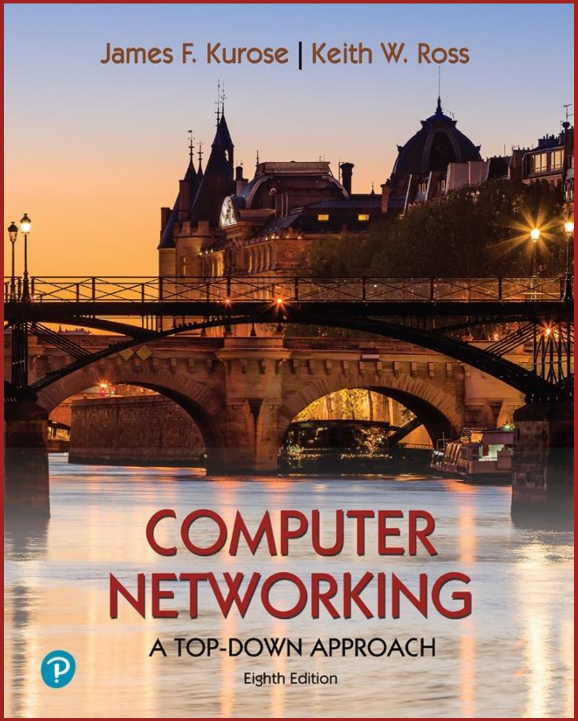
Communication Networks
W. Tavernier

Chapter 2
Application Layer



Computer Networking: A Top-Down Approach
8th Edition, 2020, Pearson,
James F. Kurose, Keith W. Ross

Chapter 2 outline

2.1 Principles of network applications

2.2 Web and HTTP, and **FTP**

2.3 electronic mail

- SMTP, POP3, IMAP

2.4 DNS

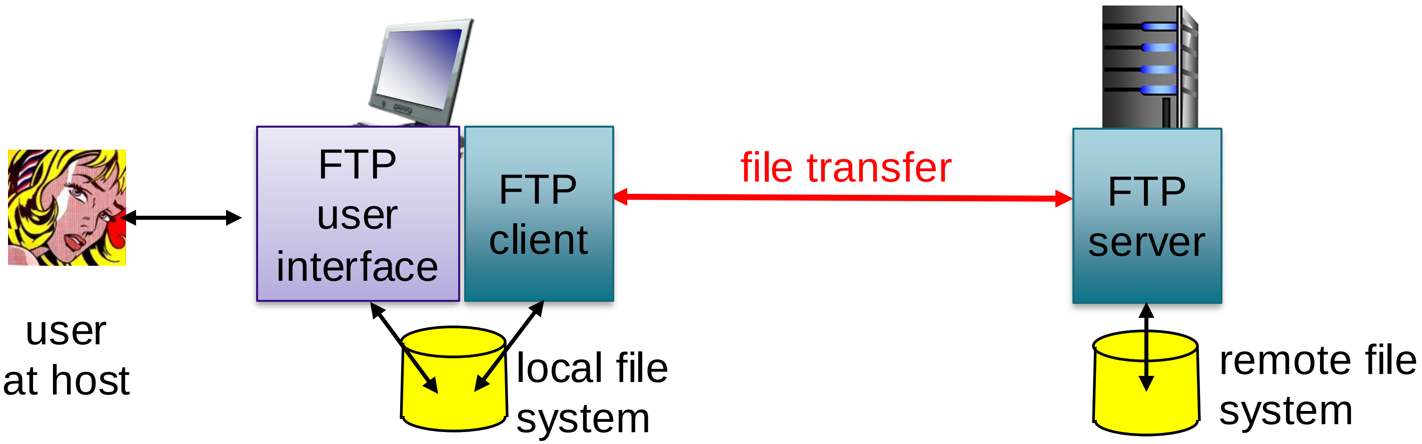
2.5 P2P applications

[**2.6** video streaming and content distribution networks]

2.7 Socket programming with UDP and TCP

FTP: the file transfer protocol

per file nieuwe tunnel aanmaken



- transfer file to/from remote host
- client/server model
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host
- ftp: RFC 959
- ftp server: port 21 (control), port 20 (data)

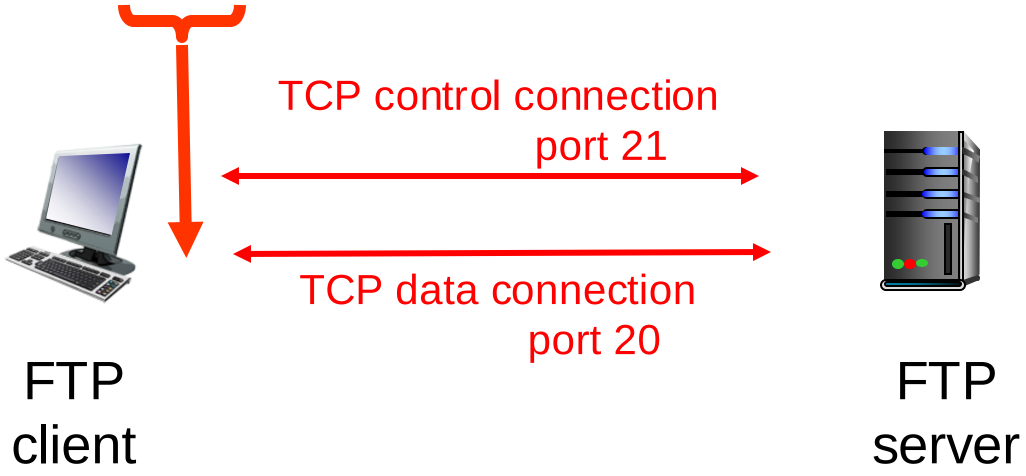
FTP: separate control, data connections

- FTP client contacts FTP server at port 21, specifying TCP as transport protocol

auth is meerwaarde van ftp, http is voor iedereen bedoed

- Client obtains authorization over control connection
- Client browses remote directory by sending commands over control connection.
- When server receives a command for a file transfer, the server opens a TCP data connection to client
- After transferring file, server closes data connection
- Server opens a new TCP data connection to transfer another file.

USER, PASS, PWD, CWD, LIST, PORT, RETR, STOR, QUIT



TCP control connection port 21

TCP data connection port 20

FTP client

FTP server

- Control connection: "out of band"
- FTP server maintains "state": current directory, earlier authentication

FTP commands, responses

Sample commands:

- sent as ASCII text over control channel
- `USER` ***username***
- `PASS` ***password***
- `LIST` return list of file in current directory
- `RETR` **filename** retrieves (gets) file
- `STOR` **filename** stores (puts) file onto remote host

Sample return codes

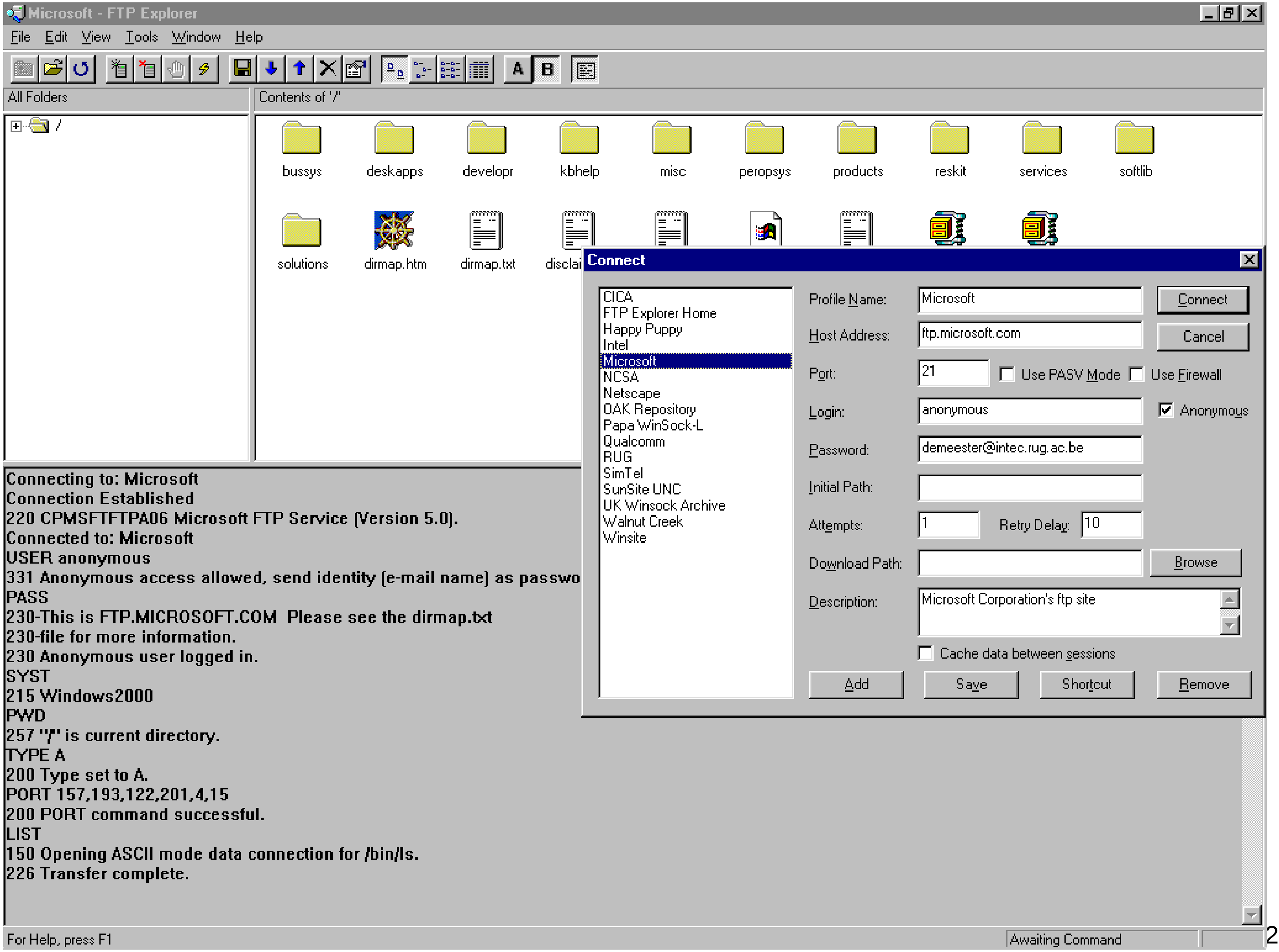
- status code and phrase (as in HTTP)
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

FTP example

```
$ telnet ftp.microsoft.com 21
220 CPMSFTFTPA06 Microsoft FTP Service (Version 5.0).
Connected to: Microsoft
USER anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
PASS
230-This is FTP.MICROSOFT.COM  Please see the dirmap.txt
230-file for more information.
230 Anonymous user logged in.
SYST
215 Windows2000
PWD
257 "/" is current directory.
TYPE A
200 Type set to A.
PORT 157,193,122,155,4,18
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for /bin/ls.
226 Transfer complete.
```

```
CWD /products/
250 CWD command successful.
TYPE A
200 Type set to A.
PORT 157,193,122,155,4,19
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for /bin/ls.
226 Transfer complete.
```

FTP example



CICA
FTP Explorer Home
Happy Puppy
Intel
Microsoft
NCSA
Netscape
OAK Repository
Papa WinSock-L
Qualcomm
RUG
SimTel
SunSite UNC
UK Winsock Archive
Walnut Creek
Winsite

Profile Name: Microsoft
Host Address: ftp.microsoft.com
Port: 21 ☐ Use PASV Mode ☐ Use Firewall
Login: anonymous ☑ Anonymous
Password: demeester@intec.rug.ac.be
Initial Path:
Attempts: 1 Retry Delay: 10
Download Path:
Description: Microsoft Corporation's ftp site
☐ Cache data between sessions

```
Connecting to: Microsoft
Connection Established
220 CPMSFTFTPA06 Microsoft FTP Service (Version 5.0).
Connected to: Microsoft
USER anonymous
331 Anonymous access allowed, send identity (e-mail name) as passwo
PASS
230-This is FTP.MICROSOFT.COM  Please see the dirmap.txt
230-file for more information.
230 Anonymous user logged in.
SYST
215 Windows2000
PWD
257 "/" is current directory.
TYPE A
200 Type set to A.
PORT 157,193,122,201,4,15
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for /bin/ls.
226 Transfer complete.
```

Chapter 2 outline

2.1 Principles of network applications

2.2 Web and HTTP

2.3 electronic mail

- **SMTP, POP3, IMAP**

2.4 DNS

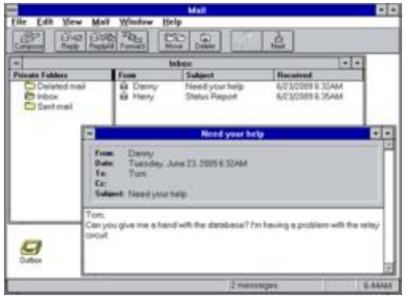
2.5 P2P applications

[**2.6** video streaming and content distribution networks]

2.7 Socket programming with UDP and TCP

History of email

ARPANET



Mail

Private Folders
- Deleted mail
- Inbox
- Sent mail

From	Subject	Received
Danny	Need your help	6/23/2009 6:32AM
Harry	Status Report	6/23/2009 6:35AM

Need your help

From: Danny
Date: Tuesday, June 23, 2009 6:32AM
To: Tom
Cc:
Subject: Need your help

Tom,
Can you give me a hand with the database? I'm having a problem with the relay circuit.

2 messages 6:44AM

1971: Invention of email

1976: The Queen's first email

1978: First spam email

1982: SMTP

1988: MSMail (Outlook v0)

1991: First email from space

1992: MIME

1993: Webmail

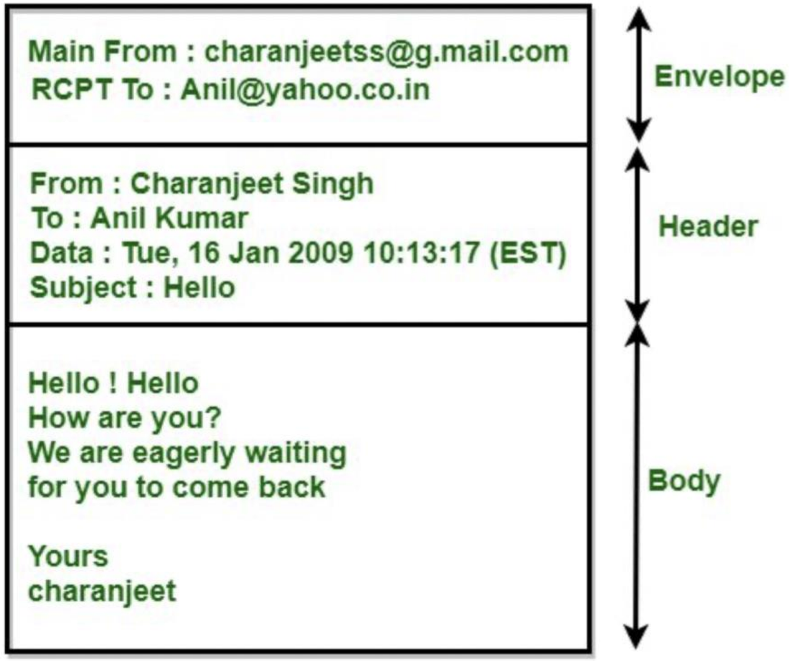
2004: Gmail

> 2010: fighting spam

"Hello Earth! Greetings from the STS-43 Crew. This is the first AppleLink from space. Having a GREAT time, wish you were here,... send cryo and RCS! Hasta la vista, baby,...we'll be back!"

