

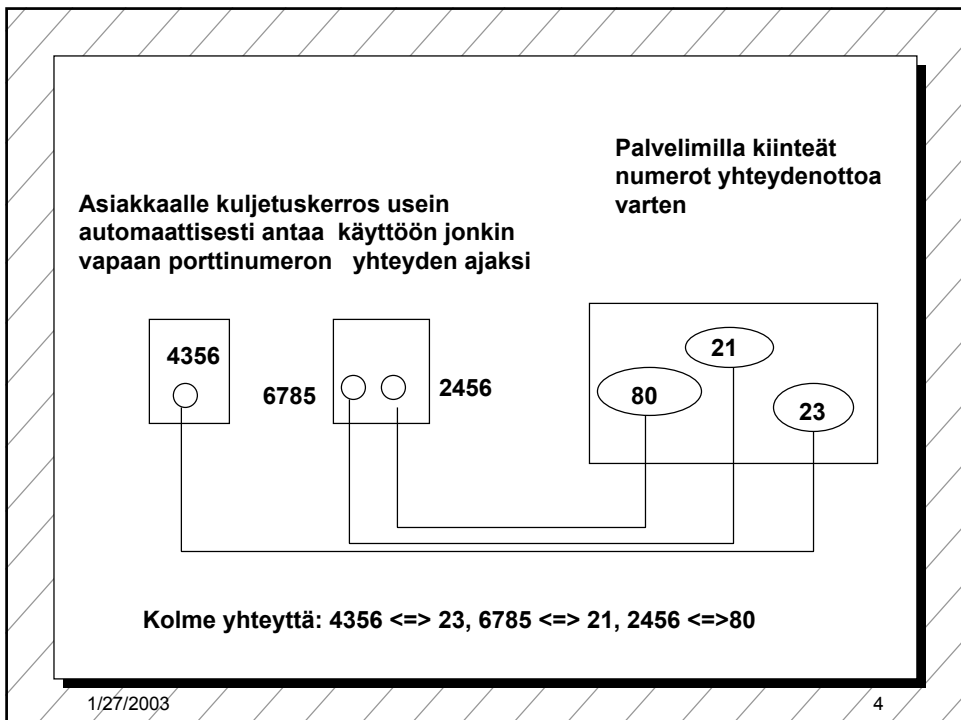
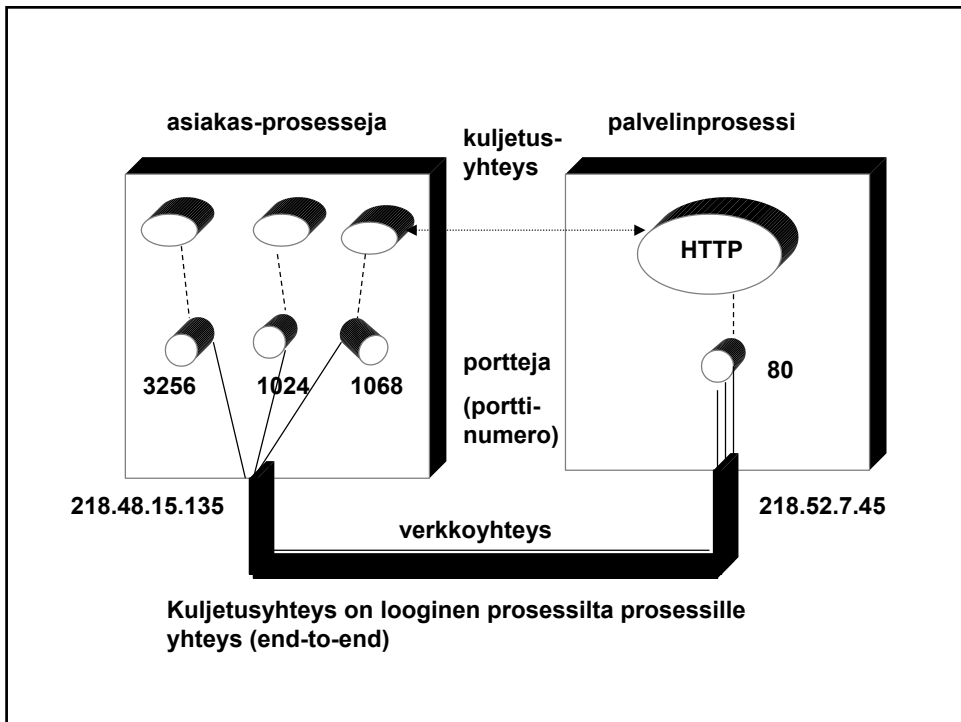
3. Kuljetuskerros

