



Aalto-yliopisto
Teknillinen korkeakoulu

Sovelluskerros: Socket API, WWW, sähköposti, DNS

Tancred Lindholm

T-110.2100 Johdatus tietoliikenteeseen
kevät 2010

Luennon sisältö

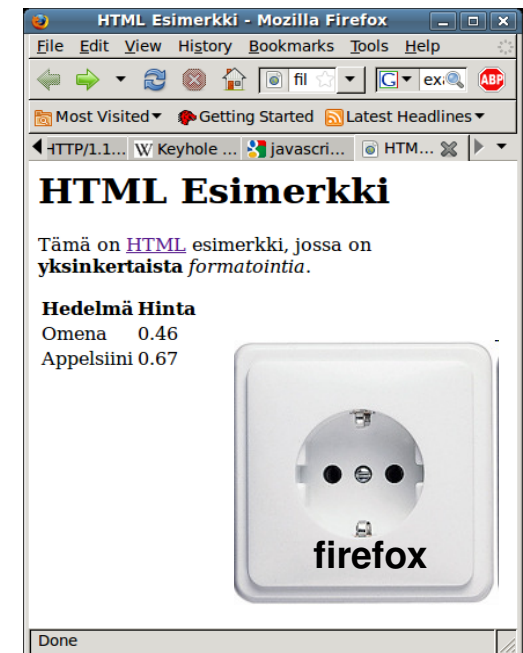
1. Socket-rajapinta
2. HTTP: Miten edellisen luennon viestit siirtyvät koneesta (solmusta) toiseen?
3. Yhteyskäytäntö eli protokolla
4. Tiedon koodaaminen
5. Domain Name System (DNS)
6. Simple Mail Transfer Protocol (SMTP)
7. Yhteenvedo

Sockets-rajapinta

- käyttöjärjestelmä tarjoaa rajapintoja IP -verkkoon
- Yleisin rajapinta nimeltä (Berkeley) "Sockets API" (Applications Programmin Interface)
- Ohjelmille rajapinta TCP- ja UDP- pohjaiseen tiedonsiirtoon
- Alkuperäinen versio BSD-Unixissa v 1983
- Nykyään jokaisessa käyttöjärjestelmässä jossa on tuki IP-verkoille

Soketti (Socket)

- Tietoliikenteessä termiä käytetään merkityksessä "yhteyden päätepiste" (communication endpoint)
- Sokettiin liitetään (bind) yksilöivä osoite
- Soketin avulla voidaan lähettää ja vastaanottaa dataa toisesta soketista
- Kun avataan yhteys sokettiin x käytetään x:n osoite
- Koneessa monta sokettia, tyypillisesti 10-10000 sokettia käytössä yhtäaikaan



Soketin osoite

- Soketin osoite koostuu **IP-osoitteesta** ja **porttinumerosta**
 - Kirjoitetaan IP:portti, esim. 127.0.0.1:80
- IP-osoite
 - Koostuu neljästä luvusta 0-255 joita erottaa piste
 - Internetissä jokaisella tietokoneella IP, tästä tarkemmin seuraavassa luennossa
 - IP:n avulla verkko voi reitittää liikennettä oikeaan koneeseen
- Porttinumero
 - Välillä 0 ja 65535 (16 bittiä)
 - Tietyt portit yhdistetään tiettyihin palveluihin
 - esim portilla 80 on yleensä nettipalvelin

