

# Tietoliikenteen perusteet



## VERKKOSOVELLUKSIA SOVELLUSPROTOKOLLIA

Kurose, Ross: Ch 2



## Sisältöä

- n Verkkosovellusten periaatteet
- n World Wide Web ja HTTP
- n Tiedostonsiirto ja FTP
- n Sähköposti ja SMTP, IMAP, POP3
- n Nimipalvelu ja DNS
- n Vertaistoimijat (peer-to-peer)
- n Pistoke ja sen käyttö

### Oppimistavoitteet:

- Osata selittää asiakas-palvelija –malliin perustuvien verkkosovellusten toimintaperiaatteet
- Tuntea sovellusprotokollien syntaksia ja semantiikkaa
- Osata selittää nimipalvelun, www:n ja sähköpostin toimintaideat
- Tunnistaa pistokkeiden käytön periaatteet





## Verkkosovellus

# Verkkosovellusten periaatteet



## Verkkosovellus

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

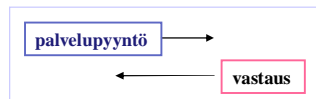


# Sovellusarkkitehtuuri



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

- n Aina toiminnassa oleva palvelijaohjelma, jolla kiinteä, tunnettu IP-osoite
- n Asiakasohjelmat halutessaan ottavat yhteyttä palvelimeen ja pyytävät siltä palvelua



## n Vertaistoimijamalli (esim. Gnutella)

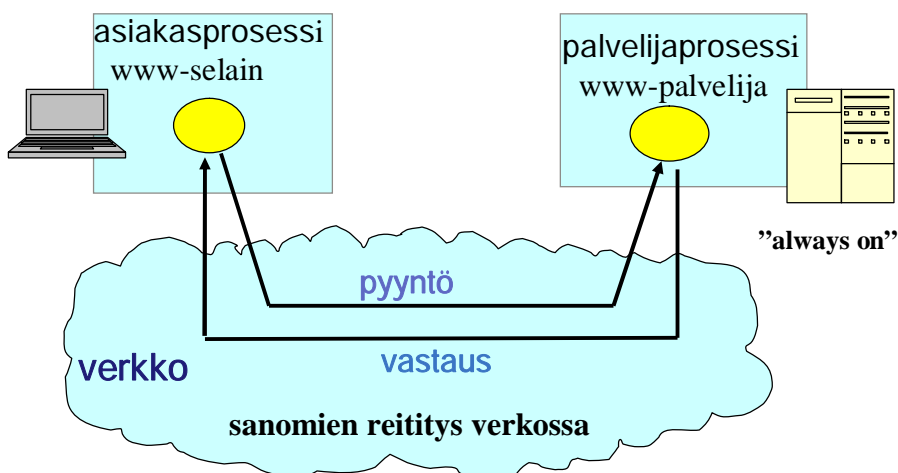
- n Vertaisiännät kommunikoivat suoraan keskenään
- n Ei tarvitse olla aina toiminnassa, IP-osoite voi muuttua
- n Jokainen toimii sekä palvelijana että asiakkaana



## n Hybridimalli (esim. Napster)



# Asiakas-palvelija-malli



Oikea kone, oikea prosessi



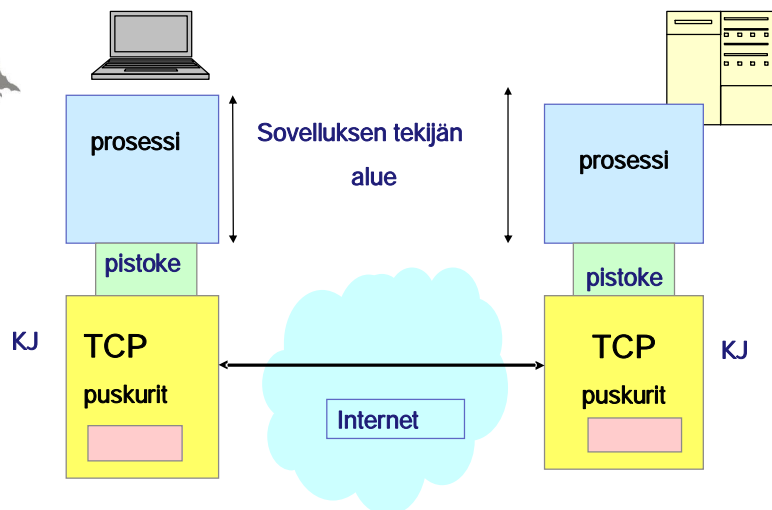
## Sovelluksen rajapinta tietoliikenteeseen

- 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!

Tietoliikenteen perusteet /2007/ Liisa Marttinen

7



KJ = käyttöjärjestelmä

Prosessien kommunikointi TCP-pistokkeita käyttäen

Tietoliikenteen perusteet /2007/ Liisa Marttinen

8



## KJ:n rajapinta laitteistoon

n KJ:n kannalta tietoliikenne normaalia siirrantää (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)

Protokollan mukaan

### Tuleva liikenne

Verkkokortti ottaa vastaa 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**.

Tietoliikenteen perusteet /2007/ Liisa Marttinen

9



## Osoittaminen

n Sanomissa oltava lähettäjän ja vastaanottajan IP-osoite ja porttinumero

n **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

n **porttinumero à oikea prosessi**

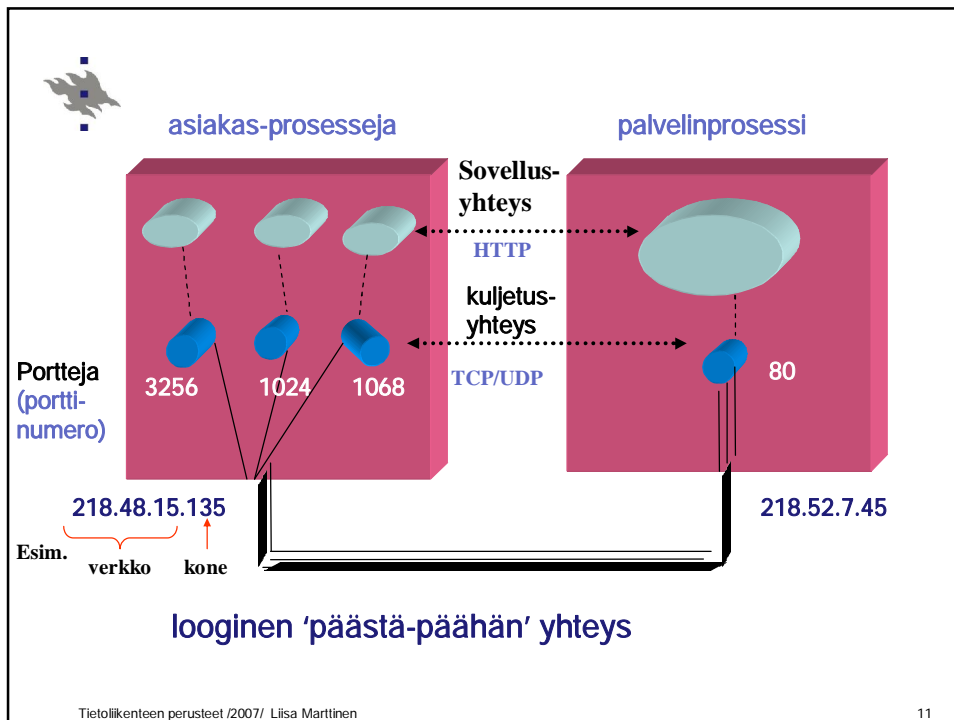
Yleisillä palveluilla standardoidut tunnetut porttinumerot:

