

Tietoliikenteen perusteet

SOVELLUSKERROS

(Application layer)

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 Sovelluksen ohjelmat eri isäntäkoneissa**
www-selain ja www-palvelin, postiohjelma ja postipalvelin, ...,
vertaisverkkosovellukset
- n Sovellusprotokolla kuvaa näiden sanomanvälityksen**
DNS, HTTP, SMTP, FTP,
Syntaksi, semantiikka, järjestys
- n Sanomat välitetään käyttäen verkon tarjoamaa
kuljetuspalvelua**
osa järjestelmän perusrakennetta
sovelluksista riippumatonta
- n Reititys tapahtuu vasta verkkotasolla, mutta
sovellutasolla tiedettävä osoite**



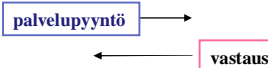
Sovellusarkkitehtuuri



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

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

Google, e-Bay, Facebook, YouTube, Amazon, ..



n Vertaistoimijamalli (esim. BitTorrent, eMule, Skype)

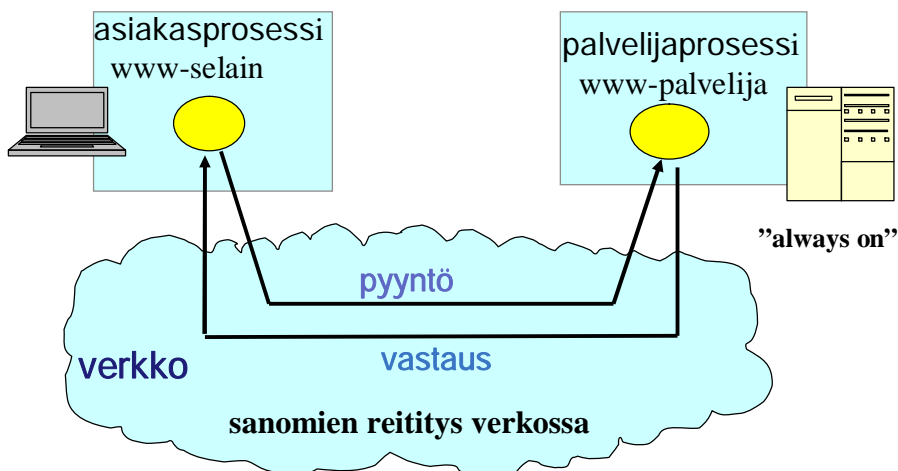
- n Vertaisiännät kommunikoivat suoraan keskenään
- n Ei tarvitse olla aina toiminnassa, IP-osoite voi muuttua
- n Jokainen toimii sekä palvelijana että asiakkaana



n Hybridimalli (esim. Napster, pikaviestimet)



Asiakas-palvelija-malli



Oikea kone, oikea prosessi



Sovelluksen rajapinta tietoliikenteeseen

nPistoke (socket) (verkkosovelluksen ohjelmointirajapinta, API)

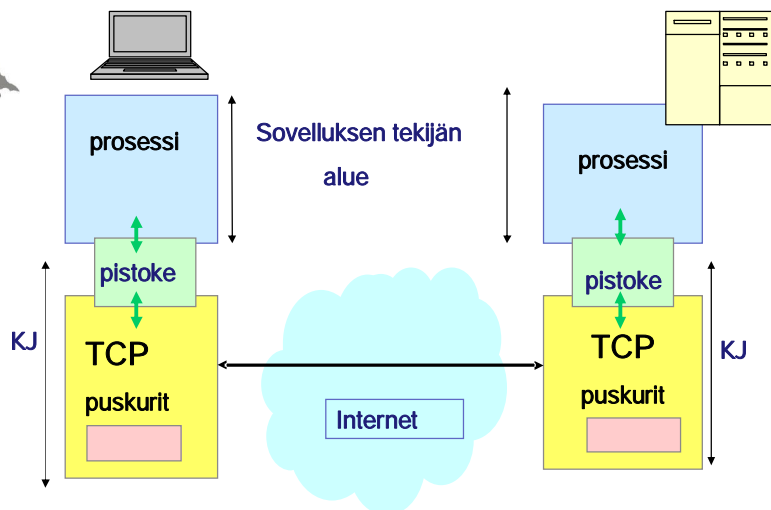
- yhteyden muodostaminen
- lue /kirjoita sanoma
prosessi kirjoittaa verkkoon ja lukee verkosta lähes samalla tavoin kuin kirjoittaa tiedostoon ja lukee tiedostosta
- 'luukku' tai 'ovi', josta dataa sisään /ulos

nLähetys (send): anna sanoma KJ:lle

nVastaanotto (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

n KJ:n kannalta tietoliikenne normaalia siirräntää (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 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 /2009/ Liisa Marttinen

9



Kuljetuspalvelun laatuvaatimuksia

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 /2009/ Liisa Marttinen

10



Kuljetusprotokollat: TCP

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: UDP

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, virheelliset yleensä hylätään

Ei vuonvalvontaa, ei ruuhkanvalvontaa

UDP voi lähettää niin paljon kuin haluaa

Huom! Kummassakaan ei ole takuita siirtonopeudelle eikä

viipeelle => ei mitään aikatakuuta (ns. 'best effort'-palvelu)

Ei myöskään datan salakirjoitusta => SSL (Secure Socket Layer)



Kumpi?

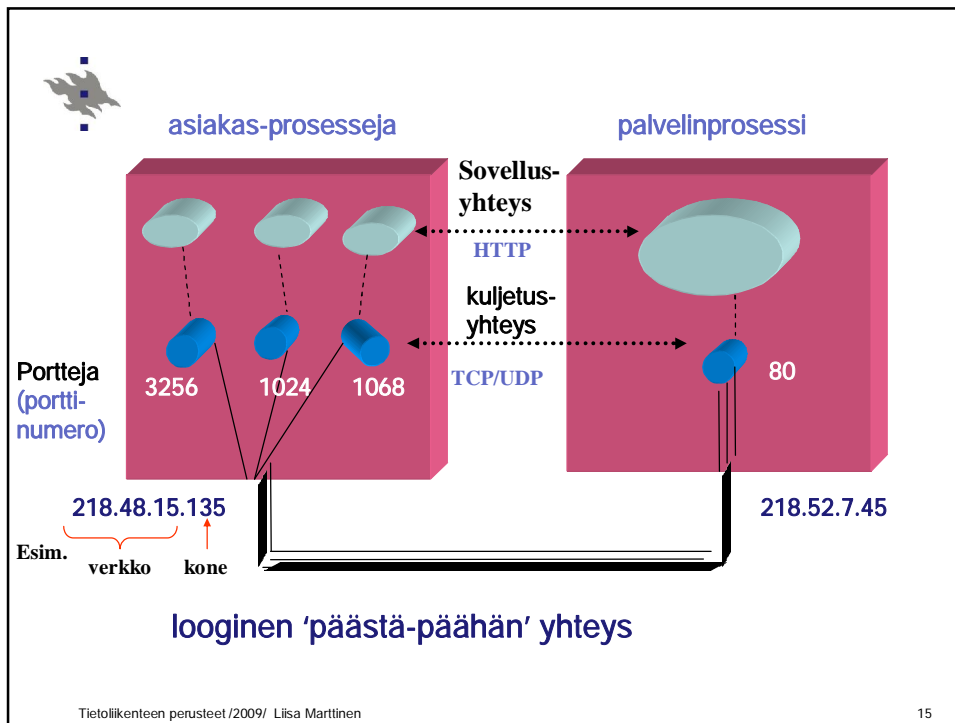
Applications	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 2821]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Remote file server	NFS [McKusik 1996]	UDP or TCP
Streaming multimedia	Often proprietary (e.g.,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



Osoittaminen

- n Sanomissa oltava lähettäjän ja vastaanottajan IP-osoite ja porttinumero
- n **IP-osoite à oikea kone** 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 25KJ osaa liittää porttinumeron prosessiin





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) (kevät 2008)



HTTP (HyperText Transfer Protocol)

(RFC 1945, RFC 2616)

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



Selaimen toiminta

Kun käyttäjä kirjoittaa/klikkaa [url-linkkiä](#) tai siihen on viitattu sivulla:

Muodosta **TCP-yhteys** palvelinkoneeseen

Yhteyspyyntö porttiin **80**, odota hyväksymisvastaus

Laita HTTP-pyyntö TCP-yhteyteen liitettyyn pistokkeeseen

Ota pistokkeesta palvelimen lähettämä HTTP-vastaus

Palvelin sulkee TCP-yhteyden (*nonpersistent connection*)

Tutki sivu

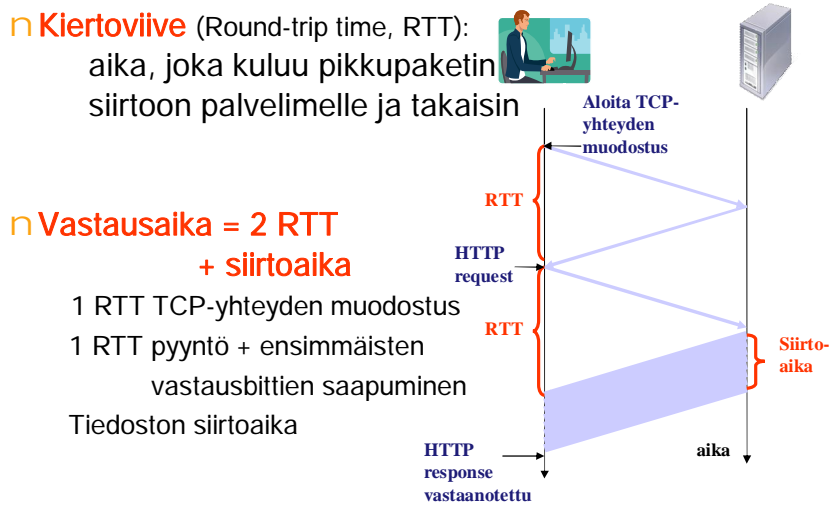
Etsi uudet viitteet ja hae ne samalla tavalla

Näytä sivu käyttäjälle

Lopullinen ulkoasu on kiinni selaimen kyvyistä



Vastausaika (response time)

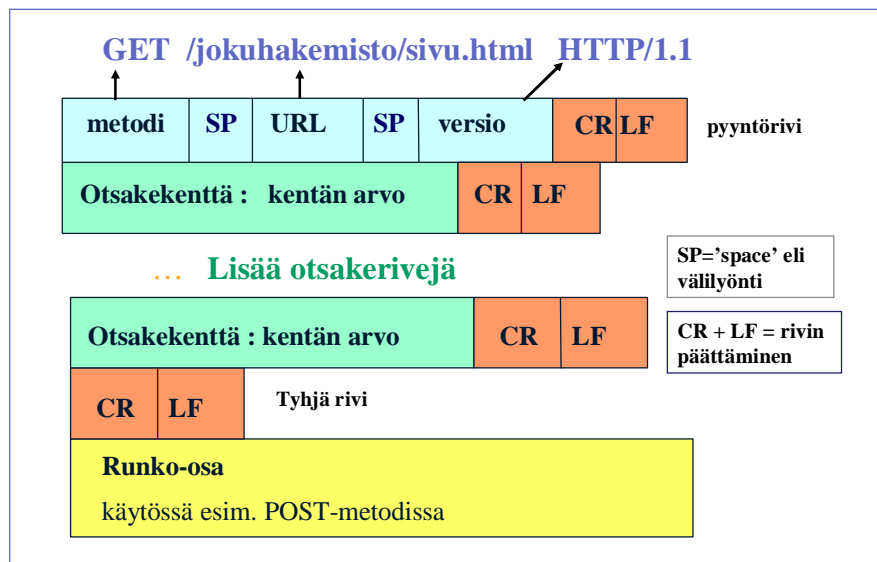


Suorituskyky?

- n Jos sivulla viitataan 10 objektiin**
- 11 peräkkäistä TCP-yhteyden muodostusta ja purkua?
KJ varaa ja vapauttaa puskuritilaa;
muodostuksiin kuluu kaikkiaan 22 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.



HTTP-pyyntö: yleinen rakenne

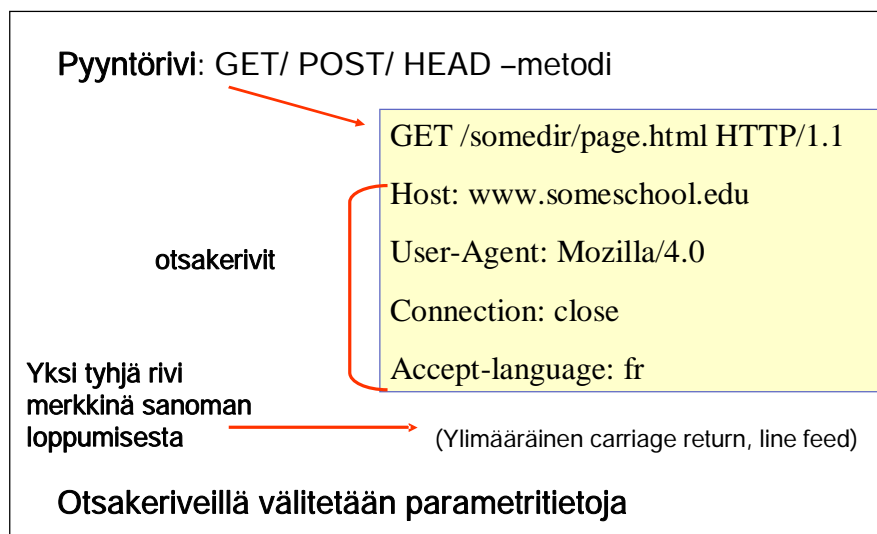


Tietoliikenteen perusteet /2009/ Liisa Marttinen

23



Esimerkki: HTTP-pyyntö



Tietoliikenteen perusteet /2009/ Liisa Marttinen

24



HTTP-pyyntömetodeja

(HTTP/1.1: <http://www.w3.org/protocols/rfc2616/rfc2616.html>)

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	} Web-julkaisu - työkalut käyttävät
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 pyyntörivillä, talletettava runko-osassa	
DELETE	Poista objekti palvelimelta	



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

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

Connection: **close** sulje yhteys lähetyksen jälkeen

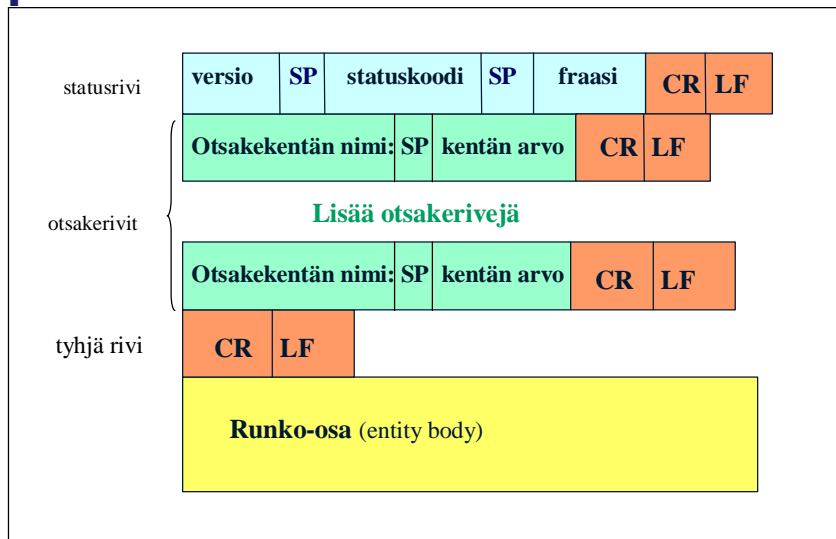
User-agent: **Mozilla/4.0** selaimen tyyppi

