

## Tietoliikenteen perusteet

# VERKKOSOVELLUKSIA SOVELLUSPROTOKOLLIA

Kurose, Ross: Ch 2

## Sisältöä

- 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 -mallin perustuvien verkkosovellusten toimintaperiaatteet
- Tuntee sovellusprotokollien syntaksia ja semantiikkaa
- Osata selittää nimipalvelun, www:n ja sähköpostin toimintaideat
- Tunnistaa pistokkeiden käytön periaatteet



## Verkkosovellus

# Verkkosovellusten periaatteet

## Verkkosovellus

- Käyttäjä ja palvelija eri koneissa  
www-selain ja www-palvelija  
postiohjelma ja postipalvelin, ...
- Sovellusprotokolla kuvaa näiden sanomanvälityksen  
DNS, HTTP, SMTP, FTP, ....  
Syntaksi, semantiikka, järjestys
- Sanomanvälitys käyttää verkon tarjoamaa kuljetuspalvelua  
osa järjestelmän perusrakennetta  
sovelluksista riippumatonta
- Reititys tapahtuu vasta verkkotasolla, mutta sovellutasolla tiedettävä osoite

## Sovellusarkkitehtuuri

### Asiakas-palvelija-malli (esim. selain ja www-palvelin)

- Alina toiminnassa oleva palvelijaohjelma, jolla klinteä, tunnettu IP-osoite
- Asiakasohjelmat halutessaan ottavat yhteyttä palvelimeen ja pyytävät siltä palvelua



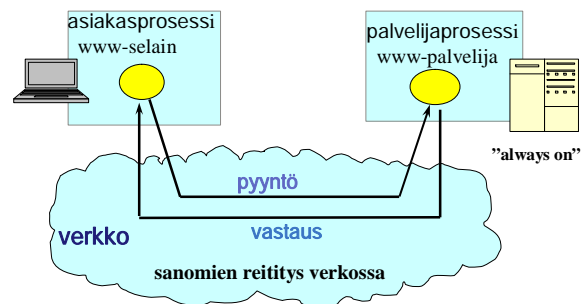
### Vertaistoimijamalli (esim. Gnutella)

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



### Hybridimalli (esim. Napster)

## Asiakas-palvelija-malli



Oikea kone, oikea prosessi

## Sovelluksen rajapinta tietoliikenteeseen

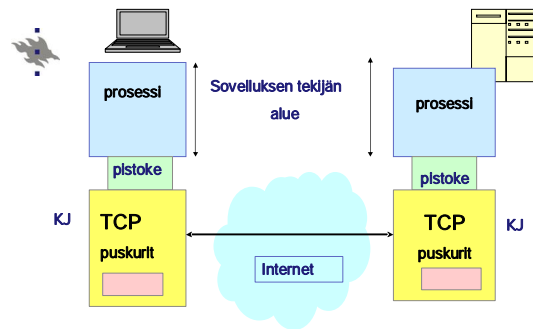
- Pistoke (socket)** (verkkosovelluksen ohjelmointirajapinta)
  - yhteyden muodostaminen
  - lue /kirjoita sanoma
    - prosessi kirjoittaa verkkoon ja lukee verkosta lähes samalla tavoin kuin kirjoittaa tiedostoon ja lukee tiedostosta
  - 'luukku' tai 'ovi', josta dataa sisään /ulos

**Lähetys (send):** anna sanoma KJ:lle

**Vastaanotto (receive):** ota sanoma KJ:ltä.

Sovellus odottaa, jos sanoma ei ole vielä saapunut

**Ohjelmoija valitsee käyttääkö KJ kuljetuskerroksella yhteydellistä vai yhteydetöntä palvelua!**



KJ = käyttöjärjestelmä

Prosessien kommunikointi TCP-pistokkeita käyttäen

## KJ:n rajapinta laitteistoon

KJ:n kannalta tietoliikenne normaalia siirtää (I/O:ta)

### Lähtevä liikenne:

Protokollan mukaan

Sovellus pyytää kuljetuspalvelua KJ:n palvelupyynnöllä **send**  
Kuljetuskerros hoitaa omat tehtävänsä ja kutsuu verkkokerroksen rutiinia  
Verkkokerros tekee hommansa ja kutsuu laiteajurin rutiinia  
**Laiteajuri** vie datan ja komennot verkkokortin ohjaimen rekistereihin  
Verkkokortti siirtää bitit linkille (linkkikerros ja fyysinen siirto)

### Tuleva liikenne

Protokollan mukaan

Verkkokortti ottaa vastaan linkiltä tulevat bitit (fyysinen siirto ja linkkikerros) ja aiheuttaa **keskeytyksen**.  
KJ:n laiteajuri siirtää bitit verkkokortilta keskusmuistiin  
Ajuri kutsuu verkkokerroksen rutiinia, joka suorittaa omat toimintonsa  
Verkkokerros kutsuu kuljetuskerroksen rutiinia, joka tekee omat toimensa  
Sanoma sovellukselle vasta, kun se sitä pyytää **palvelupyynnöllä receive**.

## Osoittaminen

Sanomissa oltava lähettäjän ja vastaanottajan IP-osoite ja porttinumero

IP-osoite à oikea kone

[www.iana.org](http://www.iana.org)

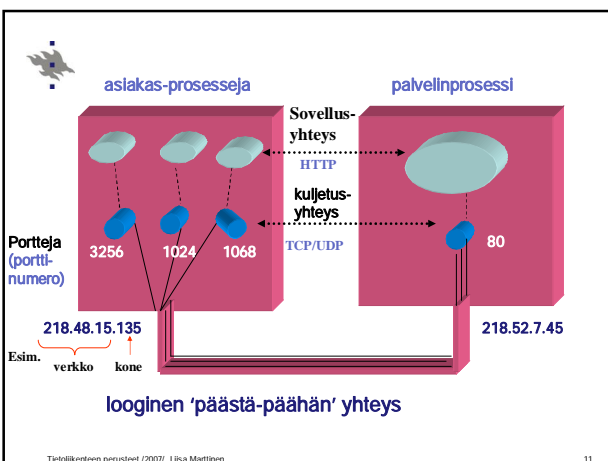
koneen (verkkokortin) yksilöivä 32-bittinen tunniste  
osoitteen verkko-osa yksilöi verkon  
osoitteen koneosa yksilöi koneen verkossa

porttinumero à oikea prosessi

Yleisillä palveluilla standardoidut tunnetut porttinumerot:

- www-palvelin kuuntelee porttia 80,
- Postipalvelin kuuntelee porttia 25

KJ osaa liittää porttinumeron prosessiin



## Kuljetuspalvelun laatu

Application	Data Loss	Bandwidth	Time-Sensitive
File transfer	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Real-time audio/video	Loss-tolerant	Audio: few kbps-1Mbps Video: 10 kbps-5 Mbps	Yes: 100s of msec
Stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps-10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