3.1. Kuljetuspalvelu

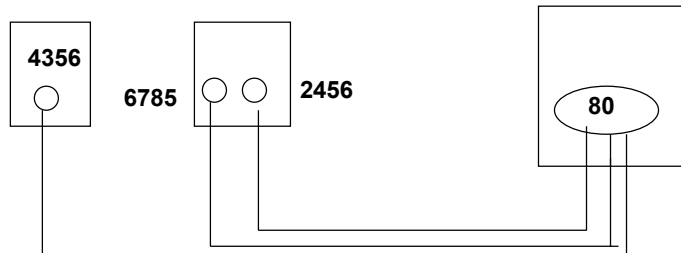
- **'End- to- end'**
 - prosessilta prosessille looginen yhteys
 - portti
 - verkkokerros koneelta koneelle
 - IP-osoite
- **peittää verkkokerroksen puutteet**
 - jos verkkopalvelu ei ole riittävän hyvä, sitä voidaan parantaa kuljetuskerroksella
 - kuljetuskerros huomaa verkkokerroksen kadottamat paketit ja pyytää niiden uudelleenlähetystä

Sovelluksien datavirtojen erottaminen

- **IP-osoite**
 - osoittaa koneen yksikäsitteisesti
- **Sovellusprosessi tunnistetaan porttinumerosta (16 bittiä =>0-65535)**
 - jokaisessa lähetetyssä segmentissä on
 - lähettäjänsä porttinumero
 - vastaanottajan porttinumero
- **Yleisillä palvelimilla omat varatut porttinumerot (0-1023)**
 - SMTP 25, HTTP 80, jne



Tarvitaan sekä lähteen että kohteen porttinumerot

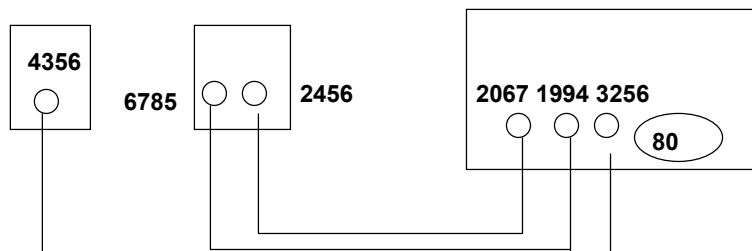


Kolme yhteyttä: 4356 \Leftrightarrow 80, 6785 \Leftrightarrow 80, 2456 \Leftrightarrow 80

1/27/2003

5

Palvelimessa yhteyksille uudet porttinumerot, jotta portti 80 voi ottaa vastaan uusia yhteyspyyntöjä

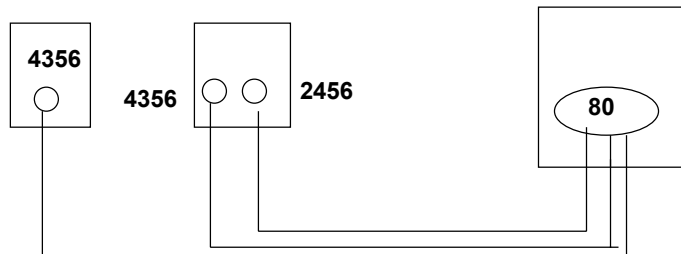


Kolme yhteyttä: 4356 \Leftrightarrow 3256, 6785 \Leftrightarrow 1994, 2456 \Leftrightarrow 2067

1/27/2003

6

Eri koneissa voidaan ottaa sama numero!