Accept-language: **fi** dokumentin kieli

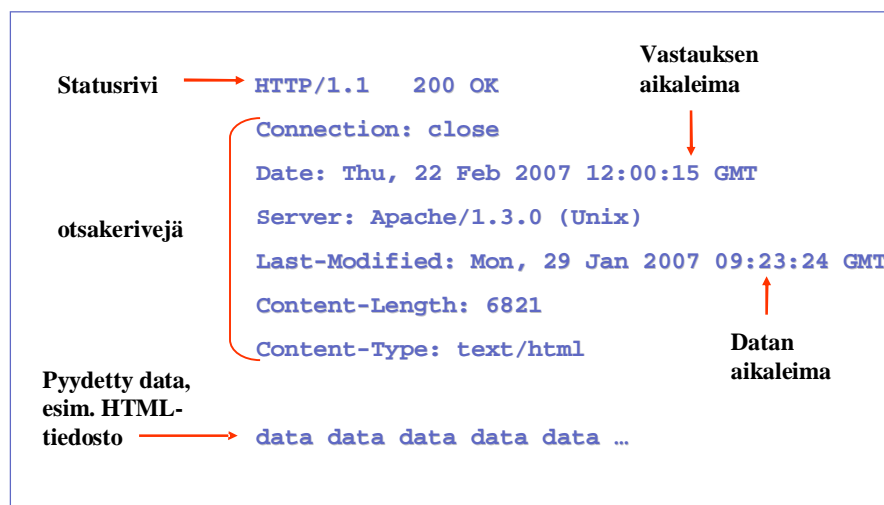
http://en.wikipedia.org/wiki/List_of_HTTP_headers



HTTP-vastaus: yleinen rakenne



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:sta

302 Moved Temporarily Siirretty tilapäisesti

400 Bad Request Palvelija ei ymmärtänyt pyyntöä

403 Forbidden Ei ole oikeutta lukea pyydettyä tiedostoa

404 Not Found Pyydettyä objektia ei löydetty

500 Internal Server Error Virhe palvelimessa

505 HTTP Version Not Supported Palvelija ei tue asiakkaan käyttämää HTTP-versiota. Syntaksissa on jotain liian uutta tai liian vanhaa.



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 (= evästeen) Lähettää 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-pyyntöissä 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?

<http://www.cookiecentral.com>



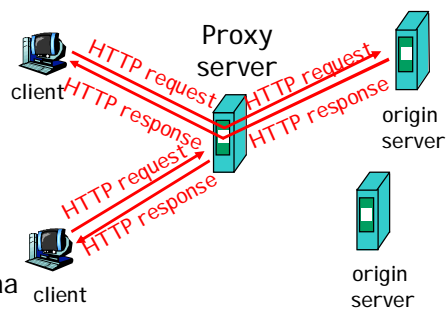
Proxy-palvelin eli verkkovälimuisti

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



[Myös asiakaskone voi ylläpitää välimuistia!]

KuRo08: Fig 2.11



Proxy-palvelimen käytöstä

KuRo08: Fig 2.12

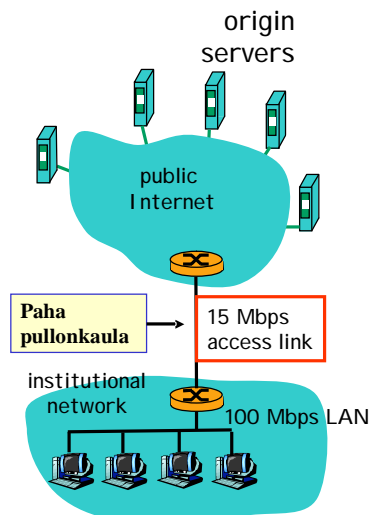
n Oletetaan

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

n Tällöin

paikallisverkon käyttöaste = 15%
ei ruuhkautunut => siirtoaika muutamia kymmeniä ms

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



Proxy-palvelimen käytöstä

n Parannus?

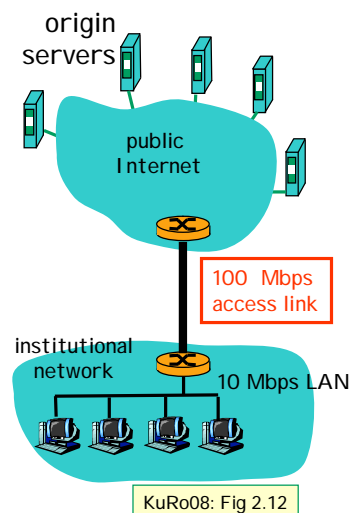
Hankitaan nopeampi yhteys, esim. 100 Mbps

n Tällöin

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

n Mitähän nopeampi linkki maksaa?

Voi olla kallis ratkaisu!



KuRo08: Fig 2.12



Proxy-palvelimen käytöstä

KuRo08: Fig 2.13

n Parannus?

Asennetaan proxy-palvelin

n Oletetaan,

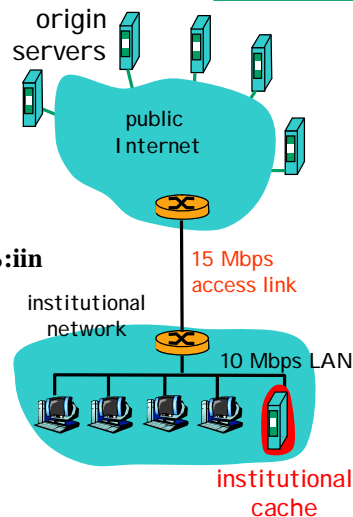
osumatodennäköisyys (hit rate)= 0,4.
(tyypillisesti välillä 0,2-0,7)

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 /2009/ 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

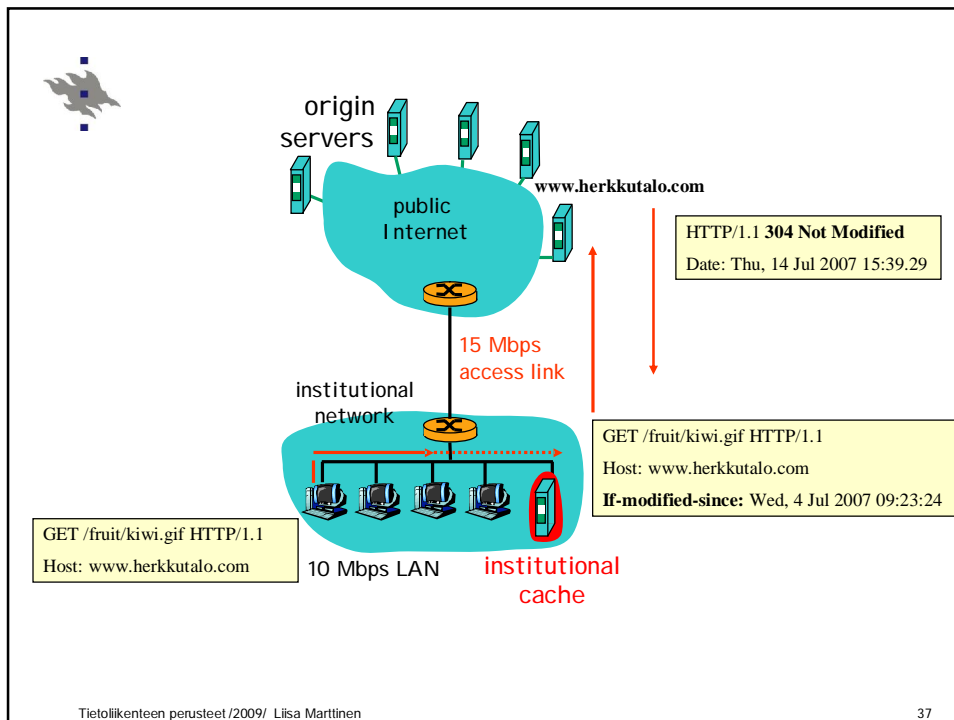
HTTP/1.0 304 Not Modified'

Eikä objektia mukana

n Muuten objekti mukana normaalisti

Tietoliikenteen perusteet /2009/ 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

