



Sovelluskerros

Kirja sivut 354-399



- Nämä protokollat yhdistävät sovelluksia ja käyttävät alempien tasojen protokollia hyväkseen
 - Sovellukset kommunikoivat käyttäen sovellusprotokollaa
 - Sovellukset hyödyntävät TCP:tä, UDP:ta ja niiden kautta IP:tä palveluksina
 - Sovelluksen on ymmärrettävä jotain portti- ja IP-osoitteista
- Sovelluskerroksen protokollat on yleensä suunniteltu vastaamaan sovelluksen tietoliikennetarpeita
 - Tosi aikaisuus tai asynkroninen viestintä...
 - Viestinvälitys, tiedostonkopiointi, pääteistunto, mittausdata...



- Sovellusarkkitehtuuri määrittelee sovelluksen rakenteen
 - Mitä informaatiota on missäkin
 - Mikä tehtävä sovelluksen eri osilla on
 - Mitä protokollaa käytetään tiedon siirtoon
 - Mitä dataformaatteja käytetään
- Tämän kurssin puitteissa tyydytään tarkastelemaan olemassaolevia arkkitehtuureja, kuten
 - Asiakas-palvelin
 - Vertaisverkot



- World Wide Web on 1990-luvun alussa yleistynyt hypertekstipohjainen multimediajärjestelmä
 - On sekä sovellus itsessään, että alusta sovelluksille
 - Perustui alkuvaiheessa HTML-kielisten tiedostojen kopiointiin HTTP-protokollalla
 - HyperText Markup Language
 - HyperText Transfer Protocol
 - Tiedostot sijaitsevat palvelimessa, asiakasohjelma kopioi tiedostoja ja näyttää ne käyttäjällä
- Toteuttaa asiakas-palvelin arkkitehtuuria
 - Palvelin odottaa palvelupyyntöjä
 - Asiakas ottaa yhteyttä palvelimeen ja pyytää palvelua



Asiakkaan HTTP-viestintä

- Asiakas (selain) saa URL-osoitteen ja tulkitsee sen
protokolla://koneen_nimi:portti/hakupolku
- Asiakas avaa TCP-yhteyden palvelimeen
 - DNS-kysely nimestä IP-osoitteeksi
 - Komento käyttöjärjestelmän TCP:lle avata yhteys
- Asiakas antaa HTTP-protokollan mukaisen hakupyynnön

```
GET hakupolku HTTP/1.0
Host: koneen_nimi
...
```



Palvelimen HTTP-viestintä

- Palvelin on varannut tietyn TCP-portin tai portteja ja kuuntelee näitä portteja, odottaen yhteyksiä
- Käyttöjärjestelmä yhdistää tulevan yhteyden ja viestii palvelimelle uudesta yhteydestä
- Palvelin tulkitsee pyynnön ja antaa pyyntöön vastauksen
 - HTTP/versio status-koodi viesti
 - Jos hakupyyntöä vastaava tietoalkio löytyy
 - HTTP/1.1 200 OK
 - Date: ...
 - Sekä ko. tietoalkio
- Palvelimen vastaus saattaa olla HTML-tiedosto, kuva, äänitiedosto jne.



Asiakas-palvelinmalli

- Asiakas-palvelinmallissa palvelimella on jokin tarvittava resurssi (dataa, laskentakapasiteettia) ja palvelin odottaa passiivisena asiakkaan palvelupyyntöjä
- Usein puhutaan ohuista (thin) ja paksuista (fat) asiakkaista
 - Ohut asiakas ei prosessoi dataa kovinkaan paljoa
 - Esim. pääte-emulaattori, WWW-selain
 - Paksu asiakas prosessoii dataa
 - Esim. WWW-liittymä sähköpostiin, joka on sähköpostijärjestelmän näkökulmasta asiakas