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

KJ osaa liittää porttinumeron prosessiin

Tietoliikenteen perusteet /2007/ Liisa Marttinen

10



## 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

Tietoliikenteen perusteet /2007/ Liisa Marttinen

12



## 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 hiljentää vauhtia, jos **vastaanottaja** ei ehdi käsitellä

#### Ruuhkanvalvonta (congestion control)

Lähettäjä hiljentää 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 viipeelle (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 (Hyper Text Markup Language)

### n WWW-sivu, WWW-dokumentti

HTML-tekstiä, jossa viittauksia muihin objekteihin  
muu HTML-tiedosto, kuva- tai äänitiedosto, Java  
applet, ...

Sivu muodostuu usean tiedoston sisällöstä, jotka  
noudetaan palvelijalta

### n Viittaus URL-osoitteella (Uniform Resource Location)

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

koneen nimi

Viitatus objektin  
polkunimi



## HTML (HyperText Markup Language)

### n Standardi siitä, kuinka sivun rakenne kuvataan

Muotoilut, eri osien sijoittelu sivuille  
Viittaukset muihin objekteihin

### n SGML (Standard Generalized Markup Language)

yleinen merkkuskieli  
kertoo, kuinka dokumentit muotoillaan - ladontamerkinnot

### n XML (Extensible Markup Language)

rakenteellinen tietosisällön kuvaus, myös merkitys kuvattu

### n Näistä enemmän kurssilla:

582304 XML-metakieli (4 op/2ov)



# HTTP (HyperText Transfer Protocol)

PC, jossa on Explorer-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



Palvelin, jossa on Apache-www-palvelija

HTTP Request  
HTTP Response

HTTP Response  
HTTP Request

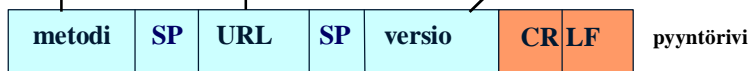


Linux-kone, jossa on Firefox-selain

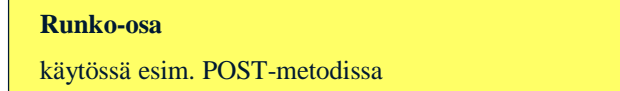
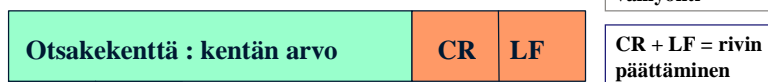


# HTTP-pyyntö: yleinen rakenne

GET /jokuhakemisto/sivu.html HTTP/1.1



... Lisää otsakerivejä



SP='space' eli välilyönti

CR + LF = rivin päättäminen



## Esimerkki: HTTP-pyyntö

Pyyntöriivi: GET/ POST/ HEAD -metodi

GET /somedir/page.html HTTP/1.1

otsakerivit

Host: www.someschool.edu

User-Agent: Mozilla/4.0

Connection: close

Accept-language: fr

Yksi tyhjä rivi  
merkinä sanoman  
loppumisesta

(Ylimääräinen carriage return, line feed)

Otsakeriveillä välitetään parametritietoja



## 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-julkaisu -  
työkalut käyttävät



Otsakekenttä : kentän arvo	CR	LF
----------------------------	----	----

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

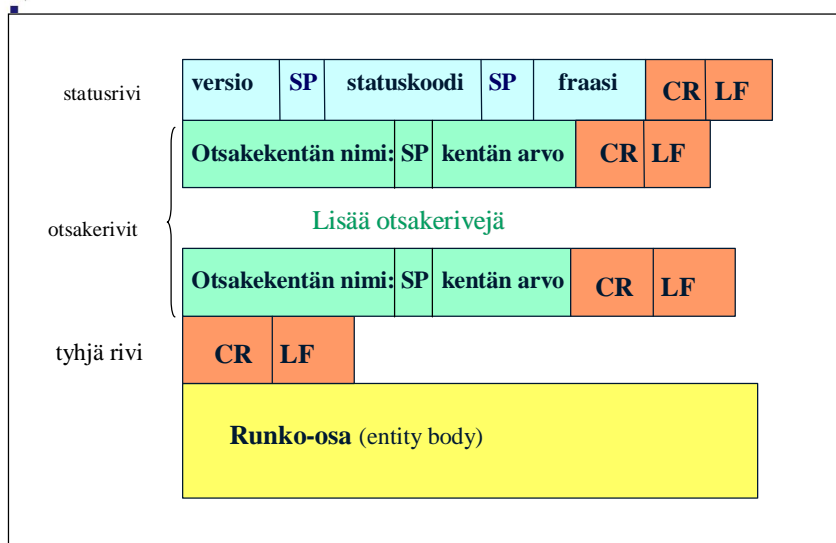
**Connection:** **close** sulje yhteys lähetyksen jälkeen