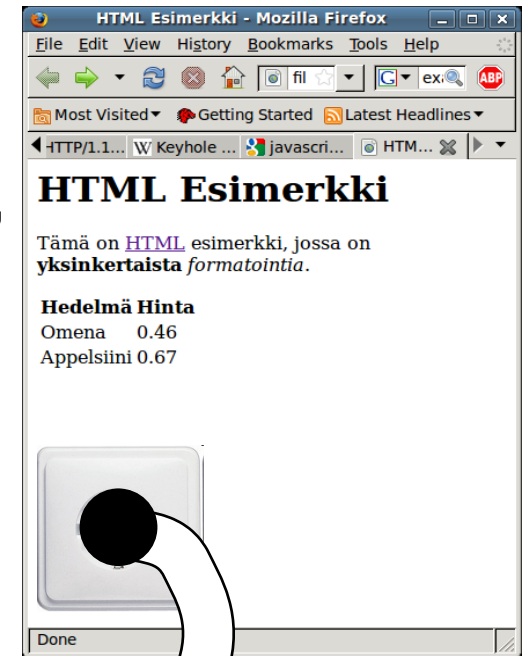


Soketin tyyppi ja tila

- On muutamia erityyppisiä soketteja, joista tavallisimmat:
 - TCP-soketti: luotettava jatkuva tiedonsiirtokanava toiseen koneeseen
 - UDP-soketti: lyhyiden viestien (n. 1000 tavua) lähettäminen toiseen koneeseen ilman että vastaanotto varmistetaan
- Tässä TCP (Transmission Control Protocol) ja UDP (User Datagram Protocol) ovat kunkin sokettityypin käyttämiä siirtoprotokollia (ensi luennolla)
- Myös "raw" ja "reliable sequenced packet service" -tyypit
- Sokettiin joka on kuuntelutilassa (listen) voidaan avata yhteys
 - Käyttöjärjestelmä ja palvelinohjelmisto siirtävät heti yhteyden toiseen sokettiin, jotta voidaan ottaa vastaan seuraava yhteydenpyyntö

TCP-tyyppinen soketti

- Yleisin tapa siirtää tietoa internetissä
- Sovellukset: netti, tiedostonsiirto, etäyhteydet (VPN, etänäyttö), sähköposti, jne.
- Toisiinsa kytketyt soketit voidaan mieltää tavuputkena (bittiputkena). Lähetetään tavuja toisesta päästä, ilmestyvät hetken jälkeen samassa järjestyksessä toisessa päässä



GET /index.html HTTP/1.0

Vertaus puheluun

- Puhelin ↔ soketti
- Puhelimen numero ↔ soketin osoite
- Puhelimesta avataan yhteys toiseen puhelimeen käyttämällä vastaanottavan puhelimen numeroa
↔ Soketista avataan yhteys toiseen käyttämällä vastaanottavan soketin osoitetta
- Listen-soketti toimii yrityksen puhelinvaihteen tavoin
 - sisääntulevat puhelut kytketään jollekin asiakaspalvelijalle (↔ toinen soketti), jotta voidaan ottaa vastaan seuraava puhelu



Socket API:n kutsut

- Alla tärkeimmät kutsut C-kielisestä API:sta, muissa ohjelmointikielissä samankaltaisia

socket() luo uusi soketti annetulla tyypillä

bind() sido soketti porttiin ja paikalliseen IP osoitteeseen
(koneella voi olla monta osoitetta)

listen() kytke soketti kuuntelutilaan

connect() avaa yhteys (TCP)

accept() ota vastaan yhteyspyyntö kuuntelutilassa olevalta soketilta

send(), recv()/write(),read() tiedon lähettäminen ja vastaanottaminen

close() sulje yhteys (TCP) ja vapauta resursseja

setsockopt() muuta socketin asetuksia, "socketin tuunaus"

Esimerkki: työaseman soketit

```
sudo netstat -tuwlanp4
```

Active Internet connections (servers and established)

Proto	Local Address	Foreign Address	State	PID/Program name
tcp	0.0.0.0:2628	0.0.0.0:*	LISTEN	2675/0
tcp	127.0.0.1:80	0.0.0.0:*	LISTEN	2659/boa
tcp	0.0.0.0:22	0.0.0.0:*	LISTEN	2757/sshd
tcp	127.0.0.1:631	0.0.0.0:*	LISTEN	9309/cupsd
tcp	127.0.0.1:6010	0.0.0.0:*	LISTEN	19966/sshd: nx@nott
tcp	0.0.0.0:7005	0.0.0.0:*	LISTEN	1924/nxagent
tcp	127.0.0.1:32811	127.0.0.1:5005	ESTABLISHED	20184/nxssh
tcp	130.233.193.40:38189	130.233.192.7:22	ESTABLISHED	8832/ssh
tcp	127.0.0.1:22	127.0.0.1:53185	ESTABLISHED	1848/sshd: ctl [pri
tcp	130.233.193.40:44826	128.214.113.133:22	ESTABLISHED	7256/ssh
tcp	127.0.0.1:53185	127.0.0.1:22	ESTABLISHED	1847/nxssh
tcp	130.233.193.40:56205	130.233.192.7:22	ESTABLISHED	28506/ssh
tcp	127.0.0.1:5005	127.0.0.1:32811	ESTABLISHED	1924/nxagent
tcp	130.233.193.40:41422	74.125.39.100:80	ESTABLISHED	8810/firefox
tcp	130.233.193.40:50029	130.233.192.7:22	ESTABLISHED	24755/ssh
udp	0.0.0.0:41190	0.0.0.0:*		3106/avahi-daemon:
udp	0.0.0.0:5353	0.0.0.0:*		3106/avahi-daemon:
udp	172.16.111.1:123	0.0.0.0:*		5744/ntpd
udp	192.168.235.1:123	0.0.0.0:*		5744/ntpd
udp	130.233.193.40:123	0.0.0.0:*		5744/ntpd
udp	127.0.0.1:123	0.0.0.0:*		5744/ntpd
udp	0.0.0.0:123	0.0.0.0:*		5744/ntpd
raw	0.0.0.0:1	0.0.0.0:*	7	9556/vmnet-natd