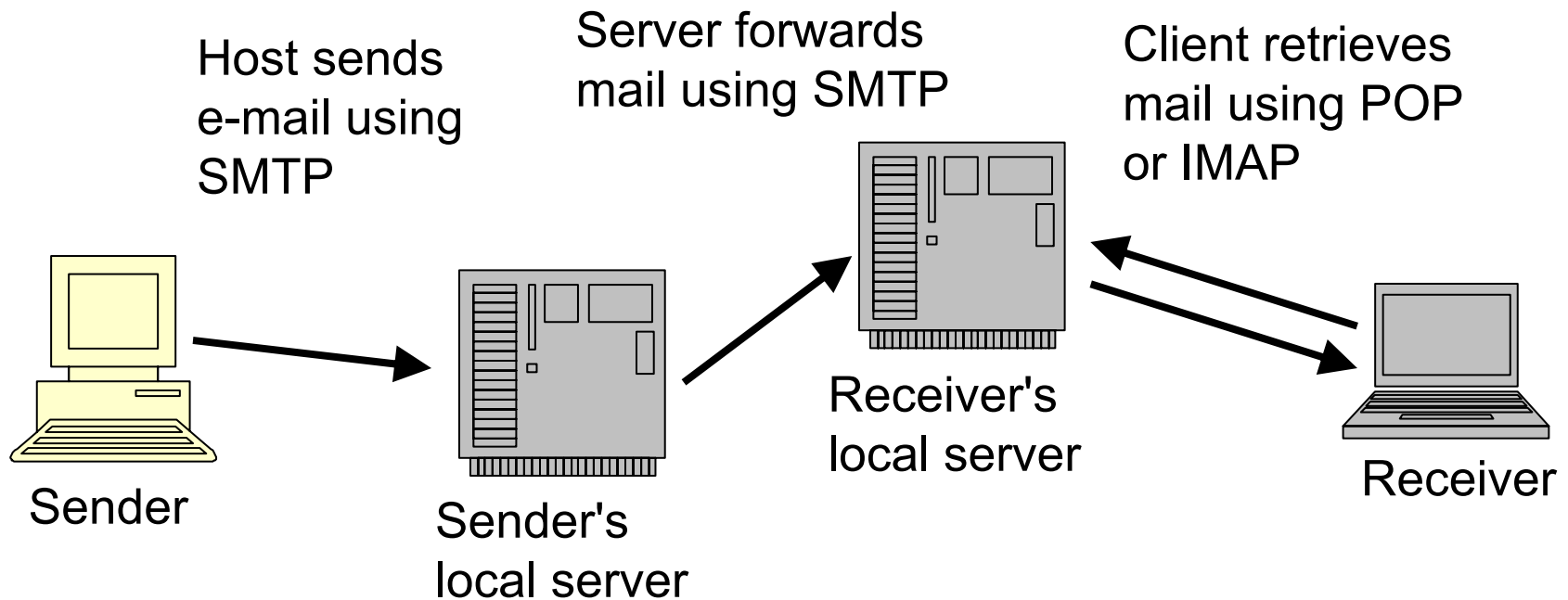


Internetin sähköposti

- Sähköposti on WWW:tä vanhempi palvelu
 - Verkkokäytössä vuodesta 1972
- Sähköposti perustuu myös palvelimiin, eroaa WWW:stä
 - Multimediatiedostojen sijaan välitetään viestejä
- Sähköpostiviestiä työnnetään (push) SMTP-protokollalla kohti vastaanottajan palvelinta, josta vastaanottaja hakee (pull) sen POP tai IMAP-protokollalla
 - Simple Mail Transfer Protocol
 - Post Office Protocol
 - Internet Message Access Protocol



Sähköpostin matka





Asiakas vai palvelin

- Sähköpostipalvelin ottaa samanaikaisesti sekä asiakkaan että palvelimen roolin
 - TCP-portissa 25 tulevaa postia odottava prosessi on palvelin
 - Kun sama prosessi ottaa yhteyden toiseen palvelimeen se on asiakas
- Roolit eivät siis ole välttämättä kiinteitä
- Sama prosessi voisi myös toimia IMAP ja POP-palvelimena
 - Yleensä sähköposti talletetaan käyttäjän postilaatikkoon (tiedosto) ja eri ohjelmisto tarjoaa lukemispalvelun
- Meillä on siis vaihtuvat roolit ja ero palvelimen sisäisen ja ulkoisen arkkitehtuurin välillä



Työntäminen ja vetäminen

- Kun viestiä välitetään SMTP:llä eteenpäin, sitä työnnetään (push)
 - Vastaanottava palvelin ei voi tietää mistä viesti olisi tulossa
- Sähköpostin lukuohjelma voisi myös olla postia odottava palvelin
 - Mutta työasemat eivät ole luotettavasti päällä ja käyttäjä saattaa haluta käyttää useita asiakasohjelmia
 - Joten posti jää vastaanottajan palvelimelle, josta se haetaan (pull)
 - Myös WWW:n HTTP on pull-protokolla
 - Asiakasohjelma käy säännöllisesti kyselemässä (poll) palvelimelta uutta postia



Internet-sähköpostin roolit tarkemmin

- Mail User Agent on käyttöliittymän tarjoava ohjelma
 - Ohjelma, josta posti lähtee ja jonne se tulee
 - Pine, Microsoft Outlook, MH, Mozilla, Elm, mail, Firefox jne.
- Mail Transfer Agent siirtää postia verkossa
 - Reitittää viestin MUA:den välillä vastaanottajan osoitteen perusteella
 - Sovellustason reititystä, ei suoraan suhteessa IP-reititykseen
 - MTA saattaa muuttaa vastaanottajan osoitetta ja ottaa viestin uuteen käsittelyyn
 - Esim. aliakset, .forward



Sähköpostin matka tarkemmin

- SMTP-viestissä on kaksi osaa
 - Kuori, jota SMTP-protokolla käyttää viestin välitykseen
 - Sisältö, viesti otsikkoineen
- MUA vastaanottaa viestin käyttäjältä ja luo kuoren
- MUA välittää viestin palvelevalle MTA:lle (asetusten mukaan)
 - MTA ottaa viestin vastaan ja yleensä tallettaa sen spool-hakemistoon odottamaan edelleen välitystä
 - Lopullisen vastaanottajan MTA tallettaa viestin vastaanottajan postilaatikkoon
 - Postilaatikko on tiedosto, tietokanta tms.



Store and Forward-konsepti

- Suom. etappivälitys
- Vastaanottaja tallettaa viestin ennen eteenpäinlähetytstä
 - Muistiin tai levyille
- IP-reittimet ottavat IP-paketin vastaan ennen kuin katsovat mihin lähtevään jonoon se sijoitetaan
 - Jos jono on täynnä, paketti tuhotaan
- Sähköpostipalvelimet ottavat viestin vastaan, tallettavat spool-hakemistoon ja kuittaavat saaneensa viestin
 - Vastuu siirtyy, vikatilanteessa voidaan käydä palvelimien lokitiedostot läpi ja etsiä missä viesti katosi



Esimerkki SMTP-istunnosta

```
morphine ~ 1$ telnet mail.tml.hut.fi 25
Connected to mail.tml.hut.fi (130.233.47.34).
220 mail.tml.hut.fi ESMTP
helo morphine.tml.hut.fi
250 mail.tml.hut.fi
mail from: joulupukki@korvatunturi.fi
250 Ok
rcpt to: kiravuo@tml.hut.fi
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
From: Joulupukki@Korvatunturi.fi
To: Kaikki kiltit lapset
Subject: Joulu tulee

Muistakaa olla kiltteja, pukki valvoo.
T. joulupukki
.
250 Ok: queued as 9C5773A2CCC
quit
221 Bye
```



Vastaanotettu viesti

Return-Path: <joulupukki@korvatunturi.fi>
Received: from mail.tml.hut.fi (mail.tml.hut.fi
[130.233.47.34]) by tml-yp-4.tml.hut.fi (Cyrus v2.2.12)
with LMTPA; Tue, 20 Feb 2007 14:11:19 +0200
Received: from morphine.tml.hut.fi (morphine.tml.hut.fi
[130.233.45.7]) by mail.tml.hut.fi (Postfix) with SMTP id
9C5773A2CCC for <kiravuo@tml.hut.fi>; Tue, 20 Feb 2007
14:10:11 +0200 (EET)
From: Joulupukki@Korvatunturi.fi
To: Kaikki@tml.hut.fi, kiltit@tml.hut.fi, lapset@tml.hut.fi
Subject: Joulu tulee
Message-Id: <20070220121011.9C5773A2CCC@mail.tml.hut.fi>
Date: Tue, 20 Feb 2007 14:10:11 +0200 (EET)

Muistakaa olla kiltteja, pukki valvoo.