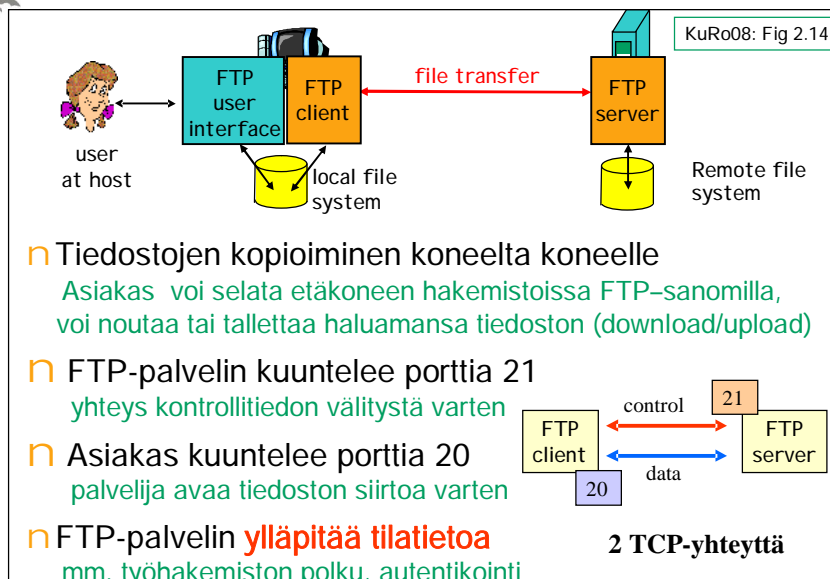
Tietoliikenteen perusteet /2009/ Liisa Marttinen 38



Tiedostonsiirto FTP



FTP: File Transfer Protocol (RFC 959)





FTP-pyyntöjä ja -vastauksia

n Kaikki sanomat 7 bitin ASCII-muodossa

n Asiakkaan pyyntöjä

USER **username**

PASS **password**

LIST

RETR **filename**

STOR **filename**

n palvelimen 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

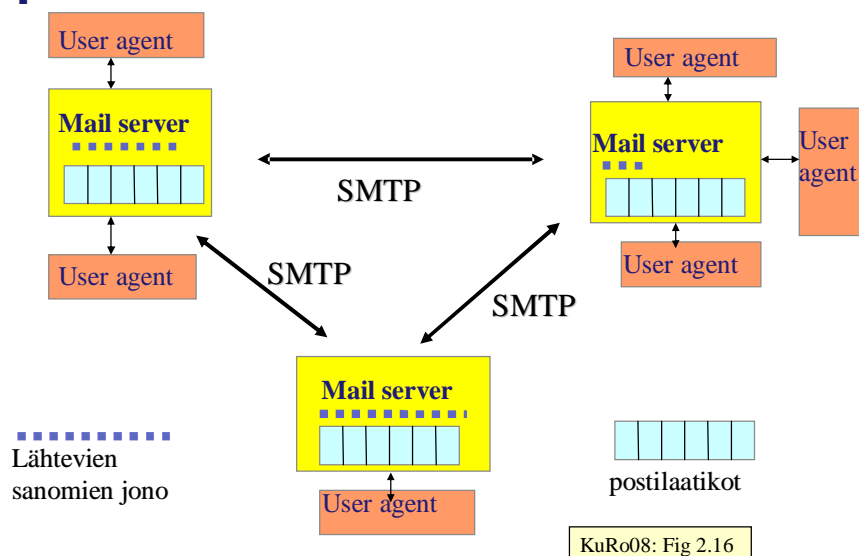


Sähköpostin komponentit

- 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



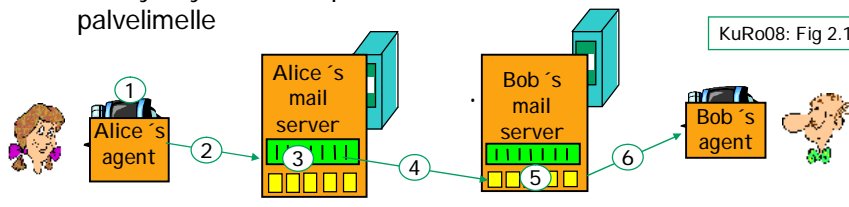
Sähköpostin komponentit





Esimerkki: Alice Bobille

1. Alice kirjoittaa viestin posti-ohjelmalla: "to:"
`bob@someschool.edu`
2. Alicen postiohjelma lähettää viestin omalle postipalvelimelle
3. Alicen postipalvelin avaa TCP-yhteyden Bobin postipalvelimelle
4. Alicen postipalvelin siirtää viestin SMTP-protokollalla Bobin postipalvelimelle käyttäen TCP-yhteyttä
5. Bobin postipalvelin laittaa viestin Bobin postilaatikkoon
6. Bob lukee viestin omalla postiohjelmalla



Tietoliikenteen perusteet /2009/ Liisa Marttinen

45



SMTP (Simple Mail Transfer Protocol) (RFC 821)

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



Tietoliikenteen perusteet /2009/ 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

Sähköpostiviestin rakenne

Eri asia kuin SMTP: eri standardit (RFC 822)

Esim.





SMTP:n rajoitteita

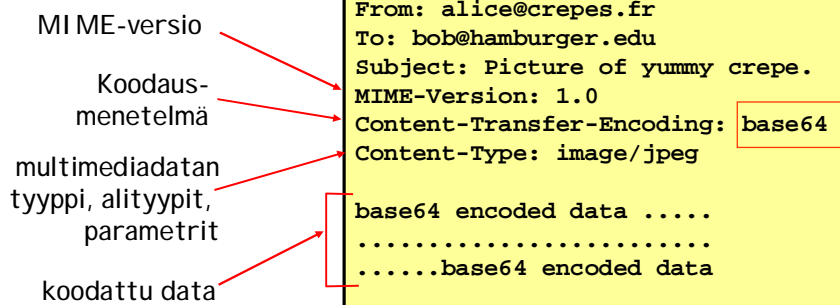
- n Kaikki esitettävä 7-bittisenä ASCII:na
 - = IRA, International Reference Alphabet
 - Myös binääridata, esim. kuvat ja ääni
- n Yksittäinen viesti loppuu omalla rivillä olevaan pisteeseen
 - eli lopussa ASCII-merkit: **CRLF.CRLF**
 - Vanha protokolla!
- n Binääridata on koodattava s.e. siinä ei esiinny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

CR = carriage return
LF = line feed



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

- n Kaikki on koodattava 7-bittiseksi ASCII-koodiksi
- n 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!

.

Nykyisin yleensä
 linkki www-
 sivulle, josta
 kuvan voi hakea!

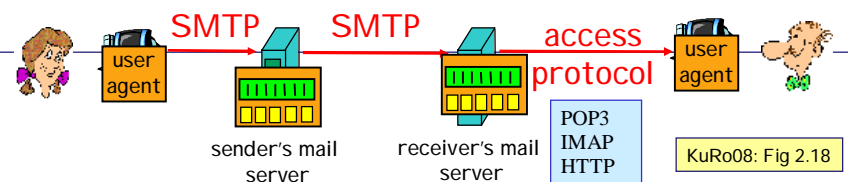


Postinnoutoprotokollat (mail access protocols)

Posti omalta postipalvelimelta postiohjelmaan

- n **POP3:** Post Office Protocol versio 3
Viestien lataamiseen omalle koneelle, ei postikansioita
- n **IMAP:** Internet Mail Access Protocol
Monipuolisempi: postikansiot (folders), lataa vain otsikot, viestien säilytys postipalvelimelle
- n **HTTP:** Esim. TKTL:lla käytettävä IlohaMail, Hotmail, ...
Web-palvelija käyttää IMAP-palvelijaa

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



ESMTP (Extended Simple Mail Transfer Protocol) RFC 2821 (uusin versio RFC 5321 (lokakuu 2008))

n Runsaasti laajennoksia jo 1995 (RFC 1868)