Kolme yhteyttä: 4356 \Leftrightarrow 80, 4356 \Leftrightarrow 80, 2456 \Leftrightarrow 80!

Kuljetusyhteydellä käytetään apuna myös IP-osoitetta:

=> koneilla eri IP-osoitteet, joten yhteydet pystytään erottamaan

Sovelluksen vaatimuksia kuljetuspalvelulle:

- Virheetön, luotettava
- järjestyksen säilyttävä
- kaksoiskappaleet karsiva
- mielivaltaisen pitkiä sanomia salliva
- vuonvalvonnan mahdollistava

Verkkokerros kuitenkin voi

- kadottaa sanomia
- toimittaa sanomat epäjärjestyksessä
- viivyttää sanomia satunnaisen pitkän ajan
- luovuttaa useita kopioita samasta sanomasta
- rajoittaa sanomien kokoa

kuljetuspalvelut parantavat verkkopalveluja

**Sovelluksen näkemä palvelun laatu
(Quality of Service, QoS)**

kuljetuskerroksen palvelut

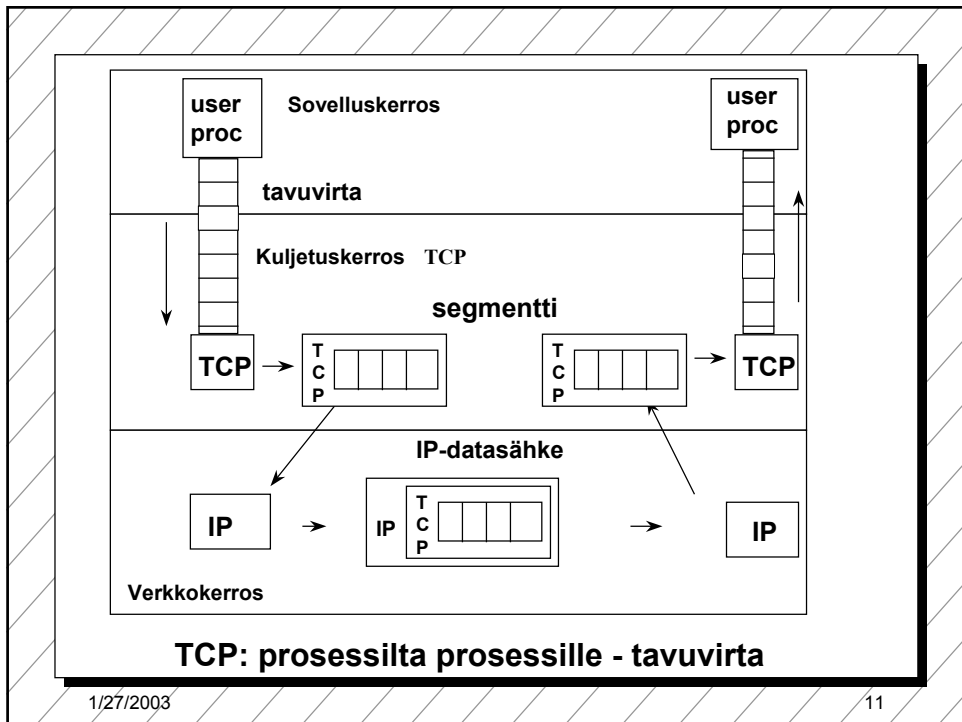
verkkokerroksen palvelut

kuljetuskerroksen palvelut

verkkokerroksen palvelut

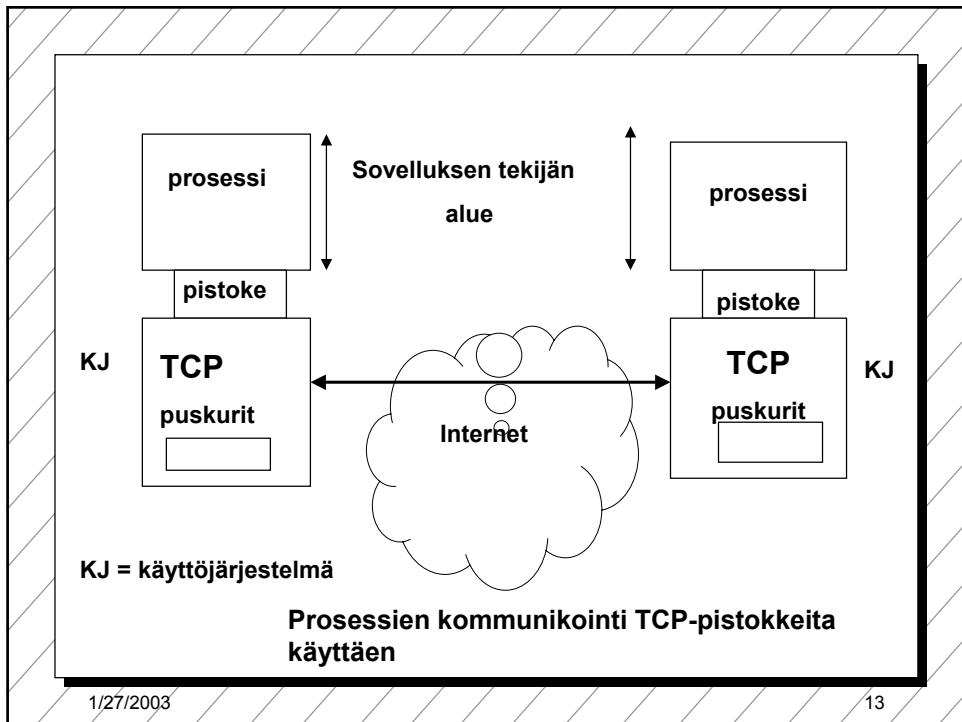
Internetin kuljetuskerros

- **UDP (User Datagram Protocol)**