Figure 2.4 ♦ Requirements of selected network applications



## Kuljetusprotokollat

### nTCP (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 hilljittää vauhtia, jos vastaanottaja ei ehdi käsitellä

#### Ruuhkanvalvonta (congestion control)

Lähettäjä hilljittää vauhtia, jos reitittimet eivät ehdi käsitellä



## Kuljetusprotokollat

### nUDP (User Datagram Protocol) [RFC768]

Kevyt kuljetuspalvelu, pieni yleisrasite

Ei yhteyden muodostusta eikä purkua

Ei takuita sanoman perillemenosta

Sanoman segmentit vain lähetetään verkkoon

Sanoman segmenttejä voi puuttua ja ne voivat

saapua epäjärjestyksessä

Ei vuonvalvontaa, ei ruuhkanvalvontaa

UDP-pistokkeeseen voi lähettää niin paljon kuin haluaa

**Huom!** Kummassakaan ei ole takuita siirtonopeudelle eikä vilpeelle (vain ns. 'best effort'-palvelu)



## Kumpi?

Applications	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 2821]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Remote file server	NFS [McKusik 1996]	UDP or TCP
Streaming multimedia	Often proprietary (e.g., Real Networks)	UDP or TCP
Internet telephony	Often proprietary (e.g., Net2phone)	Typically UDP

**Figure 2.5** ♦ Popular Internet applications, their application-layer protocols, and their underlying transport protocols



## Verkkosovelluksia, sovellusprotokollia

World Wide Web  
HTTP



## WWW ja HTML (HyperText Markup Language)

### nWWW-sivu, WWW-dokumentti

HTML-tekstää, jossa viittauksia muihin objekteihin

muu HTML-tiedosto, kuva- tai äänitiedosto, Java

applet, ...

Sivu muodostuu usean tiedoston sisällöstä, jotka noudetaan palvelijalta

### nViittaus URL-osoitteella (Uniform Resource Location)

<http://www.someschool.edu/someDept/pic.gif>

koneen nimi

Viitatus objektin polkunimi



## HTML (HyperText Markup Language)

### nStandardi siitä, kuinka sivun rakenne kuvataan

Muotoilut, eri osien sijoittelu sivuille

Viittaukset muihin objekteihin

### nSGML (Standard Generalized Markup Language)

yleinen merkkuskieli

kertoo, kuinka dokumentit muotoillaan - ladontamerkinnot

### nXML (Extensible Markup Language)

rakenteellinen tietosisällön kuvaus, myös merkitys kuvattu

### nNäistä enemmän kurssilla:

582304 XML-metakieli (4 op/2ov)

## HTTP (HyperText Transfer Protocol)

PC, jossa on Explorer-selain

Palvelin, jossa on Apache-www-palvelija

Linux-kone, jossa on Firefox-selain

**WWW:N sovellusprotokolla**  
Tekstimuotoiset sanomat  
pyyntö – vastaus

**Asiakas**  
Selain: FireFox, Internet Explorer, Opera, Apple Safari, ...  
pyytää, noutaa ja näyttää objektit

**Palvelija**  
etsii objektin (tiedoston) koneen hakemistosta ja lähettää sen vastauksena asiakkaalle

**Tilaton protokolla**  
Palvelija ei muista mitään edellisistä pyynnöistä

Tietoliikenteen perusteet /2007/ Liisa Marttinen 19

## HTTP-pyyntö: yleinen rakenne

GET /jokuhakemisto/sivu.html HTTP/1.1

metodi SP URL SP versio CR LF pyyntöriivi

Otsakekenttä : kentän arvo CR LF

... Lisää otsakerivejä

Otsakekenttä : kentän arvo CR LF

CR LF Tyhjä rivi

Runko-osa  
käytössä esim. POST-metodissa

SP='space' eli välilyönti

CR + LF = rivin päättäminen

Tietoliikenteen perusteet /2007/ Liisa Marttinen 20

## Esimerkki: HTTP-pyyntö

Pyyntöriivi: GET/ POST/ HEAD -metodi

otsakerivit

Yksi tyhjä rivi merkinä sanoman loppumisesta

Otsakeriveillä välitetään parametritietoja

(Ylimääräinen carriage return, line feed)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-Agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

Tietoliikenteen perusteet /2007/ Liisa Marttinen 21

## HTTP-pyyntömetodeja

**GET** Nouda objekti (download),  
nouda objekti vain jos annettu ehto pätee (**conditional GET**):  
If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, or If-Range

**HEAD** Nouda vain otsaketiedot

**POST** Voidaan myös lähettää tietoa lomakkeen täyttö se. kenttien sisällöt annetaan mukana olemassa olevien dokumenttien kommentointi sanomien lähettäminen uutisryhmiin tai ilmoitustauluille tiedoston lisääminen hakemistoon; yhteisjulkaisun laajentaminen

**PUT** Talleta objekti palvelimelle (upload)  
polkunimi pyyntöriivillä, talletettava runko-osassa

**DELETE** Poista objekti palvelimelta

Web-julkaisun otsakkeita käytetään

Tietoliikenteen perusteet /2007/ Liisa Marttinen 22

Otsakekenttä : kentän arvo CR LF

**Host: WWW.jokupaikka.fi** kone, jossa dokumentti on

**Connection: close** sulje yhteys lähetyksen jälkeen

**User-agent: Mozilla/4.0** selaimen tyyppi

**Accept-language: fi** dokumentin kieli

Tietoliikenteen perusteet /2007/ Liisa Marttinen 23

## HTTP-vastaus: yleinen rakenne

statusriivi

versio SP statuskoodi SP fraasi CR LF

Otsakekentän nimi: SP kentän arvo CR LF

Lisää otsakerivejä

Otsakekentän nimi: SP kentän arvo CR LF

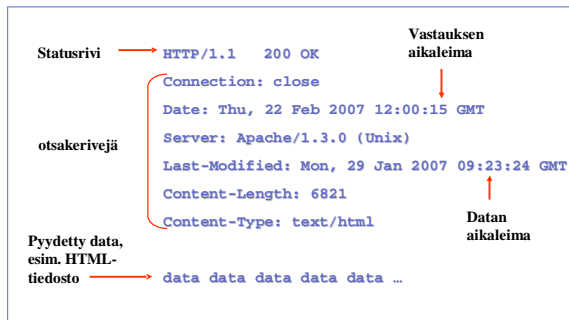
tyhjä rivi

CR LF

Runko-osa (entity body)

Tietoliikenteen perusteet /2007/ Liisa Marttinen 24

## Esimerkki: HTTP-vastaus



## HTTP: statuskoodeja ja fraaseja

Vastaussanomien 1. rivillä esim.:

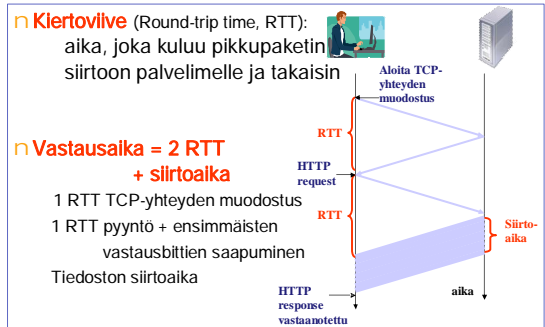
- 200 OK**  
pyyntö onnistui, pyydetty objekti mukana vastauksessa
- 301 Moved Permanently**  
Objekti on siirretty, uusi URL on mukana vastauksen otsakekentässä **Location**.  
Asiakas tekee uuden noudon uudesta URL:ista
- 400 Bad Request**  
Palvelija ei ymmärtänyt pyyntöä
- 404 Not Found**  
Pyydettyä objektia ei löydetty
- 505 HTTP Version Not Supported**  
Palvelija ei tue asiakkaan käyttämää HTTP-versiota  
Syntaksissa on jotain liian uutta tai liian vanhaa

## Selaimen toiminta

Kun käyttäjä kirjoittaa/klikkaa URL:n tai URL:iin on viitattu sivulla:

- Muodosta TCP-yhteys palvelinkoneeseen
- Yhteyspyyntö porttiin 80, odota hyväksymisvastaus
- Laita HTTP-pyyntö TCP-yhteyteen liitettyyn pistokkeeseen
- Ota pistokkeesta palvelimen lähettämä vastaussanoma
- Palvelin sulkee TCP-yhteyden (nonpersistent connection)
- Tutki sivu
- Etsi uudet viitteet ja hae ne samalla tavalla
- Näytä sivu käyttäjälle
- Lopullinen ulkoasu on kiinni selaimen kyvyistä

## Vastausaika (response time)



## Suorituskyky?

- Jos sivulla viitataan 10 objektiin
- 10 peräkkäistä TCP-yhteyden muodostusta ja purkua?  
KJ varaa ja vapauttaa puskuritilaa; muodostuksiin kuluu kaikkiaan 20 RTT
- Avataan useita rinnakkaisia yhteyksiä?  
Puskuritilat yhteyksille
- Käytetään säilyvää TCP-yhteyttä (persistent)  
Oletus uusimmissa standardeissa: Palvelin jättää yhteyden (toistaiseksi) sulkematta. Ajastin on säädettävissä.  
Seuraavat samalle palvelimelle kuuluvat pyynnöt ja vastaukset käyttävät samaa yhteyttä
- Liukuhinnoitettu (pipelining) / liukuhinnoittamaton: seuraava pyyntö lähtee jo ennenkuin edelliseen on saatu vastaus / ei lähde.

## Evästeet (cookies)



- HTTP on tilaton protokolla  
Palvelija ei talleta mitään istuntoon liittyvää
- Selain  
Tallettaa asiakaskoneelle (tiedostoon) palvelimen pyynnöstä ja sen tarpeita varten käyttäjäkohtaista tietoa  
Lähetää tiedot palvelijalle joka pyynnön yhteydessä.
- Palvelin  
Ylläpitää tietokantaa käyttäjistä (back-end database) yksikäsitteiset käyttäjätunnisteet (tav. numero)
- Evästeiden talletus ja lähetys  
HTTP-vastauksessa otsakerivi: **Set-cookie: tieto**  
HTTP-pyyntöissä otsakerivi: **Cookie: tieto**

## Mihin evästeitä käytetään?

### Käyttäjien tunnistamiseen

Palveluntarjoaja muistaa käyttäjän edellisestä sanomasta  
Ensimmäisellä käyttökerralla tietojen kyselyä  
Jatkossa tunnistuseväste mukana sanomissa

### Istunnon vaiheen tallentamiseksi

Autentikointi vain kertaalleen esim. www-postinluohjelman yhteydessä

### Ostoskorina

Selaile palveluntarjoajan sivuilla ja kerää ostokset koriin.  
Lähetä lopuksi tilaus

### Yksityisyys?

Palveluntarjoaja saa koottua tietoa käyttäjästä  
Hakukoneilla voi kerätä lisää.  
Väärinkäyttö? Mainosposti?

## Proxy-palvelin

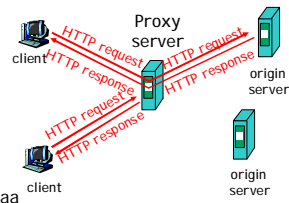
### Verkkovälimuisti

Säilyttää kopioita haetuista objekteista

Pyyntö ohjautuu ensin välimuistiin  
haetaan verkon yli vasta,  
jos ei löydy välimuistista

### Etuja

lyhentää vastausaikaa  
vähentää verkkoliikennettä  
vähentää palvelimen kuormaa



[Myös asiakaskone voi ylläpitää välimuistia!]

KuRo05: Fig 2.10

## Proxy-palvelimen käytöstä

KuRo05: Fig 2.11

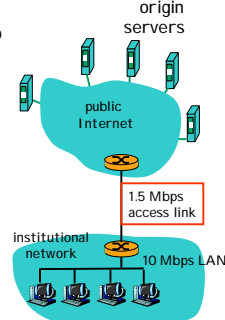
### Oletetaan

Haettavan objektin koko on 100 Kb  
15 pyyntöä/sek => 1,5 Mbps  
Viive Internetin reitittimeltä palvelimelle ja takaisin = 2 sec

### Tällöin

paikallisverkon käyttöaste = 15%  
ei ruuhkautunut siirtoaika 10 ms

Reititinlinkin käyttöaste = 100%  
Saantiaika = Internet delay +  
Access delay + LAN delay  
= 2 sec + mins + msec



## Proxy-palvelimen käytöstä

### Parannus?

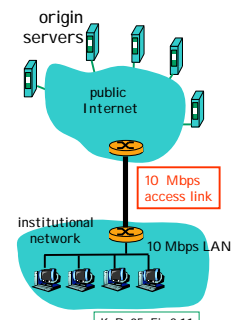
Hankitaan nopeampi yhteys,  
esim. 10 Mbps

### Tällöin

Paikallisverkon käyttöaste = 15%  
Reititinlinkin käyttöaste = 15%  
Saantiaika = Internet delay +  
Access delay + LAN delay  
= 2 sec + msec + msec

### Mitähän nopeampi linkki maksaa?

Voi olla kallis ratkaisu



KuRo05: Fig 2.11

## Proxy-palvelimen käytöstä

KuRo05: Fig 2.12

### Parannus?

Asennetaan proxy-palvelin

### Oletetaan,

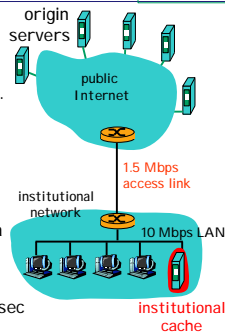
osumatodennäköisyys (hit rate) = 0,4.