T. joulupukki



- Kuoressa on MTA:n näkemys lähettäjistä ja vastaanottajista
 - Eri asia kuin viestissä oleva lähettäjä (From:) ja vastaanottaja
 - Virukset ja spam väärinkäyttävät tätä ominaisuutta
 - SMTP:n MAIL FROM ja RCPT TO hyödyntävät näitä
- Otsakkeet
 - Viestin alusta ensimmäiseen tyhjään riviin saakka
- Viestin runko
 - Otsakkeiden jälkeen



- MX-tietueet
 - Mail eXchanger - tietueet DNS-järjestelmässä
 - Kertovat minne tietyn osoitteen sähköposti ohjataan
 - Mahdollistavat palvelimien priorisoinnin ja varapalvelimien esittelemisen
 - Esim: sral.fi:n MX-tietueen
 - sral.fi. IN MX 10 bar.foo.fi.
 - sral.fi. IN MX 20 smtp3.kolumbus.fi.
- Logiikka: postia siirretään MX-listassa pienimmälle arvon omaavalle palvelimelle, johon saadaan yhteys
 - Jos yhtään MX-tietuetta ei ole määritelty, kokeillaan A-tietuetta (IP-osoite)



Postin lukeminen verkossa

- Saman koneen (esim. Unix-palvelin) sisällä posti luetaan suoraan postilaatikosta
 - Tiedosto, esim. /var/spool/mail-hakemistossa
- MUA ottaa yhteyttä palvelimeen ja pyytää asiakkaan viestejä
 - Välittää käyttäjätunnuksen ja salasanan
 - Saattaa käyttää SSL-salausta
- POP, Post Office Protocol
 - TCP-portissa 110 (versio 3)
 - Käytetään yleensä vain postin noutamiseen
- IMAP, Internet Message Access Protocol
 - TCP-portissa 143
 - Tukee useita kansioita (hakemistoja) palvelimella



Sähköpostin kokonaisarkkitehtuuri

- Sähköpostin arkkitehtuuri koostuu siis monista osista ja sitä voidaan tarkastella eri tavoin
 - Abstraktisti, MTA, MUA, viestin rakenne
 - Konkreettisesti, SMTP, POP, IMAP, RFC-2822:n määrittelemä viestin muoto
 - Roolien ja ohjelmaprosessien vaatimuksien mukaan
- Arkkitehtuurin olennainen hyöty on osien vaihdettavuus ja sisäisen rakenteen piilottaminen
 - Postilaatikon sisäinen rakenne ei vaikuta muiden ohjelmien toimintaan, jos sen kanssa asioidaan vain SMTP:llä ja IMAP:lla
 - Kun järjestelmällä on arkkitehtuuri sen yksittäisiä osia voidaan kehittää koskematta muihin



Keskusteluryhmät ((Usenet) News)

- Sähköposti on pienen ryhmän kaksisuuntaista viestintää ja WWW suuren joukon yksisuuntaista viestintää
- News-järjestelmä kehittyi 1980-luvulla (ennen WWW:tä) laajaksi hajautetuksi keskustelufoorumiksi
- Keskustelu on jaettu ryhmiin ja artikkeleihin
 - Sama artikkeli voidaan lähettää useaan ryhmään
 - Ryhmillä on nimet
 - sfnet.harrastus.retkeily
 - Artikkeleilla on uniikit tunnisteet
Message-ID: <Pine.GS0.1.64.0702014510.4033@hila.hut.fi>
- Palvelimet vaihtavat artikkeleita keskenään



News-järjestelmän arkkitehtuuri

- Palvelimille on annettu tieto toisista palvelimista
- Ne ottavat säännöllisesti yhteyksiä toisiinsa ja vaihtavat artikkeleita
 - Message-ID:n perusteella palvelin voi päätellä onko sillä jo kyseinen artikkeli
 - Käytetty protokolla on NNTP, Network News Transfer Protocol
- Palvelimet muodostavat vertaisverkon, jossa uusi artikkeli leviää verkon kaikkiin palvelimiin (flooding)
- Käyttävät lukevat artikkeleita asiakasohjelmilla vertaisverkon palvelimilta



- P2P (Peer to Peer) verkoissa koneet ovat suhteellisen tasa-arvoisia keskenään
 - Eroaa siis asiakas-palvelinmallista
- Käytetyt protokollat saattavat erottaa asiakkaan ja palvelimen roolin
 - Tyypillisesti yksi kone ottaa yhteyttä toiseen
 - Sisällöllisesti viestintä on kuitenkin tasa-arvoista
 - Yleensä yhdessä sovelluksessa on sekä asiakkaan että palvelimen ominaisuudet ja se voi toimia kummassa tahansa roolissa
- Arkkitehtuuri on yleensä hajautettu ja dynaaminen
 - News-palvelimet tuntevat vain naapurinsa, uusien liittyminen on helppoa



Tiedostonjakoverkot

- Uudempaa kehitystä kuin News-järjestelmä
- Tavalliset käyttäjät voivat jakaa tiedostoja keskenään
- Halutun tiedoston löytäminen on arkkitehtuurin ydinkysymys
 - P2P-mallin rikkova indeksipalvelin
 - Tiedostoon viittaava tunniste ja algoritmi, joka kyselee muilta palvelimilta tietoa tiedostosta
 - Tiedoston pilkkominen palasiin ja (redundanttien) palasten löytäminen kuten tiedoston, kunnes koko tiedosto on koossa
- Tiedostonjakoverkkojen teknologiat kehittyvät parhaillaan ja mahdollistavat mm. tiedoston salaamisen ja jakamisen siten, että kukaan ei tiedä missä tiedoston osat ovat, mutta se löytyy vertaisverkosta