HTTP:n toteutus soketeilla

- Viime luennolla nettiselailu, joka rakentui HTTP-viestien lähettämiseen ja vastaanottamiseen
- Nettisivut nimetty URL:llä (Uniform resource locator)
 - Muotoa protokolla://palvelin/sivunimi, esim
`http://www.aalto.fi/index.html`
- Tärkein viesti GET sivunimi → sivun sisältö HTML-muodossa
- Miten toteuttaa HTTP sokettien avulla?
- Perusosa nettiselaimen toteutuksesta

HTTP

- Yksinkertainen kaava (versio 1.0 HTTP-protokollasta)
- Halutaan hakea sivu `http://n:p/y`
 - `n`=palvelinkoneen IP, `p`=HTTP-palvelun portti, `y` sivu
 - esim `http://130.233.225.254:80/index.html`
 - Yleensä nettiosoitteessa palvelimen nimi eikä `n:p`-tyylistä osoitetta, palataan tähän hetken päästä!
- 1. Avataan TCP sokettiyhteys osoitteeseen `n`, portilla `p`
- 2. Lähetetään sivupyyntö
- 3. Luetaan vastausta (tilakoodi, sivun sisältö)
- 4. Suljetaan yhteys
- Tämän jälkeen näytetään sivun sisältö jos tilakoodi näyttää onnistunutta hakua, muuten näytetään virheilmoitus

HTTP viestien koodaus

- Miten koodataan HTTP-viesti verkon yli?

Pyynnön muoto

Yhteyden kesto {

```
metodi resurssi protokolla/versio↵
otsake1: arvo1↵
otsake2: arvo2↵
↵
datadatadata....
```

Esimerkki

```
GET /index.html HTTP/1.0↵
User-Agent: Mozilla/5.0 (Linux i686; ...↵
Accept: text/html,application/xhtml+xml↵
Accept-Encoding: gzip, deflate↵
Accept-Charset: ISO-8859-1,utf-8↵
↵
```

↵ = Rivinvaihtomerkki
(arvo lukuna on 10)

Tässä ei lähetetty
dataa pyynnön
mukana

HTTP-viestien koodaus

- Miten koodataan HTTP-viesti verkon yli?

Vastauksen muoto

protokolla koodi selitys

otsake1: arvo1

otsake2: arvo2

datadatadata....

Esimerkki

```
HTTP/1.1 200 OK↵
Date: Thu, 28 Jan 2010 13:33:53 GMT ↵
Server: Apache/1.3.41 (Unix) PHP/4.4.7 ↵
Content-Length: 4242↵
Content-Type: text/html; charset=iso-8859-1 ↵
↵
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
:
:
:
```

Esimerkki HTTP-liikenteestä

----- Read from: 127.0.0.1:36452 -----

GET / HTTP/1.1

Host: localhost:8001

User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.17) Gecko/2010010604 Ubuntu/9.04 (jaunty)

Firefox/3.0.17

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Pragma: no-cache

Cache-Control: no-cache

----- Read from: 192.0.32.10:80 -----

HTTP/1.1 200 OK

Date: Mon, 01 Feb 2010 17:22:04 GMT

Server: Apache/2.2.3 (CentOS)

Last-Modified: Tue, 15 Nov 2005 13:24:10 GMT

ETag: "b300b4-1b6-4059a80bfd280"

Accept-Ranges: bytes

Content-Length: 438

Connection: close

Content-Type: text/html; charset=UTF-8

<HTML>

<HEAD>

<TITLE>Example Web Page</TITLE>

</HEAD>

<body>

<p>You have reached this web page by typing "example.com";

"example.net";

or "example.org" into your web browser.</p>

<p>These domain names are reserved for use in documentation and are not available for registration. See RFC 2606, Section 3.</p>