E-mail protocols

- Collection of different protocols for sending, forwarding and downloading/viewing e-mails
- E-mail protocols are text-based
  - All e-mail content should be text (including HTML support)
  - Attachments are possible through MIME extensions converting binary formats to text



Main From : charanjeetss@g.mail.com
RCPT To : Anil@yahoo.co.in

From : Charanjeet Singh
To : Anil Kumar
Data : Tue, 16 Jan 2009 10:13:17 (EST)
Subject : Hello

Hello ! Hello
How are you?
We are eagerly waiting
for you to come back

Yours
charanjeet

Electronic mail protocols/formats

(E)SMTP = (Extended) Simple Mail Transfer Protocol :

transfer e-mail message from UA to MTA or between MTAs

POP3 = Post Office Protocol 3

retrieve e-mail from MTA

haalt de mails van een server shit, vroeger niet altijd verbonden dus if online
-> zit er iets in me postvak een haal ze op als er zijn

IMAP = Internet Message Access Protocol

advanced retrieve of e-mail from MTA
intelligence in MTA (also advanced database structure)

RFC 822 (message format) ➔ not a protocol !

format of a plain text message

MIME (Multipurpose Internet Mail Extensions) ➔ not a protocol !

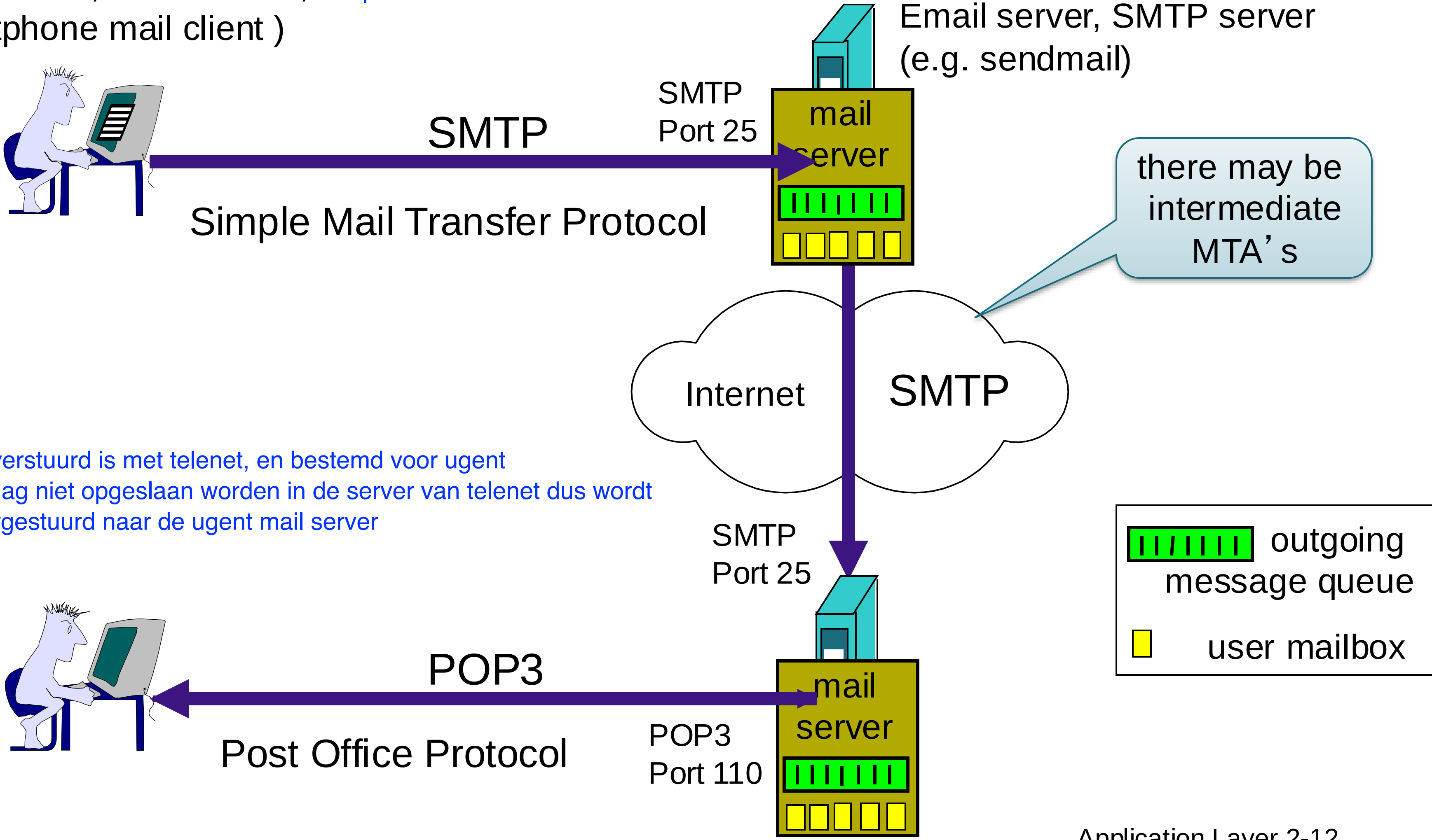
format and coding of non plain text messages (e.g. JPEG, Word)
and split into several sub-messages (e.g. attachments)

Electronic mail

User agent (UA)
Email client
(e.g. Outlook, Thunderbird,
smartphone mail client)

via / met smtp -> naar een eigenmail service
pusht de email

Message Transfer Agent (MTA)
Email server, SMTP server
(e.g. sendmail)



SMTP

SMTP
Port 25

Simple Mail Transfer Protocol

mail
server

there may be
intermediate
MTA’s

Internet

SMTP

als verstuurd is met telenet, en bestemd voor ugent
-> mag niet opgeslaan worden in de server van telenet dus wordt
doorgestuurd naar de ugent mail server

SMTP
Port 25

🟩 outgoing
message queue

🟨 user mailbox

POP3

Post Office Protocol

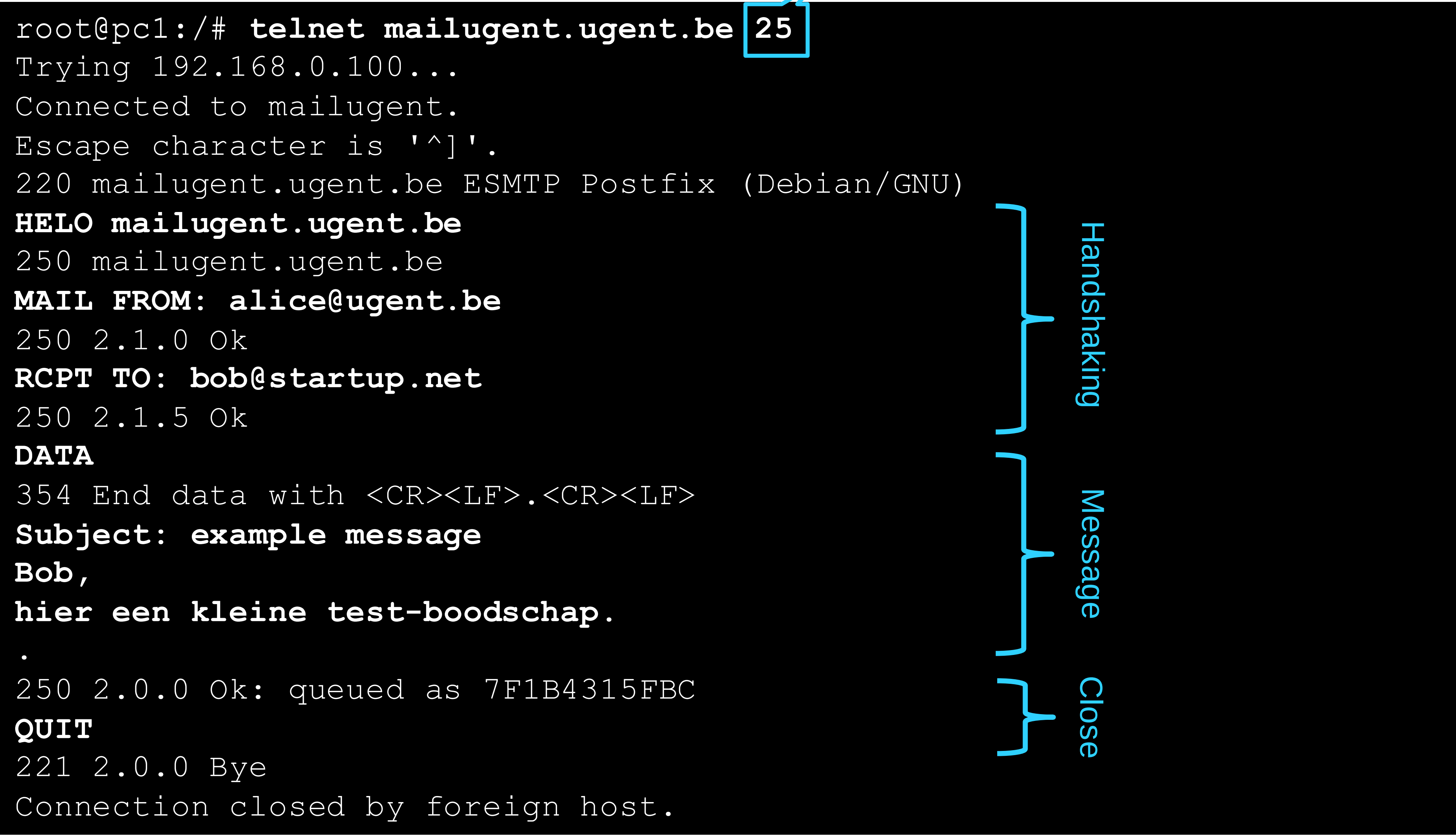
POP3
Port 110

mail
server

Submitting and forwarding e-mail

- The Simple Mail Transfer Protocol (SMTP) is used for submitting or forwarding an e-mail to an MTA
- The SMTP server will:
  1. Verify if the submitting party has necessary permissions (username and password)
  2. Receive e-mail and put it in its outgoing queue
  3. Perform a spam check (e.g. using AI matching algorithms)
  4. Forward the e-mail towards the next MTA by checking DNS record matching the mail domain of receiver (cfr. DNS section)

Electronic mail : SMTP [RFC 2821]



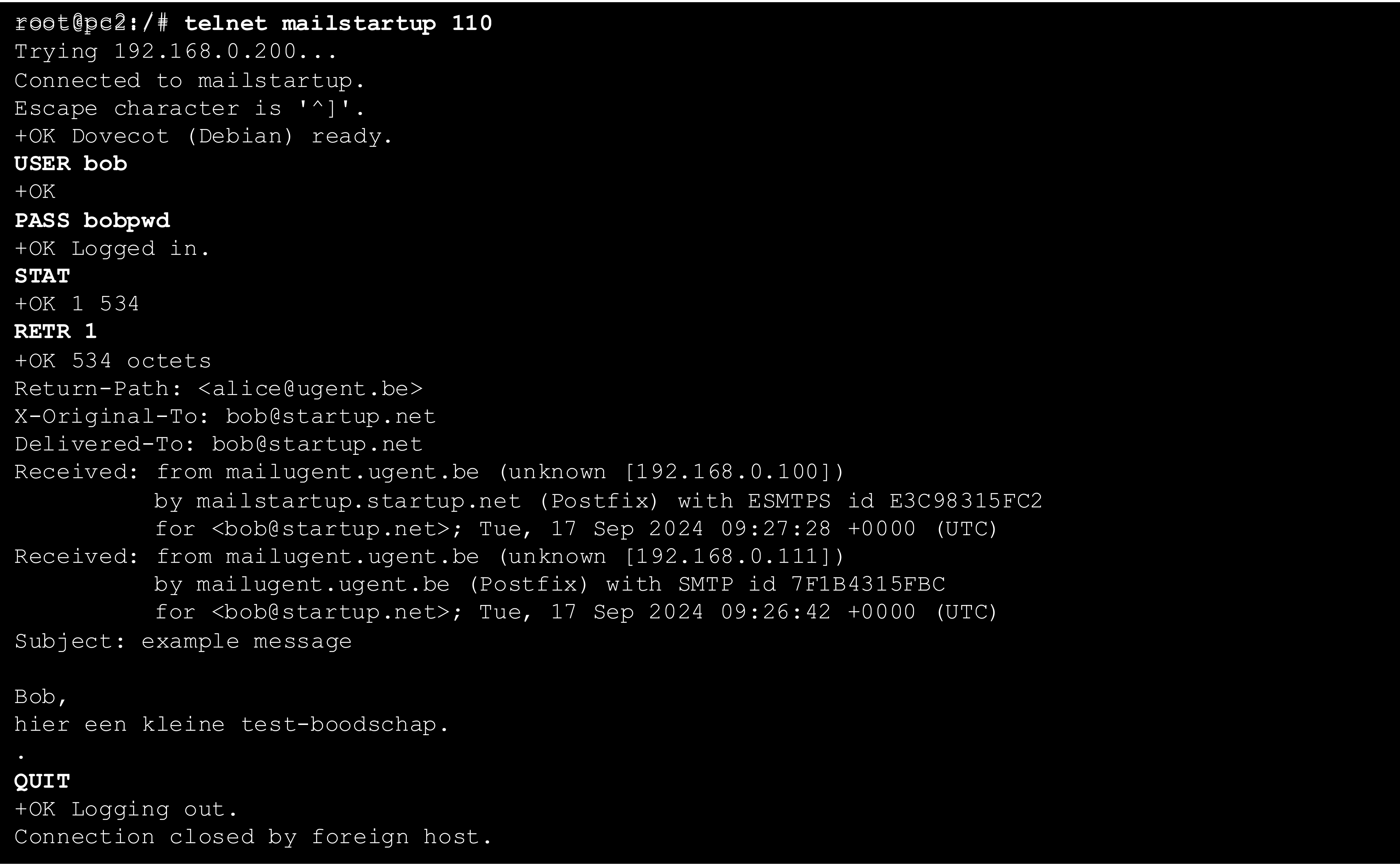
```
root@pc1:/# telnet mailugent.ugent.be 25
Trying 192.168.0.100...
Connected to mailugent.
Escape character is '^]'.
220 mailugent.ugent.be ESMTP Postfix (Debian/GNU)
HELO mailugent.ugent.be
250 mailugent.ugent.be
MAIL FROM: alice@ugent.be
250 2.1.0 Ok
RCPT TO: bob@startup.net
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: example message
Bob,
hier een kleine test-boodschap.
.
250 2.0.0 Ok: queued as 7F1B4315FBC
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

ESMTP adds secure authentication to SMTP amongst other features (RFC 2821)
Note: you can redo this experiment in Kathará.

Accepting and receiving e-mails

- Receiving email servers accept messages if:
  - The server is configured for the domain of the receiver
  - If the receiver username is known (has a registered mailbox)
  - Checked for spam amongst other security checks
- Once accepted, the e-mail is stored in a local database or filesystem, ready to be read by users requesting their e-mails
  - Using the Post Office Protocol (POPv3) or the Internet Access Protocol (IMAP) through:
    - a mail client (e.g., Outlook or Thunderbird)
    - a webserver having access to the mailbox (webmail)

Electronic mail : POP3



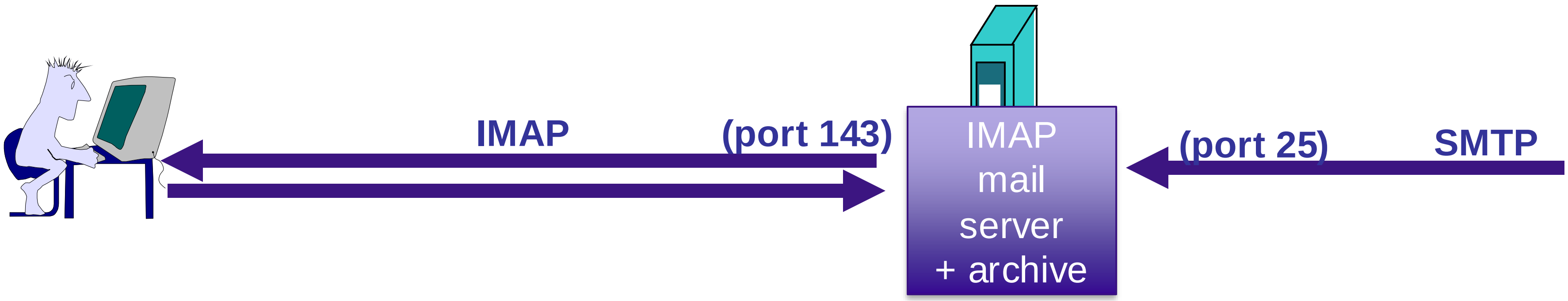
```
root@pc2:/# telnet mailstartup 110
Trying 192.168.0.200...
Connected to mailstartup.
Escape character is '^]'.
+OK Dovecot (Debian) ready.
USER bob
+OK
PASS bobpwd
+OK Logged in.
STAT
+OK 1 534
RETR 1
+OK 534 octets
Return-Path: <alice@ugent.be>
X-Original-To: bob@startup.net
Delivered-To: bob@startup.net
Received: from mailugent.ugent.be (unknown [192.168.0.100])
          by mailstartup.startup.net (Postfix) with ESMTPS id E3C98315FC2
          for <bob@startup.net>; Tue, 17 Sep 2024 09:27:28 +0000 (UTC)
Received: from mailugent.ugent.be (unknown [192.168.0.111])
          by mailugent.ugent.be (Postfix) with SMTP id 7F1B4315FBC
          for <bob@startup.net>; Tue, 17 Sep 2024 09:26:42 +0000 (UTC)
Subject: example message