Protokollat ja rajapinnat

- Protokollat mahdollistavat verkon olioiden tiedonvaihdon
 - Firefox OS X:ssä voi lähettää sähköpostia Sendmailille Unixissa, josta vastaanottaja lukee Outlookilla Windowsissa
 - Yhteinen kieli peittää toteutuksien erot
- Protokollat käyttävät alempien tasojen **palveluita rajapintojen** kautta
 - Sovellukset puhuvat sovellusprotokollia socket-rajapinnan kautta TCP/IP:lle
 - Ethernet-käsittelijä puhuu ethernet-kortille laiteohjainrajapinnan kautta
 - Rajapinnat ovat yleensä käyttöjärjestelmäkohtaisia
 - Rajapinnalla ei ole merkitystä protokollan toiminnalle, vaikuttaa lähinnä ohjelmien siirrettävyyteen
- Internet-protokollaperhe tarjoaa abstraktiokerroksen, joka peittää eri fyysisten tietoliikennekerrosten ominaisuudet
- Ylemmän tason protokollat luottavat alemman tason protokollien palveluihin



TEKNILLINEN KORKEAKOULU

Tietoliikenneohjelmointi



Berkeleyyn pistoke-rajapinta

- C-kielinen Berkeley Socket API, joka julkaistiin 4.2 BSD Unixissa on muodostunut yleiseksi rajapinnaksi käyttöjärjestelmän TCP/IP-pinon palveluihin
 - Application Programming Interface, API
 - Berkeley Software Distribution on tärkeä ja merkityksellinen versio Unixin historiassa
 - Käytössä Unixissa ja Windowsissa
 - Käytössä suunnilleen saman näköisenä muissakin kielissä
- Rajapinta erottaa teknisen toteutuksen sitä abstrahoivasta mallista
 - Sovellusohjelmoijan ei tarvitse tietää miten tekniikka on toteutettu



Miten API:a käytetään?

- Käyttöjärjestelmä- ja kielikohtainen asia
- Yleensä
 - Lisätään ohjelmakoodiin API:n kuvaavia otsaketiedostoja
 - Kuvataan tieto API:n määrittelemässä muodossa
 - Käytetään API:n määrittelemiä ohjelmakutsuja
 - Linkataan API:n toteuttava binäärikirjasto ohjelmaan mukaan
- Rajapinnat eristävät ohjelmistojen osia toisistaan mahdollistaen
 - Osien uudelleenkäytön
 - Integroinnin erilaisiin muihin järjestelmiin
 - Vastuiden määrittelyn
- Integrointi on tärkein syy API:n olemassaoloon



Pistokkeen käyttö

- Luodaan pistoke: `int socket(int domain, int type, int protocol)`
 - Tyyppi voi olla mm.
 - `SOCK_STREAM` (TCP)
 - `SOCK_DGRAM` (UDP)
 - `SOCK_RAW` (raw IPv4)
- Määritellään tietue osoitteen kuvaamiseksi

```
struct in_addr {
    in_addr_t s_addr; /* IPv4 address, network byte order */ };
struct sockaddr_in {
    sa_family_t sin_family; /* AF_INET */
    in_port_t sin_port;      /* 16-bit port, network byte order*/
    struct in_addr sin_addr; /* IPv4 address */
    ...};
```
- Avataan yhteys: `connect()`
- Käytetään pistoketta: `read()`, `write()`
- Lopuksi suljetaan yhteys: `close()`



Esimerkki: HTTP-asiakas (1/4)

```
/*  
 * Simple HTTP client program, version 1.  
 * Written by Pasi.Eronen@nixon.fi.  
 */  
#include <arpa/inet.h>  
#include <netinet/in.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/socket.h>  
#include <sys/types.h>  
#include <unistd.h>
```



Esimerkki: HTTP-asiakas (2/4)

```
void die(const char* message)
{
    fprintf(stderr, "%s\n", message);
    exit(1);
}

int main(int argc, char *argv[])
{
    int sockfd, n;
    struct sockaddr_in addr;
    unsigned char buffer[4096];

    if (argc != 4)
        die("usage: geturl ip-address port local-url");
```



Esimerkki: HTTP-asiakas (3/4)

```
/* Open socket */
if ((sockfd = socket(AF_INET, SOCK_STREAM, 0))
    == -1)
    die("socket error");
/* Parse address */
memset(&addr, 0, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_port = htons(atoi(argv[2]));
addr.sin_addr.s_addr = inet_addr(argv[1]);
if (addr.sin_addr.s_addr == -1)
    die("bad address");
/* Connect to remote host */
if (connect(sockfd, (struct sockaddr*) &addr,
            sizeof(addr)) == -1)
    die("connect error");
```



Esimerkki: HTTP-asiakas (4/4)

```
/* Send HTTP request */
write(sockfd, "GET ", 4);
write(sockfd, argv[3], strlen(argv[3]));
write(sockfd, " HTTP/1.0\r\n\r\n", 13);
/* Read response */
while ((n = read(sockfd, buffer,
                  sizeof(buffer))) > 0)
    write(STDOUT_FILENO, buffer, n);
/* Close and exit */
close(sockfd);
return 0;
}
```



```
$ ./httpclient1 192.168.3.4 80 /  
HTTP/1.1 200 OK  
Date: Sat, 24 Apr 1999 17:08:25 GMT  
Server: Apache/1.3.4  
Last-Modified: Fri, 26 Feb 1999 15:28:20 GMT  
Connection: close  
Content-Type: text/html  
<html><head><title>Example Inc.</title></head>  
<body>  
<h1>Welcome to Example Inc's web server!</h1>  
...  
$
```