- * 8BITMIME — 8 bit data transmission, RFC 1652
- * ATRN — Authenticated Turn, RFC 2645
- * SMTP-AUTH — Authenticated SMTP, RFC 2554
- * CHUNKING — Chunking, RFC 3030
- * DSN — Delivery status notification, RFC 1891
- * ETRN — Extended Turn, RFC 1985
- * HELP — Supply helpful information, RFC 821
- * PIPELINING — Command pipelining, RFC 2920
- * SIZE — Message size declaration, RFC 1870
- * STARTTLS — Transport layer security, RFC 3207

n EHLO aloittaa



Verkkosovelluksia, sovellusprotokollia

Internetin nimipalvelu DNS

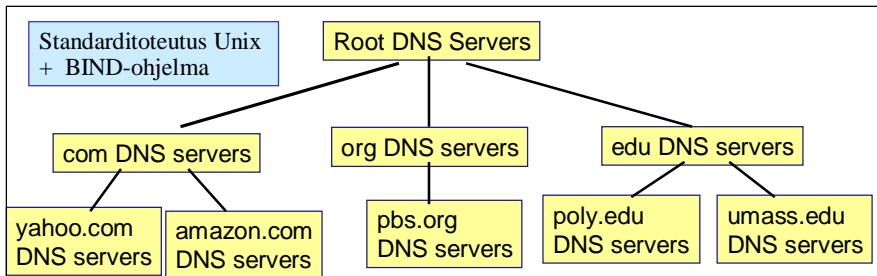


DNS (Domain Name System)

- n Hakemistopalvelu ja sovelluskerroksen protokolla
Isännät ja nimipalvelimet käyttävät
Käyttää itse UDP-kuljetuspalvelua DNS-sanomien kuljettamiseen
- 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, hierarkkinen tietokanta (hakemisto)
Toteutettu useiden replikointien nimipalvelimien yhteistyönä
skaalautuvuus, kuormantasaus, ylläpito, vikasietoisuus, ..
Jos oma nimipalvelija ei tunne, se kysyy muilta.



Hajautettu, hierarkinen tietokanta



KuRo08: Fig 2.19

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

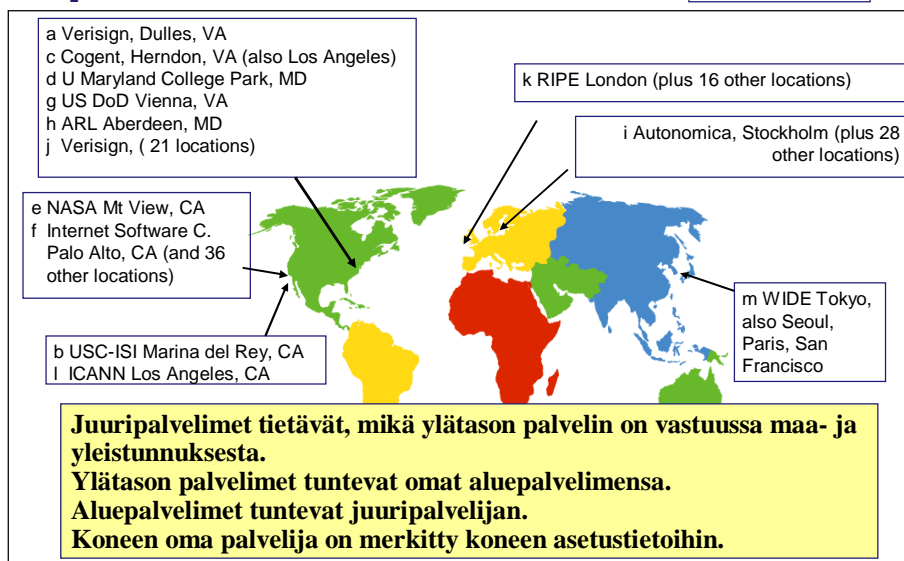
Tietoliikenteen perusteet /2009/ Liisa Marttinen

57



Juuripalvelimet (2007)

KuRo08:Fig 2.20



Juuripalvelimet tietävät, mikä ylätason palvelin on vastuussa maa- ja yleistunnuksesta.

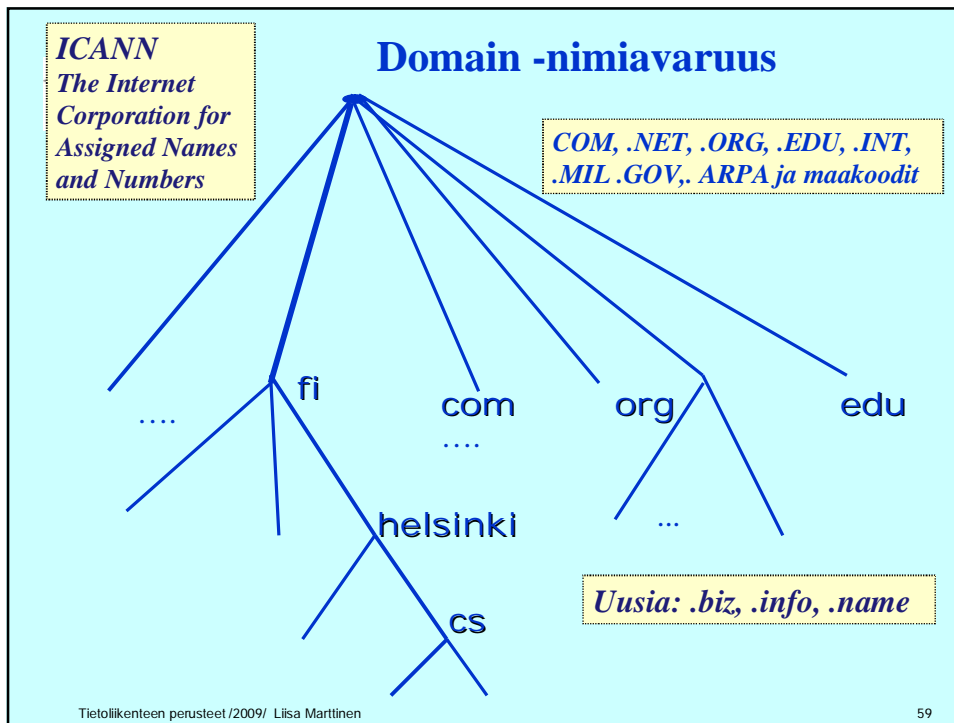

Ylätason palvelimet tuntevat omat aluepalvelimensä.

Aluepalvelimet tuntevat juuripalvelijan.

Koneen oma palvelija on merkitty koneen asetustietoihin.

Tietoliikenteen perusteet /2009/ Liisa Marttinen

58

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

Tietoliikenteen perusteet /2009/ Liisa Marttinen 60



IP-nimen selvittäminen

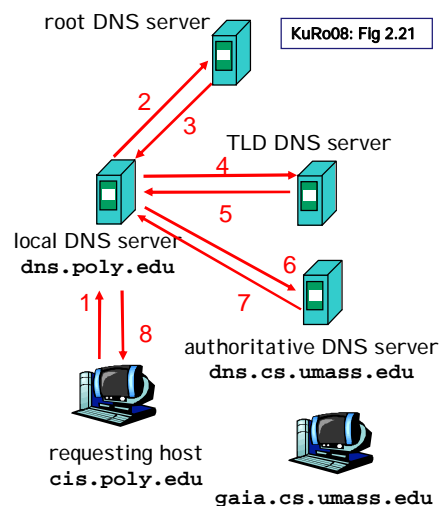
- sovellusohjelma kutsuu kirjastorutiinia parametrina nimi merkkijonona
 - esim Unix:ssa `gethostbyname()`
- kirjastorutiini lähettää UDP-datasähkeen paikalliselle DNS-palvelimelle, joka etsii nimeä vastaavan IP-osoitteen ja palauttaa sen kirjastorutiinille
 - etsinnässä tarvitaan usein monien palvelimien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



Iteratiivinen kysely: "kerro keneltä pitää kysyä"

`gaia.cs.umass.edu`

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





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

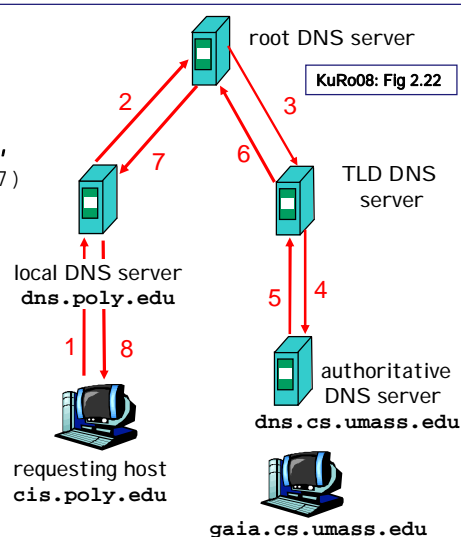
Kysy ensin omalta aluepalvelimelta. (1, 8)

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.