### Tällöin

40% pyynnöistä löytyy heti läheltä  
Reititinlinkin käyttöaste putoaa 60%:iin  
ei jonotusviilpeltä, saantiaika 10 ms

60% pyynnöistä palvelimelta saakka

Saantiaika = Internet delay +  
Access delay + LAN delay  
= 0,6 \* (2 + 0,01) sec + 0,4 \* 0,01 sec  
= 1,2 secs



## Conditional GET

Välimuistiin talletettu objekti haetaan verkosta vain,  
jos objektia on muutettu  
- Aikaleima silti tarkistettava

### GET-pyyntönsä otsakkeessa

If-modified-since: aikaleima

esim. Mon, 5 Feb 2007 09:23:24

### Jos ei muutettu, vastauksen otsakkeessa

HTTP/1.0 304 Not Modified

Eikä objektia mukana

### Muuten objekti mukana normaalisti

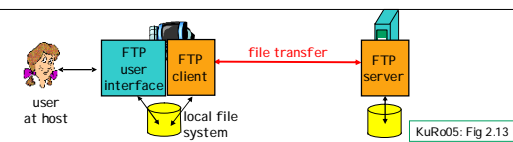
## Muita URL-osoitteita

- file:///C:/webs/html/mottle.gif  
Avaa paikallinen tiedosto (asiakkaan tiedostojärjestelmässä)  
Selain ei generoi HTTP -pyyntöä, KJ huolehtii
- ftp://usc.edu/pubs/myfile.doc  
Hae tiedosto ftp-protokollaa käyttäen
- news:hy.opiskelu.tht.tili  
Avaa uutistenluohjelman käyttöliittymä ja muodosta yhteys uutispalvelimeen
- mailto:oskari.olematon@cs.helsinki.fi  
Avaa postiohjelman käyttöliittymä, välitä sähköposti postipalvelimelle
- mms:video.avi  
Avaa multimediasoitin  
Nouda MultiMedia Streaming -protokollaa käyttäen

## Verkkosovelluksia, sovellusprotokollia

# Tiedostonsiirto FTP

## FTP: File Transfer Protocol



- Tiedostojen kopiointi koneelta koneelle  
Asiakas voi selata etäkoneen hakemistoissa FTP-sanomilla, voi noutaa tai tallettaa haluamansa tiedoston (download/upload)
- FTP-palvelin kuuntelee porttia 21  
yhteys kontrollitiedon välitystä varten
- Asiakas kuuntelee porttia 20  
palvelija avaa tiedoston siirtoa varten
- FTP-palvelin ylläpitää tilatietoa  
mm. työhakemiston polku, autentikointi

## FTP-pyyntöjä ja -vastauksia

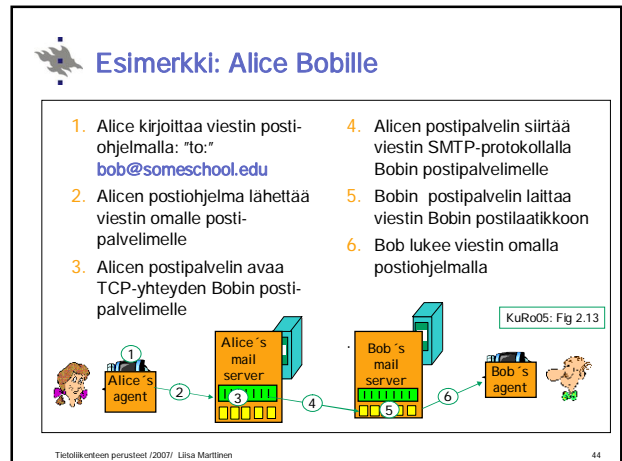
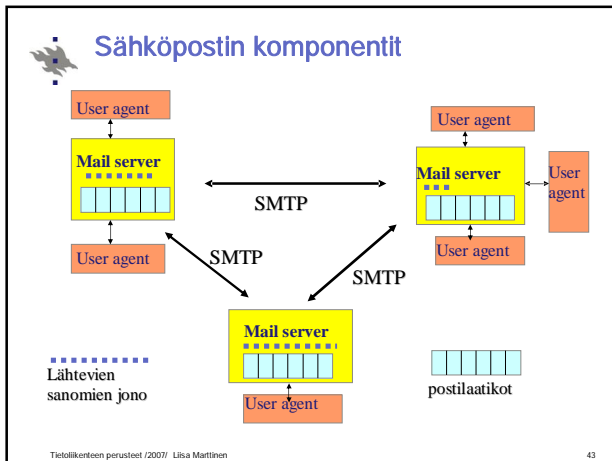
- Kaikki sanomat ASCII-muodossa
- Pyyntöjä
  - USER username
  - PASS password
  - LIST
  - RETR filename
  - STOR filename
- Vastauksia
  - 331 Username OK, password required
  - 125 Data connection already open, transfer starting
  - 424 Can't open data connection
  - 452 Error writing file

## Verkkosovelluksia, sovellusprotokollia

# Sähköposti SMTP, IMAP, POP3

## Sähköpostin komponentit

- Postiohjelma (user agent)  
Postin lukeminen ja lähettäminen  
Eudora, Outlook, elm, pine, Messenger, Pegasus, Kmail, ...  
Posti talletettuna omalle postipalvelimelle
- Postipalvelin (mail server)  
Kullakin käyttäjällä on oma saapuvien postien laatikko  
Yhteinen lähtevien postien laatikko
- Postiprotokolla SMTP  
Protokolla, jolla postipalvelin välittää postin suoraan vastaanottajan postipalvelimelle  
asiakas = lähettävä postipalvelin  
palvelin = vastaanottava postipalvelin



- ### SMTP (Simple Mail Transfer Protocol)
- Postipalvelimet kuuntelevat portilla 25
  - Säilyvä TCP-yhteys
    - luotettava
    - yksi yhteys: kaikki samalle palvelimelle menevät viestit
  - Lähetyksessä: Kättely, Viestien välitys, Lopetus
  - Pyyntö-vastaus-protokolla
    - Pyyntö: ASCII-tekstiä
    - Vastaus: status-koodi ja fraasi
  - Push-protokolla: työntää tietoa vastapäähän
    - vt. HTTP on ns. pull-protokolla
- 
- Tietoliikenteen perusteet /2007/ Liisa Marttinen 45

- ### SMTP
- Kaikki esitettävä 7-bittisenä ASCII:na
    - = IRA, International Reference Alphabet
    - Myös binääridata, esim. kuvat ja ääni
  - Yksittäinen viesti loppuu omalla rivillä olevaan plsteeseen
    - eli lopussa ASCII-merkit: **CRLF.CRLF**
    - Vanha protokolla!
  - Binääridata on koodattava s.e. siinä ei esiinny **CRLF.CRLF**
    - MIME**-laajennus
- CR = carriage return  
LF =line feed
- Tietoliikenteen perusteet /2007/ Liisa Marttinen 46