 - yhteydetön, epäluotettava palvelu
- **TCP (Transmission Control Protocol)**
 - yhteydellinen, luotettava palvelu
 - **virhevalvonta**
 - havaitsee ja korjaa siirrossa syntyneet virheet
 - **vuonvalvonta**
 - ei ylikuormita vastaanottajaa
 - **ruuhkanvalvonta**
 - huolehtii ettei verkko pääse ruuhkautumaan



Pistokerajapinta (Socket interface)

- **Verkkopalvelun ja sitä käyttävän sovelluksen rajapinta**
 - yleensä käyttöjärjestelmän tarjoama palvelu
 - pistokerajapinta alunperin Berkeley Unixin mukana, nyt lähes kaikissa käyttöjärjestelmissä
 - miten verkkoprotokollan tarjoamiin palveluihin päästään käsiksi sovelluksesta



- **pistoke (socket)**

- TCP-yhteyden päätepiste sovellukselle
 - lähettäjällä ja vastaanottajalla oma pistoke
- pistokenumero 48 bittiä
 - koneen 32 bitin IP-osoite
 - 16 bitin porttinumero

TCP-yhteys

- **kaksisuuntainen** (full-duplex)
kaksipisteyhteys
- **tunnistetaan päätepisteinä olevien pistokkeiden tunnuksista** (pistoke1, pistoke2)