Bob,
hier een kleine test-boodschap.
.
QUIT
+OK Logging out.
Connection closed by foreign host.
```

POP3 (as HTTP) : pull protocol <> SMTP : push protocol

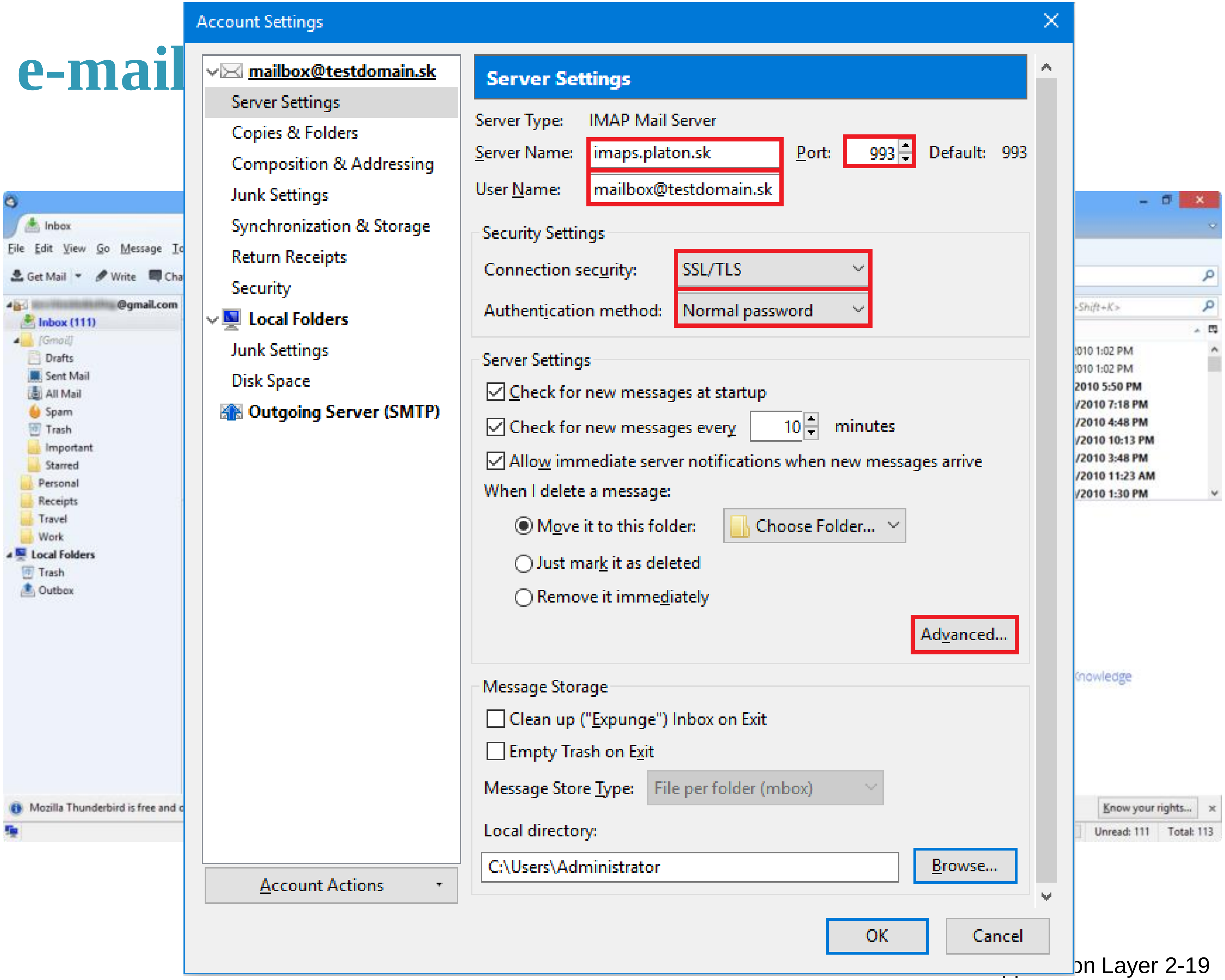
IMAP (Internet Message Access Protocol)

- Keep all messages in one place: the server
- Allows user to organize messages in folders
- IMAP keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name



Web based e-mail access : HTTP protocol – e.g. gmail.com

e-mail



Account Settings

mailbox@testdomain.sk
- Server Settings
- Copies & Folders
- Composition & Addressing
- Junk Settings
- Synchronization & Storage
- Return Receipts
- Security

Local Folders
- Junk Settings
- Disk Space

Outgoing Server (SMTP)

Account Actions

Server Settings

Server Type: IMAP Mail Server

Server Name: imaps.platon.sk Port: 993 Default: 993

User Name: mailbox@testdomain.sk

Security Settings

Connection security: SSL/TLS

Authentication method: Normal password

Server Settings

☑ Check for new messages at startup

☑ Check for new messages every 10 minutes

☑ Allow immediate server notifications when new messages arrive

When I delete a message:

● Move it to this folder: Choose Folder...

○ Just mark it as deleted

○ Remove it immediately

Advanced...

Message Storage

□ Clean up ("Expunge") Inbox on Exit

□ Empty Trash on Exit

Message Store Type: File per folder (mbox)

Local directory:

C:\Users\Administrator Browse...

OK Cancel

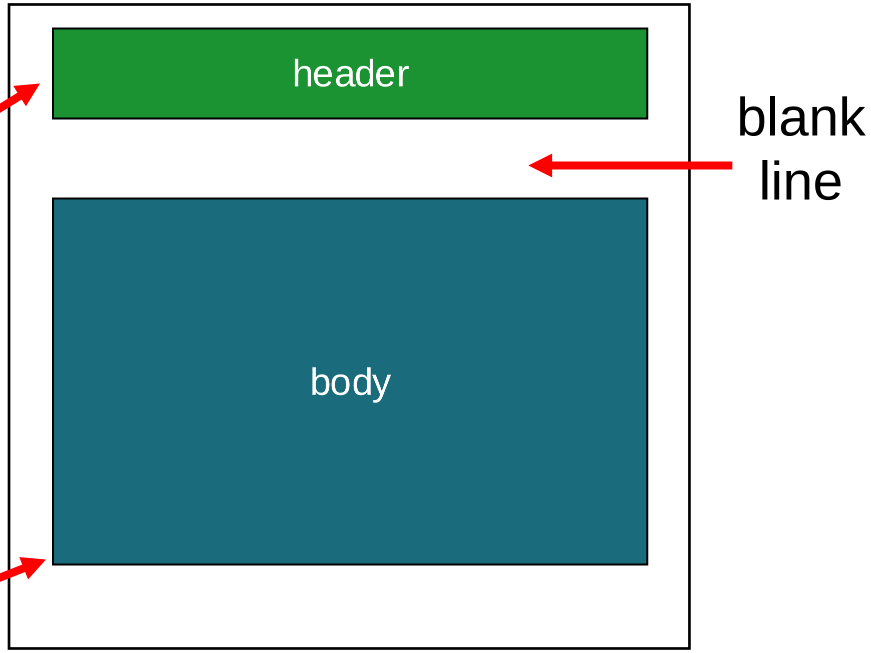
Mail message format

[SMTP: protocol for exchanging email msgs]

RFC 822: standard for text message format:

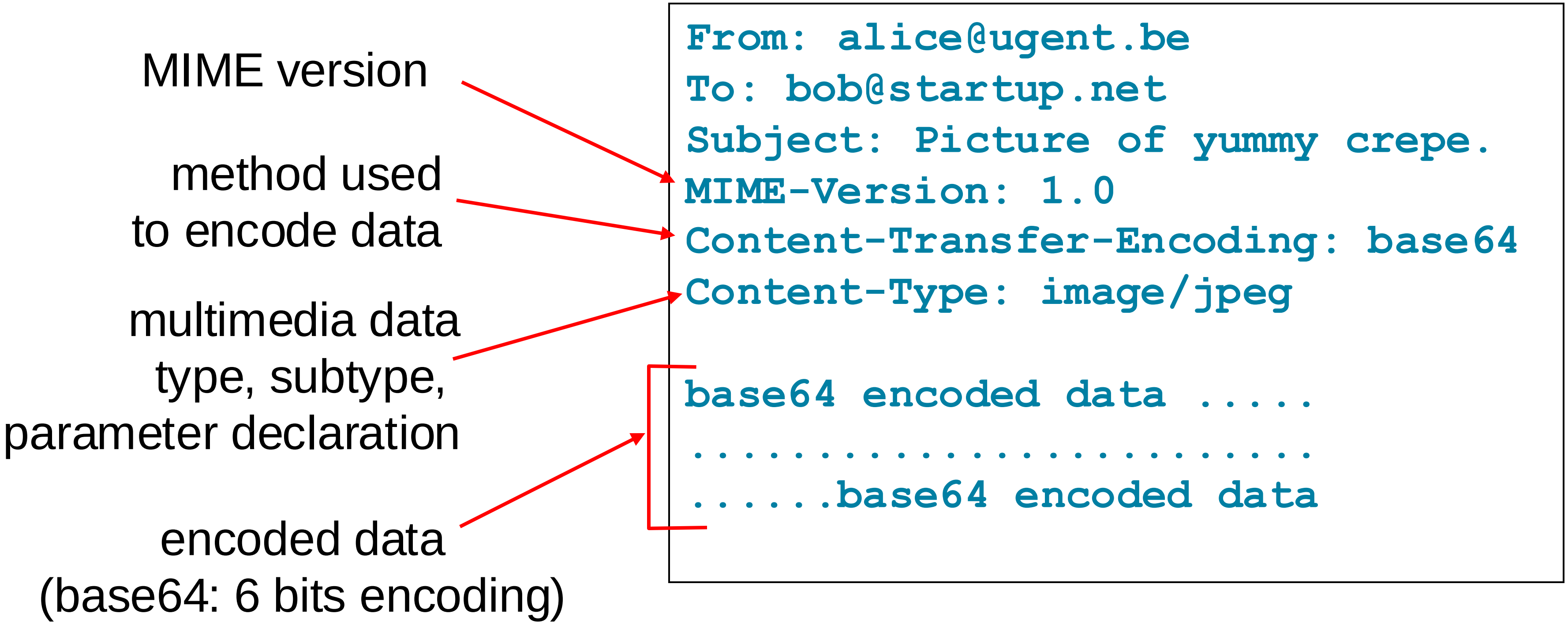
- header lines, e.g.,
  - To:
  - From:
  - Subject:

  different from SMTP commands !
- body
  - the “message”, ASCII characters only



Message format: multimedia extensions

- MIME (Multipurpose Internet Mail Extensions) : multimedia mail extension, RFC 2045, 2056
- additional lines in msg header declare MIME content type



MIME version

method used to encode data

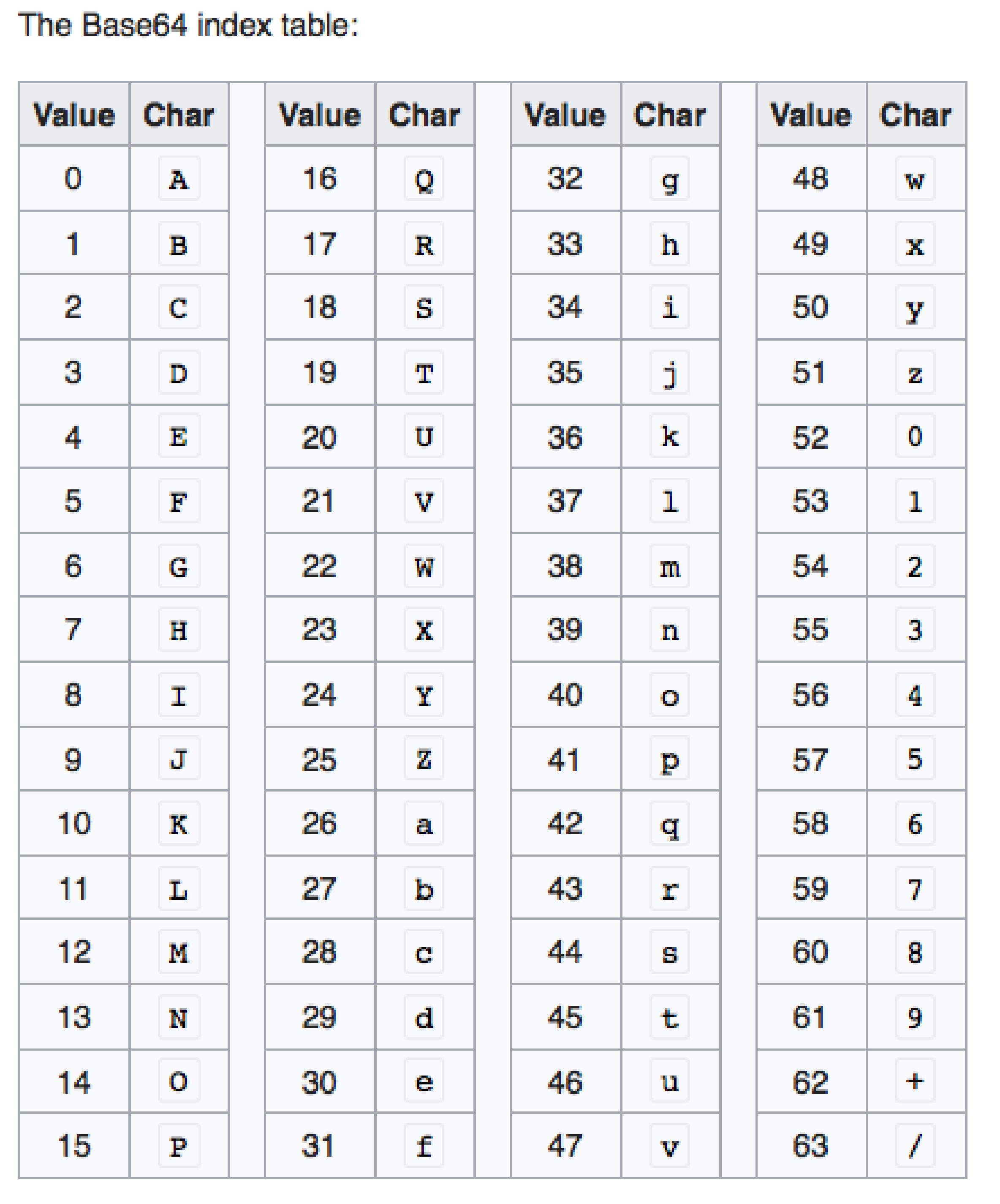
multimedia data type, subtype, parameter declaration

encoded data (base64: 6 bits encoding)