Selaimen sivuhaun toteutus*

```
public static boolean httpGet(String ip, int port, String resource,
    StringBuilder otsakkeet, StringBuilder data) throws IOException {
    // Luo paikallinen soketti
    Socket socket = new Socket();
    // Soketin osoite 127.0.0.1:55555
    InetSocketAddress localAddress = new InetSocketAddress("127.0.0.1", 55555);
    socket.bind(localAddress);
    // Avaa yhteys nettipalvelimeen
    socket.connect(new InetSocketAddress(ip, port));
    print(socket, "GET " + resource + " HTTP/1.0\n");
    print(socket, "User-Agent: jotinetti/1.0\n");
    print(socket, "Accept: text/html\n");
    print(socket, "\n");
    //socket.getOutputStream().close();
    String statusLine = readline(socket);
    if (!statusLine.startsWith("HTTP/1.0 200") &&
        !statusLine.startsWith("HTTP/1.1 200")) {
        return false; // Virhe
    }
    // Lue otsakkeet
    for (String line=null; (line=readline(socket)) != null &&
        !line.equals("\n");) {
        otsakkeet.append(line);
    }
    // Lue data-osuus
    for (String line=null; (line=readline(socket)) != null;) {
        data.append(line);
    }
    return true;
}
```

*) Hyvin rajoitettu mutta toimii perustapauksessa

Missä oltiinkaan?

Lähtevä viesti

Vastaanottaja:
www.cse.tkk.fi
Viesti:
GET /fi/index.html

Saapuva viesti

Vastaanottaja:
selain
Viesti:
<html>Hello world...

SELAIMEN RAJAPINTA:
VIESTIN LÄHETYS JA VASTAANOTTO

Siirtokerros: TCP, UDP, ...

Verkkokerros: IPv4, IPv6

Linkkikerros: Ethernet, MPLS,
WLAN, GPRS ...

```
public static boolean httpGet(String ip, int port, String resource,
    StringBuilder otsakkeet, StringBuilder data) throws IOException {
    // Luo paikallinen soketti
    Socket socket = new Socket();
    // Soketin osoite 127.0.0.1:8080
    InetAddress localAddress = new InetAddress("127.0.0.1", 8080);
    socket.bind(localAddress);
    // Avaa yhteys nettipaivalimeen
    socket.connect(new InetAddress(ip, port));
    int(socket, "GET " + resource + " HTTP/1.0\n");
    print(socket, "User-Agent: jotinetti/1.0\n");
    print(socket, "Accept: text/html\n");
    print(socket, "\n");
    //socket.getOutputStream().close();
    String statusLine = readLine(socket);
    if (!statusLine.startsWith("HTTP/1.0 200") &&
        !statusLine.startsWith("HTTP/1.1 200")) {
        return false; // Virhe
    }
    // Lue otsakkeet
    for (String line=null; (line=readLine(socket)) != null &&
        !line.equals("\n");) {
        otsakkeet.append(line);
    }
    // Lue data-osuus
    for (String line=null; (line=readLine(socket)) != null;) {
        data.append(line);
    }
    return true;
}
```

Protokolla (yhteyskäytäntö)

- Viestit solmujen (koneiden) välillä noudattavat määrättyä kaavaa jota kutsutaan yhteyskäytännöksi, eli **protokollaksi**
- Esim: selaimen ja nettipalvelimen väliset viestit noudattavat HTTP-protokollaa
- Protokollassa voidaan määritellä
 - Viestien muuttaminen biteiksi verkkolähetystä varten
 - Eri viestit ja niiden merkitykset
 - Viestien järjestys ja suhde toisiinsa
- Muita esimerkkejä protokollista tällä luennolla: DNS, SMTP



In Western Society, a handshake is part of the standard protocol when making a new acquaintance

Protokolla

- Hyvä protokolla on riippumaton toteutuksesta, jolloin voi olla monta toteutusta
 - Esim. nettipalvelinohjelmistoa saa monelta eri toimittajalta, HTTP on yhteinen ”kieli”
- Internet-protokollien standardointi
 - Internet Engineering Task Force (IETF)
 - Word Wide Web Consortium (W3C)
- Hyvin toimivan protokollan suunnittelu on haasteellista
 - Onko kaikissa tilanteissa määritelty miten jatketaan?
 - Edistyykö protokolla aina?
 - Onko tietoturva kunnossa?
 - Tutkimusaiheet: formalisointi ja verifiointi



Tiedon koodaaminen (encoding)

- Verkossa siirretään tavuja (1 tavu = 8 bittiä)
- Tavuja on $2^8 = 256$ kpl (0-255)
- Verkkosiirtoa varten tieto on koodattava tavuiksi
- Esim 2: ASCII-koodaus, tavulle annetaan merkitys merkkinä
 - Esim. $65 \leftrightarrow 'A'$, $66 \leftrightarrow 'B'$,
 - Myös erikoismerkkejä kuten rivinvaihto
 - Koska tavuja on vain 256, vain yleisimmät länsieurooppalaiset merkit käytössä
 - Miten koodataan kiinaa, ääkkösiä (ääö),...?
 - Joillekin merkille monta tavua (tästä esim UTF-8-koodaus)
 - Entäs tekstin formatointi?
 - Esim. HTML!