■ Ei tarvitse aina hakea uudestaan

- Kuormittaa vähemmän ylemmän tason nimipalvelimia
- Nopeuttaa tavallisimpia kyselyjä: löytyy läheltä

■ Tiedon oikeellisuus

- Tietueelle määrätty elinaika (TTL, time to live) kertoo voimassaoloajan (yleensä muutama päivä)
- Kun umpeutuu, tieto poistetaan.
- Yleensä muutokset paikallisia: koneen lisäys, koneen poisto, joskus uusi verkko

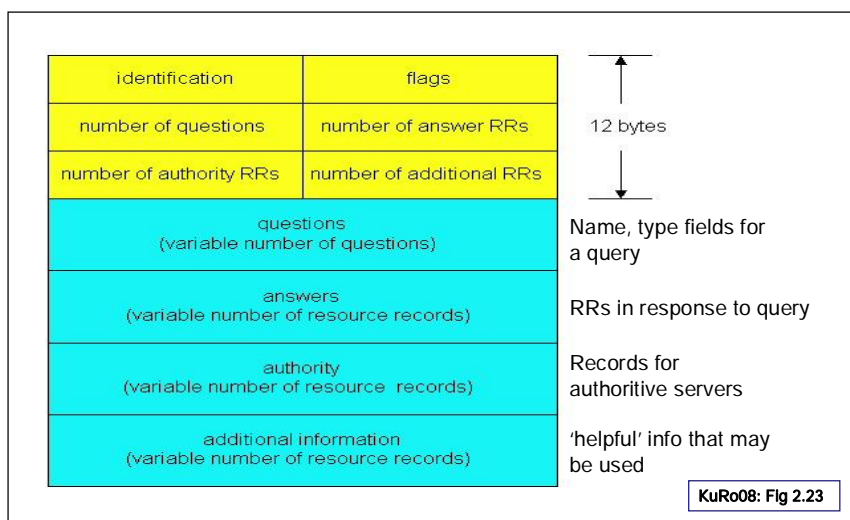


DNS- resurssitietue

- n Kentät: (nimi, arvo, tyyppi, elinaika)
- 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 = NS** (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)



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



Verkkosovelluksia, sovellusprotokollia

Vertaistoimijat
peer-to-peer



Vertaistojijat: file sharing

Isäntäkoneet asiakkaan ja palvelijan roolissa

Jaetaan uusi versio käyttöjärjestelmästä, korjaustiedosto ohjelmaan, MP3-tiedostoja, videoleikkeitä, ...

Jokainen vertainen voi toimija jakelijana

BitTorrent-liikenne
jo 30% Internetin
koko liikenteestä?

Skaalautuvuus, kuormantasaus

Kone on satunnaisesti Internetissä

IP-osoitekin voi vaihdella kerrasta toiseen

Miten löytää vertaistojija(t)?

Keskitetty hakemisto: kiinteä IP-osoite, josta voi kysellä

Kyselyn tulvus: kysellään potentiaalisilta toimijoilta

Hiukan keskitetty hakemistopalvelu, joka tekee jatkokyselyt

Kun kohde löytynyt, kopiointi suoraan sieltä

Kyselyn tuloksena IP-osoite

Nouto HTTP-protokollaa käyttäen

Tietoliikenteen perusteet /2009/ Liisa Marttinen

69



Skaalautuvuus

KuRo08: Fig. 2.24

Asiakas-palvelinmalli:

Palvelimen siirrettävä
 $n * F$ bittia => siirtoaika =
 nF/u_s

Hitain asiakas d_{min} saa
tiedoston ajassa F/d_{min}

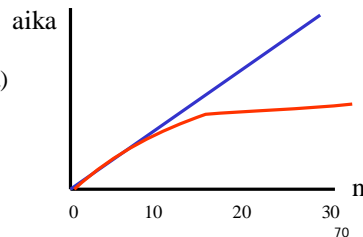
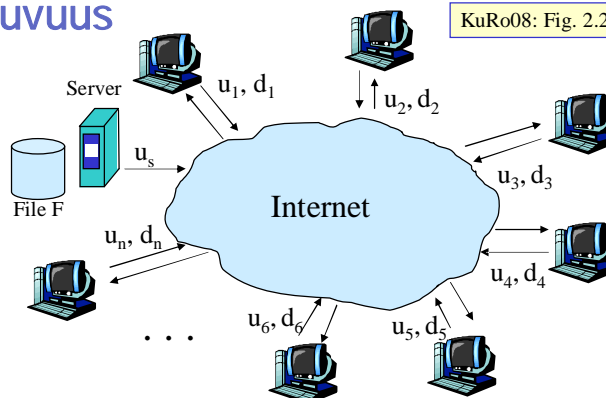
Siirtoaika =
 $\max(nF/u_s, F/d_{min})$

Kun n kasvaa, palvelimen
kuorma kasvaa ja siirtoaika
kasvaa.

Vertaistojijamalli (alussa tiedosto on palvelimella)

Siirtoaika = $\max(F/u_s, F/d_{min}, nF/(u_s + \sum u_i))$

Summamerkki



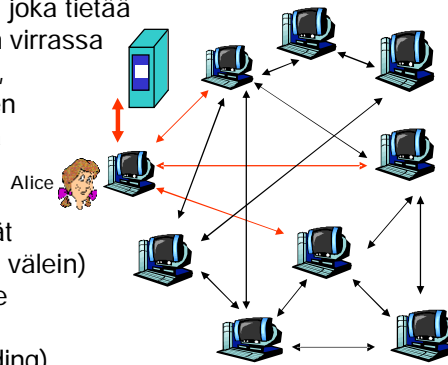
Tietoliikenteen perusteet /2009/ Liisa Marttinen

70



BitTorrent

- n Ladataan ja samaan aikaan jaellaan yhdenkokoisia lohkoja (esim. 256 KB) bittivirtaa (torrent)
- n Rekisteröidytään "tracker"-solmuun, joka tietää mitkä koneet ovat mukana missäkin virrassa
- n Trackerilta saa muiden IP-osoitteita, joihin voi yrittää ottaa TCP-yhteyden
- n Naapureilta kysellään lohkolistoja ja pyydetään lähettämään lohkoja (harvinaisimmat ensin)
- n Itse lähetetään 4:lle, jotka lähettävät suurimmalla nopeudella (arvio 10 s välein) ja 30 s välein lähettää satunnaiselle naapurille kokeeksi
- n Vapaa matkustus -ongelma (free-riding)
- n BitTorrentissa paljon muita piirteitä!



Keskitetty hakemisto

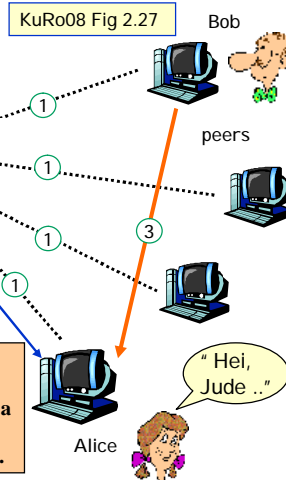
Keskitetty hakemisto, hajautettu siirto

1. Aina kun liitytään 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

Pikaviestintäjäjärjestelmä pitää kirjaa, millä koneella käyttäjät ovat aktiivisina ja ilmoittaa tiedon siitä kiinnostuneille (buddy list).