```
From: alice@ugent.be
To: bob@startup.net
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

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

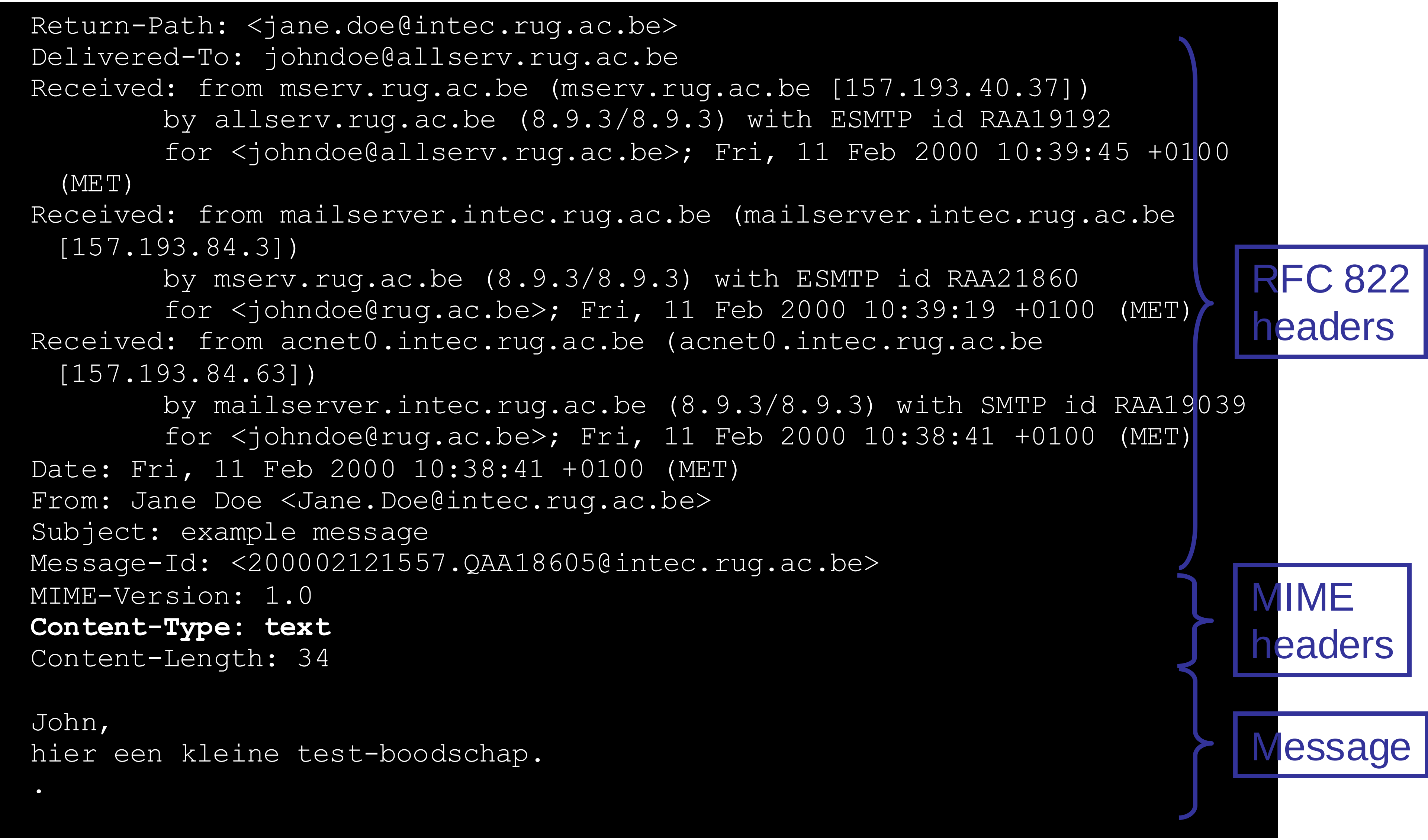
BASE64



The Base64 index table:

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

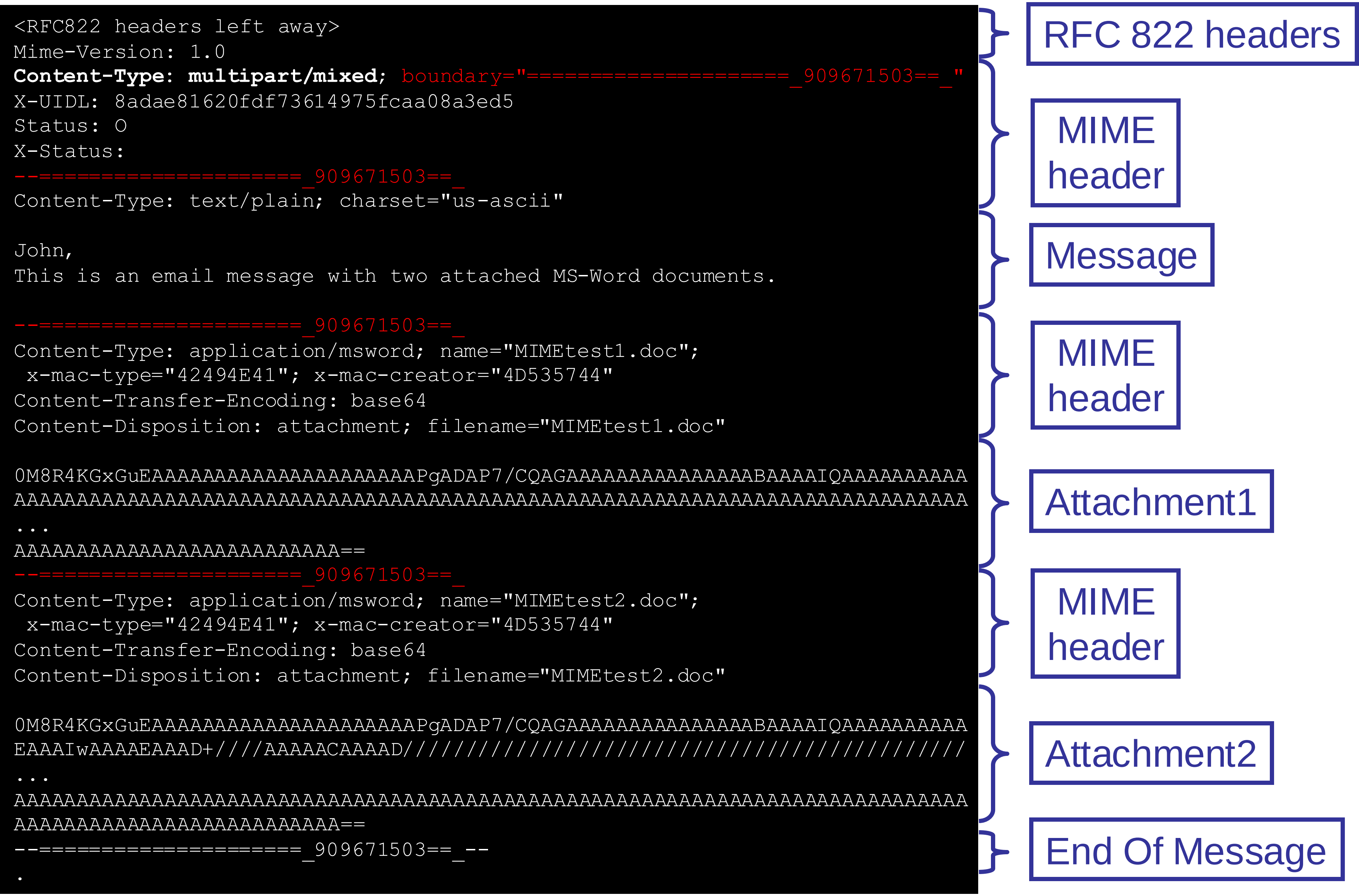
Mail message format example



```
Return-Path: <jane.doe@intec.rug.ac.be>
Delivered-To: johndoe@allserv.rug.ac.be
Received: from mserv.rug.ac.be (mserv.rug.ac.be [157.193.40.37])
        by allserv.rug.ac.be (8.9.3/8.9.3) with ESMTP id RAA19192
        for <johndoe@allserv.rug.ac.be>; Fri, 11 Feb 2000 10:39:45 +0100
  (MET)
Received: from mailserver.intec.rug.ac.be (mailserver.intec.rug.ac.be
  [157.193.84.3])
        by mserv.rug.ac.be (8.9.3/8.9.3) with ESMTP id RAA21860
        for <johndoe@rug.ac.be>; Fri, 11 Feb 2000 10:39:19 +0100 (MET)
Received: from acnet0.intec.rug.ac.be (acnet0.intec.rug.ac.be
  [157.193.84.63])
        by mailserver.intec.rug.ac.be (8.9.3/8.9.3) with SMTP id RAA19039
        for <johndoe@rug.ac.be>; Fri, 11 Feb 2000 10:38:41 +0100 (MET)
Date: Fri, 11 Feb 2000 10:38:41 +0100 (MET)
From: Jane Doe <Jane.Doe@intec.rug.ac.be>
Subject: example message
Message-Id: <200002121557.QAA18605@intec.rug.ac.be>
MIME-Version: 1.0
Content-Type: text
Content-Length: 34

John,
hier een kleine test-boodschap.
.
```

Mail message format example (with 2 attachments)



```
<RFC822 headers left away>
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary="=====================_909671503==_"
X-UIDL: 8adae81620fdf73614975fcaa08a3ed5
Status: O
X-Status:
--=====================_909671503==_
Content-Type: text/plain; charset="us-ascii"

John,
This is an email message with two attached MS-Word documents.

--=====================_909671503==_
Content-Type: application/msword; name="MIMEtest1.doc";
 x-mac-type="42494E41"; x-mac-creator="4D535744"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="MIMEtest1.doc"

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAABAAAAIQAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
...
AAAAAAAAAAAAAAAAAAAAAAAAAA==
--=====================_909671503==_
Content-Type: application/msword; name="MIMEtest2.doc";
 x-mac-type="42494E41"; x-mac-creator="4D535744"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="MIMEtest2.doc"

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAABAAAAIQAAAAAAAAAAAA
EAAAIwAAAAEAAAD+////AAAAACAAAAD/////////////////////////////////////////////
...
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA==
--=====================_909671503==_--
.
```

RFC 822 headers

MIME header

Message

MIME header

Attachment1

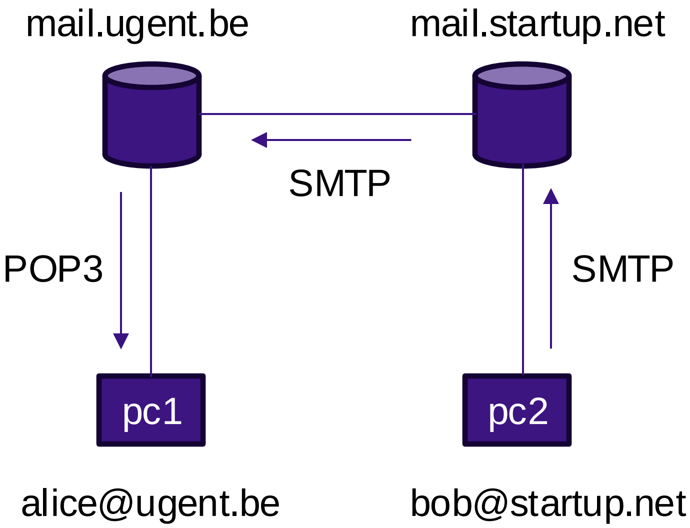
MIME header

Attachment2

End Of Message

Experiment in Kathará

- kathara-chapter2->email experiment
  - Involves commonly used postfix SMTP server and dovecot POP3/IMAP server
- Write and receive e-mail from bob@startup.net to alice@ugent.be
  - Capture SMTP packets between mail servers using tcpdump
  - Capture POP3 packets when Bob fetches mail