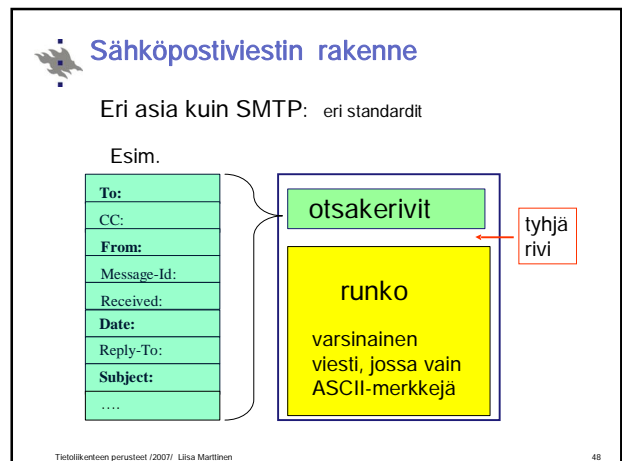
### Esimerkki

```

S: 220 helsinki.fi
C: HELO princeton.edu
S: 250 Hello princeton.edu
C: MAIL FROM: <Bob@princeton.edu>
S: 250 <Bob@princeton.edu> OK
C: RCPT TO: <pekka.puupaa@cs.helsinki.fi>
S: 250 <pekka.puupaa@cs.helsinki.fi> OK
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: dataa ... dataa
C: dataa ... dataa
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 princeton.edu closing connection
  
```

SMTP:n kättely  
 Viesti(t)  
 SMTP:n lopetus

Tietoliikenteen perusteet /2007/ Liisa Marttinen 47







## MIME (Multipurpose Internet Mail Extension, RFC 2045, 2056)

- Kaikki on koodattava 7-bittiseksi ASCII-koodiksi
- Lisää kenttiä otsakkeeseen: vastaanottajan postiohjelma osaa käynnistää oikean sovelluksen viestin näyttämiseksi.

MIME-versio  
Koodausmenetelmä  
multimediadatan tyyppi, alityypit, parametrit  
koodattu data

```

From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.....base64 encoded data

```



## MIME

MIME-sisältötyyppiä

```

text/plain; charset=us-ascii
text/html
image/gif, image/jpeg,
video/mpeg
application/postscript
application/msword
application/octetstream
multipart/mixed

```

MIME-versio:  
Content-Transfer-Encoding: ---  
Content-Type: ---

Base-64-koodaus

Sanoma 24 bitin ryhmät on jaettu 6 bitin osiksi, jotka kukin on koodattu ASCII-merkiksi. 64 eri vaihtoehtoa



## Moniosainen MIME-viesti

```

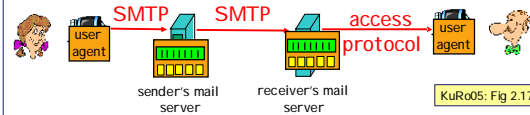
...
Content-Type: multipart/mixed; Boundary=StartOfNextPart
-- StartOfNextPart
Hel Allu,
sinulle kaunis kuva kissastani Villestä.
-- StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: Image/jpeg
base64 encoded data .....
.....base64 encoded data
-- StartOfNextPart
Haluatko multa kuvia!

```



## Postinnoutoprotokollat (mail access protocols)

- Posti omalta postipalvelimelta postiohjelmaan
- Koska SMTP on 'PUSH'-protokolla, sitä ei voi käyttää sanomia haettaessa ('PULL').
- POP3: Post Office Protocol versio 3  
Viestin lataamiseen omalle koneelle, ei postikansioita
  - IMAP: Internet Mail Access Protocol  
Monipuolisempi: postikansiot (folders), lataa vain otsikot, viestien säilytys postipalvelimelle
  - HTTP: Esim. TKTL:lla käytettävä IlohaMail, Hotmail, ...  
Web-palvelija käyttää IMAP-palvelijaa



## Verkkosovelluksia, sovellusprotokollia

# Internetin nimipalvelu

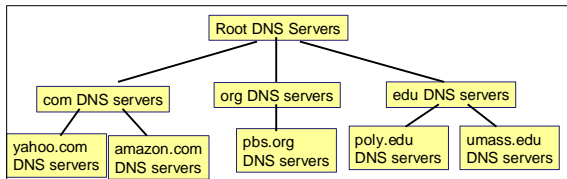
## DNS



## DNS (Domain Name System)

- Sovellusprotokolla
- Isännät ja nimipalvelimet käyttävät  
Käyttää itse UDP-kuljetuspalvelua
  - 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 allasnimet, palvelijan replikoinnin  
Esim. WWW.cs.helsinki.fi ja cs.helsinki.fi ovat allasnimiä  
Esim. www-palvelijaan voi liittyä useita IP-osoitteita, rotaatio
  - Hajautettu, hierarkinen tietokanta (hakemisto)  
Toteutettu uselden replikoitujen nimipalvelmien yhteistyönä  
skaalautuvuus, kuormantasaus, ylläpito, vikasietoisuus, ..  
Jos oma nimipalvelija ei tunne, se kysyy mullta.

## Hajautettu, hierarkinen tietokanta

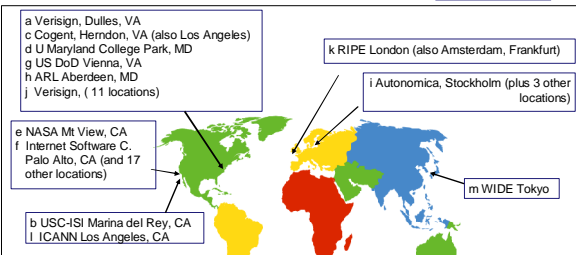


KuRo05: Fig 2.18

- n 13 juuritason nimipalvelija Replikoituja, kaikilla samat tiedot
- n Ylätason palvelimet maa- ja yleistunnuksille (n. 265 kpl) ..., fi, fr, uk, ... edu, net, com, org, ...
- n Autorisoidut aluepalvelimet (domain) (2-taso) [www.iana.org](http://www.iana.org)  
Isoilla yliopistoilla ja firmoilla omansa, pienet käyttävät jonkun muun ylläpitämää

## Juuripalvelimet

KuRo05: Fig 2.21

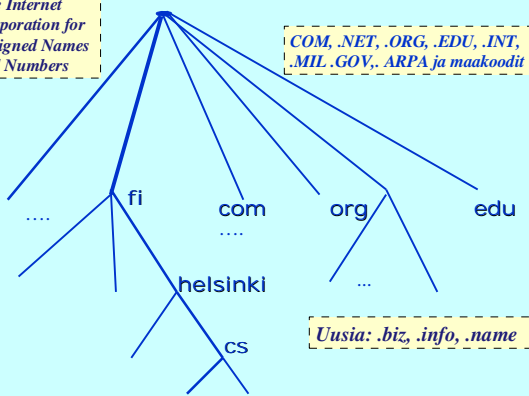


Juuripalvelimet tietävät, mikä ylätason palvelin on vastuussa maa- ja yleistunnuksesta. Ylätason palvelimet tuntevat omat aluepalvelimensä. Aluepalvelimet tuntevat juuripalvelijan. Koneen oma palvelija on merkitty koneen asetustietoihin.