8-bit

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0-																
1-																
2-		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7-	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8-																
9-																
A-		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
B-	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C-	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D-	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E-	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F-	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Missä

©?

%o?

ב?

fi?

Koodaus protokollassa

- Viesteissä olevaa tietoa pitää koodata ettei se riko protokollaa
- Esim HTTP:llä haetaan `http://example.com/name with space`
- Välilyönnit pitää koodata, muuten sanat "with space" tulkitaan kuuluvan seuraavaan kenttään (protokolla ja versio)

Koodataan esimerkiksi ' ' = %20 (ja %= %25):

`http://example.com/name%20with%20spaces`

HTTP-pyyntö

```
metodi resurssi protokolla/versio ↵
otsake1: arvo1 ↵
otsake2: arvo2 ↵
↵
datadatadata....
```

Esimerkki

Virhetulkinta!

```
GET name with space HTTP/1.0
User-Agent: Mozilla/5.0 (Linux i686; ... ↵
....

GET name%20with%20 space HTTP/1.0
User-Agent: Mozilla/5.0 (Linux i686; ... ↵
....
```

Domain Name System (DNS)

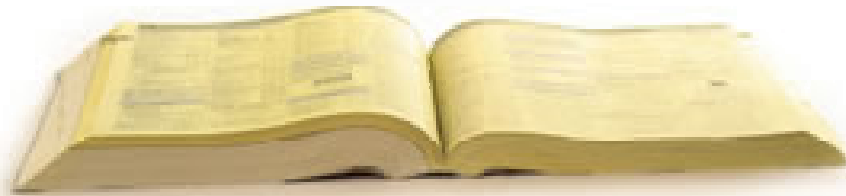
- HTTP-esimerkissä palvelin nimettiin IP osoitteella ja portilla
- Yleensä käytetään kuitenkin nettisaitin nimi, esim `www.aalto.fi`
- Nettipalvelin on yleensä portilla 80 (443 jos suojattu yhteys), joten voidaan olettaa että portti on 80 ellei sitä anneta erikseen
- Miten saadaan saittinimestä sokettien käyttämä IP-osoite? (`www.aalto.fi` → `130.233.224.254`)
- Tähän on olemassa Domain Name System (DNS) palvelu, [RFC 1034, RFC 1035]

DNS-nimet Internetissä

- Sokset ja IP -tietoliikenneverkko tarvitsevat IP -osoitetta voidakseen reitittää verkkoliikennettä oikeaan verkkosolmuun (koneeseen)
- Numerosarjojen käyttäminen koneen nimeämiseen kuitenkin hankala ihmisille
- DNS -nimi on hierarkinen nimi (vrt sukunimi etunimi), jossa tasot erotellaan pisteellä
 - nic.fi, kosk.hut.fi, morgul.cs.hut.fi, jne.
- DNS-nimellä voidaan myös nimetä palveluita
 - Esim www-etuliitteellä nimetty on yleensä nettipalvelin
 - www.aalto.fi, www.google.com

DNS-nimet Internetissä

- DNS-haulla haetaan nimeä vastaava IP-osoite
- myös muita tietoja
- IP-osoite annetaan ns A-tietueessa
- palvelun nimen haku voi antaa vastaukseksi koneen nimen (josta seuraavalla kyselyllä saadaan IP). Tämä on ns. CNAME-tietue
- mihin palvelimeen sähköposti lähetetään (ns. MX-tietue)
- NS (name server) tietue kertoo DNS-palvelun käyttämistä DNS-palvelimista
- "Internetin puhelinluettelo"



DNS-hakuja host-kommennolla

```
host -a kosh.hut.fi
```

```
;; ANSWER SECTION:
```

```
kosh.hut.fi.          695      IN        A        130.233.228.12
```

```
Host -a www.google.com
```

```
;; ANSWER SECTION:
```

```
www.google.com.      431747 IN        CNAME    www.l.google.com.
```

```
Host -a www.l.google.com
```

```
;; ANSWER SECTION:
```

```
www.l.google.com.    300      IN        A        74.125.39.103
```

```
host -a hut.fi
```