Chapter 2 outline

2.1 Principles of network applications

2.2 Web and HTTP

2.3 electronic mail

- SMTP, POP3, IMAP

2.4 DNS (Domain Name System)

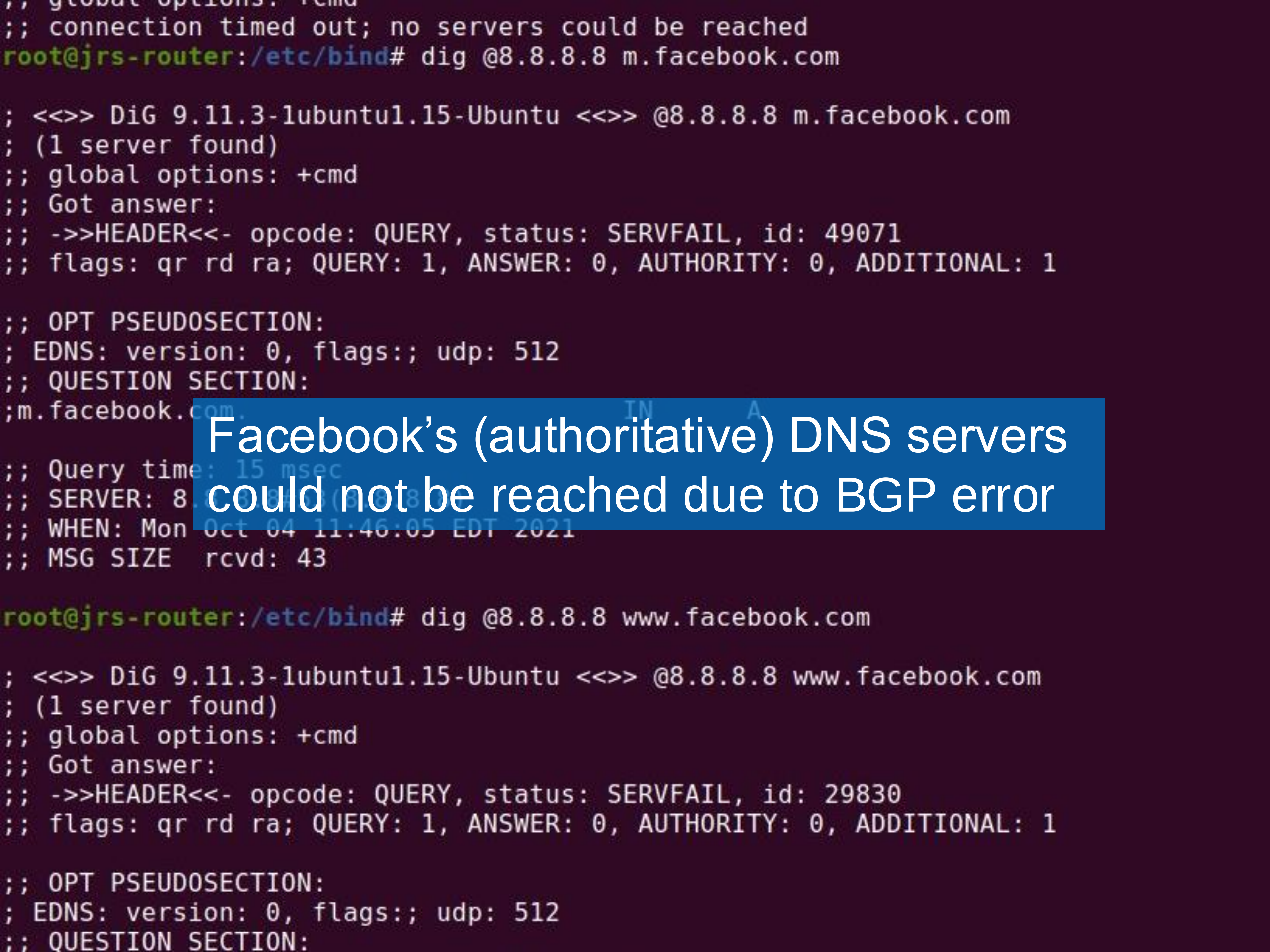
2.5 P2P applications

[**2.6** video streaming and content distribution networks]

[**2.7** socket programming with UDP and TCP]

Facebook unreachable on Oct 4, 2021

Outage lasting > 5 hours



```
;; global options: +cmd
;; connection timed out; no servers could be reached
root@jrs-router:/etc/bind# dig @8.8.8.8 m.facebook.com

; <<>> DiG 9.11.3-1ubuntu1.15-Ubuntu <<>> @8.8.8.8 m.facebook.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 49071
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;m.facebook.com.                IN      A

;; Query time: 15 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Oct 04 11:46:05 EDT 2021
;; MSG SIZE  rcvd: 43

root@jrs-router:/etc/bind# dig @8.8.8.8 www.facebook.com

; <<>> DiG 9.11.3-1ubuntu1.15-Ubuntu <<>> @8.8.8.8 www.facebook.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 29830
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
```

Facebook’s (authoritative) DNS servers could not be reached due to BGP error

Domain Name System (DNS)

Two possible network identifications in the Internet : name (used by Internet users) OR address (used by hardware)

Address : 4 bytes (4 numbers with values between 0 and 255)
example : 157.193.40.41 (corresponding to *allserv.ugent.be* server)
advantage : fixed limited length, hierarchical, easy to handle in the network, physical structure of the network

Name : mnemonic : xxx.xxx. … .xxx
example : intec.ugent.be
advantage : readable, independence of name and address, logical structure of an organization

DNS = application layer protocol using *distributed database*
to provide name to address translation using a *client/server* architecture

Some examples :
google.be ⬄ 142.250.179.195
ugent.be ⬄ 157.193.43.50

When to use DNS ?

Send e-mail to :

rik.vandewalle@ugent.be

alexander.decroo@premier.fed.be

SMTP server of ugent.be domain ?
157.193.49.14
cedar.ugent.be

SMTP server of premier.fed.be domain ?
157.193.214.245
relay-shs.fed.be

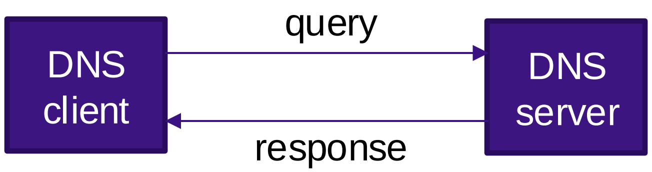
Access web-site :

www.atlantis.ugent.be

IP address of HTTP server ?
157.193.215.4
pegasus.atlantis.ugent.be

DNS basics:

- Client/server protocol
- Application layer
- Uses UDP in the transport layer as default (TCP also possible)



Domain Name System (DNS)

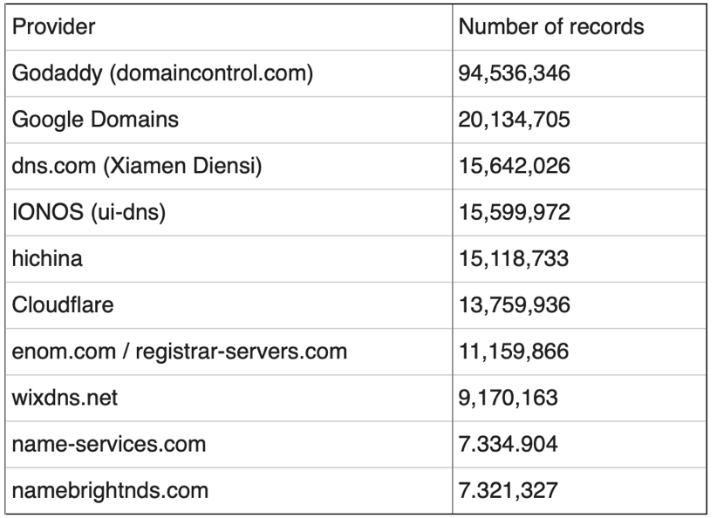
- Who determines the mapping and where is it stored?
  - Organisations can request domain names (e.g. ugent.be) to a DNS registrar, and if available, they need to set up **authoritative DNS server** which stores mappings between their servers and IP addresses, e.g.:

ugent.be	IN	A	157.193.43.50
allserv.ugent.be	IN	A	157.193.40.41
cedar.ugent.be	IN	A	157.193.49.14
ugent.be	IN	MX	cedar.ugent.be

DNS Resource Records

Some internet DNS server statistics in 2020

- World wide:
  - 477 million DNS records
  - ~ 2.7 million authoritative name servers
- only 0.35 % =~ 9400 servers responsible for 90 % of the domain names
  - Reason: use of cloud-based DNS service

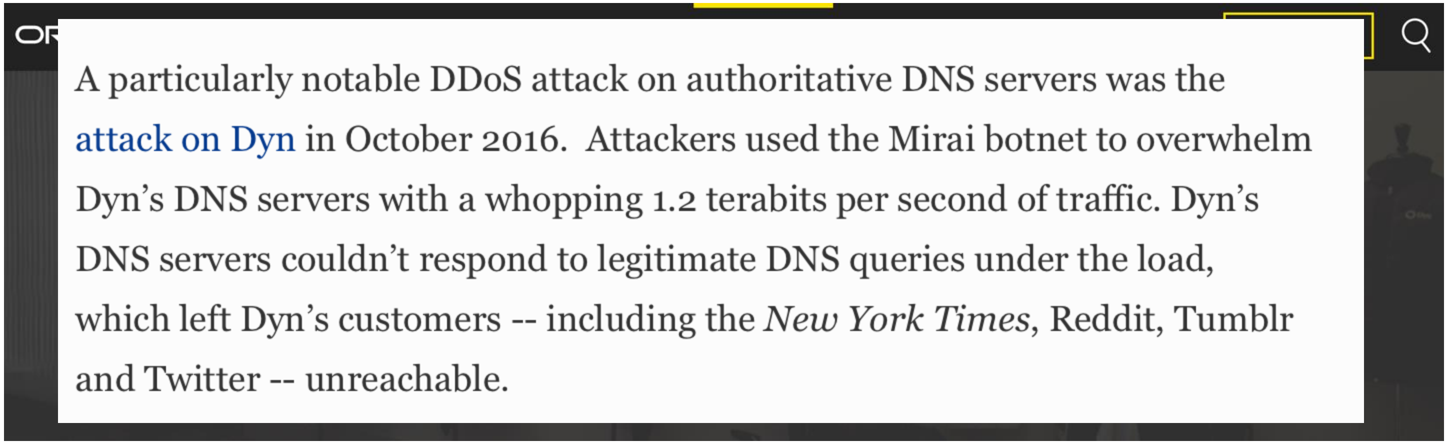


Provider	Number of records
Godaddy (domaincontrol.com)	94,536,346
Google Domains	20,134,705
dns.com (Xiamen Diensi)	15,642,026
IONOS (ui-dns)	15,599,972
hichina	15,118,733
Cloudflare	13,759,936
enom.com / registrar-servers.com	11,159,866
wixdns.net	9,170,163
name-services.com	7.334.904
namebrightnds.com	7.321,327

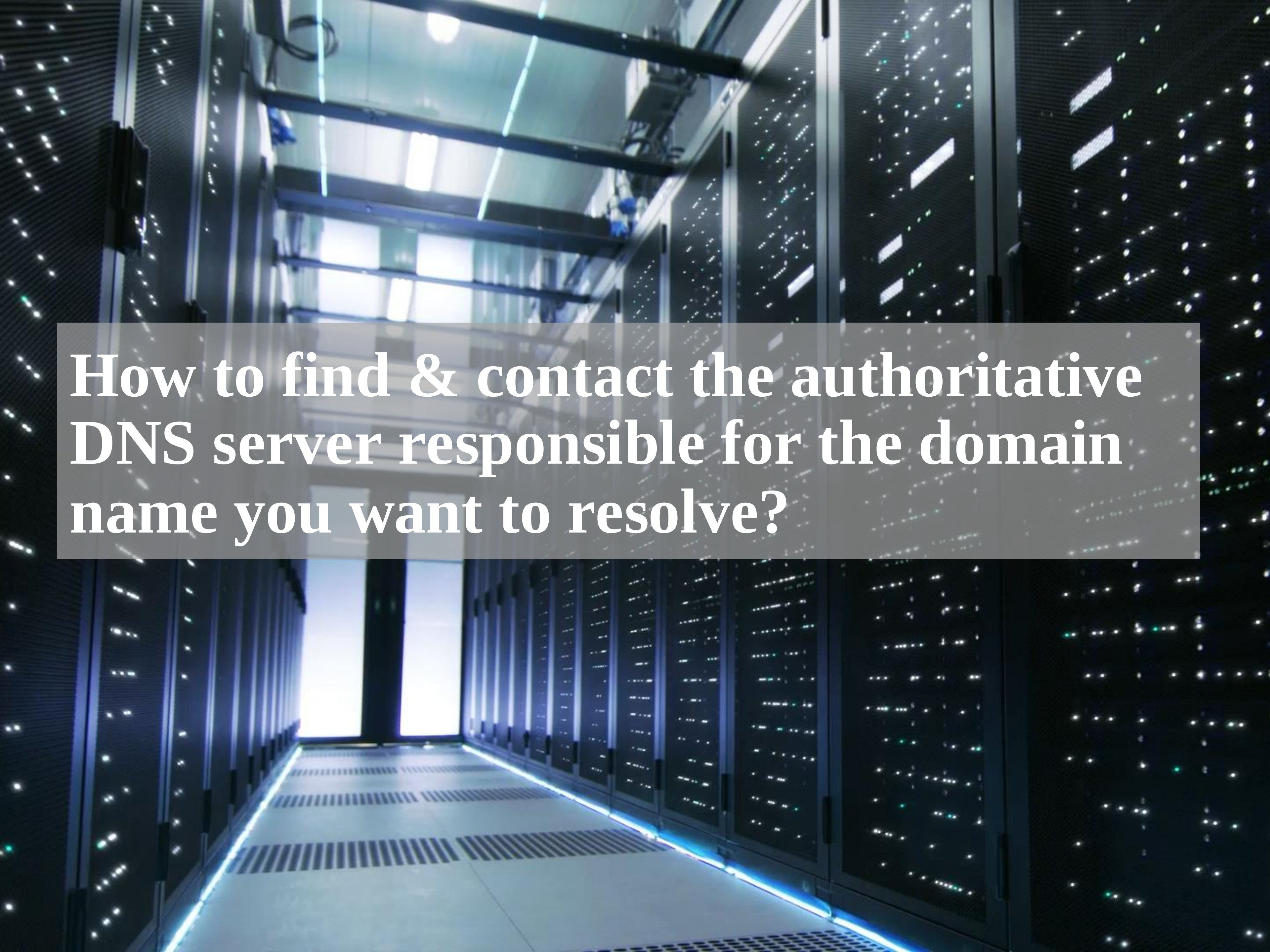
Source: https://isc.sans.edu/diary/Internet+Choke+Points%3A+Concentration+of+Authoritative+Name+Servers/26428

DDOS attack

Mirai botnet generating tens of millions of DNS queries



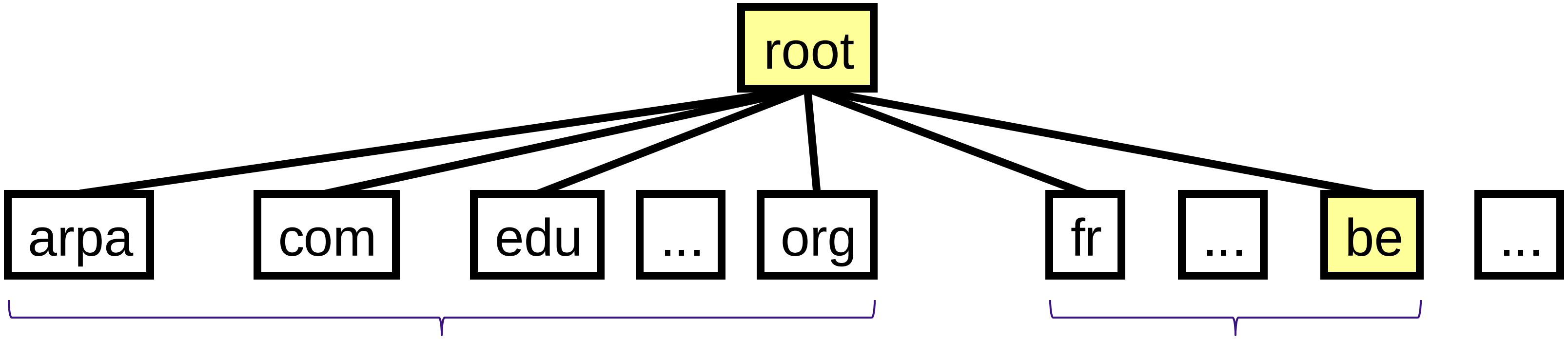
A particularly notable DDoS attack on authoritative DNS servers was the attack on Dyn in October 2016. Attackers used the Mirai botnet to overwhelm Dyn's DNS servers with a whopping 1.2 terabits per second of traffic. Dyn's DNS servers couldn't respond to legitimate DNS queries under the load, which left Dyn's customers -- including the *New York Times*, Reddit, Tumblr and Twitter -- unreachable.