Esim. Napster





Kyselyn tulvitus

Esim. Gnutella

Kysely tulvitettuna, kopiointi suoralla yhteydellä

- n Käyttää pystyssäolevia TCP-yhteyksiä kyselyn hajauttamiseen (**overlay network**)
- n Välittää kyselyn edelleen, jollei itse pysty sitä täyttämään
- n Jos pystyy, niin lähettää tästä tiedon samaa reittiä, jota kyselykin tuli
- n Kopiointi suoralla yhteydellä

Tulvitus ei saa jatkua loputtomiin!
=> Kyselyssä hyppylaskuri

Miten uusi tulija löytää kumppanit?
• lista, keskuskone => TCP-yhteys näihin
• ping-sanoma ja pong-sanoma (IP-osoite)

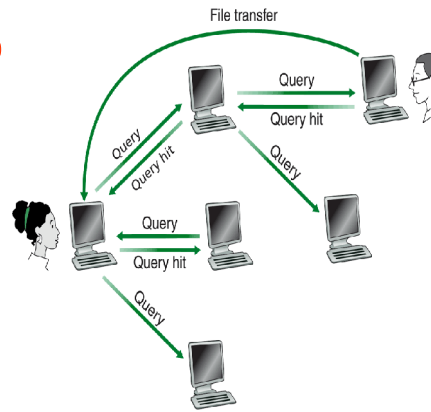


Fig. 2.28 Search and file transfer in Gnutella

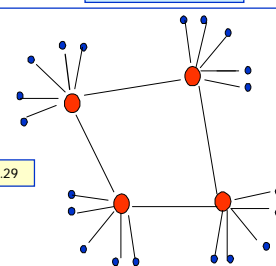


Hierarkkiset verkot: heterogeeniset toimijat

Esim. KaZaA,
Morpheus

- n Joko **superjäsen** tai ryhmän jäsen jäsenet tuntevat oman superjäsenensä super tuntee muita supereita
- n Superjäsen tietää jäseniensä tiedostot kertovat yhteyttä ottaessaan
- n Superjäsen kysyy multa tiedostoja tuntemiltaan toisilta superjäseniltä
- n Tiedoston kopiointi suoralla yhteydellä

KuRo08: Fig 2.29



• ordinary peer
• super peer
— neighboring relationships in overlay network

Skype?

DHT (Distributed Hash Table) hajautettu hajautustaulu

hajautettu hakemisto, joka kuvaa tiedostotunnisteet IP-osoitteeksi siten, että kaikki haetun tiedoston sijaintipaikat voidaan selvittää ilman yletöntä tietoliikennettä.



Pistoke Verkkosovelluksen ohjelmointia

Perusteellisemmin kurssilla:
Verkkosovellusten toteuttaminen



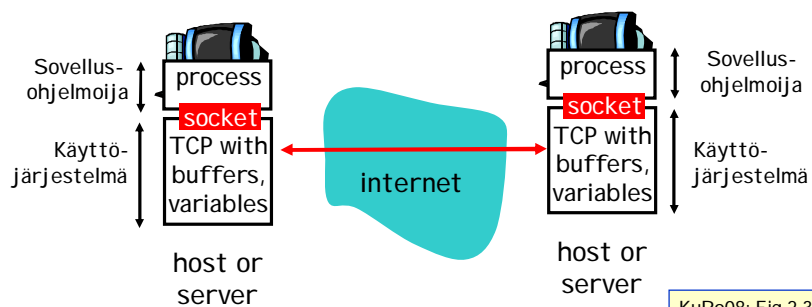
Pistoke (socket)

• Kuljetuspalvelun ja sitä käyttävän sovelluksen rajapinta isäntäkoneessa

Sovelluksen tietoliikenne = KJ:n palvelupyyntöjä

Pistoke on "palveluluukku"

• 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 Palvelimella on pysyvä (standardi)portti ja kutakin asiakasyhteyttä varten luotu tilapäinen portti (yhteysoportti)
 - n Asiakasohjelmalle tilapäinen KJ:n valitsema
- n Kaksisuuntainen (full duplex)
 - n Samaan pistokkeeseen sekä kirjoitetaan ja siitä luetaan
- n Lähetys (send)
 - n Kirjoita pistokkeeseen
- n Vastaanotto (receive)
 - n Lue pistokkeesta



TCP-kuljetuspalvelu

Welcoming socket = vastaanottopistoke?
Connection socket = yhteysoportti?

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

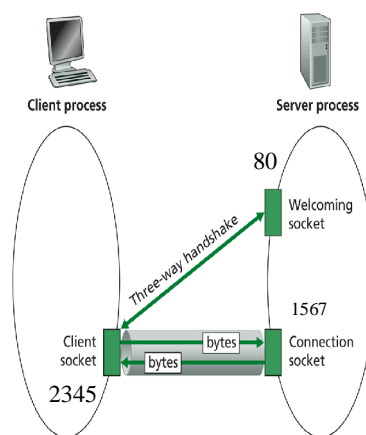
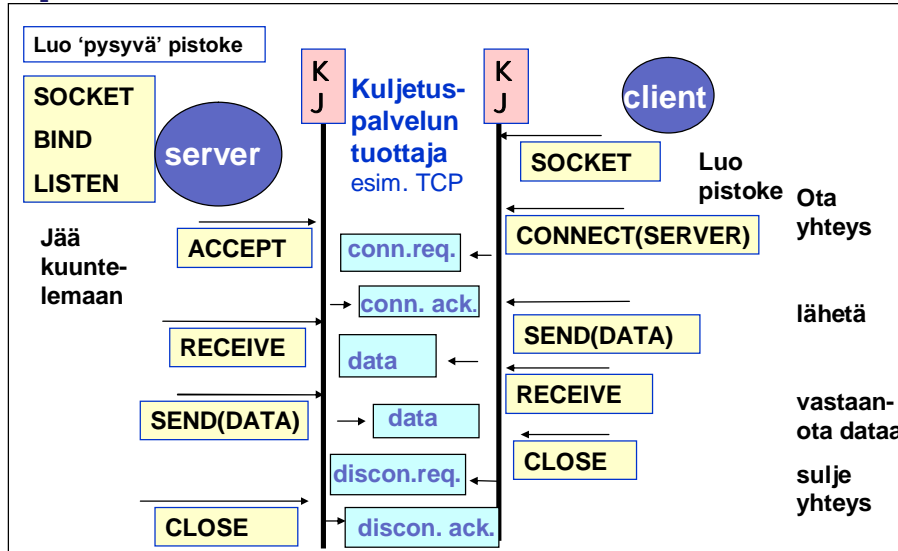