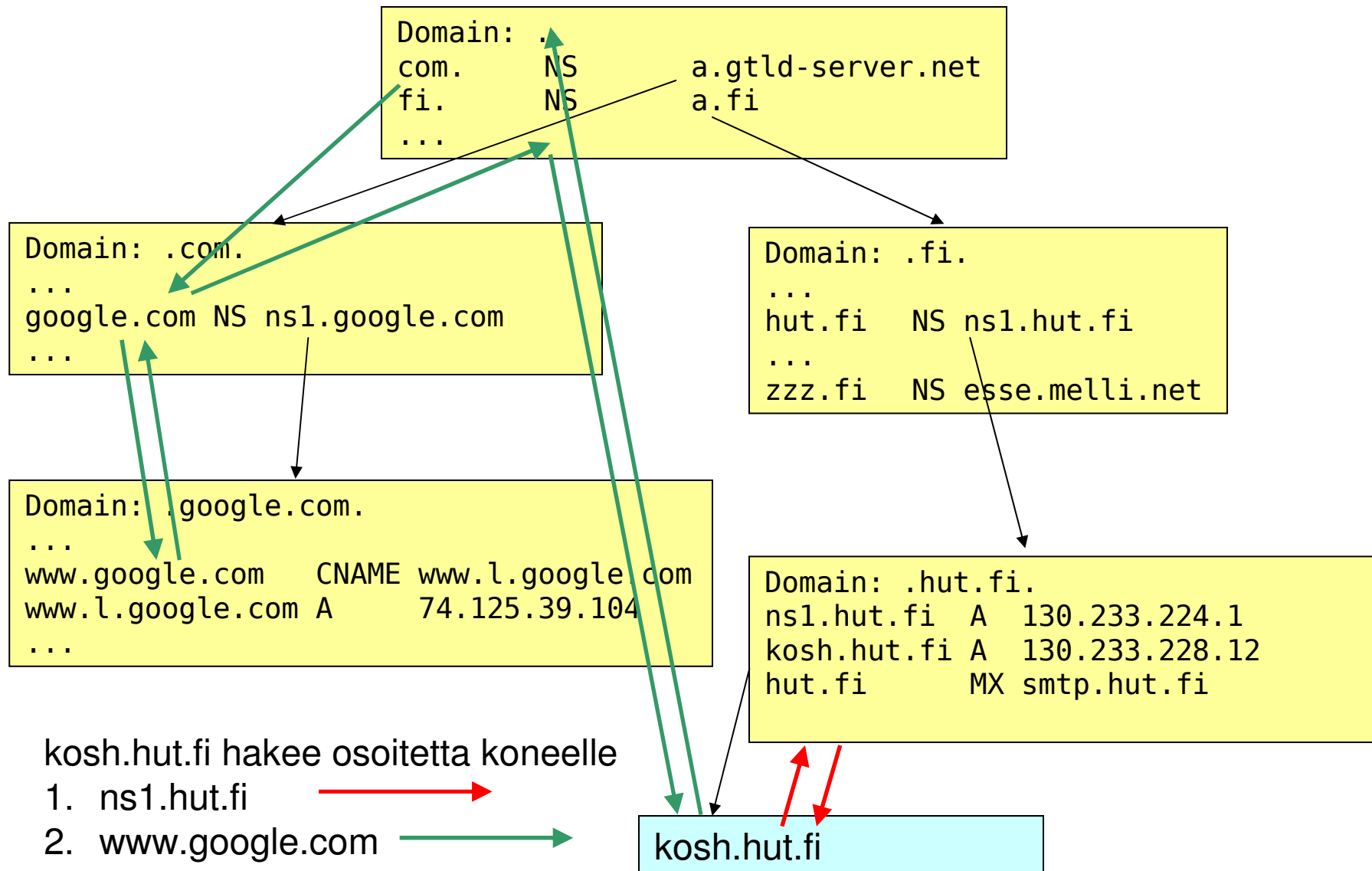
```
;; ANSWER SECTION:
```

```
hut.fi.              3032     IN        MX        9 smtp.hut.fi.  
hut.fi.              3248     IN        NS        ns1.hut.fi.  
hut.fi.              3248     IN        NS        ns2.hut.fi.  
hut.fi.              3248     IN        NS        ns-secondary.funet.fi.
```