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

**Accept-language:** **fi** dokumentin kieli

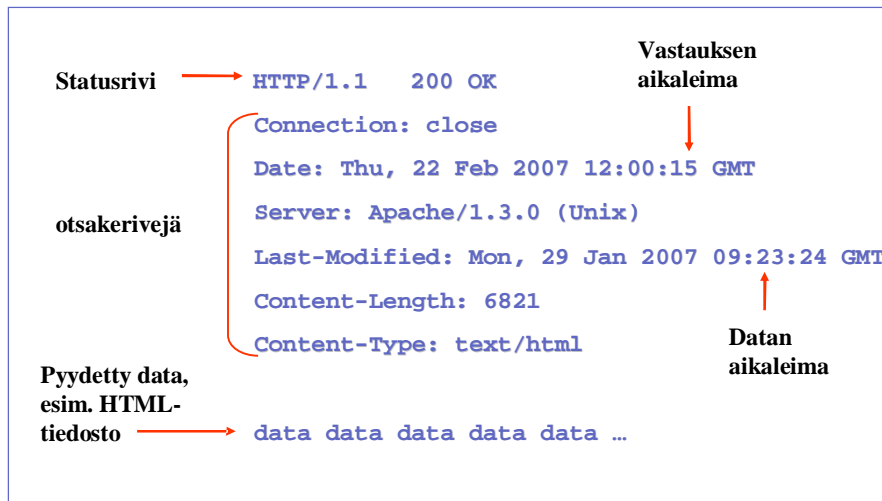


## HTTP-vastaus: yleinen rakenne





## Esimerkki: HTTP-vastaus



## HTTP: statuskoodeja ja fraaseja

Vastaussanoman 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:sta

### 400 Bad Request

Palvelija ei ymmärtänyt pyyntöä

### 404 Not Found

Pyydettyä objektiä 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**



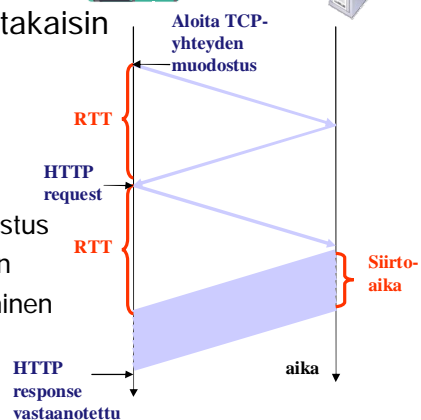
## Vastausaika (response time)

**n Kiertoviive** (Round-trip time, RTT):  
aika, joka kuluu pikkupaketin  
siirtoon palvelimelle ja takaisin



**n Vastausaika = 2 RTT  
+ siirtoaika**

1 RTT TCP-yhteyden muodostus  
1 RTT pyyntö + ensimmäisten  
vastausbittien saapuminen  
Tiedoston siirtoaika





## Suorituskyky?

n 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

n 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

**Liukuhihnoitettu** (pipelining) / liukuhihnoittamaton: seuraava pyyntö  
lähtee jo ennenkuin edelliseen on saatu vastaus / ei lähde.



## Evästeet (cookies)



n **HTTP on tilaton protokolla**

Palvelija ei talleta mitään istuntoon liittyvää

n **Selain**

Tallettaa asiakaskoneelle (tiedostoon) palvelimen pyynnöstä ja  
sen tarpeita varten käyttäjäkohtaista tietoa  
Lähetää tiedot palvelijalle joka pyynnön yhteydessä.

n **Palvelin**

Ylläpitää tietokantaa käyttäjistä (back-end database)  
yksikäsitteiset käyttäjätunnisteet (tav. numero)

n **Evästeiden talletus ja lähetys**

HTTP-vastauksessa otsakerivi: **Set-cookie: tieto**

HTTP-pyynnössä otsakerivi: **Cookie: tieto**



## Mihin evästeitä käytetään?

### n Käyttäjien tunnistamiseen

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

### n Istunnon vaiheen tallentamiseksi

Autentikointi vain kertaalleen esim. www-postinlukuohjelman yhteydessä

### n Ostoskorina

Selaile palveluntarjoajan sivuilla ja kerää ostokset koriin.  
Lähetä lopuksi tilaus

### n Yksityisyys?

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



## Proxy-palvelin

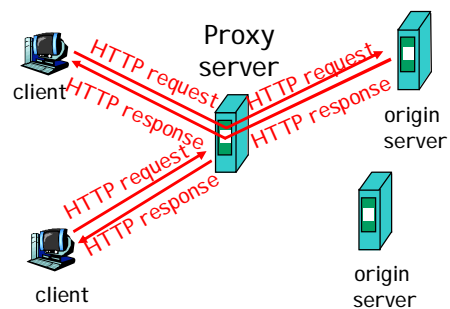
### Verkkovälimuisti

Säilyttää kopioita haetuista objekteista

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

### n Etuja

lyhentää vastausaikaa  
vähentää verkkoliikennettä  
vähentää palvelimen kuormaa



KuRo05: Fig 2.10

[Myös asiakaskone voi ylläpitää välimuistia!]





## Proxy-palvelimen käytöstä

KuRo05: Fig 2.11

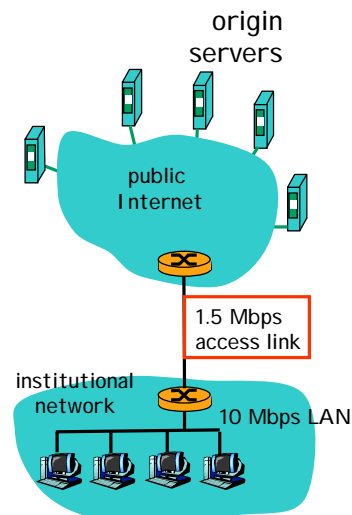
### n Oletetaan

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

### n Tällöin

paikallisverkon käyttöaste = 15%  
ei ruuhkautunut siirtoaika 10 ms

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



Tietoliikenteen perusteet /2007/ Liisa Marttinen

33



## Proxy-palvelimen käytöstä

### n Parannus?

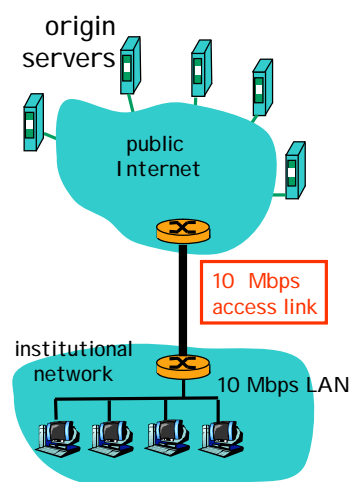
Hankitaan nopeampi yhteys,  
esim. 10 Mbps

### n Tällöin

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

### n Mitähän nopeampi linkki maksaa?

Voi olla kallis ratkaisu



KuRo05: Fig 2.11

Tietoliikenteen perusteet /2007/ Liisa Marttinen

34



## Proxy-palvelimen käytöstä

KuRo05: Fig 2.12

### n Parannus?

Asennetaan proxy-palvelin

### n Oletetaan,

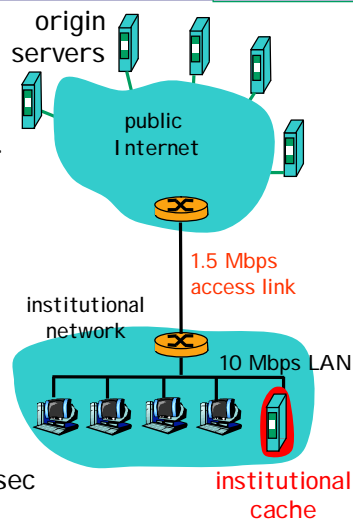
osumatodennäköisyys (hit rate)= 0,4.