Figure 2.31: Client socket, welcoming socket, and connection socket

TCP-kuljetuspalvelu



Tietoliikenteen perusteet /2009/ Liisa Marttinen

79

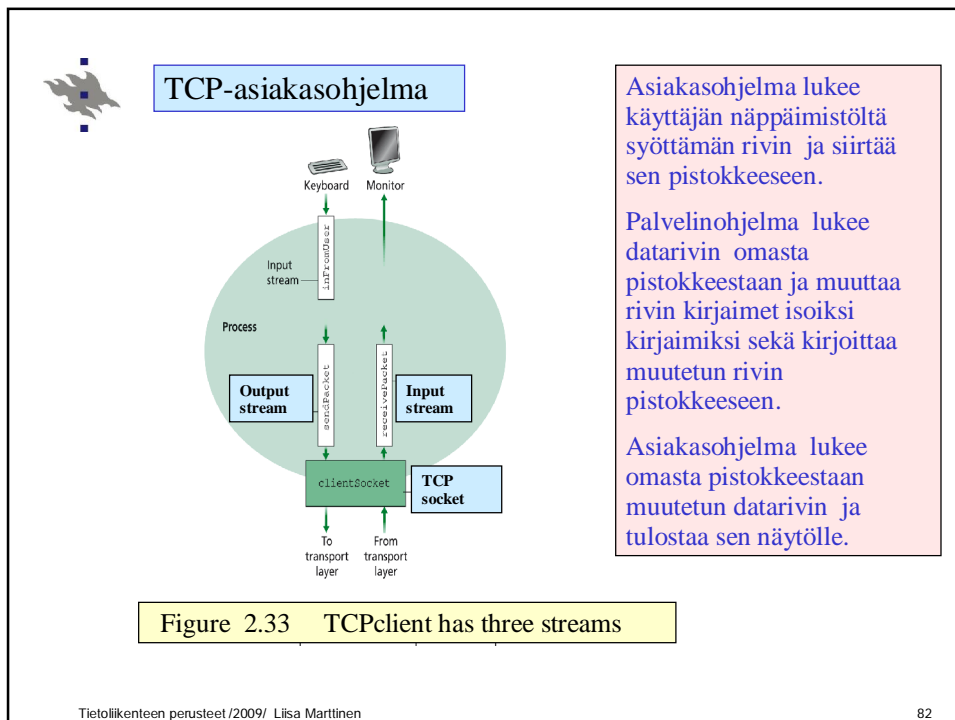
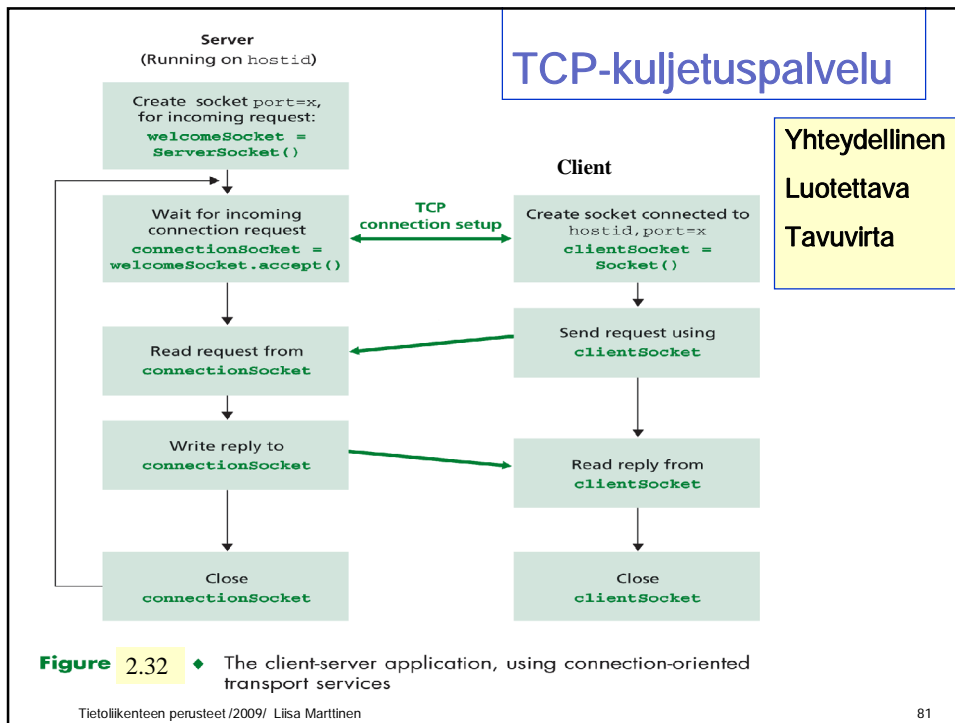
Posix: Pistokerajapinta (API)

```

socfd = 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)
    
```

Tietoliikenteen perusteet /2009/ Liisa Marttinen

80





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); yhteyspyyntö
        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(); Sulkee myös TCP-
        yhteyden
    }
}
```



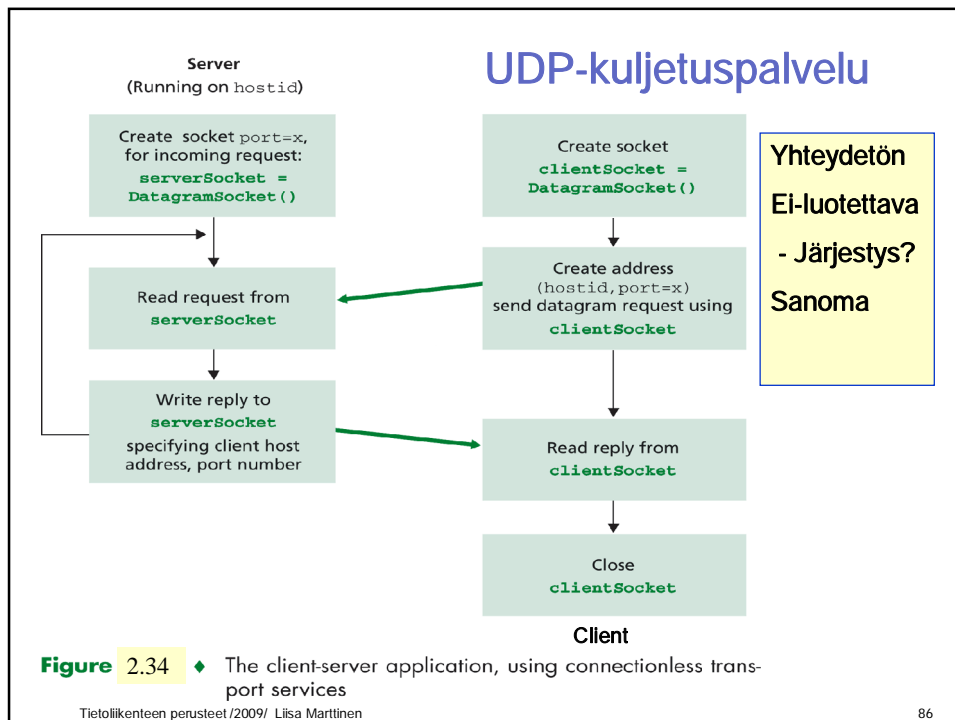
Esimerkki: TCP-palvelija (Java)

```
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(); Yhteyspistokkeen
            luonti
            BufferedReader inFromClient = new BufferedReader(
                new InputStreamReader(connectionSocket.getInputStream()));
            DataOutputStream outToClient =
                new DataOutputStream(connectionSocket.getOutputStream());
            clientSentence = inFromClient.readLine();
            capitalizedSentence = clientSentence.toUpperCase() + '\n'; Muuttaa isoiksi
            kirjaimiksi!
            outToClient.writeBytes(capitalizedSentence);
        }
    }
}
```

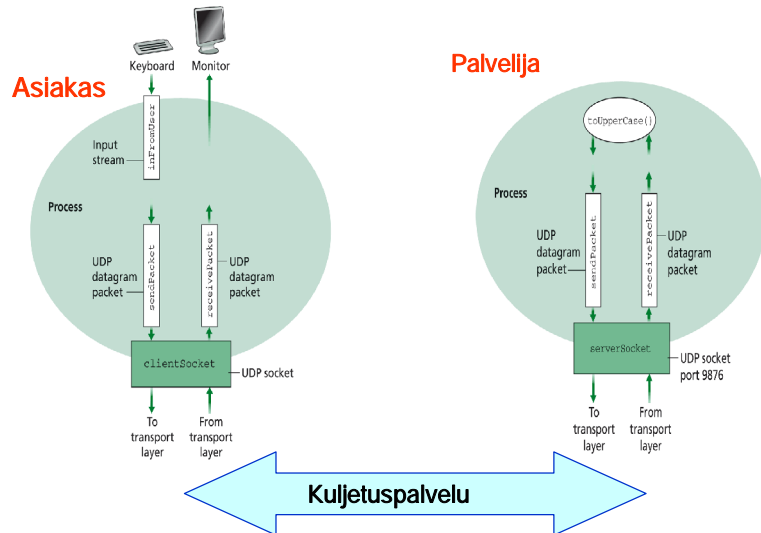


UDP-kuljetuspalvelu

- n Ei kättelyä, yhteydenmuodostusta /purkua
- n Ei-luotettava
- n Sovellusprosessi lukee ja kirjoittaa kokonaisia yksittäisiä 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)`



UDP-esimerkki



KuRo08: Fig 2.35 ja 2.36

Tietoliikenteen perusteet /2009/ Liisa Marttinen

87

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();
    }
}
```

IP-osoitteen
selvittäminen!

Vapauttaa pistokkeen

Tietoliikenteen perusteet /2009/ Liisa Marttinen

88



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);
        }
    }
}
```

Pura paketti:
data, IP-osoite
ja portti

Lähetä muokattu data.



Kertauskysymyksiä

- n Asiakas-palvelija-malli? Vertaisverkkomalli?
- n Kuinka asiakas löytää palvelimen?
- n Miten KJ osaa antaa bitit oikealle sovellukselle?
- n Miten koneen nimestä saadaan selville sen IP-osoite?
- n Miten HTTP-protokolla toimii?
- n Miksi SMTP ei riitä, vaan tarvitaan POP3 tai IMAP?
- n Mitä hyötyä on proxy-palvelimesta?
- n Miksi käytetään evästeitä?
- n Mikä on pistoke ja missä sitä käytetään?

Ks. myös kurssikirja s.170.