How to find & contact the authoritative DNS server responsible for the domain name you want to resolve?

Domain Name System (DNS)

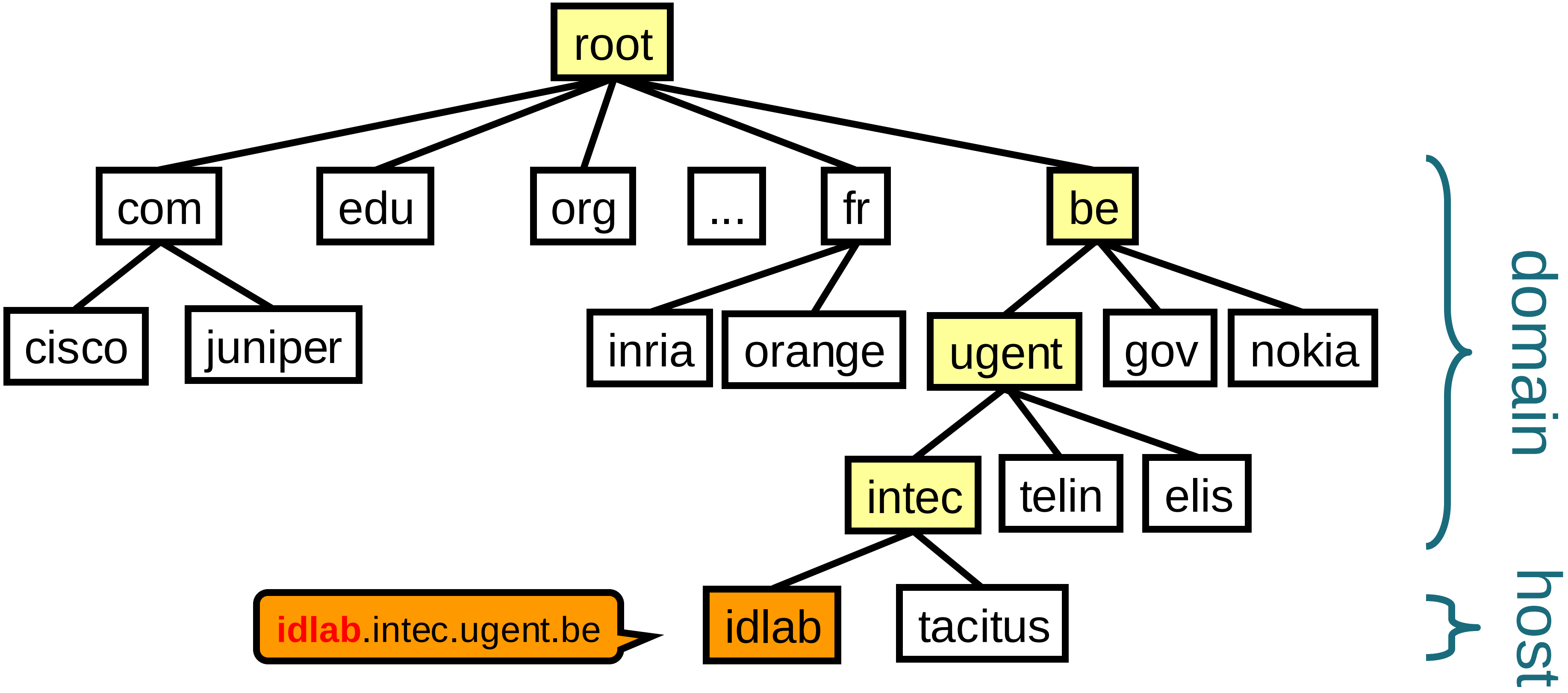
- Historically (<1982)
  - Single `hosts.txt` file stored at the NIC (Network Information Center) in the US storing list of <name,IP> address mappings
  - All hosts needed to fetch the file and regularly sync it
  - Quickly became unscalable
- Since the ‘80s: hierarchical tree structure for domain names
  - Top-Level Domain (TLD) names (managed by ICANN)
  - Each TLD is managed by organization deciding on sub-domain names



Early introduced TLD domains

Domain per ISO country code (RFC 1032)

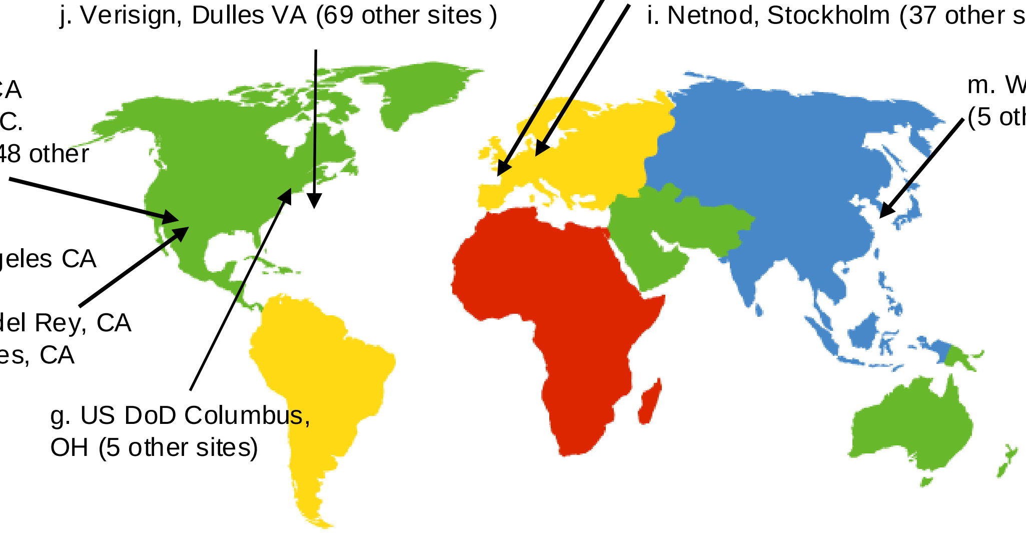
DNS : Domain Name System



- hierarchical & relative distinguished names
- sub-domain hierarchy usually reflects logical structure of organization
- **every domain has at least one name server** able to answer following queries:
    1. Give IP address of any server/host inside its own domain
    2. Give name server responsible for any direct sub-domain

Root name servers

- Contacted by local name server that can not resolve name
- Has reference to TLD name servers



c. Cogent, Herndon, VA (5 other sites)
d. U Maryland College Park, MD
h. ARL Aberdeen, MD
j. Verisign, Dulles VA (69 other sites)

k. RIPE London (17 other sites)

i. Netnod, Stockholm (37 other sites)

e. NASA Mt View, CA
f. Internet Software C.
Palo Alto, CA (and 48 other sites)

m. WIDE Tokyo
(5 other sites)

a. Verisign, Los Angeles CA
(5 other sites)
b. USC-ISI Marina del Rey, CA
l. ICANN Los Angeles, CA
(41 other sites)

g. US DoD Columbus, OH (5 other sites)

13 logical root name "servers" worldwide

each "server" replicated many times

TLD, Authoritative Servers

Top-Level Domain (TLD) servers:

- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp, be, tk
- **Network Solutions** maintains servers for .com TLD
- **Educause** for .edu TLD

Authoritative DNS servers:

- Organization’s own DNS server(s), providing authoritative hostname to IP mappings for organization’s named hosts
- can be maintained by organization or service provider

Inserting Records into DNS

- example: new startup “Network Utopia”
- register name networkuptopia.com at **DNS registrar** (e.g., Network Solutions)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into .com TLD server:

```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
(networkutopia.com, dns2.networkutopia.com, NS)
(dns2.networkutopia.com, 212.212.215.1, A)
```

- In **`dns1.networkutopia.com`** **`(& dns2.networkutopia.com):`** create authoritative server
  - type A record for www.networkuptopia.com (web server);
  - type MX record for networkutopia.com (mail server)

DNS Belgium is responsible for registration of .be, .vlaanderen and .brussels domains
Via > 380 DNS registrars, you can register your domain name on their registration system

DNS : Resource Records (RR)

What to store in a DNS :

- list of the worldwide root servers
- list of names (host, name server, mail server, …) and their corresponding IP address
- alias names and their canonical name
- list of IP addresses and their corresponding names (for inverse look-up)
- ...

How to store information in the DNS databases ?

<u>Resource Record (RR) :</u>

[name], [TTL], [class], record-type, record-data

name : name to be resolved
TTL : how long record may be cached
class : IN (for Internet)
record-type : e.g.: NS, A, MX, CNAME
record-data : e.g.: IP address

DNS : Resource Records (RR)

name can have multiple A records for redundancy/load balancing

Record-Types / Record-Data :

A **:** the *name* is a **hostname** and the *record-data*

webserver1.intec.ugent.be IN A 157.193.135.37

Name server is not necessarily located in same domain

NS **:** the *name* is a **domain** and the *record-data* is the **hostname of a server** that knows how to obtain the IP addresses in that domain

ugent.be IN NS ugdns1.ugent.be (authoritative name server for ugent.be)

intec.ugent.be IN NS dns1.intec.ugent.be (authoritative name server for intec.ugent.be)

CNAME **:** the *name* is an **alias for a hostname** and the *record-data* is the corresponding **canonical hostname**

www.intec.ugent.be IN CNAME webserver1.intec.ugent.be

MX **:** the *name* is a **domain name** and the *record-data* is the corresponding **name of a mail server** (MTA), *preference* indicates the primary, secondary, … mail servers for the domain

