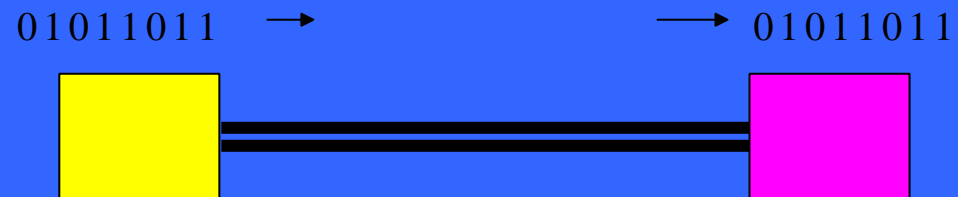
– yhteisen kanavan käyttö

5.1. Kaksipisteystehtydet

Virhevalvonta

- * virheiden havaitseminen
- * virheiden korjaus

Vuonvalvonta



Bittivirta \Leftrightarrow kehyksiä

- tavoite
 - bittivirheiden hallinta
 - muuttuu
 - katoaa
 - monistuu
- bittivirta kehyksinä
- kehys tarkistettavissa
 - tarkistustietoa

Kehysten kuljetus

- **tavoite**
 - kaikki kehykset
 - kukin kehys virheettömästi
 - lähetysjärjetyksessä
- **vastaanottaja kertoo lähettäjälle**
 - **ACK: kehys vastaanotettu ok**
 - tietty kehys
 - kaikki kehykset tähän asti
 - **NAK: kehyksessä vikaa => lähetettävä uudelleen**
 - **Saako lähettää lisää vai pitääkö keskeyttää**
 - **vuonvalvonta**

Virheet

- **Kahdenlaisia virheitä:**
 - yhden bitin virheet
 - usean peräkkäisen bitin vääristyminen (burst error)
- **Virheiden esiintymistiheys**
 - BER (bit error rate)
 - mitä suurempi BER, sitä lyhyempiä kehyksiä kannattaa käyttää

Missä virhe hoidetaan?

- **kuittaava linkkikerros havaitsee virheet ja korjaa ne**
- **yhteydetön, kuittaamaton & virhe**
=> kuljetuskerros havaitsee ja korjaa
- **ja jos ei, niin sovelluskerros havaitsee ja korjaa**
- **ja jos ei, niin asiakas havaitsee ja korjaa**

Virheiden havaitseminen ja korjaaminen

Virheiden takia dataan lisäinformaatiota:

- **virheen korjaamiseksi** (error-correcting code, forward error correction (FEC))
 - lisäinformaatiota niin paljon, että vastaanottaja sekä havaitsee että kykenee itse korjaamaan virheen
- **virheen havaitsemiseksi** (error-detecting code, feedback/backward error control)
 - lisäinformaatiota, jotta vastaanottaja havaitsee virheen tapahtuneen => korjauksena **uudelleenlähetys**

Virheen korjaus/havaitseminen

- **virheen korjaava koodaus**
 - **kallis koko ajan**
 - paljon lisäinformaatiota
 - **rajoitettu korjauskyky**
 - esim. kokonaan kodonnut kehys
- **virheen havaitseva koodaus**
 - **virheen sattuessa kallis**
 - uudelleen lähettäminen maksaa
 - uudelleen lähettäminen on hidasta

Virheen korjaus

- Käytetään esim.

- CD- ja DVD-levyissä, digitaalitelevisiossa
- nopeissa modeemeissa, kannettavissa puhelimissa
- satelliittiyhteyksissä, avaruusluotaimissa

- Esimerkkejä

- Hamming-pariteettitarkistus (Tito-kurssilla)
 - pystyy korjaamaan yhden virheellisen bitin
 - virheryöpyn, jos se jaetaan yhden bitin virheiksi
- Reed-Solomon -koodit
 - lohkokoodija , jotka pystyvät korjaamaan virheryöppyjä

Virheen havaitseminen

- Pariteettibitti
 - parillinen pariteetti
 - pariton pariteetti
- horisontaaliset ja vertikaaliset pariteetit
- Internet tarkistussumma
- CRC (Cyclic redundancy code (tai check))
 - yleisesti käytetty virheen paljastusmenetelmä
 - perustuu polynomien aritmetiikkaan (modulo2-aritmetiikkaan, XOR)
 - useita tarkistusbittejä => havaitaan usean bittivirheen ryöppy