- Eroaa tavallisista ohjelmista
 - Suoritetaan taustalla
 - Yleensä ei varsinaista käyttöliittymää
 - Toteuttaa verkosta tulevia palvelupyyntöjä
 - Tarvittaessa lähettää palvelupyyntöjä muille palvelimille
- Usein samanaikaisten palvelupyyntöjen käsitleminen on vaatimus
 - Moniajoa toteuttamalla ei-estäviä systeemikutsuja samanaikaisesti tai
 - säikeillä tai
 - luomalla uusia palvelinprosesseja
- Tehokkaan palvelinprosessin suunnittelemisen on kohtuullisen vaativaa



Palvelin pistokeympäristössä

- Palvelinprosessi kuuntelee tiettyä porttia: `bind()`
 - Tiettyä porttia tai myös portti/IP-osoite yhdistelmää
- Ja ottaa yhteyksiä vastaan: `accept()`
- `accept()`-kutsulla palvelinprosessi saa kahvan tiedostoon (TCP-yhteys) lukemista ja kirjoittamista varten
 - `getpeername()` palauttaa vastapään portin ja IP-osoitteen
 - `getsockname` kertoo portin ja IP-osoitteen palvelimen päässä



Hajautettu tietojenkäsittely

- Datat ja prosessoinnin hajauttaminen verkon olioille läpinäkyvästi on ollut pitkään tietojenkäsittelyn tavoitteena
 - Ideaalisesti ohjelmoijan ei tarvitse tietää hajautuksesta
- Sun RPC oli ensimmäinen suosittu menetelmä
 - Remote Procedure Call
 - Aliohjelmakutsu toteutetaan toisessa tietokoneessa
- Oliopohjainen CORBA on yleistynyt sittemmin
 - Common Object Request Broker Architecture
- Web Services lähestyy asiaa hivenen toisin
 - Palvelukutsut esitetään XML-muodossa
- Hajautuksen etuna on jaettujen resurssien läpinäkyvä hyödyntäminen ja arkkitehtuurin joustavuus
- Haittana on lisääntynyt monimutkaisuus ja tietoliikenteen vaatimukset
 - Kaksi toisistaan riippuvaa tietokonetta ei kaksinkertaista luotettavuutta vaan puolittaa sen



Perustuu Bengt Sahlinin materiaaliin



WA-7



WA-9



WA-10



WA-10L



WA-11A



WA-12A



WA-14



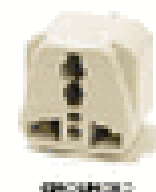
WA-15



WA-16



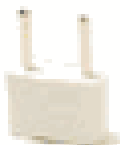
WA-20



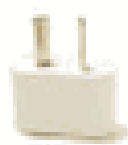
GROUNDED
BRACE



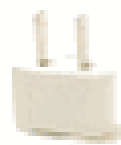
UNGROUNDED
BRACE



#1



#2



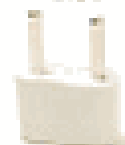
#3



#4



#5



#6



- Standardi on yleisesti hyväksytty mahdollisimman **yksiselitteinen** määrittely
- Standardointi edistää eri valmistajien tuotteiden
 - Ennustettavuutta
 - Yhteensopivuutta
 - Vaihtokelpoisuutta
- Standardointi voi vakiinnuttaa olemassa olevia käytäntöjä tai luoda pohjan uusien tuotteiden suunnittelulle
- Liki kaikkea voi standardoida
 - Ohjelmistorajapintoja
 - Tietoliikenneprotokollia
 - Fyysisiä liittimiä, jännitetasoja, virtarajoja
 - Tuotekehitysprosesseja
 - Hyväksi havaittuja käytäntöjä (best practice -standardit)



De facto -standardit

- Tyypillisesti valmistaja, jolla on merkittävä asema markkinoilla tai syntyy luonnostaan
 - Tuote on ensimmäinen laatuaan
 - Avaintuote
 - Markkinajohtajuus
 - Varsinaisten standardien puute
- Esimerkiksi:
 - Windows, PC-arkkitehtuuri kokonaisuutena
- Edut:
 - Tässä ja nyt käytössä oleva menettely
- Riskit:
 - Usein puutteellisesti määriteltä
 - Standardin haltija voi hyötyä tilanteesta



Internet Engineering Task Force (IETF)

- Organisaatio, joka standardoi Internetiin liittyviä asioita
 - Etenkin protokollia
- Koostuu työryhmistä
 - Määritelty tavoite
 - Ryhmä lakkauttaa itsensä kun tavoite on saavutettu
- Avoin prosessi
 - Kuka tahansa voi osallistua IETF:n toimintaan ja työryhmiin
 - Suurin osa työstä sähköpostilistoilla
 - Tapaamisia kolmesti vuodessa
- Alkuperäinen ohjausperiaate: “running code and rough consensus”
 - Nykyään IETF kärsii kasvusta
- <http://www.ietf.org/>



Internet Engineering Task Force (IETF)

- Kaikki julkaisut ovat saatavilla maksuttomasti
 - Request for Comments -sarja (RFCt)
 - <ftp://ftp.funet.fi/pub/doc/rfc/>
 - Standardit
 - Tiedoksi-julkaisut
 - Tutkimustuloksia
 - Vanhentuneet versiot pysyvät mukana
 - Hyväksi havaittuja käytäntöjä dokumentoivat
 - Internet Draftit
 - Työdokumentteja, yleensä ideoita tai ehdotuksia protokolliksi tai arkkitehtuureiksi
 - Kuka tahansa voi julkaista Draftin
 - Työryhmä voi jatkaa kehitystä



3rd Generation Partnership Program (3GPP)

- Määrittelee kolmannen sukupolven mobiileja solukkoverkkoja (WCDMA)
- Julkaisee kokonaisia standardiperheitä jaksoissa (parhaillaan määritellään release 7:ää)
- <http://www.3gpp.org/>
- Jäsenet ovat organisaatioita
 - Yrityksiä, julkishallintoa
- Määrittelyt luodaan vaiheittain
 - Vaihe 1 (stage 1): Vaatimukset
 - Vaihe 2 (stage 2): Arkkitehtuuri
 - Vaihe 3 (stage 3): Yksityiskohtainen määrittely