1/27/2003

15

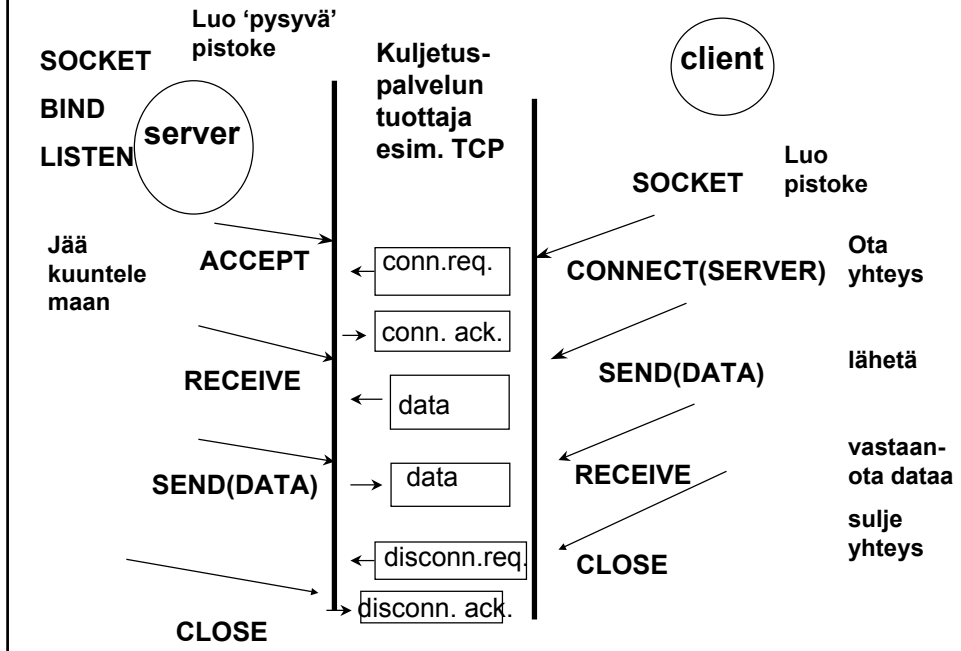
TCP:n pistokeprimitiivit

- **SOCKET** luo uusi yhteydenpäätepiestepistoke
- **BIND** anna pistokkeelle osoite
- **LISTEN** halukas vastaanottamaan yhteyksiä
- **ACCEPT** jää odottamaan yhteysyhteyksiä
- **CONNECT** yritä muodostaa yhteys
- **SEND** lähetä dataa yhteyttä pitkin
- **RECEIVE** vastaanota dataa yhteydeltä
- **CLOSE** pura yhteys (symmetrinen)

1/27/2003

16

Kuljetusyhteyden muodostus ja käyttö



3.3 UDP

■ UDP (User Data Protocol)

- voidaan lähettää sanomia ilman yhteyden muodostusta

UDP-otsake

←----- 32 bittiä ----->

Source port #	Destination port #
UDP length	UDP checksum
sovelluksen dataa	

UDP-tarkistussumma

- Virheen havaitsemista varten otsakkeeseen liitetään tarkistussumma
 - kaikki segmentin 16 bitin sanat lasketaan yhteen ja summasta otetaan yhden komplementti
 - = muutetaan ykköset nolliksi ja nollat ykkösiksi
 - vastaanottaja laskee taas kaikkien segmentin sanojen (mukana myös tarkistussumma) summan
 - jos tulokseksi saadaan 16 ykköstä, niin ok!

1/27/2003

19

Esimerkki

- Lasketaan yhteen kolme 8 bitin mittaista sanaa:

- Lähettäjä

1011 0100

0111 0101

1000 1101

=====

1011 0110

0100 1001

Yhden komplementti

- vastaanottaja

1011 0100

1111 0101

1000 1101

0100 1001

=====

0111 1111

1/27/2003

20

- **Miksi tarvitaan tarkistussumma?**
 - Kaikki siirtoyhteyskerrokset eivät suorita tarkistuksia
- **UDP-tarkistussumma ei ole kovin tehokas havaitsemaan virheitä!**
- **Se ei myöskään yritä toipua virheistä!**
 - Jotkut toteutukset voivat tuhota virheellisen segmentin
 - jotkut antavat se sovellukselle varoituksen kera