### n Tällöin

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

60% pyynnöistä palvelimelta saakka

Saantiaika = Internet delay +  
Access delay + LAN delay  
=  $0.6 * (2 + 0,01) \text{ sec} + 0,4 * 0,01 \text{ sec}$   
= **1,2 secs**



Tietoliikenteen perusteet /2007/ Liisa Marttinen

35



## Conditional GET

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

### n GET-pyyntöns otsakkeessa

**If-modified-since:** aikaleima

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

n Jos ei muutettu, vastauksen otsakkeessa

**HTTP/1.0 304 Not Modified'**

Eikä objektia mukana

n Muuten objekti mukana normaalisti

Tietoliikenteen perusteet /2007/ Liisa Marttinen

36



## Muita URL-osoitteita

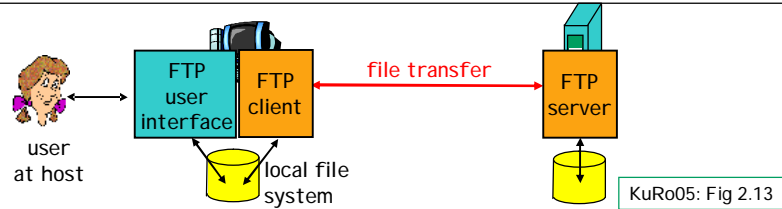
- n file:///C:/webs/html/mottle.gif  
Avaa paikallinen tiedosto (asiakkaan tiedostojärjestelmässä)  
Selain ei generoi HTTP -pyyntöä, KJ huolehtii
- n ftp://usc.edu/pubs/myfile.doc  
Hae tiedosto ftp-protokollaa käyttäen
- n news:hy.opiskelu.tht.tili  
Avaa uutistenlukuohjelman käyttöliittymä ja muodosta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



## Verkkosovelluksia, sovellusprotokollia

Tiedostonsiirto  
FTP

## FTP: File Transfer Protocol



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

## FTP-pyyntöjä ja -vastauksia

- n Kaikki sanomat ASCII-muodossa
- n Pyyntöjä
  - USER**    **username**
  - PASS**    **password**
  - LIST**
  - RETR**    **filename**
  - STOR**    **filename**
- n 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

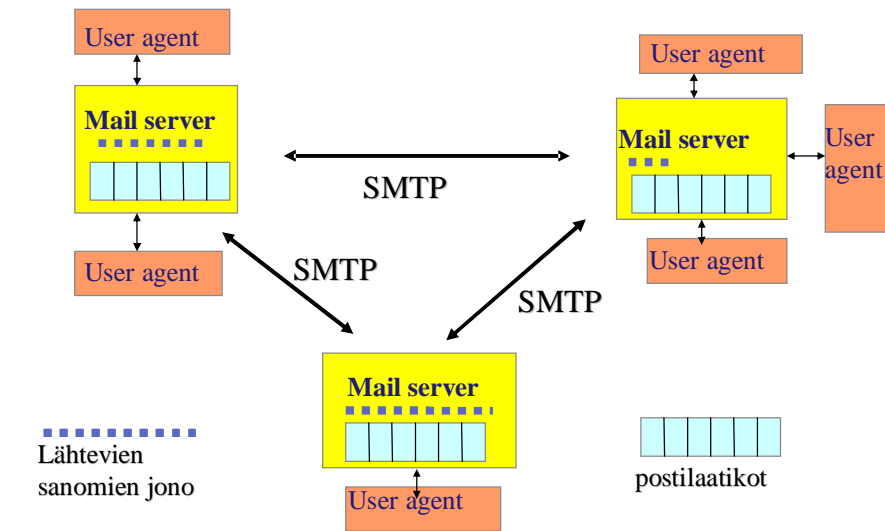
# Sähköposti



## Sähköpostin komponentit

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

## Sähköpostin komponentit

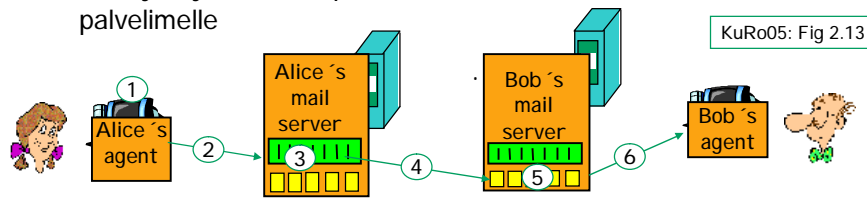


Tietoliikenteen perusteet /2007/ Liisa Marttinen

43

## Esimerkki: Alice Bobille

1. Alice kirjoittaa viestin posti-ohjelmalla: "to:" [bob@some.edu](mailto:bob@some.edu)
2. Alicen postiohjelma lähettää viestin omalle postipalvelimelle
3. Alicen postipalvelin avaa TCP-yhteyden Bobin postipalvelimelle
1. Alicen postipalvelin siirtää viestin SMTP-protokollalla Bobin postipalvelimelle
2. Bobin postipalvelin laittaa viestin Bobin postilaatikkoon
3. Bob lukee viestin omalla postiohjelmalla



Tietoliikenteen perusteet /2007/ Liisa Marttinen

44



## SMTP (Simple Mail Transfer Protocol)

- n Postipalvelimet kuuntelevat porttia 25
- n Säilyvä TCP-yhteys
  - luotettava
  - yksi yhteys: kaikki samalle palvelimelle menevät viestit
- n Lähetyksessä: Kättely, Viestien välitys, Lopetus
- n Pyyntö-vastaus-protokolla
  - Pyyntö: ASCII-tekstiä
  - Vastaus: status-koodi ja fraasi
- n Push-protokolla: työntää tietoa vastapäähän
  - vt. HTTP on ns. pull-protokolla



## SMTP

- n Kaikki esitettävä 7-bittisenä ASCII:na
  - = IRA, International Reference Alphabet
  - Myös binääridata, esim. kuvat ja ääni
- n Yksittäinen viesti loppuu omalla rivillä olevaan pisteeseen
  - eli lopussa ASCII-merkit: **CRLF.CRLF**
  - Vanha protokolla!
- n Binääridata on koodattava s.e. siinä ei esiinny **CRLF.CRLF**
  - MIME**-laajennus