Pariteetti

- esimerkki yksinkertaisesta virheen havaitsevasta koodista
- jokaiseen merkkiin lisätään yksi ylimääräinen ns. **pariteettibitti**
 - lisäyksen jälkeen kaikissa merkeissä on parillinen (tai jos niin sovitaan pariton) määrä ykkösiä
- paljastaa kaikki yhden bitin virheet
 - kehyksen pituudesta riippumatta
- ei paljasta kahden bitin virheitä

Pariteettibitin käyttö

- erityisesti asynkronisessa tiedonsiirrossa merkkejä siirrettäessä
- käytännössä paljastaa noin puolet virheellisistä bittijonoista
 - esim. modeemeissa syntyy useita virheitä
 - linjahäiriöt aiheuttavat usein pitkiä virheryöppyjä

Horisontaaliset ja vertikaaliset pariteetit

- järjestetään bittijono kaksiulotteiseen taulukkoon
- lasketaan pariteetti jokaiselle vaaka- ja pystyriville

1 0 0 1 0 1 0		1	
0 1 1 1 0 1 0		0	
1 1 1 0 0 0 1		0	horisontaaliset
1 0 0 0 1 1 1		0	pariteetit
0 0 1 1 0 0 1		1	
<hr/>			
1 0 1 1 1 1 1		0	taulukon pariteetti
			vertikaaliset
			pariteetit

Virheiden havaitseminen

- Ei löydä lyhyitä virheryöppyjä, joissa neljä bittiä vaihtuu sopivasti

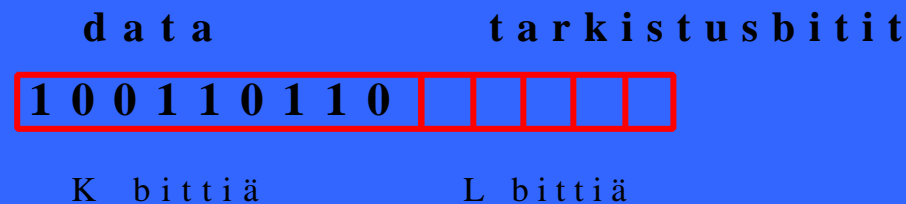
```
1 0 0 1 0 1 0
0 1 1 1 0 1 0
1 1 1 0 0 0 1
1 0 0 0 1 1 1
0 0 1 1 0 0 1
```

Internetin tarkistussumma

- lasketaan 16-bittisten sanojen yhden komplementit yhteen
- otetaan summasta yhden komplementti
- käytetään Internet-protokollissa
 - UDP- ja TCP -protokollissa
- monia virhekombinaatioita jää havaitsematta
- riittävän hyvä, jos virheitä vähän

CRC:n perusidea

- tarkistusavain (virittäjä, virittäjäpolynomi)
 - bittejä yksi enemmän kuin tarkistusbittejä
 - lähettäjä ja vastaanottaja tuntevat
- lähettäjä
 - laskee lähetettävälle datalle tarkistusavaimen avulla tarkistusbitit ja liittää ne kehykseen
- vastaanottaja
 - tarkistaa, onko koko saapunut kehys (data + tarkistusbitit) pysynyt muuttumattomana



Esimerkki: data = 101110, virittäjä = 1001, (polynomina $X^3 + 1$), tarkistusbittejä 3

Lähetettävä data = 101110??? tarkistusbitit

```

      101011
    -----
1001 101110000
      1001
    -----
      1010
      1001
    -----
        1100
        1001
    -----
          1010
          1001
    -----
            0011
  
```

**Modulo 2-
aritmetiikka:**

1+1 = 0 (XOR)

Lähetetään: **101110 011**

Vastaanottaja: jakaa saamansa kehyksen virittäjällä. Kehys on ok, jos jakojäännös on 0!

0011 = tarkistusbitit

Standardoituja virittäjäpolynomeja

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

CRC: n virheiden havaitsemiskyky

- kaikki virheryöpyt, joiden pituus $<$ tai $=$ kuin virittäjän
- useimmat virheryöpyt, joiden pituus on suurempi
 - CRC-32: $P\{\text{ryöppy} > 33 \text{ havaitaan}\} = 0.9999999998$

– Huom

- » Arvioinneissa lähtökohtana ollut täysin satunnainen bittien jakautuminen, mutta todellisuudessa näin ei ole!
- » Joten havaitsemattomien virheiden määrä on arvioitua suurempi.