UDP:n etuja:

- **Yhteydetön**
 - aikaa ei kulu yhteyden muodostamiseen ja purkamiseen
 - ei tarvita resursseja yhteyden tilatietojen ylläpitoon
- **Otsake (= 8 tavua) pieni => pieni yleisrasite => lisää tehokkuutta**
- **Ruuhkanvalvonta ei säännöstele liikennettä**

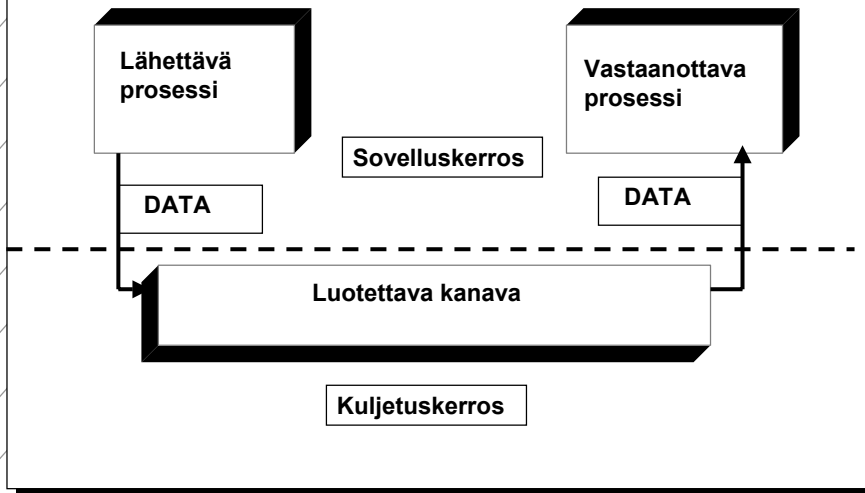
Tehtäviä:

- Lähetetään 10 tavun viesti UDP:llä.
 - Miten kauan kestää lähettäminen, jos lähetyksenopeus on 56 kbps?
 - $10 \text{ tavua} + 8 \text{ tavua} = 18 * 8 \text{ b} = 144 \text{ bittiä}$
 - $144 \text{ b} / 56\,000 \text{ b/s} = 2.57 \text{ ms}$
 - Miten suuri on etenemisviive, jos etäisyys lähettäjältä vastaanottajalle on 1000 km?
 - $1000\text{km} / 200\,000 \text{ km/s} = 5 \text{ ms}$
 - Miten suuri on UDP-otsakkeen aiheuttama yleisrasite (overhead)?
 - $8/18 = 0.44$ eli 44 %

UDP:n käyttö

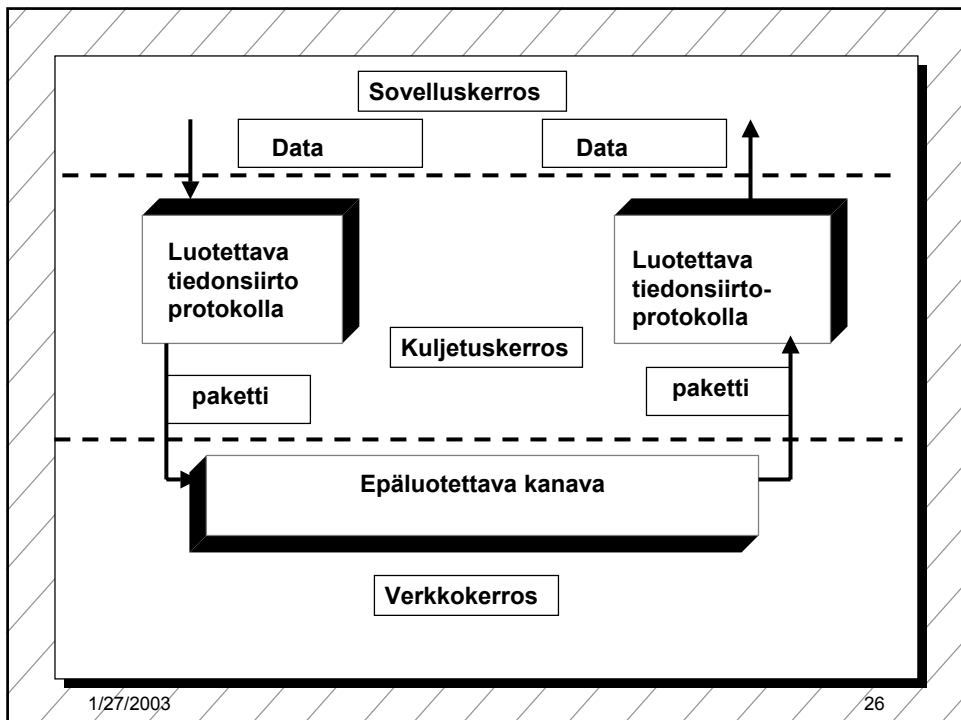
- Vaikka UDP on epäluotettava, se sopii monien sovellusten tarpeisiin:
 - Remote file server (NFS)
 - multimedia
 - Internet-puhelin
 - verkon hallinta (SNMP)
 - reititys (RIP)
 - nimipalvelu (DNS)
- Miksi nämä sovellukset suosivat UDP:tä?