CR = carriage return
LF =line feed

## 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
```

STMP:n  
kättely

Viesti(t)

STMP:n  
lopetus

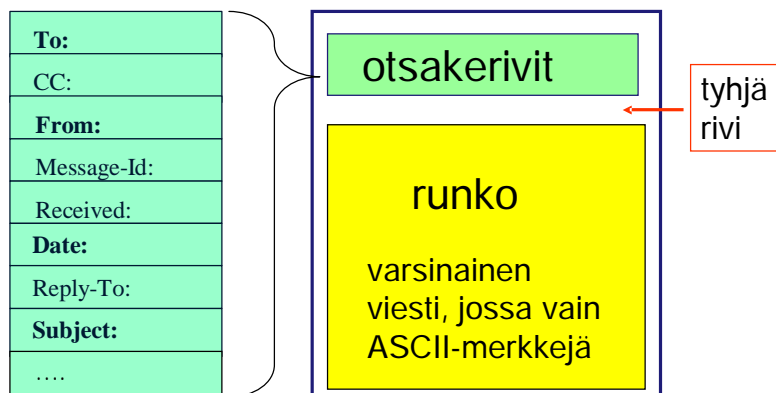
Tietoliikenteen perusteet /2007/ Liisa Marttinen

47

## Sähköpostiviestin rakenne

Eri asia kuin SMTP: eri standardit

Esim.



Tietoliikenteen perusteet /2007/ Liisa Marttinen

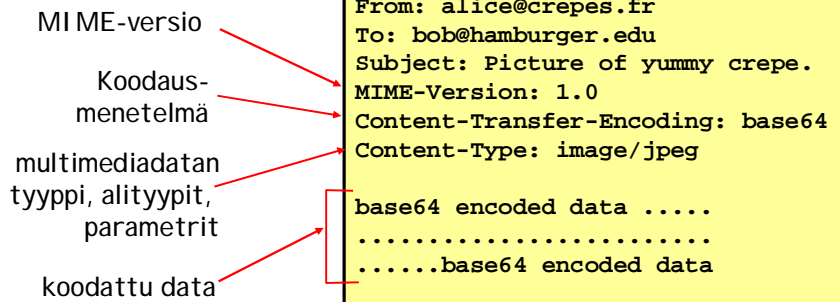
48





## 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

### • MIME-sisältötyyppejä

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
Hei Allu,
sinulle kaunis kuva kissastani Villestä.
-- StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....
.....base64 encoded data
-- StartOfNextPart
Haluatko muita kuvia!
```

Tietoliikenteen perusteet/2007/ Liisa Marttinen

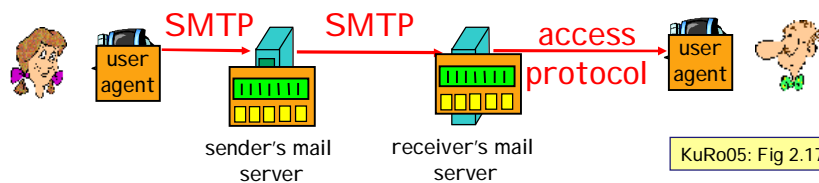


## Postinnoutoprotokollat (mail access protocols)

### Posti omalta postipalvelimelta postiohjelmaan

- POP3: Post Office Protocol versio 3  
Viestien 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

Koska SMTP on 'PUSH'-protokolla, sitä ei voi käyttää sanomia haettaessa ('PULL').



Tietoliikenteen perusteet/2007/ Liisa Marttinen

52



# Internetin nimipalvelu DNS



## DNS (Domain Name System)

### n Sovellusprotokolla

Isännät ja nimipalvelimet käyttävät  
Käyttää itse UDP-kuljetuspalvelua

### n 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

### n Sallii aliasnimet, palvelijan replikoinnin

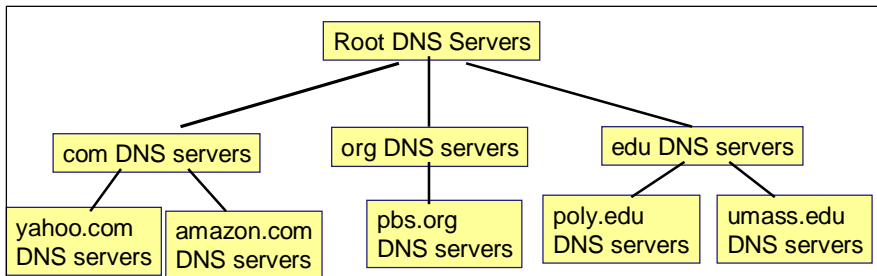
Esim. WWW.cs.helsinki.fi ja cs.helsinki.fi ovat aliasnimiä  
Esim. www-palvelijaan voi liittyä useita IP-osoitteita, rotaatio

### n Hajautettu, hierarkinen tietokanta (hakemisto)

Toteutettu useiden replikoitujen nimipalvelimien yhteistyönä  
skaalautuvuus, kuormantasaus, ylläpito, vikasietoisuus, ..  
Jos oma nimipalvelija ei tunne, se kysyy muilta.



## 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)

Isolla yliopistoilla ja firmoilla omansa, pienet käyttävät jonkun muun ylläpitämää

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

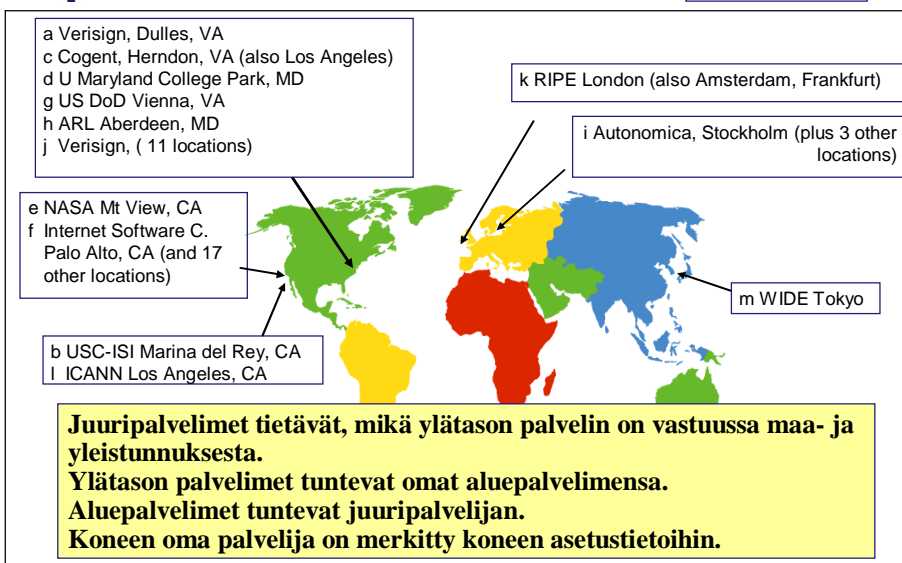
Tietoliikenteen perusteet /2007/ Liisa Marttinen