DNS: maailmanlaajuinen tietokanta

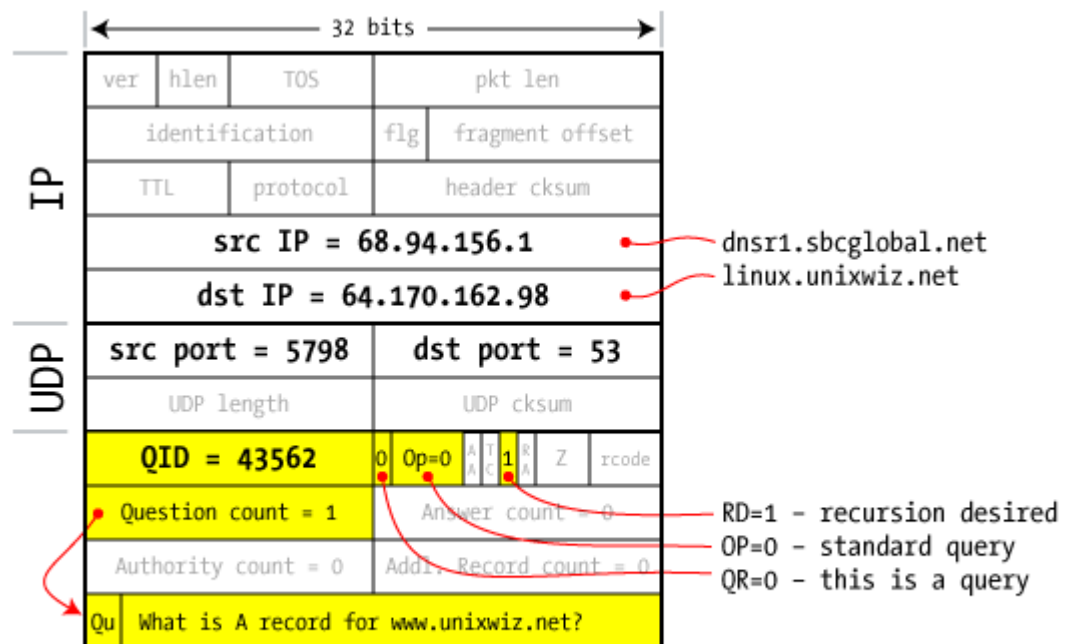
- DNS on maailmanlaajuinen tietokanta
- koostuu miljoonista palvelimista
- järjestetty puurakenteeseen sen mukaan, mitä osaa nimistä osaavat käsitellä
- ylemmän tason palvelin osaa lähettää kyselyn eteenpäin oikeaan seuraavan tason palvelimeen
- paikallinen nimipalvelin osaa oman verkon IP -osoitteet
 - Nimipalvelin sauna.cs.hut.fi tietää .cs.hut.fi -loppuisten koneiden IP:t
 - Muut kyselyt reititetään muille DNS-palvelimille
 - Jokainen kone verkossa tietää oman paikallisen nimipalvelimensa johon se lähettää DNS-kyselyt
- hakutuloksia muistetaan jotta koko hakuprosessi (seuraava kalvo) ei jouduttaisi toistamaan jokaisella haulla

DNS-hierarkia



DNS-kysely

- DNS kyselyyn käytetään UDP-soketti, eli lyhyt viesti joka voi kadota
- Kyselyviesti lähetetään paikalliselle nimipalvelimelle portille 53
- Vastaukset lähetetään takaisin kyselijän UDP-soketille
- Viestin koodauksessa käytetään yksittäisiä bittejä jotta yhteen viestiin mahtuisi mahdollisimman paljon tietoa
- DNS-viestin tarkka muoto on jatkokurssia...