3.4 Luotettava tiedonsiirto



1/27/2003

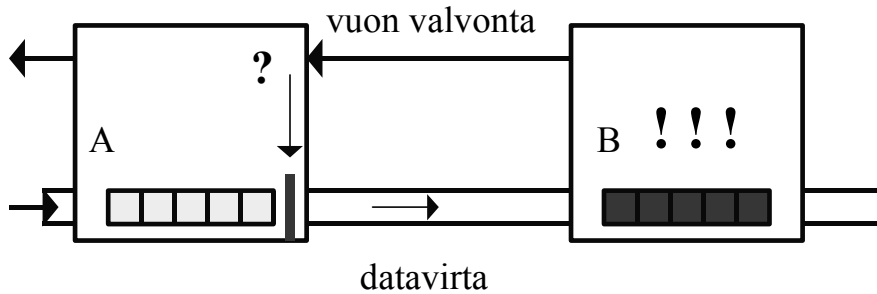
25



1/27/2003

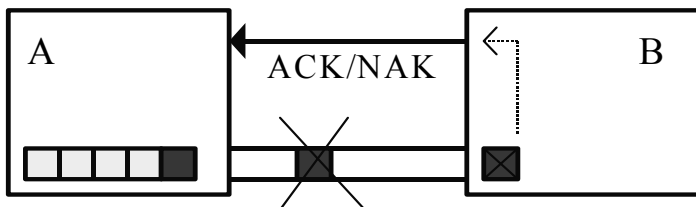
26

Vuon valvonta



- X-ON / X-OFF : GO! | STOP!

Kohinainen kanava



- sanoma vääristyy => virhetarkistus
- sanoma katoaa => numerointi, ajastin ja uudelleenlähetyks
 - duplikaattien havaitseminen
- sanoma viivästyy => rajallinen elinaika
- sanomien järjestys muuttuu => järjestäminen

Yksinkertainen Stop and wait -protokolla

■ Oletus

- virheetön siirto => ei huolta virheistä, mutta vuonvalvontaa tarvitaan

■ lähettäjä

- lähettää sanoman
- odottaa lupaa lähettää seuraava sanoma

■ vastaanottaja

- käsittelee sanoman
- lähettää tiedon (=antaa luvan) lähettäjälle

Entä jos virheitä?

- Sanomissa virheitä tai sanomat voivat puuttua kokonaan

- Myös kuittaukset voivat kadota

■ Tarvitaan

- virheen havaitseminen ja korjaaminen
 - tarkistussumma
 - kuittaus
 - uudelleenlähetys
- sanomien numerointi
- uudelleenlähetysajastin