55



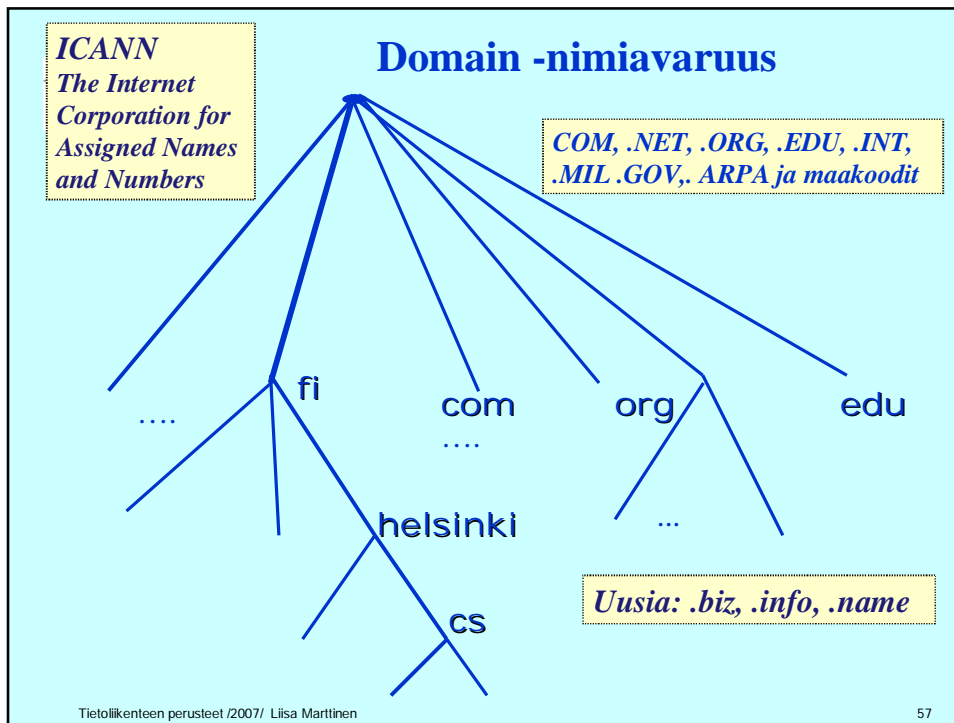
## Juuripalvelimet


KuRo05:Fig 2.21



Tietoliikenteen perusteet /2007/ Liisa Marttinen

56



 **IP-nimen selvittäminen**

- n sovellusohjelma kutsuu kirjastorutiinia parametrina nimi merkkijonona
  - 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 Välimuistin käyttö

Tietoliikenteen perusteet /2007/ Liisa Marttinen 58

## Iteratiivinen kysely: "kerro keneltä pitää kysyä"

**gaia.cs.umass.edu**

Isäntäkone (1)

Kysy omalta aluepalvelijalta

Aluepalvelija (**poly**) (2)

Ota yhteys juuritasolle

Juuripalvelin (3)

Kerro, mistä löytyy ylätason palvelin **edu**-tunnuksille

Ylätason palvelin (**edu**) (4, 5)

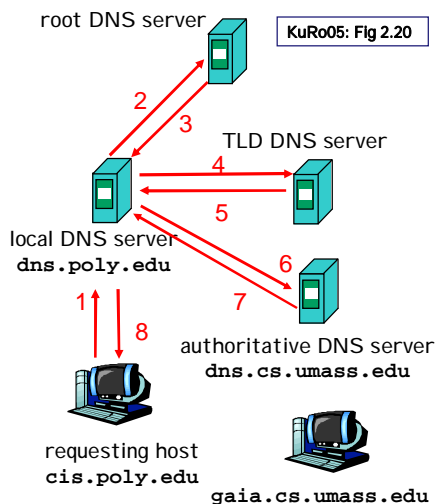
Kerro, mistä löytyy aluepalvelija

**umass.edu**-tunnuksille

Aluepalvelija (6,7)

Tuntee **cs**-verkon koneet.

Kerro koneen IP-osoite



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

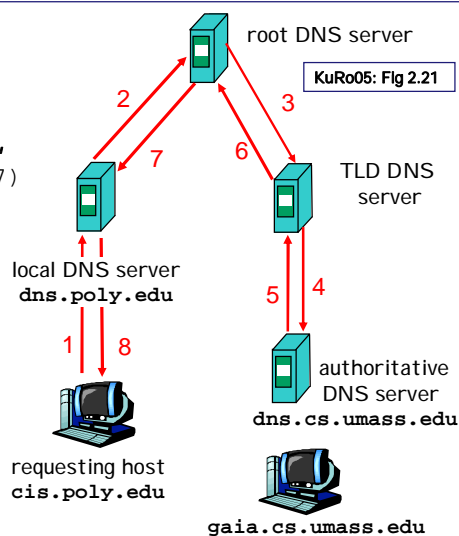
Kysy ensin omalta aluepalvelimelta. (1, 8)

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.





## DNS-välimuisti

Suorituskyvyn parantamiseksi nimipalvelijat varastoivat välimuistiinsa näkemiään DNS-resurssitietueita.

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

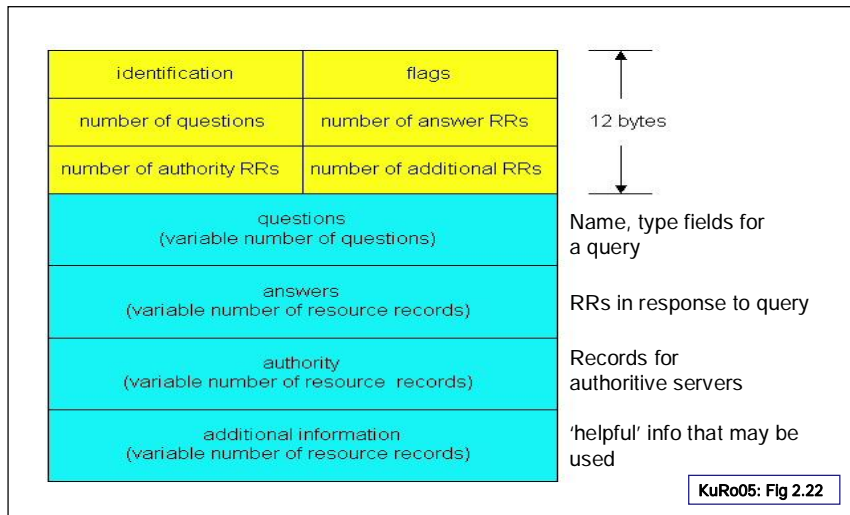


## DNS-nimiavaruuden vyöhykejako

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



## DNS-sanoman rakenne



## DNS-sanoma

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





## DNS- resurssitietue

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



## Verkkosovelluksia, sovellusprotokollia

Vertaistoimijat  
peer-to-peer



## Vertaistoimijat: file sharing

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



## Keskitetty hakemisto

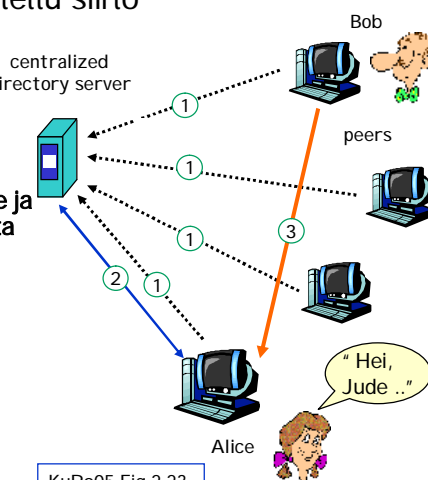
Esim. Napster

Keskitetty hakemisto, hajautettu siirto

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

Ongelmia:  
Suorituskyky  
Vikasietoisuus  
Tekijänoikeudet

centralized  
directory server



KuRo05 Fig 2.23



## Kyselyn tulvitus

Esim. Gnutella

Kysely tulvitettuna, kopiointi suoralla yhteydellä

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

Tulvitus ei saa jatkua loputtomiin!