## Domain -nimiavaruus

ICANN  
The Internet  
Corporation for  
Assigned Names  
and Numbers

COM, NET, .ORG, .EDU, .INT,  
.MIL, .GOV., ARPA ja maakoodit



Uusia: .biz, .info, .name

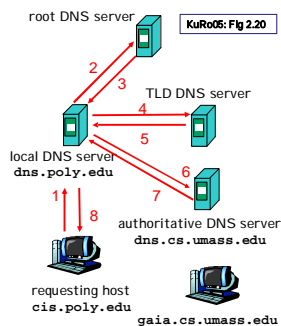
## IP-nimen selvittäminen

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

## Iteratiivinen kysely: "kerro keneltä pitää kysyä"

gaia.cs.umass.edu

- 1) Isäntäkone
- 2) Kysy omalta aluepalvelijalta
- 3) Aluepalvelija (poly) Ota yhteys juuritasolle
- 4) Juuripalvelin Kerro, mistä löytyy ylätason palvelin edu-tunnuksille
- 5) Ylätason palvelin (edu) Kerro, mistä löytyy aluepalvelija
- 6) Aluepalvelija umass.edu-tunnuksille
- 7) Aluepalvelija Tuntee cs-verkon koneet.
- 8) Kerro koneen IP-osoite



KuRo05: Fig 2.20

## Rekursiivinen kysely: "kysy muilta, jos et itse tiedä"

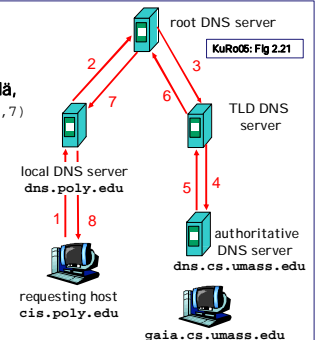
Kysy ensin omalta aluepalvelimelta. (1, 8)

Jos oma aluepalvelija ei tiedä, se kysyy juuripalvelijalta. (2, 7)

Jos juuripalvelin ei tiedä, se kysyy ylätason palvelijalta. (3, 6)

Jos ylätason palvelija ei tiedä, se kysyy aluepalvelijalta. (4, 5)

Aluepalvelija tuntee omat verkkonsa ja koneensa.



KuRo05: Fig 2.21



## DNS-väilmuisti

Suorituskyvyn parantamiseksi nimipalvelijat varastoivat väilmuistiinsa näkemiään DNS-resurssitietueita.

- Ei tarvitse aina hakea uudestaan
  - Kuormittaa vähemmän ylempään tason nimipalvelimia
  - Nopeuttaa tavallisimpia kyselyjä: löytyy läheltä
- Tiedon oikeellisuus
  - Tietueelle määrätty elinaika (TTL, time to live) kertoo voimassaoloajan.
  - Kun umpeutuu, tieto poistetaan.
  - Yleensä muutokset paikallisia: koneen lisäys, koneen poisto, joskus uusi verkko

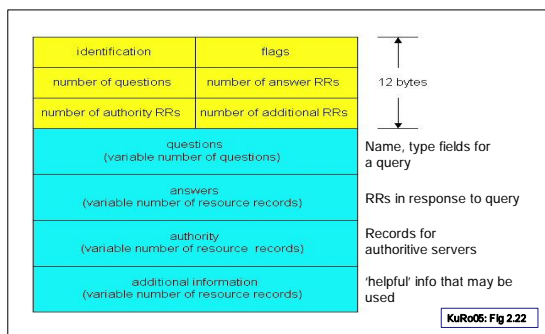


## DNS-nimiavaruuden vyöhykejako

- DNS-nimiavaruus jaettu vyöhykkeisiin (zone)
  - kukin vyöhyke kattaa osan nimipuusta
  - vyöhykkeellä on yksi siitä vastaava nimipalvelija (primary) ja yksi tai useita apunimipalvelijoita (secondary)
- Vyöhykejako on hallinnollinen
  - tarpeen mukaan nimipalvelijoita vastaamaan omasta alueestaan



## DNS-sanoman rakenne



## DNS-sanoma

- Kysely ja vastaus käyttävät samaa formaattia
  - Kyselystä voi generoitua vastaus, jossa on useita resurssitietueita
    - Esim. Palvelijafarmien kuormantasaaminen: vastauksessa on useita IP-osoitteita (rotaatio)
- Identification-kenttä
  - Kyselyn tunniste (16-bittinen numero) ja vastauksessa sama numero => kysely ja vastaus helposti yhdistettävissä toisiinsa.
- Lipukkeet (flags)
  - Pyyntö vai vastaus
  - Käytä rekursiivista kyselyä
  - Rekursiivinen kysely mahdollista
  - Vastaus tulee suoraan autorisoidulta palvelijalta



## DNS- resurssitietue

- Kentät: (nimi, arvo, tyyppi, elinaika maiden tunnuksset )
- Tyyppi määrää nimen ja arvon merkityksen:
  - **Tyyppi = A** (host address)
    - nimi = koneen nimi, arvo = IP-osoite
    - esim: (relay1.bar.foo.com, 145.37.3.126, A, TTL)
  - **Tyyppi = ND** (name server)
    - nimi = aluenimi (domain), arvo = autorisoidun palvelimen nimi
    - esim: (foo.com, ds.foo.com, NS, TTL)
  - **Tyyppi = CNAME** (canonical name)
    - nimi = koneen aliasnimi, arvo= kanoninen, oikea konenimi
    - esim: (foo.com, relay1.bar.foo.com, CNAME, TTL)
  - **Tyyppi = MX** (mail exchange)
    - nimi = koneen aliasnimi, arvo = postipalvelimen kanoninen nimi
    - esim: (foo.com, mail.bar.com, MX, TTL)



## Verkkosovelluksia, sovellusprotokolla

Vertaistoimijat  
peer-to-peer

## Vertaistoimijat: file sharing

- Isäntäkoneet asiakkaan ja palvelijan roolissa
  - Noutaa MP3-tiedostoja, videoleikkeitä, ...
  - Tarjoaa omiaan muiden noudettavaksi
- Kone on satunnaisesti Internetissä
  - IP-osoitekin voi vaihdella kerrasta toiseen
- Miten löytää vertaistoimija(t)?
  - Keskitetty hakemisto: kiinteä IP-osoite, josta voi kysellä
  - Kyselyn tulvitus: kysellään potentiaalisilta toimijoilta
  - Hiukan keskitetty hakemistopalvelu, joka tekee jatkokyselyt
- Kun kohde löytynyt, kopiointi suoraan sieltä
  - Kyselyn tuloksena IP-osoite
  - HTTP-protokollaa käyttäen
- Skaalautuvuus, kuormantasaus