Vuonvalvonta

- Liukuva ikkuna

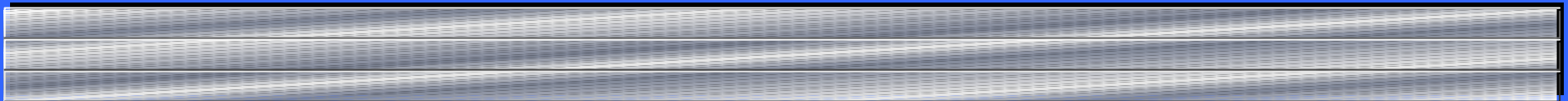
- ikkunan koko rajoittaa lähettämistä

- » jos kehyksen numero ei ole ikkunassa, sitä ei oteta vastaan

- kuittaus siirtää ikkunaa eteenpäin

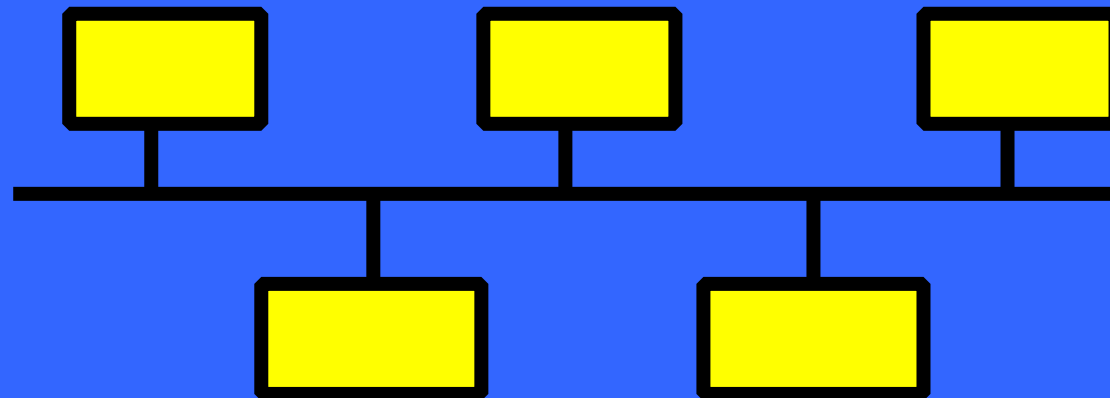
- stop-sanoma

- Receive not ready



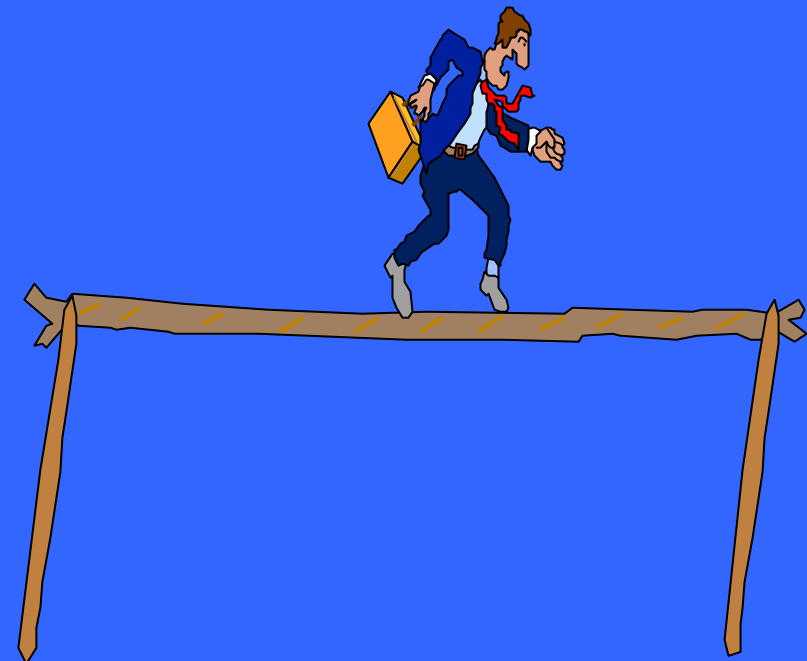
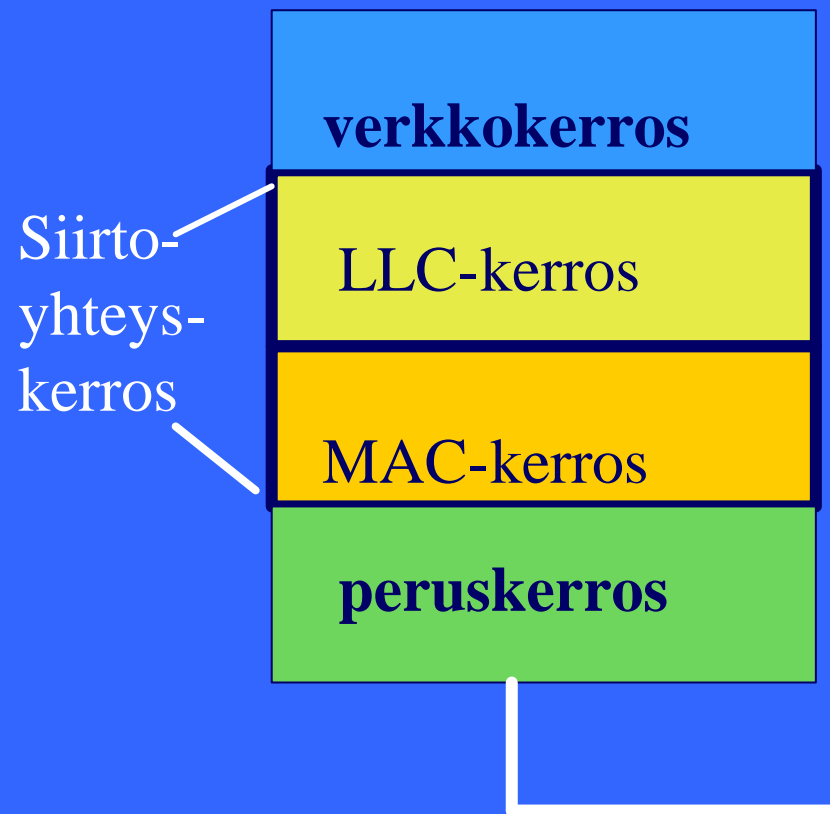
5.3. Yhteiskäyttöinen kanava

- yleislähetys (broadcast)
 - » multiaccess channel
 - » random access channel
 - LAN (Ethernet)
 - langaton
- ongelma: käyttövuoron ‘jakelu’



MAC = Medium Access Control

LLC = Logical Link Control



Vain yksi kerrallaan!

Eri yhteiskäyttötapoja on hyvin paljon:

- **kilpailu** Aloha, CSMA, **CSMA/CD**
 - 'se ottaa kun ehtii'
- **vuorotellen:** pollaus, vuoromerkki
 - 'sinä ensin ja sitten on minun vuoroni'
- **varaus:** vuorot varataan etukäteen
 - varaukseen käytetään usein kilpailua
- **kanava jaetaan:** TDMA, FDMA, **CDMA**
 - 'käytä sinä tätä puolta ja minä tätä toista'

Törmäys

- yksi yhteinen kanava lähettäjäille
- lähetys onnistuu vain, jos yksi lähettää
- Jos useampi kuin yksi lähettää, syntyy yhteentörmäys (collision)
 - kaikki törmänneet sanomat tuhoutuvat ja ne on lähetettävä uudelleen
 - vaikka törmäisivät vain yhden bitin verran
 - **kaikkien havaittavissa**
 - LAN: törmäyssignaali
 - satelliittikanava: kuuntelee oman lähetyksensä
 - WLAN: ilmoitus vastaanottajalta

Aika

- **jatkuva aika**

- lähetykset voivat alkaa milloin vain
- ei mitään synkronointi, ei yhteistä aikaa

- **viipaloitu aika (slotted time)**

- aika lokeroitu aikaviipaleiksi
- lähetys voi alkaa vain aikaviipaleen alussa
- aikaviipaleessa
 - ei kukaan lähetä => hukkaan
 - yksi lähetys => ok
 - useita lähetyksiä => törmäys
- vähentää törmäyksiin (=hukkaan) menevää aikaa
 - törmäykset täydellisiä