3rd Generation Partnership Program (3GPP)

- Toiminta on jaettu osa-alueisiin
 - RAN: 3G-radioverkko (Radio Access Network)
 - GERAN: GSM ja EDGE radioverkko
 - CT: Runkoverkko ja päätelaitteet
 - SA: Palvelut ja järjestelmäarkkitehtuuri
- Alueilla on tarkemmin fokusoituja työryhmiä
 - Esim. SA WG1 keskittyy palveluihin
 - RAN WG1 keskittyy radioverkon tasoon 1
- Kaikki dokumentit ovat virallisia määrittelyjä olennaisille toiminnoille
 - UMTS
 - IMS
 - MBMS
 - WLAN interworking



Muita standardointiorganisaatioita

- 3GPP2
 - Eri teknologia 3G:hen (USA-vetoinen)
- Institute of Electrical and Electronics Engineers (IEEE)
 - Ethernet, WLAN...
- International Tele-communication Union (ITU)
 - YK:n alainen standardointiorganisaatio
 - Jäseninä valtioita ja suuria organisaatioita
 - On standardoinut mm. ISDN:n, B-ISDN:n ja GSM:n
- European Telecommunications Standards Institute (ETSI)
- Open Mobile Alliance (OMA)
 - Mobiilipalveluita
- jne.



Standardoinnin merkitys

- Tekninen yhteensopivuus ja laatu
- Kommunikointi ostajan ja myyjän välillä
 - Standardeja noudattavilla tuotteilla yleensä kilpailuetu
- Insinööriosaimisen dokumentointi
- Lisää aiheesta ETSI:n julkaisu "Making Better Standards"
 - <http://portal.etsi.org/mbs/>
 - Esittelee mm. kuvaustyökaluja, kuten MSC



MSC-kaaviot

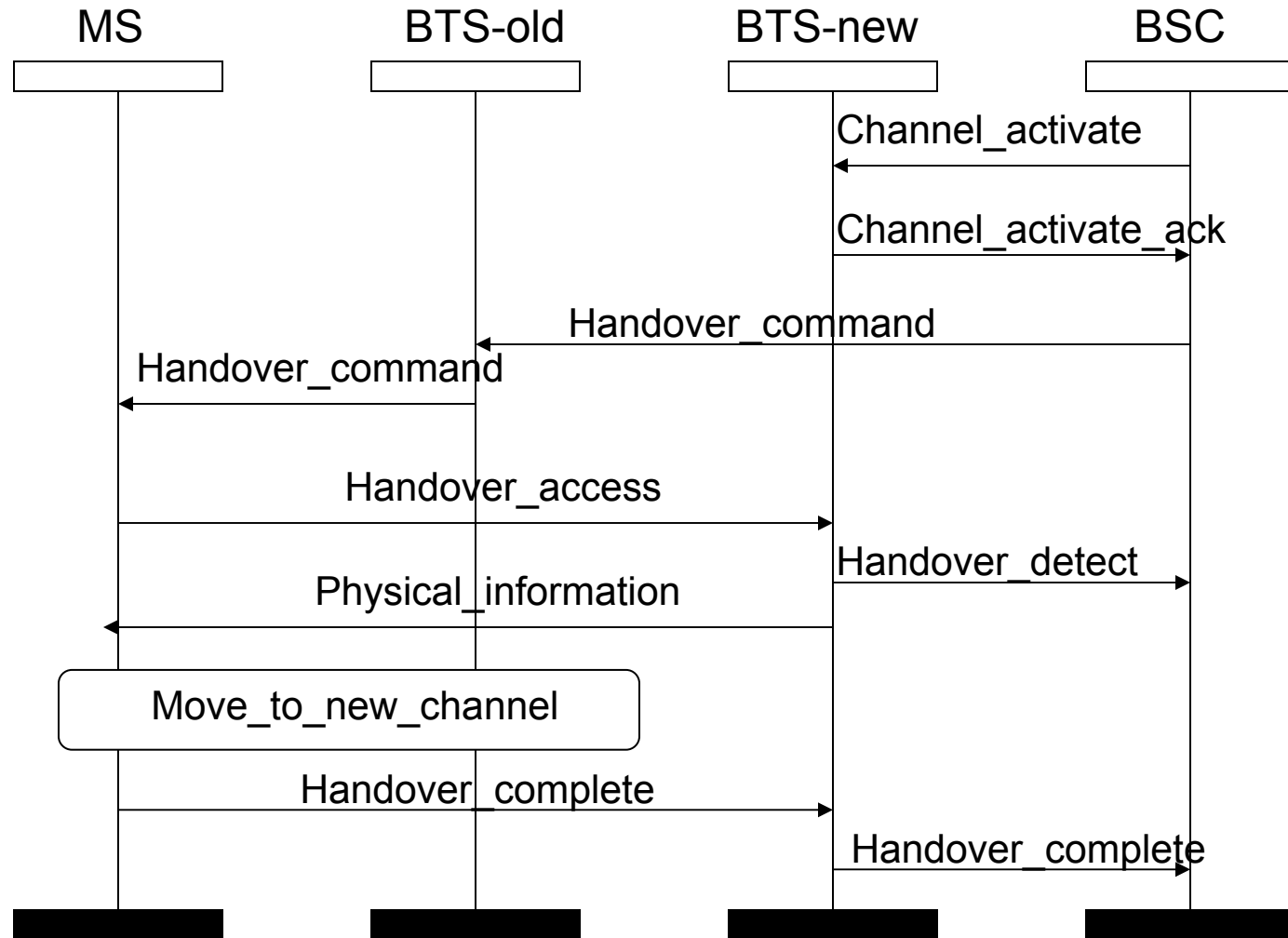


Message Sequence Chart

- Yksinkertainen ja hyödyllinen graafinen työkalu viestinnän järjestyksen ja osallistujien esittämiseen
- Notatio
 - Pystysuorat viivat esittävät osallistujia
 - Viestit ovat vaakasuoria nimettyjä nuolia
 - Viistoviivoilla voidaan esittää risteäviä viestejä
 - Aika virtaa kaaviossa alaspäin
- Käyttö
 - Tärkeimpien tapahtumien esittäminen
 - Auttavat asettamaan vaatimuksia protokollalle suunnittelussa