## Keskitetty hakemisto

Esim. Napster

Keskitetty hakemisto, hajautettu siirto

1. Aina kun liitetään Internetiin, ilmoitetaan hakemistolle IP-osoite ja jaettavat tiedostot
2. Tiedostoa haettaessa kerrotaan haetun tiedoston nimi tai tunniste ja vastauksena tulee IP-osoite, josta tiedoston voi hakea.
3. Tieto haetaan saadusta osoitteesta

**Ongelmia:**  
Suorituskyky  
Vikasietoisuus  
Tekijänoikeudet

KuRo05 Fig 2.23

## Kyselyn tulvitus

Esim. Gnutella

Kysely tulvitettuna, kopiointi suoralla yhteydellä

- Käyttää pystyssäolevia TCP-yhteyksiä kyselyn hajauttamiseen
- Välttää kyselyn edelleen, jolle itse pysty sitä täyttämään
- Jos pystyy, niin lähettää vastauksena pyydetyn tiedoston
- Kopiointi suoralla yhteydellä

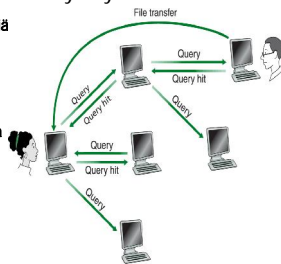
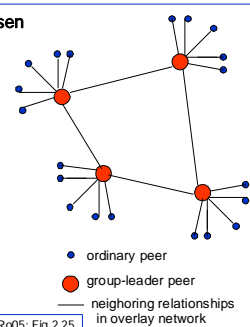


Figure 2.24 Search and file transfer in Gnutella

## Heterogeeniset toimijat

Esim. KaZaA

- Joko ryhmänjohtaja tai ryhmän jäsen
  - jäsenet tuntevat johtajan
  - johtaja tuntee muita johtajia
- Ryhmänjohtaja tietää jäseniensä tiedostot
  - kertovat yhteyttä ottaessaan
- Ryhmänjohtaja kyselee muita tiedostoja tuntemiltaan ryhmänjohtajilta
- Ei-julkinen protokolla
  - kontrollisanomat kryptattu
- Tiedoston kopiointi suoralla yhteydellä



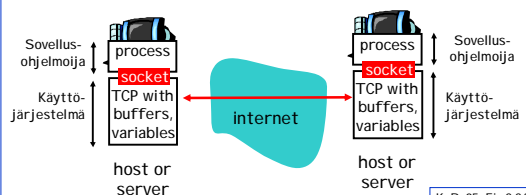
KuRo05: Fig 2.25

## Verkkosovelluksia, sovellusprotokollia

Pistoke  
Verkkosovelluksen  
ohjelmointia

## Pistoke (socket)

- Kuljetuspalvelun ja sitä käyttävän sovelluksen rajapinta isäntäkoneessa
  - Sovelluksen tietoliikenne = KJ:n palvelupyyntöjä
  - Pistoke on "palveluluukku"
- Alunperin Berkeley UNIXin (BSD) mukana



KuRo05: Fig 2.26

## Pistoke

- n Sovellus luo pistokkeen ja liittää sen porttiin tai KJ voi valita porttinumeron
  - n Yksi pistoke per porttinumero
- n Kaksisuuntainen (full duplex)
  - n Yksi pistoke, josta luetaan ja johon kirjoitetaan
- n Lähetys (send)
  - n Kirjoita pistokkeeseen
- n Vastaanotto (receive)
  - n Lue pistokkeesta

## Posix: Pistokerajapinta (API)

```

sockfd = socket(domain, type, protocol)
    luo yhteydellinen (TCP) tai yhteydetön (UDP) pistoke
bind(sockfd, addr[, addr_len])
    porttinumeron kytkeminen prosessiin
listen(sockfd, backlog)
    yhteyksipyyntöön odottaminen (welcoming socket) (TCP)
sockfd = accept(sockfd, addr[, *addr_len])
    yhteyksipyyntöön hyväksyminen, luo uusi pistoke (connection socket)
connect(sockfd, addr[, addr_len])
    yhteyksipyyntöön lähetyksen, mihin koneeseen ja porttiin (TCP)
send(sockfd, buf[, buf_len, flags])
recv(sockfd, buf[, buf_len, flags])
    tavuvirran lähetyksen ja vastaanotto (TCP)
send(sockfd, msg[, msg_len, flags, addr[, *addr_len])
recv(sockfd, msg[, msg_len, flags, addr[, *addr_len])
    sanoman lähetyksen ja vastaanotto, mukana kone ja portti (UDP)
close(sockfd), shutdown(sockfd, how)
    yhteyden lopettaminen (TCP)
    
```

## TCP-kuljetuspalvelu

- n Yhteyksipyyntö palvelun porttiin
- n Palvelija luo yhteyttä varten uuden portin
  - Voi palvella useita yhteyksipyyntöjä
- n Tavallisesti palvelija luo yhteyttä varten myös oman prosessin
- n Lue /kirjoita tavuja

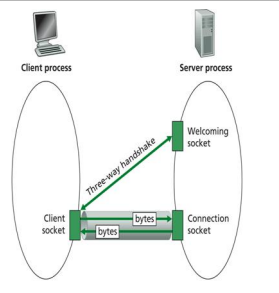


Figure 2.27 ♦ Client socket, welcoming socket, and connection socket

## TCP-kuljetuspalvelu

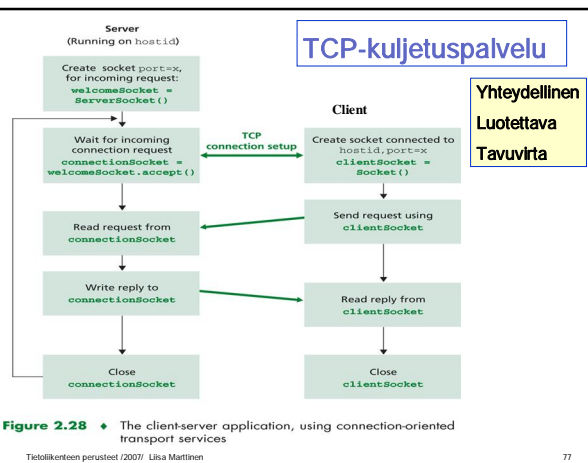
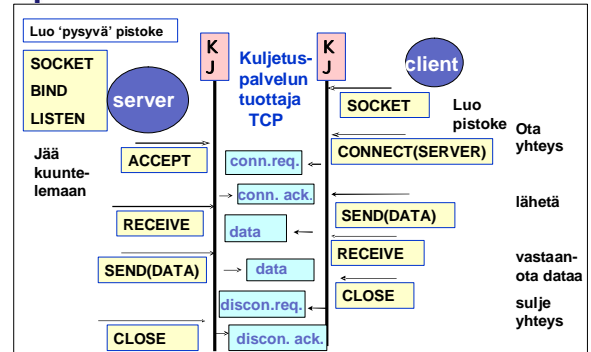