Lähetyskanavan kuuntelu (carrier sense)

- käynnissä olevan lähetyksen havaitseminen
 - asema tutkii, onko kanava jo käytössä
 - ennen lähetystä tutkitaan, onko joku muu lähettämässä
 - jos on, ei lähetetä
 - yleensä lähiverkot (CSMA)
 - asema ei tutki kanavan käyttöä
 - asema lähettää aina kun haluaa
 - lähettämisen jälkeen havaitaan onnistuiko
 - esim. satelliitilähetys

Kanavan kuuntelu

- ei aina paljasta jo alkanutta lähetystä
 - etenemisviipeen takia
- tai ole järkevää
 - esim. satelliittikanavan kuuntelu ei paljasta sitä, onko joku toinen maa-asema jo aloittanut lähetyksen
 - langattomassa lähiverkossa lähettäjän ympäristön kuuntelu ei kerro sitä, onko vastaanottaja saamassa sanomia muualta

Yleislähetysprotokollia

Esimerkkejä:

- CSMA/CD (Aloha, CSMA)
 - mm. Ethernet-verkossa käytetty kilpailuprotokolla
- CDMA
 - radiolinjoilla käytetty koodinjakoon perustuva protokolla

ALOHA

- Hawaiiilla, 70-luvulla radiotietä varten
- puhdas ALOHA:
 - asema lähettää aina, kun sillä on lähetettävää
 - ja samalla kuuntelee, onnistuiko lähetys
 - lähiverkossa törmäys havaitaan ‘heti’, sillä siirtoviive pieni
 - toisin satelliitilla!
 - jos törmäys, niin lähettäjä odottaa satunnaisen ajan ja yrittää uudelleen
 - **maksimaalinen tehokkuus ~18%**

Viipaloitu ALOHA

- lähetysaika jaettu aikaviipaleiksi
- lähetys voi alkaa vain aikaviipaleen alussa
- törmäykset täydellisiä
 - » lähetykset samassa aikaviipaleessa
 - » törmäysvaara-aika = yhden aikaviipaleen mittainen
- suorituskyky kaksinkertaistuu
 - maksimi ~ **37%**
 - siis 37% tyhjiä, 37% onnistuneita, 26% törmäyksiä

CSMA (Carrier Sense Multiple Access)

- toiminta

- **kuuntele linjaa ennen lähettämistä**
- jos linja vapaa lähetä (yleensä)
- jos linja varattu odota satunnainen aika ja yritä uudelleen

- Suorituskyky:

- törmäysvaara vain jos asemat lähettävät niin samanaikaisesti, että eivät siirtoviipeen vuoksi havaitse toista lähetystä
- ongelma, jos siirtoviive on pitkä

CSMA-protokollat

- Useita versioita, jotka hieman eroavat toisistaan
 - miten toimitaan, kun kanava varattu?
 - jäädään odottamaan ja lähetetään heti kanavan vapauduttua => jos useita odottajia, tulee varmasti törmäys
 - luovutaan ja yritetään uudestaan satunnaisen ajan kuluttua => hukkaa lähetysvuoroja
 - viipaloitu aika vai ei?
 - vaikka kanava on vapaa, ei silti aina lähetetä
 - lähetys vapaalle väylälle todennäköisyydellä p !

CSMA/CD (Collision Detection)

- keskeyttää lähettämisen heti, kun havaitsee törmäyksen tapahtuneen
 - törmäyksen aiheuttama hukka-aika pienenee
- ‘epävarmuuden aika’ on 2τ , τ on maksimi etenemisviive kahden aseman välillä
- jos törmäys
 - => havaitaan ja lopetetaan lähetys
 - => yritetään uudestaan satunnaisen ajan kuluttua

Varausprotokollat

- ei törmäyksiä!
- lähetysvuorot varataan etukäteen
- varausvaihe
 - usein kilpaillaan varauksista
 - törmäyksiä, mutta vähän
- lähetysvaihe
 - kaikki varanneet lähettävät sanomansa
- hyvin paljon erilaisia versioita
 - etenkin satelliittiyhteyksille

Vuorotteluprotokollat

- Pollaus (vuorokysely)
 - isäntäasema antaa vuorotellen muille asemille lähetyksluvan
- Vuoromerkki
 - asemilla kiertää vuoromerkki (token)
 - asema saa lähettää vain kun sillä on vuoromerkki
 - kun asema on lähettänyt tai sillä ei enää ole lähetettävää, se siirtää vuoromerkin seuravalle

Kanavan jakoprotokollat

- TDMA

- aikajako

- asemalla oma aikaviipale

- FDMA

- taajuusjako

- asemalla oma taajuusalue

- CDMA

- koodijako

- asemalla oma koodi
 - asemat voivat lähettää yhtäaikaan!