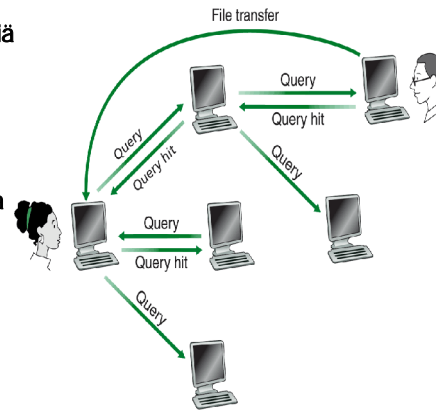


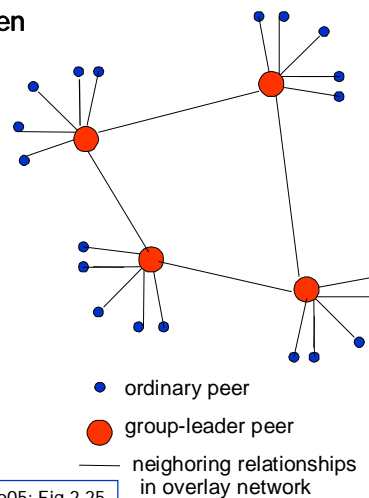
Figure 2.24 Search and file transfer in Gnutella



## Heterogeeniset toimijat

Esim. KaZaA

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



KuRo05: Fig 2.25



# Pistoke Verkkosovelluksen ohjelmointia



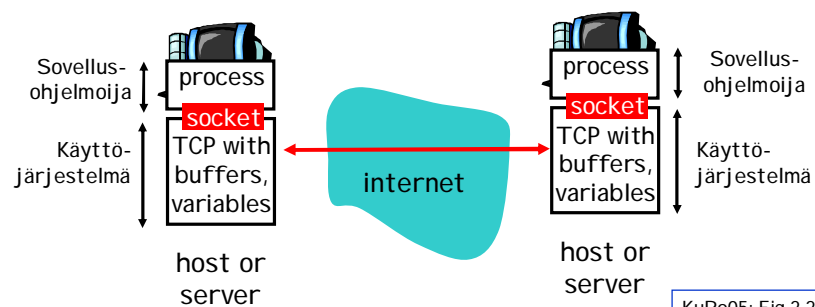
## Pistoke (socket)

n Kuljetuspalvelun ja sitä käyttävän sovelluksen rajapinta isäntäkoneessa

Sovelluksen tietoliikenne = KJ:n palvelupyyntöjä

Pistoke on "palveluluukku"

n Alunperin Berkeley UNIXin (BSD) mukana





## 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)
    yhteyspyynnön odottaminen (welcoming socket) (TCP)
sockfd = accept(sockfd, addr[], *addr_len)
    yhteyspyynnön hyväksyminen, luo uusi pistoke (connection socket)
connect(sockfd, addr[], addr_len)
    yhteyspyynnön lähetys, mihin koneeseen ja porttiin (TCP)
send(sockfd, buf[], buf_len, flags)
recv(sockfd, buf[], buf_len, flags)
    tavuvirran lähetys ja vastaanotto (TCP)
send(sockfd, msg[], msg_len, flags, addr[], addr_len)
recv(sockfd, msg[], msg_len, flags, addr[], *addr_len)
    sanoman lähetys ja vastaanotto, mukana kone ja portti (UDP)
close(sockfd), shutdown(sockfd, how)
    yhteyden lopettaminen (TCP)
```

## TCP-kuljetuspalvelu

Yhteyspyyntö palvelun porttiin

Palvelija luo yhteyttä varten uuden portin  
 Voi palvella useita yhteyspyyntöjä

Tavallisesti palvelija luo yhteyttä varten myös oman prosessin

Lue /kirjoita tavuja

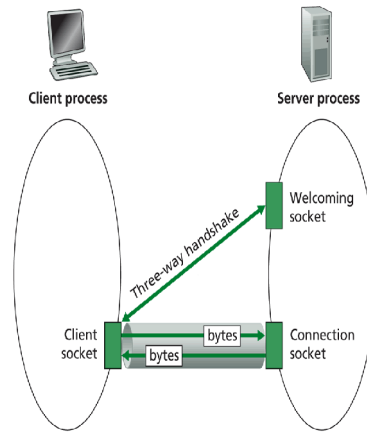
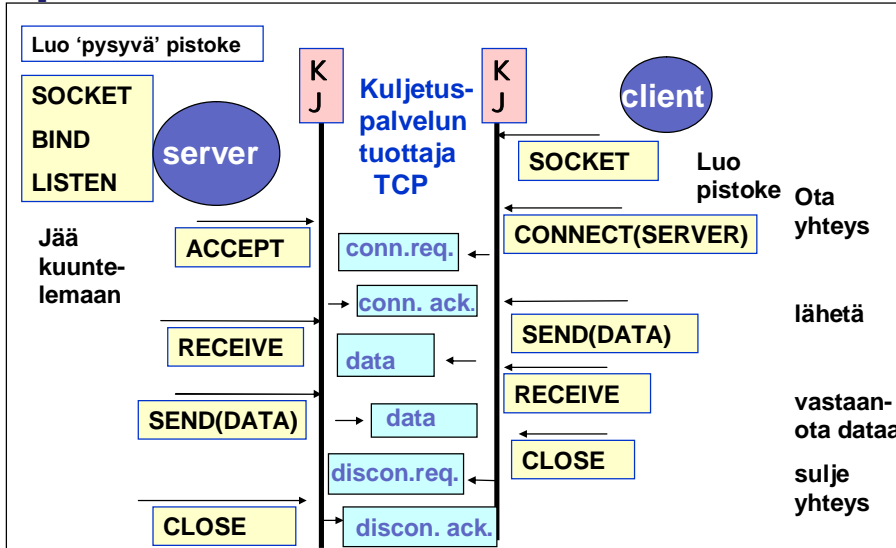
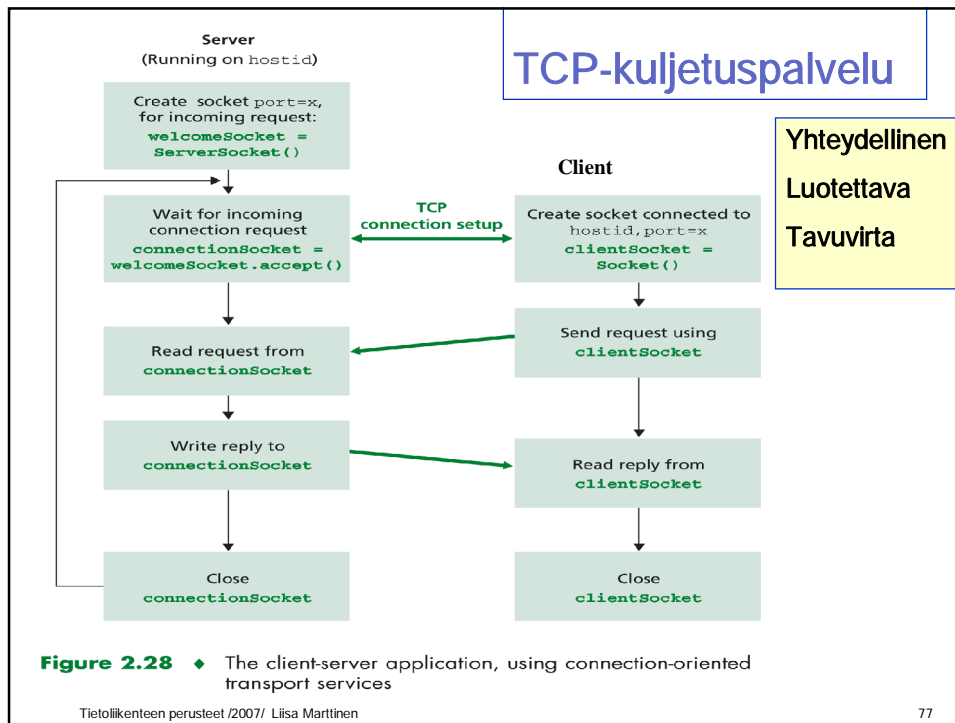


Figure 2.27 ♦ Client socket, welcoming socket, and connection socket

## TCP-kuljetuspalvelu





## 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);
}
}
}
        
```