Kuvan lähde: unixwiz.net

DNS-ohjelmointirajapinta

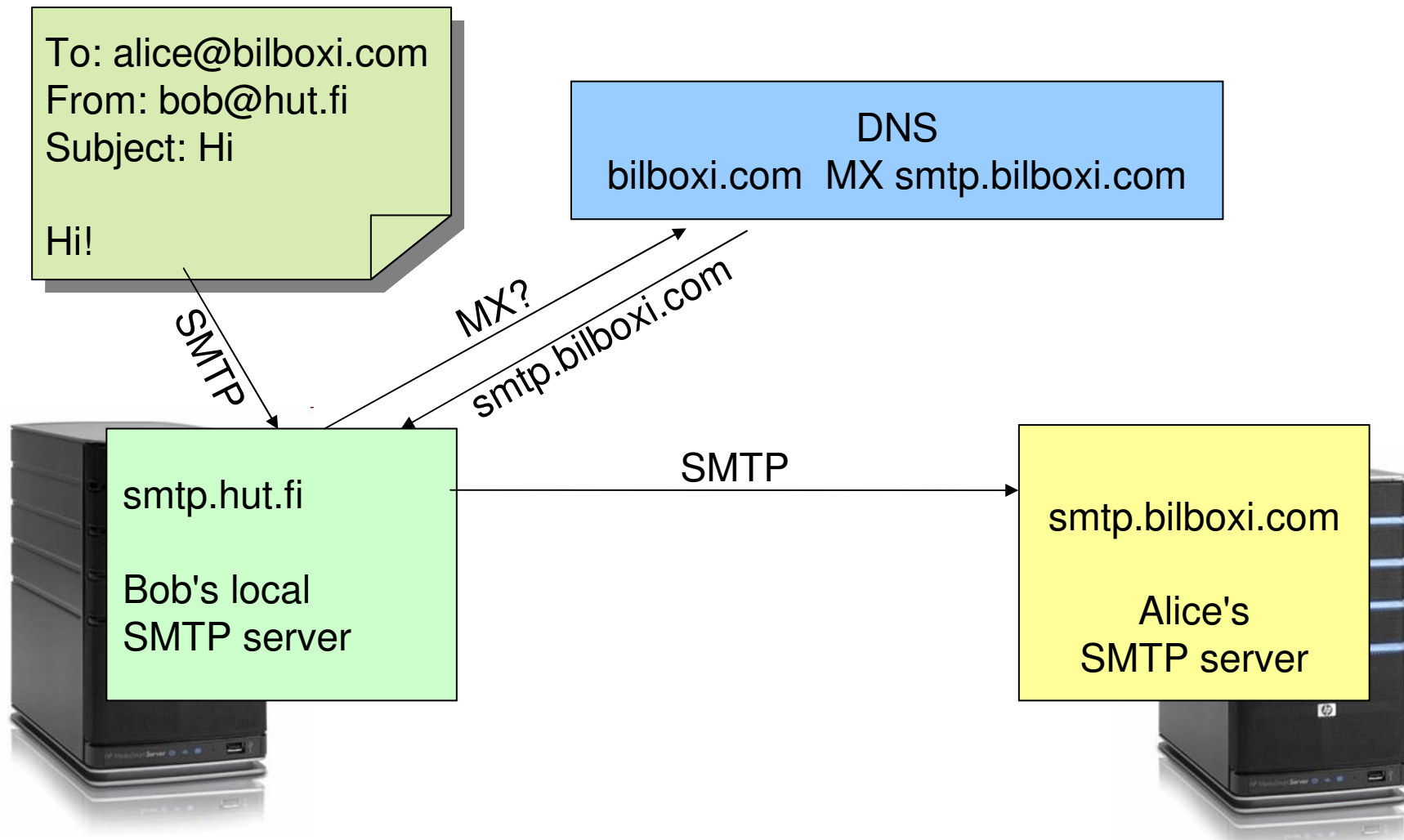
- Käyttöjärjestelmä tarjoaa ohjelmoijalle rajapinnan DNS -kyselyiden tekemiseen
- Useimmiten kyselyt tapahtuvat myös automaattisesti
 - esim. Socket API:ssa voi avata yhteyden myös DNS-nimellä, jolloin nimi → IP haku suoritetaan automaattisesti
- Esimerkki Javalla

```
Inet4Address.getAllByName("example.com")
```

Sähköposti: SMTP

- Sähköpostin lähettäminen
 - Sähköpostiohjelma avaa yhteyden paikalliseen sähköpostipalvelimeen (esim smtp.hut.fi jos TKK:n verkossa)
 - Paikallinen palvelin lähettää eteenpäin vastaanottavaan palvelimeen
 - Molemmissa tapuksissa käytetään Simple Mail Transfer Protocol (SMTP)
 - DNS:n MX tietueella selviää vastaanottavan sähköpostipalvelimen nimi

SMTP Esimerkki



SMTP-protokolla

- tekstipohjainen kuten HTTP
- käyttää TCP-sokettia, portti yleensä 25
- pitkä historia, nimellä SMTP jo 1982
- RFC 5321 (alkuperäinen RFC 821)
- sähköpostin hakemiseen sähköpostilukijaan eri protokollat
 - POP, IMAP
- webmail toimii vähän eri tavalla
 - SMTP-viesti lähtee nettipalvelimelta
 - nettipalvelin lukee viestiä ja lähettää selaimeen

SMTP-esimerkki

```
220 smtp.bilboxi.com ESMTP Postfix
HELO smtp.hut.fi
250 Hello smtp.hut.fi, I am glad to meet you
MAIL FROM:<bob@hut.fi>
250 Ok
RCPT TO:<alice@bilboxi.com>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "Bob Example" <bob@hut.fi>
To: Alice Example <alice@bilboxi.com>
Date: Tue, 15 Jan 2010 16:02:43 -0500
Subject: Hi

Hi!
.
250 Ok: queued as 12345
QUIT
221 Bye
{The server closes the connection}
```

Yhteenveto

- Sokset Internet-tiedonsiirron perustana
- Protokolla määrittelee kommunikaation muoto ja merkitys
- Tutustuttiin seuraaviin protokolleihin
 - HTTP (TCP-pohjainen)
 - DNS (UDP-pohjainen)
 - SMTP (TCP-pohjainen)