CDMA (Code Division Multiple Access)

- **yksi kanava**

- **usea samanaikainen lähetys**
- **kukin koko kanavan taajuudella!**

- yhden bitin lähetysaika jaetaan pienempiin osiin (aikasiruihin)

- » 64 tai 128 sirua bittiä kohden

- kullakin asemalla oma 'sirukuvio' 1-bitin lähetykseen

- » (0-bitti on tämän yhden komplementti)

Esimerkiksi

- aseman A 1-bitti: 00011011
0-bitti: 11100100
- aseman B 1-bitti: 00101110
0-bitti: 11010001
- aseman C 1-bitti: 01011100
0-bitti: 10100011
- aseman D 1-bitti: 01000010
0-bitti: 10111101

Ps. Oikeasti käytetään 64 tai 128 sirua

Kaikki bittikuviot parittain ortogonaalisia

- $A \bullet B = 0 = 1/m \sum A_i B_i$ (sisätulo)
- $A \bullet A = 1$
- $-A \bullet A = -1$
- \Rightarrow yhteissignaalista löydetään eri asemien omat lähetykset

- kukin asema lähettää omat 1-bittinsä ja 0-bittinsä
- kun moni lähettää samanaikaisesti tuloksena on yhteissignaali S.
 - » lähetettyjen signaalien ‘summa’
- aseman datan ‘purkaminen’ yhteissignaalista
 - » A = aseman oma bittikuvio
 - » $S \bullet A$ tuottaa aseman lähettämän bitin
 - kerrottuna bitin aikasirujen lukumäärällä

Esimerkki

» merkintä 1 =1, 0 = -1,

» helpompi laskea yhteen

● $S = (-2 \ -2 \ 0 \ -2 \ 0 \ -2 \ 4 \ 0)$

● $C = (-1 \ 1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1)$

● $S \bullet C = (2 \ -2 \ 0 \ -2 \ 0 \ -2 \ -4 \ 0)$
 $= -8 \Rightarrow -1$

● eli **C lähetti 0-bitin**

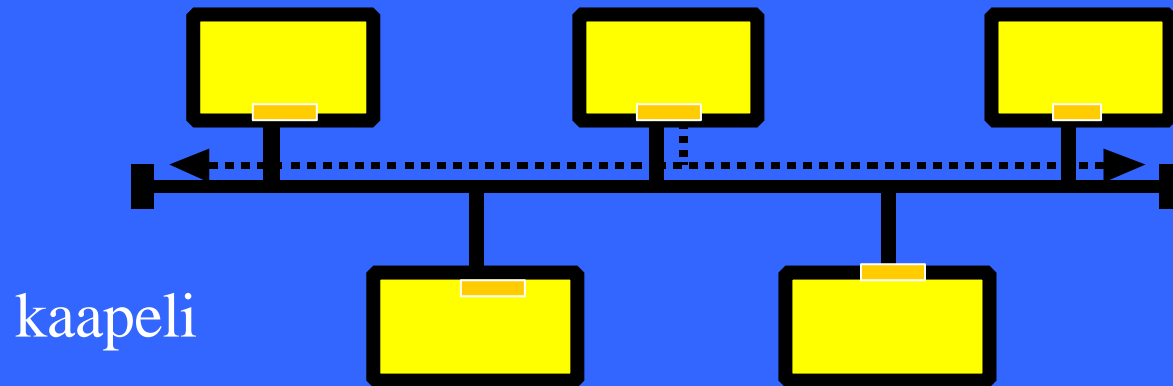
5.5 Ethernet-lähiverkko

- Yleisin lähiverkkoteknologia
- IEEE:n standardoima LAN-verkko
 - CSMA/CD (kuulosteluväylä)
- Muita lähiverkkostandardeja
 - esim.
 - Token ring (vuororengas)
 - FDDI
 - WLAN (langaton lähiverkko)