Figure 2.28 ♦ The client-server application, using connection-oriented transport services

## Esimerkki: TCP-palvelija (Java)

```

import java.io.*; import java.net.*;
class TCPServer {
    public static void main(String argv[]) throws Exception {
        String clientSentence;
        String capitalizedSentence;
        ServerSocket welcomeSocket = new ServerSocket(6789);
        while(true) {
            Socket connectionSocket = welcomeSocket.accept();
            BufferedReader inFromClient = new BufferedReader(
                new InputStreamReader(connectionSocket.getInputStream()));
            DataOutputStream outToClient =
                new DataOutputStream(connectionSocket.getOutputStream());
            clientSentence = inFromClient.readLine();
            capitalizedSentence = clientSentence.toUpperCase() + '\n';
            outToClient.writeBytes(capitalizedSentence);
        }
    }
}
    
```

## Esimerkki: TCP-asiakas (Java)

```
import java.io.*; import java.net.*;
class TCPClient {
    public static void main(String argv[]) throws Exception {
        String sentence;
        String modifiedSentence;
        BufferedReader inFromUser =
            new BufferedReader(new InputStreamReader(System.in));
        Socket clientSocket = new Socket("hostname", 6789);
        DataOutputStream outToServer =
            new DataOutputStream(clientSocket.getOutputStream());
        BufferedReader inFromServer =
            new BufferedReader(new InputStreamReader(
                clientSocket.getInputStream()));
        sentence = inFromUser.readLine();
        outToServer.writeBytes(sentence + '\n');
        modifiedSentence = inFromServer.readLine();
        System.out.println("FROM SERVER: " + modifiedSentence);
        clientSocket.close();
    }
}
```

Tietoliikenteen perusteet /2007/ Liisa Marttinen

79

## UDP-kuljetuspalvelu

- n Ei kättelyä, yhteydenmuodostusta /purkua
- n Ei-luotettava
- n Sovellusprosessi lukee ja kirjoittaa kokonaisia sanomia
- n Lähettäjä kertoo KJ:lle sanoman lisäksi kohteen IP-osoitteen ja portin  
 POSIX: `send(sockfd, msg[], msg_len, flags, addr[], addr_len)`
- n Vastaanottaja saa KJ:ltä mahdollista vastausta varten lähettäjän IP-osoitteen ja portin  
 POSIX: `recvfrom(sockfd, msg[], msg_len, flags, addr[], *addr_len)`

Tietoliikenteen perusteet /2007/ Liisa Marttinen

80

## UDP-kuljetuspalvelu

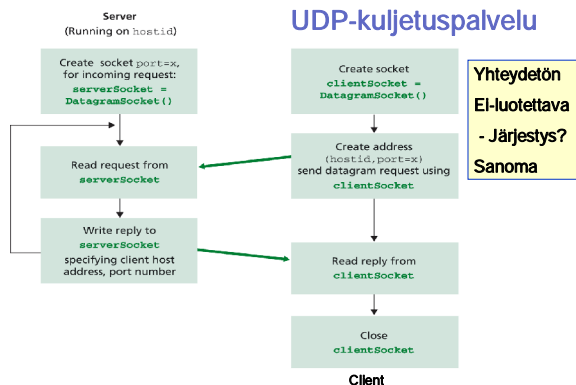
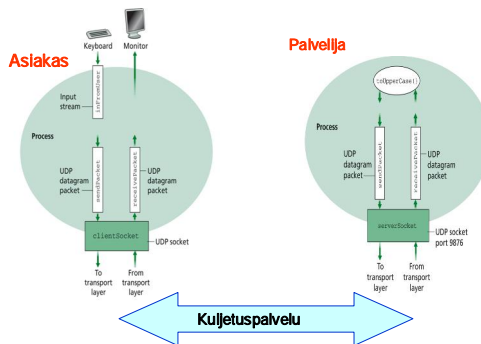


Figure 2.30 ♦ The client-server application, using connectionless transport services

Tietoliikenteen perusteet /2007/ Liisa Marttinen

81

## UDP-esimerkki



Tietoliikenteen perusteet /2007/ Liisa Marttinen

82

## Esimerkki: UDP-asiakas (Java)

```
import java.io.*; import java.net.*;
class UDPClient {
    public static void main(String args[]) throws Exception {
        BufferedReader inFromUser =
            new BufferedReader(new InputStreamReader(System.in));
        DatagramSocket clientSocket = new DatagramSocket();
        InetAddress IPAddress = InetAddress.getByName("hostname");
        byte[] sendData = new byte[1024];
        byte[] receiveData = new byte[1024];
        String sentence = inFromUser.readLine();
        sendData = sentence.getBytes();
        DatagramPacket sendPacket =
            new DatagramPacket(sendData, sendData.length, IPAddress, 9876);
        clientSocket.send(sendPacket);
        DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
        clientSocket.receive(receivePacket);
        String modifiedSentence = new String(receivePacket.getData());
        System.out.println("FROM SERVER: " + modifiedSentence);
        clientSocket.close();
    }
}
```

Tietoliikenteen perusteet /2007/ Liisa Marttinen

83

## Esimerkki: UDP-palvelin (Java)

```
import java.io.*; import java.net.*;
class UDPServer {
    public static void main(String args[]) throws Exception {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];
        while(true) {
            DatagramPacket receivePacket =
                new DatagramPacket(receiveData, receiveData.length);
            serverSocket.receive(receivePacket);
            String sentence = new String(receivePacket.getData());
            InetAddress IPAddress = receivePacket.getAddress();
            int port = receivePacket.getPort();
            String capitalizedSentence = sentence.toUpperCase();
            sendData = capitalizedSentence.getBytes();
            DatagramPacket sendPacket =
                new DatagramPacket(sendData, sendData.length, IPAddress, port);
            serverSocket.send(sendPacket);
        }
    }
}
```

Tietoliikenteen perusteet /2007/ Liisa Marttinen

84



## Kertauskysymyksiä

- n Asiakas-palvelija-malli? Peer-to-peer?
- n Kuinka asiakas löytää palvelimen?
- n Miten KJ osaa antaa bitit oikealle sovellukselle?
- n Miten koneen nimestä saadaan selville sen IP-osoite?
- n Mitä hyötyä on proxy-palvelimesta?
- n Miksi käytetään evästeitä?
- n Mikä on pistoke ja missä sitä käytetään?

Ks. myös kurssikirja s.170.