Tietoliikenteen perusteet /2007/ Liisa Marttinen 78



## Esimerkki: TCP-asiakas (Java)

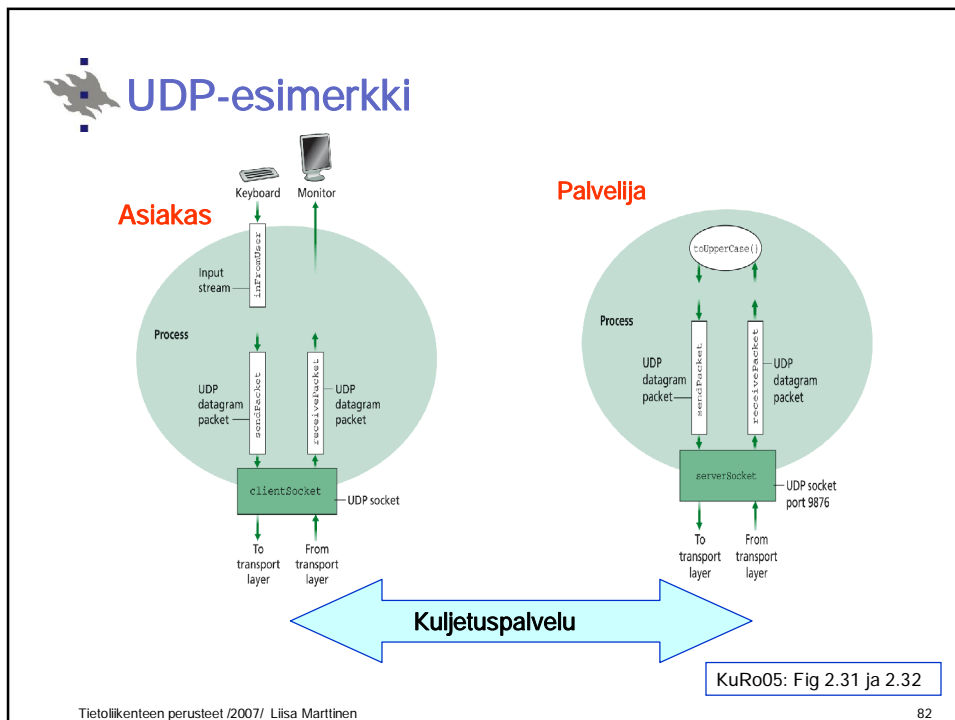
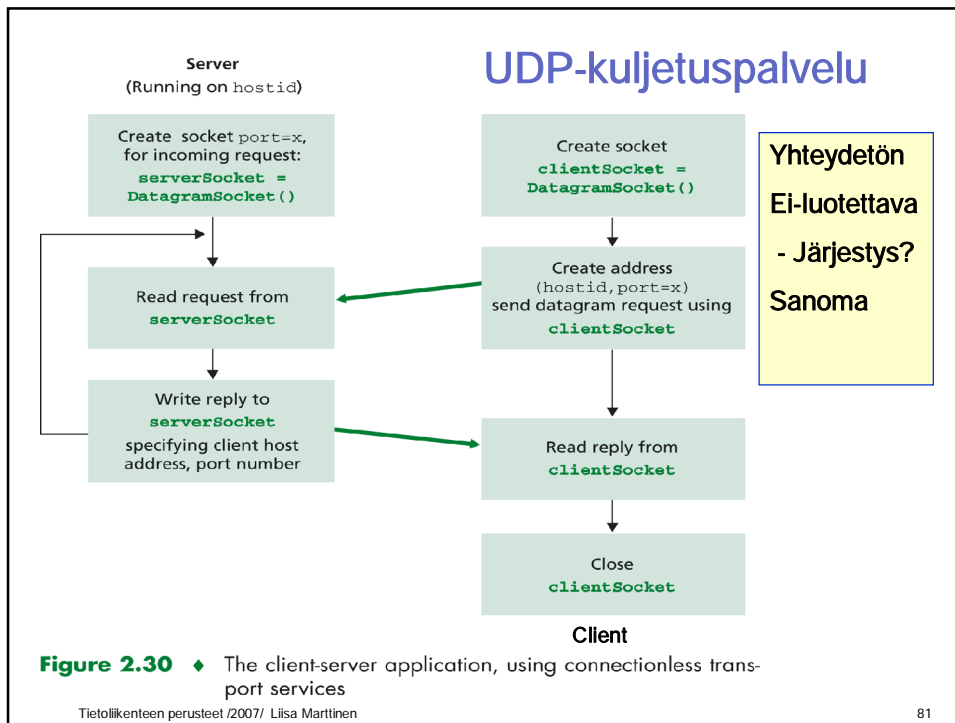
```
import java.io.*; import java.net.*;
class TCPClient {
    public static void main(String argv[]) throws Exception {
        String sentence;
        String modifiedSentence;
        BufferedReader inFromUser =
            new BufferedReader(new InputStreamReader(System.in));
        Socket clientSocket = 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();
    }
}
```



## 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)`







## 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.