ei käsitellä tällä kurssilla

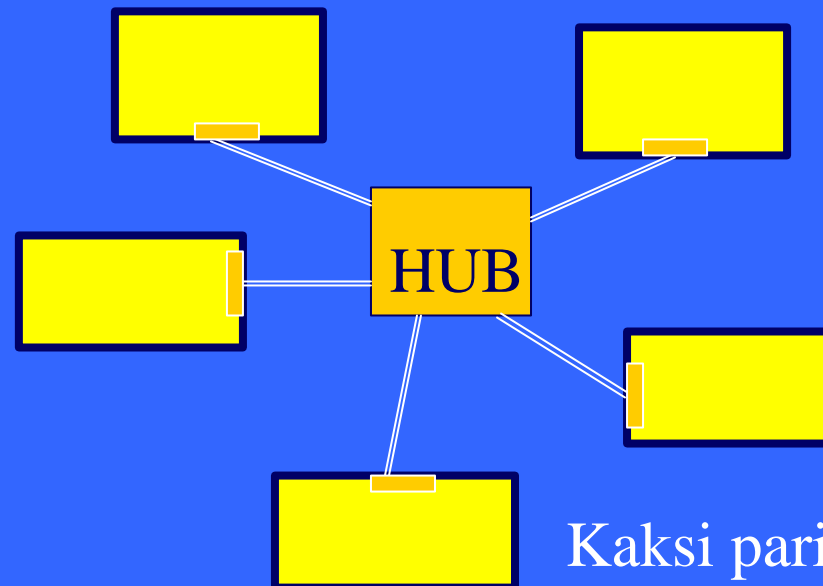
Eetteriverkon rakenne

- väylä



- ◆ tähti

- hub toimii toistimen tavoin



Kaapelit

10Base2 ohut kaapeli

- » 10 => 10 Mbps
- » Base => kantataajuus
- » 2 => 200 m

- 10Base-T kierretty pari & central hub

- » helppo hallita, kallis, suosio kasvaa

- 10Base-F valokaapeli

- » kallis, luotettava, tehokas

- 100Base-T, 100 Base-F

- » Fast Ethernet

- 1000Base-T, 1000Base-X

- » Gigabit Ethernet

Lyhyet etäisyydet, pieni määrä laitteita

- sovittimesta keskittimeen (hub) maks. 100 m
- väylä
 - pituus maks. < 200 metriä,
 - syynä vaimeneminen
 - solmuja maks. 30 kpl
 - syynä CSMA/CD => liikaa törmäyksiä
 - maks. 5 väylää voidaan yhdistää **toistimilla**
 - => ~1000 m, 150 laitetta
- valokuitua käytettäessä hieman pitemmät etäisyydet

Signaalin koodaus

- Manchester-koodaus

- tahdistus

- » jännitteen muutos keskellä bittiä

- ei kellopulssia

- mutta lisää kaistanleveyttä

CSMA/CD

- jos väylä vapaa, lähetetään heti

- muuten jäädään odottamaan ja lähetetään heti linjan vapauduttua

- entä kun tapahtuu törmäys?

Törmäyksen jälkeinen uudelleenlähetys

● Binary exponential backoff

- törmäyksen jälkeen aika jaetaan lokeroiksi
 - 51.2 μ s vastaten 512 bittiä eli 64 tavua
- 1. törmäyksen jälkeen asema odottaa satunnaisesti joko 0 tai 1 lokeron ajan ennen kuin yrittää uudelleen
- 2. törmäyksen jälkeen odotus on 0, 1, 2 tai 3 lokeroa
- n. törmäyksen jälkeen valitaan odotusaika väliltä:
0 - 2^{n-1} lokeroa
 - 10. törmäyksen jälkeen väliä [0-1023] ei enää kasvateta
 - 16. törmäyksen jälkeen luovutaan ja ilmoitetaan 'asiakkaalle' (eli verkkokerrokselle) epäonnistumisesta

- binäärinen eksponentiaalinen perääntymien on joustava
 - kuorma kasvaa \Rightarrow väli kasvaa
- vaihtoehtona kiinteä valintaväli
 - » aina [0- 1023]
 - » aina [0-1]
 - » aina [a-n]
 - entä suorituskyky?

Ehternet-kehys