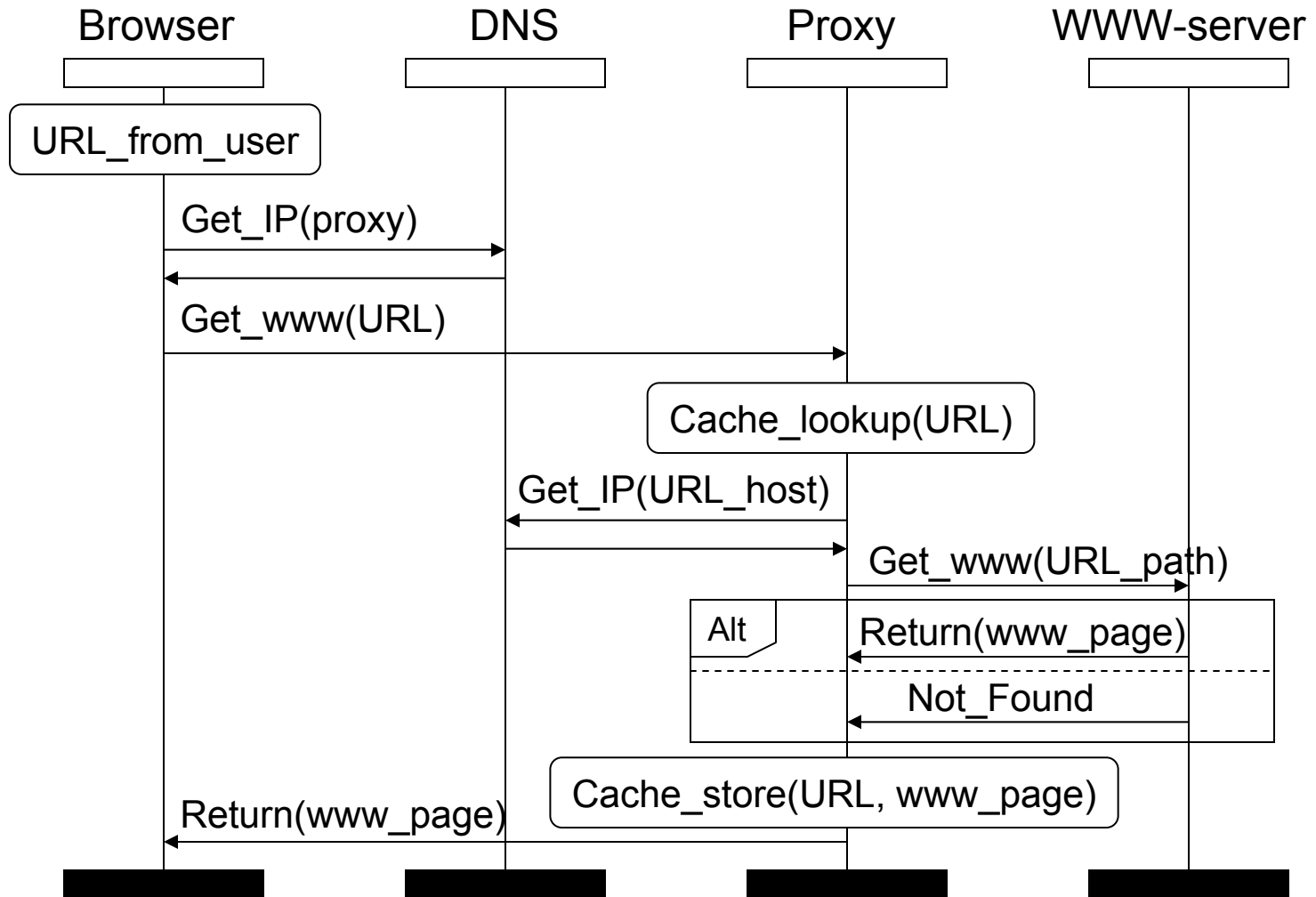


Esimerkki MSC-kaaviosta





Toinen esimerkki MSC- kaaviosta (osittainen)





TEKNILLINEN KORKEAKOULU

Web 2.0 ja uusi maailmanjärjestys



- Web 2.0 on Internetin uusiin mahdollisuuksiin viittaava muotitermi
 - Kuka tahansa voi olla sisällöntuottaja
 - Sosiaaliset verkot ja yhteistyö ovat avainsanoja
 - Joukko WWW-pohjaisia työkaluja tasa-arvoistaa verkon käyttöä
- Muotitermit saattavat vanheta muutamassa vuodessa
 - Muutos on kuitenkin todellista ja havaittavaa
 - Uusia sovelluksia tulee jatkuvasti
 - Myös vanhoja sovelluksia yhdistelemällä tehtyjä "mashups"-sovelluksia
 - Esim. yhdistämällä 112.fi ja Google Maps



Historiaa: Korkeakoulujen Internet

- Vuosiin 1992-1995 saakka Internet oli akateeminen tutkimusverkko
 - Voimakas yhteistyön ja ei-kaupallisuuden kulttuuri
- Joukko uusia konsepteja kehittyi
 - Keskustelun moderointi ja netiketti
 - FAQ-listat (Frequently Asked Questions)
 - "Rough consensus and running code" hallinnan asenteena
 - Jos se toimii ja suurin osa ihmisistä hyväksyy sen, käytetään sitä
- Sitten Internet aukesi massoille
 - WWW ja etenkin NCSA Mosaic 1993 on selvä rajapyykki
- Joukko uusia työkaluja, WWW toimii usein sovellusalueena tai käyttöliittymänä