ugent.be IN MX preference=20 smtpfltrp1.ugent.be

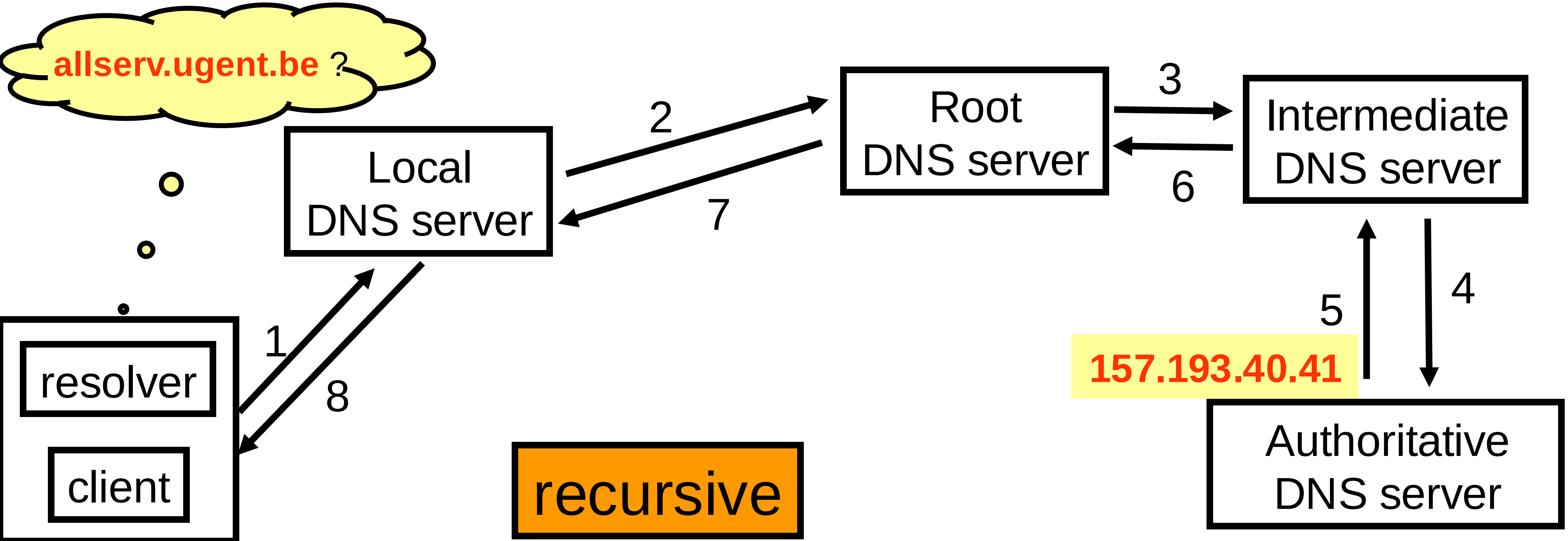
preference=20 smtpfltrp2.ugent.be

Local DNS Name Server

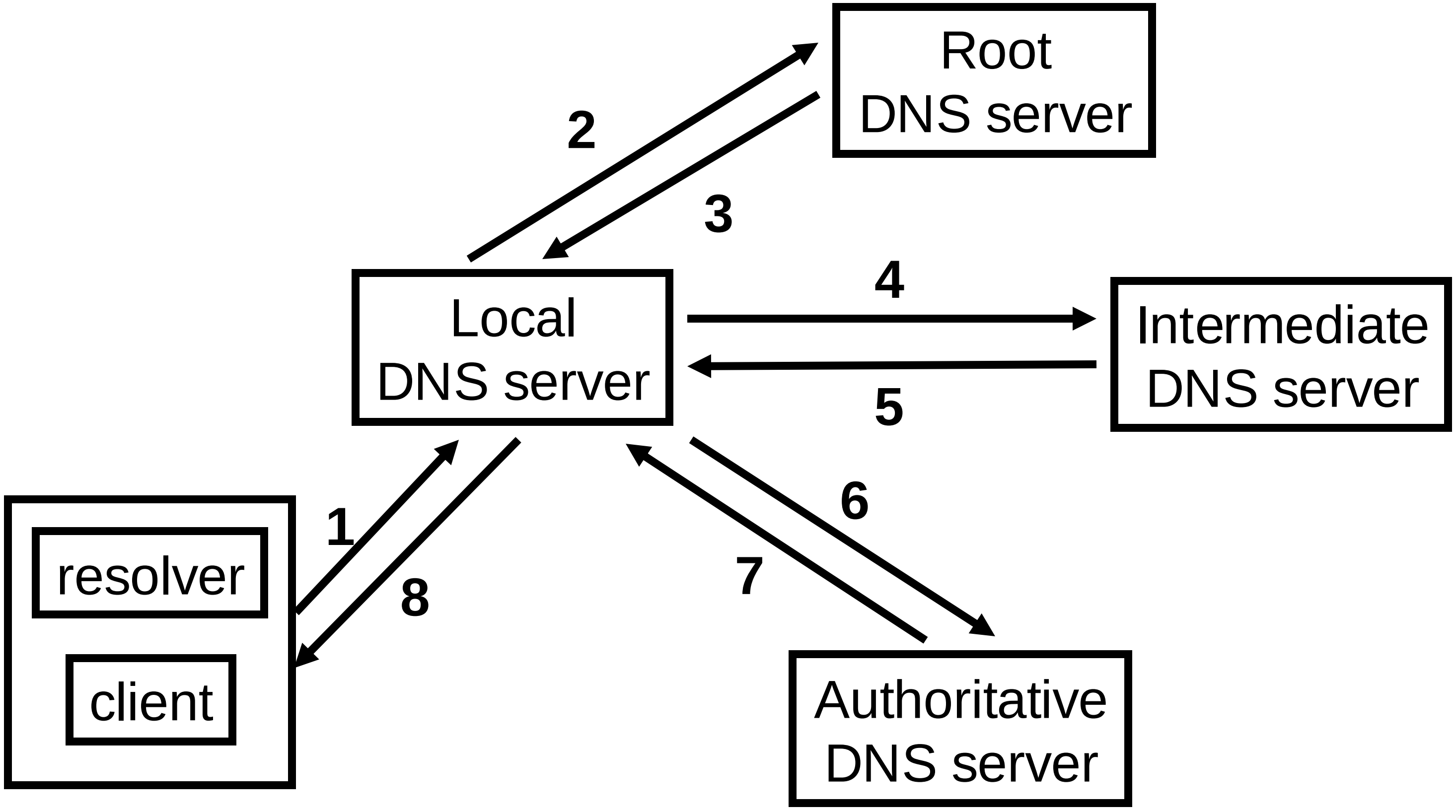
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy

How do DNS servers interact?

- Resolver : local program (client side) sending out a mapping request (allserv.ugent.be ?) to local name server
- Local name server (default name server) : handles request from client contacts other name server(s) to resolve the name
- Root name server : top level root server (13 in total)
- Authoritative name server : where host (requested name) is registered (at least two authoritative name servers for each host)

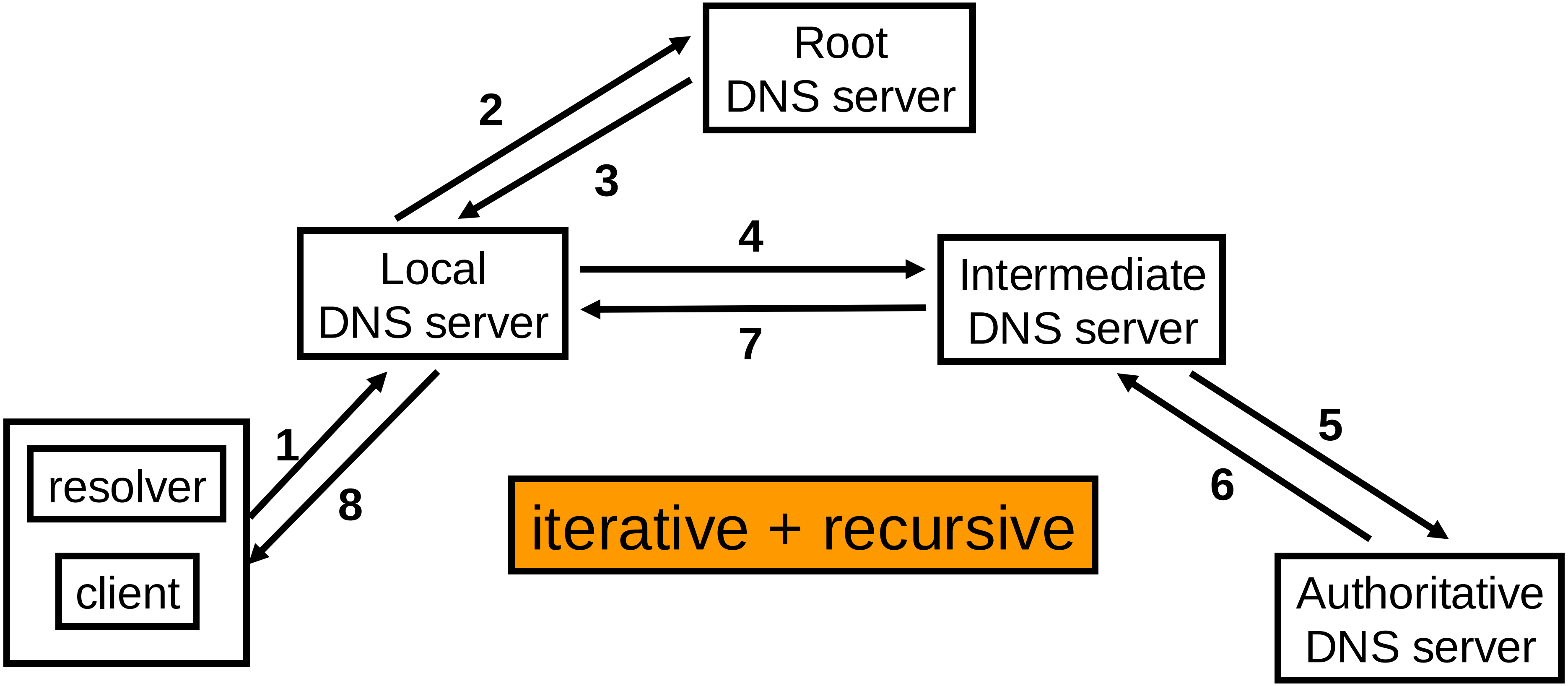


DNS : mapping name to address



iterative

DNS : mapping name to address

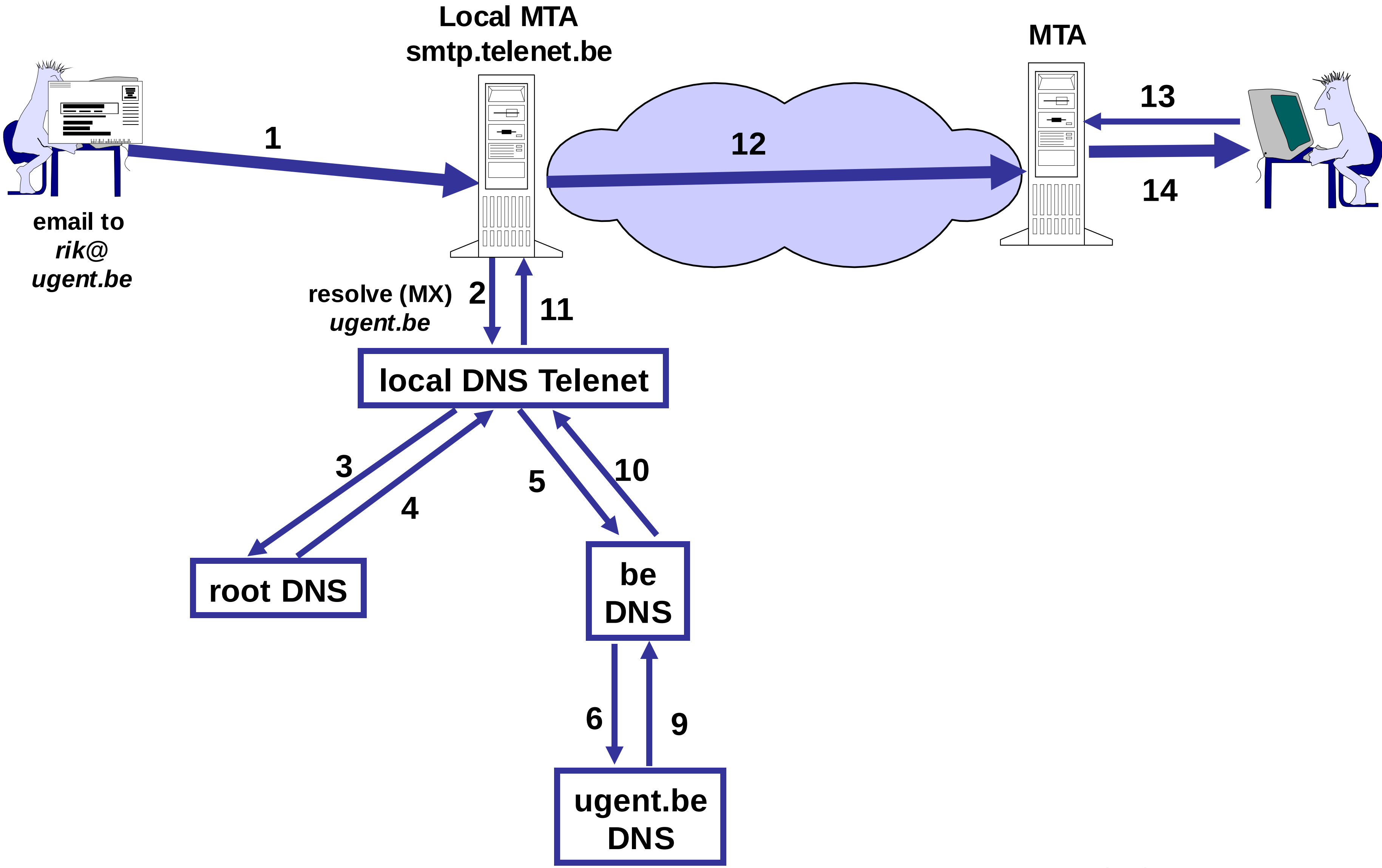


iterative + recursive

caching of name/address translation pairs

- caching in intermediate name servers
- improve delay performance of name/address translation
- reduce number of DNS queries on the network
- cached record is valid limited in time (few days ➔ TTL)
- very limited number of requests towards root servers

Interaction between e-mail and DNS

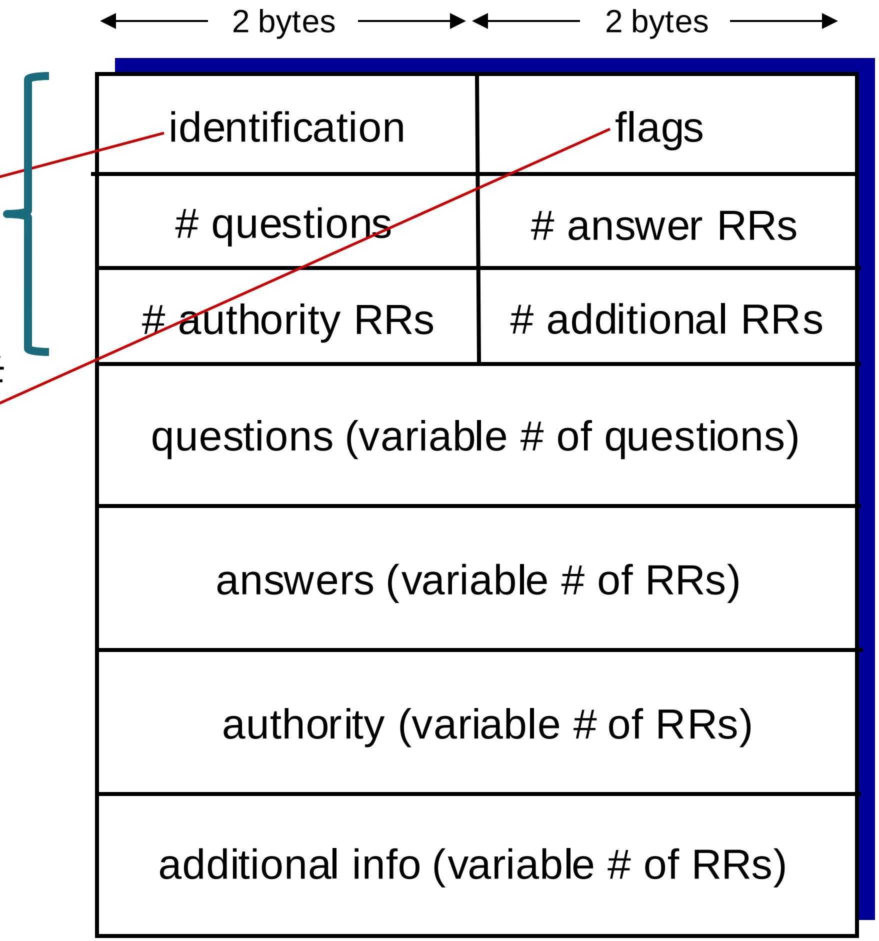


DNS : Messages (in UDP)

query and *reply* messages, both with same *message format*

message header

- identification:
  16 bit # for query,
  reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