- Wiki mahdollistaa WWW-palvelimen sisällön helpon muokkaamisen
 - Erona yhdelle ylläpitäjälle, joka muokkaa HTML-tiedostoja
 - Wiki on lähellä Tim Berners-Leen alkuperäisiä ideoita yhteistyön mahdollistavista tietojärjestelmistä
- Wikit ovat suosittuja yhteistyöalustoja
 - Järjestelmä tallettaa muutokset ja mahdollistaa paluun aikaisempiin versioihin
- Wikipedia on Wiki-teknologiaan pohjaava tietosanakirja



- "Really Simple Syndication" ja muutama muu selitys tälle akronyymille
- XML-pohjainen tiedostomuoto WWW-palvelimen päivittyvän informaation kuvaamiselle
 - XML on kieli, tehtäväkohtaisten kielten luomiseen datan kuvaamiseksi
- Kun palvelimen sisältöä päivitetään, ylläpitäjä päivittää RSS-kuvauksen
 - Käsillä tai ohjelmallisesti
- Käyttäjällä on RSS-lukija, joka seuraa (pollaa) WWW-palvelimia ja kertoo käyttäjälle päivityksistä
 - Kuten sähköpostin tai keskusteluryhmien lukuohjelma
- RSS:n XML-kuvauksessa on yleensä lyhyt kivaus päivityksestä ja linkki WWW-sivulle



- Blogit ovat sosiaalinen ilmiö
- Käyttäjä pitää päiväkirjaa WWW-sivustolla ja päivittää sitä säännöllisesti
 - Käsien tai ohjelmallisesti
 - Blogin yhteydessä saattaa olla keskustelufoorumi
- Blogi-järjestelmä tarjoaa usein RSS-tietoa päivitetystä artikkelista
 - RSS lisää blogien mielekkyyttä merkittävästi



- Peer to Peer -arkkitehtuuri
- Paradigmamuutos
 - Sähköposti, News, IRC, WWW jne. vaativat palvelimen
 - Tämän takia asiakas-palvelinarkkitehtuuri on suosittu
 - P2P olettaa, että palvelimia ei ole, vaan käyttäjien tietokoneet ottavat toisiinsa yhteyttä suoraan
 - Käytännössä P2P-ohjelmistot toteuttavat sekä palvelimen että asiakkaan TCP-yhteyden näkökulmasta
- P2P-verkkojen koneet tarjoavat informaatiota tai palveluita
 - Tiedoston tai palvelun löytäminen on haaste
 - Ratkaistavissa indeksipalvelimilla tai
 - Käyttämällä tunnisteita ja hakualgoritmia
- P2P tunnetaan tiedostojaosta, mutta teknologialla on paljon enemmän potentiaalia



- Yhteyksien teknisen tason kehittyessä niitä voidaan käyttää yhä kasvavien laatuvaatimusten viestintään
 - Sähköposti → IRC ja IM → yksisuuntainen ääni → kaksisuuntainen ääni ja kuva
- Voice over IP (VoIP)
- IETF-pohjainen VoIP-arkkitehtuuri käyttää kahta protokollaa
 - Signaalointiprotokolla (esim. Session Initiation Protocol)
 - UDP-pohjainen siirtoprotokolla (esim. Real Time Protocol)
 - Rakenne jäljittelee perinteisen puhelinverkon kaksikerroksista rakennetta



- Skype on suljettu VoIP-tuote
 - Käytetty protokola ei ole julkinen
- Skype on suosittu, koska IETF:n protokollat eivät vastanneet todellisuuden asettamia vaatimuksia
 - NAT/PAT muunnos estää P2P-viestinnän teokkaasti, SIP/RTP jämähtää tähän
 - Skype käyttää kolmatta käyttäjää releoidakseen kahden muun keskustelut
- Käyttäjien löytäminen ja yhteyden muodostus perustuu keskuspalvelimiin, jotka tietävät kaikkien kirjautuneiden käyttäjien sijainnin



- Selvästi havaittava trendi
 - Ohjelmistoarkkitehtuurit eivät enää tue "loppukäyttäjä asiakkaana yrityksen omistamalle palvelimelle" -mallia
 - Teknisesti edistyneet käyttäjät ilmaisevat itseään ja tuottavat niin sisältöä kuin palveluitakin
- Tämän päivän käyttöliittymä kaikkeen on WWW
 - Matkaviestimet saattavat muuttaa tämän
 - Pitäisikö sovelluksen tarjota:
 - WWW-liityntä (mukava satunnaiseen käyttöön) vai
 - Java-pohjainen GUI (täsmälleen räätälöity) vai
 - protokollarajapinta ("tee itse asiakasohjelma", IRC, IMAP, NNTP...)
- Mitä tämä tarkoittaa
 - Kirjoittaminen, kirjapaino, kirjoituskone, monistaminen, valokopiointi, julkaisuohjelmistot ovat jo sallineet kenen tahansa sisällöntuotannot
 - Luultavammin tämä on evoluutio, ei revoluutio
 - Maailma muuttuu



- Minä vai me?
 - Haluatko ihmisten tietävän kaiken sinusta?
 - Mahdollisten tulevien työnantajien
 - Informaation määrän ja hakutekniikkojen kasvaessa, tietoturva ja -suoja ovat yhä tärkeämpiä vaatimuksia
 - Kaikki tallentuu nykyään jo arkistoihin
- Kuka minä olen verkossa?
 - Identiteetinhallinta
 - Montako identiteettiä tai roolia henkilöllä on, onko niiden oltava linkitettävissä?
- Luotettavan informaation löytäminen?
 - Wikipedia on yhteistyöprojekti, keiden kanssa?
 - Onko Youtube totta?



Conclusion

If the label on the cable on the table at your house,
Says the network is connected to the button on your mouse,
But your packets want to tunnel on another protocol,
That's repeatedly rejected by the printer down the hall,
And your screen is all distorted by the side effects of gauss,
So your icons in the window are as wavy as a souse,
Then you may as well reboot and go out with a bang,
'Cause as sure as I'm a poet, the sucker's gonna hang!

<http://www.netfunny.com/rhf/jokes/96q1/seuss.html>