2 bytes | 2 bytes

identification | flags

# questions | # answer RRs

# authority RRs | # additional RRs

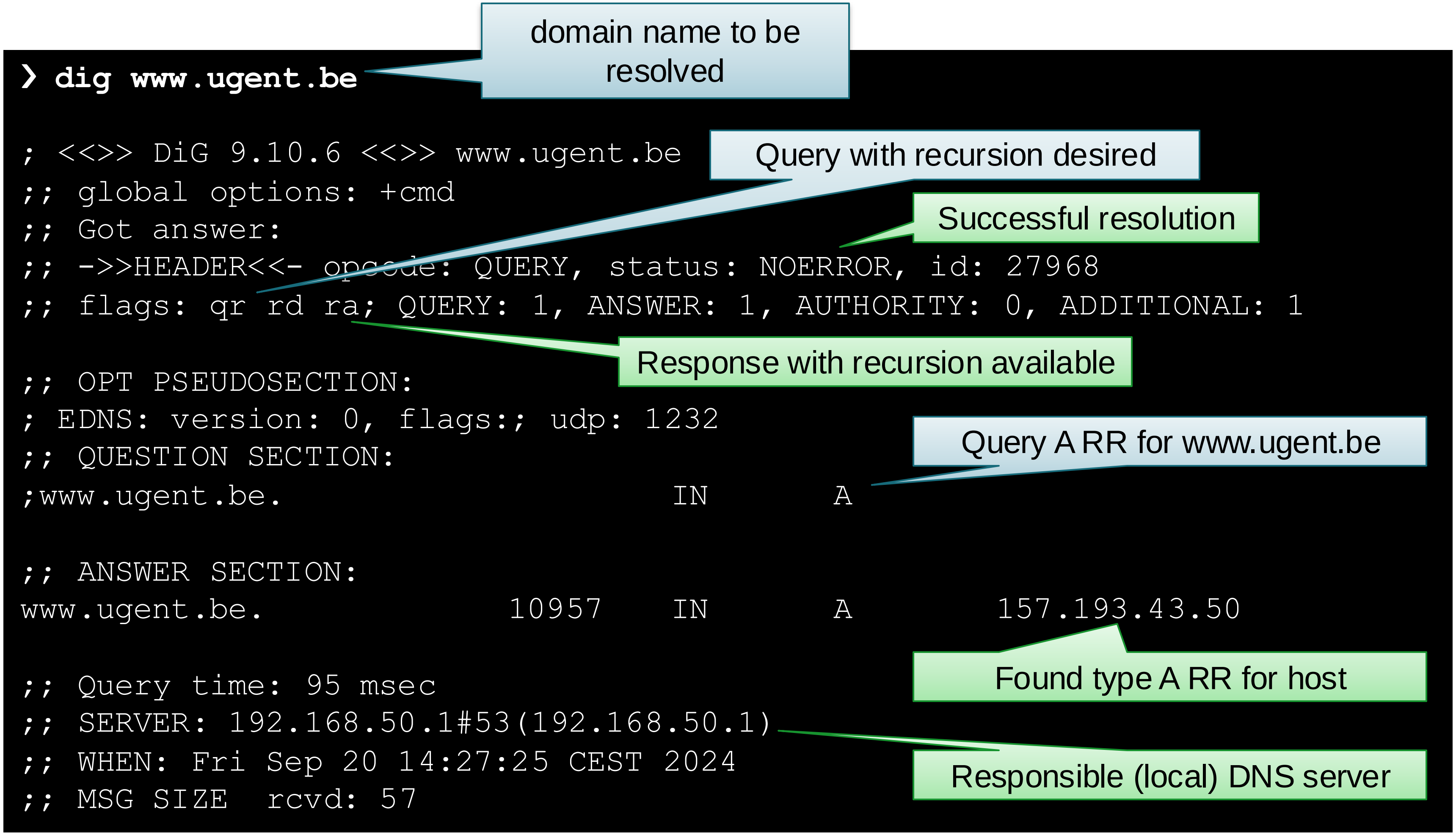
questions (variable # of questions)

answers (variable # of RRs)

authority (variable # of RRs)

additional info (variable # of RRs)

Resolving domains from command line



```
❯ dig www.ugent.be

; <<>> DiG 9.10.6 <<>> www.ugent.be
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27968
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.ugent.be.                    IN       A

;; ANSWER SECTION:
www.ugent.be.            10957    IN       A        157.193.43.50

;; Query time: 95 msec
;; SERVER: 192.168.50.1#53(192.168.50.1)
;; WHEN: Fri Sep 20 14:27:25 CEST 2024
;; MSG SIZE  rcvd: 57
```

domain name to be resolved

Query with recursion desired

Successful resolution

Response with recursion available

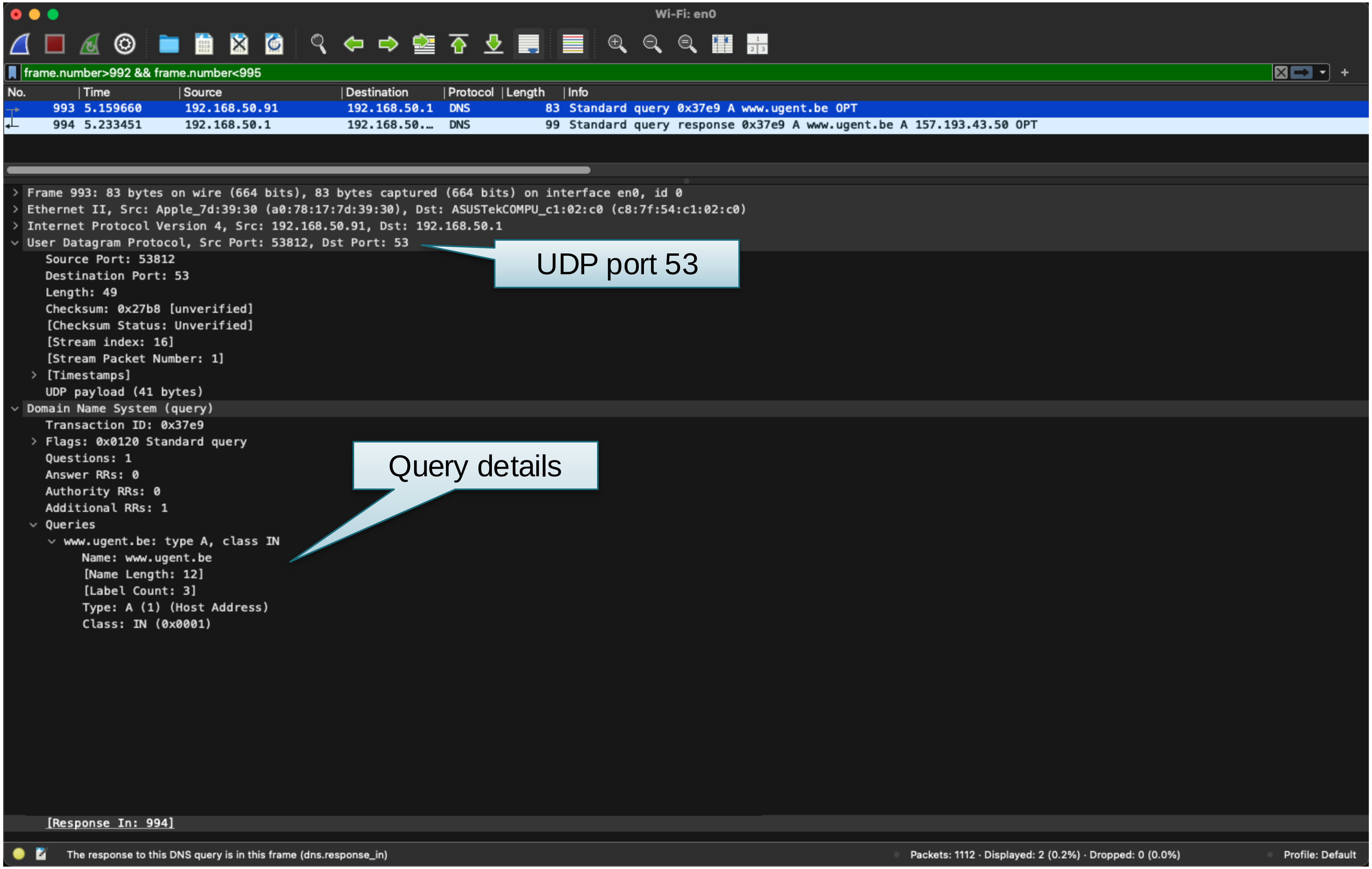
Query A RR for www.ugent.be

Found type A RR for host

Responsible (local) DNS server

Alternatively the tool `nslookup` can be used from command line.

dig in Wireshark: DNS query



```
frame.number>992 && frame.number<995
```

```
No.    Time        Source           Destination     Protocol  Length  Info
993    5.159660    192.168.50.91    192.168.50.1    DNS       83      Standard query 0x37e9 A www.ugent.be OPT
994    5.233451    192.168.50.1     192.168.50.…    DNS       99      Standard query response 0x37e9 A www.ugent.be A 157.193.43.50 OPT
```

```
> Frame 993: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface en0, id 0
> Ethernet II, Src: Apple_7d:39:30 (a0:78:17:7d:39:30), Dst: ASUSTekCOMPU_c1:02:c0 (c8:7f:54:c1:02:c0)
> Internet Protocol Version 4, Src: 192.168.50.91, Dst: 192.168.50.1
v User Datagram Protocol, Src Port: 53812, Dst Port: 53
    Source Port: 53812
    Destination Port: 53
    Length: 49
    Checksum: 0x27b8 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 16]
    [Stream Packet Number: 1]
  > [Timestamps]
    UDP payload (41 bytes)
v Domain Name System (query)
    Transaction ID: 0x37e9
  > Flags: 0x0120 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  v Queries
    v www.ugent.be: type A, class IN
        Name: www.ugent.be
        [Name Length: 12]
        [Label Count: 3]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
    [Response In: 994]
```

UDP port 53

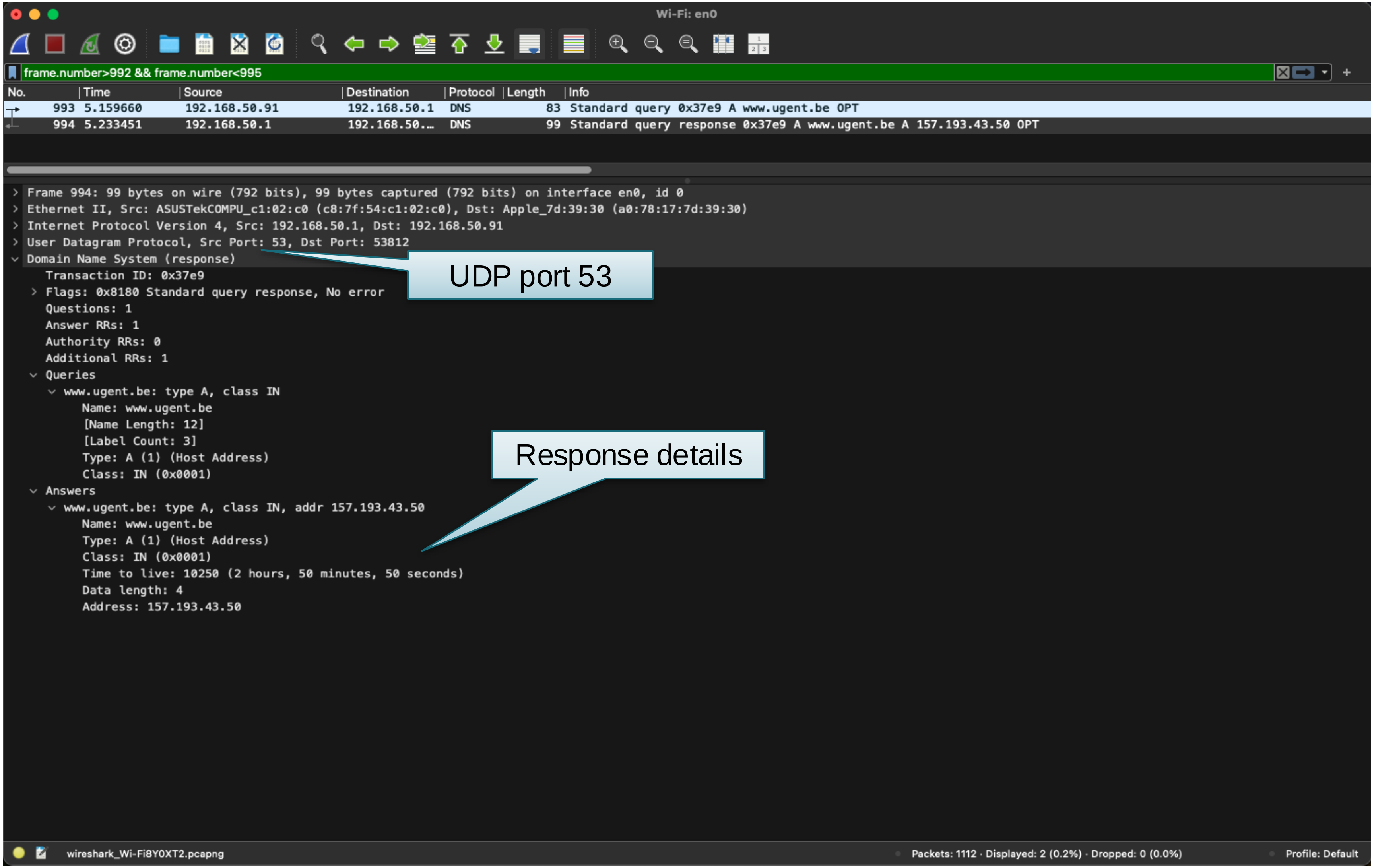
Query details

The response to this DNS query is in this frame (dns.response_in)

Packets: 1112 · Displayed: 2 (0.2%) · Dropped: 0 (0.0%)

Profile: Default

dig in Wireshark: DNS response



```
Wi-Fi: en0

frame.number>992 && frame.number<995

No.   Time       Source          Destination    Protocol Length Info
  993 5.159660   192.168.50.91   192.168.50.1   DNS      83 Standard query 0x37e9 A www.ugent.be OPT
  994 5.233451   192.168.50.1    192.168.50.…   DNS      99 Standard query response 0x37e9 A www.ugent.be A 157.193.43.50 OPT

> Frame 994: 99 bytes on wire (792 bits), 99 bytes captured (792 bits) on interface en0, id 0
> Ethernet II, Src: ASUSTekCOMPU_c1:02:c0 (c8:7f:54:c1:02:c0), Dst: Apple_7d:39:30 (a0:78:17:7d:39:30)
> Internet Protocol Version 4, Src: 192.168.50.1, Dst: 192.168.50.91
> User Datagram Protocol, Src Port: 53, Dst Port: 53812
v Domain Name System (response)
    Transaction ID: 0x37e9
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  v Queries
    v www.ugent.be: type A, class IN
        Name: www.ugent.be
        [Name Length: 12]
        [Label Count: 3]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
  v Answers
    v www.ugent.be: type A, class IN, addr 157.193.43.50
        Name: www.ugent.be
        Type: A (1) (Host Address)
        Class: IN (0x0001)
        Time to live: 10250 (2 hours, 50 minutes, 50 seconds)
        Data length: 4
        Address: 157.193.43.50

wireshark_Wi-Fi8Y0XT2.pcapng     Packets: 1112 · Displayed: 2 (0.2%) · Dropped: 0 (0.0%)     Profile: Default
```

UDP port 53

Response details

DNS and IPv6: AAAA record

- Dual network layer:
  IPv6 addresses next to IPv4 addresses

- DNS extended with a new Resource Record type: AAAA
  - Similar to A record in IPv4
  - E.g. zone file facebook.com

```
facebook.com. IN     A          179.60.195.36                        ; IPv4
              IN     AAAA       2a03:2880:f121:83:face:b00c::25de    ; IPv6
```

Upgrade of _all_ DNS records needed!

Attacking DNS

DDoS attacks

- bombard root servers with traffic
  - not successful to date
  - traffic filtering
  - local DNS servers cache IPs of TLD servers, allowing root server bypass
- bombard TLD servers
  - potentially more dangerous

redirect attacks

- man-in-middle
  - Intercept queries
- DNS poisoning
  - Send bogus replies to DNS server, which caches

exploit DNS for DDoS

- send queries with spoofed source address: target IP
- requires amplification