COSC264
Introduction to Computer Networks and the Internet

Email, DNS

Dr. Barry Wu
Wireless Research Centre
University of Canterbury
barry.wu@canterbury.ac.nz

Outline

- Electronic Mail
  - SMTP, POP3, IMAP
- DNS

Outline

- Electronic Mail
  - SMTP, POP3, IMAP
- DNS

Email

- Invented by Ray Tomlinson, first used in 1960s [wiki];
- One of the Internet's most important applications to date!

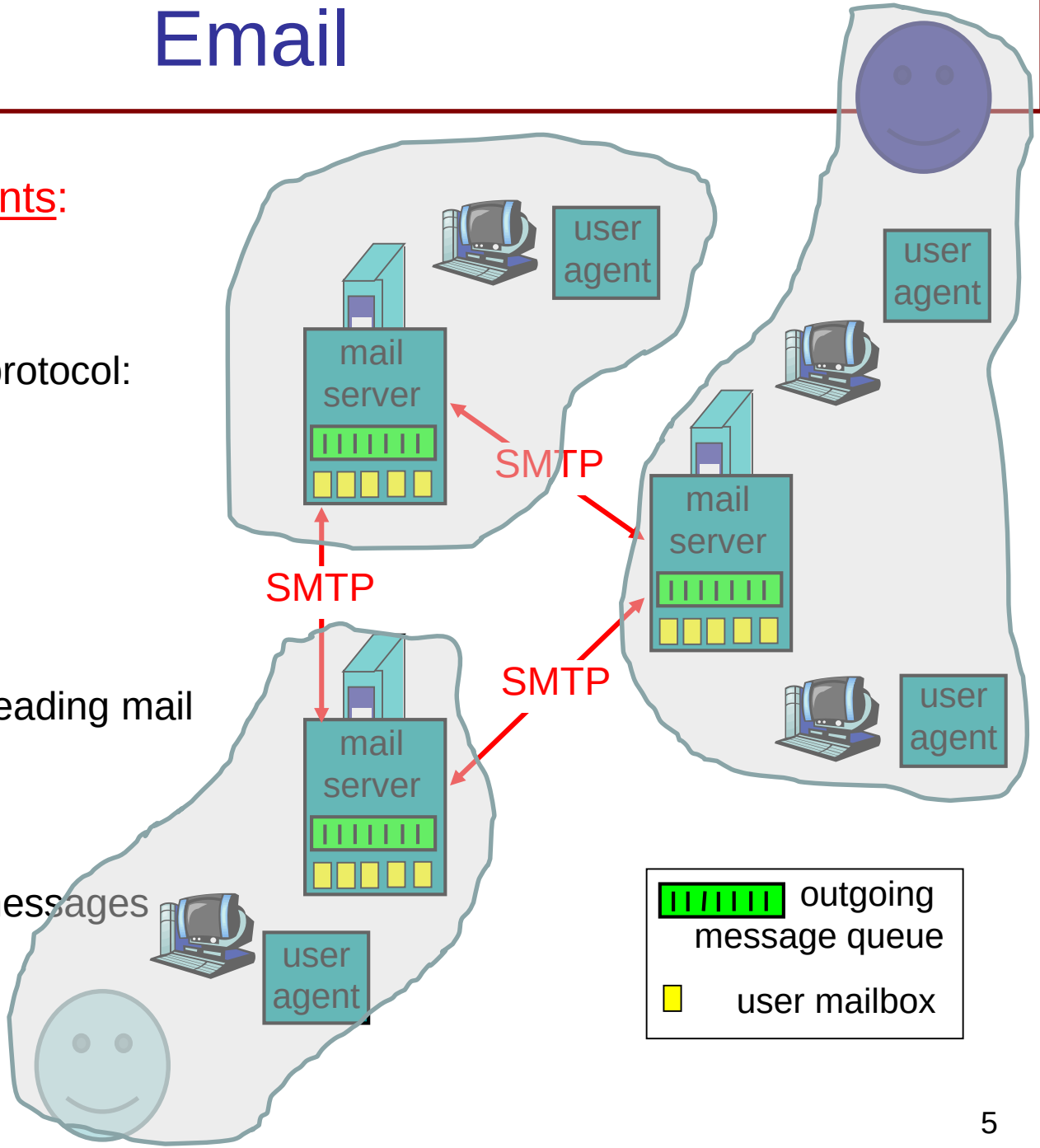
Email

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

User Agent

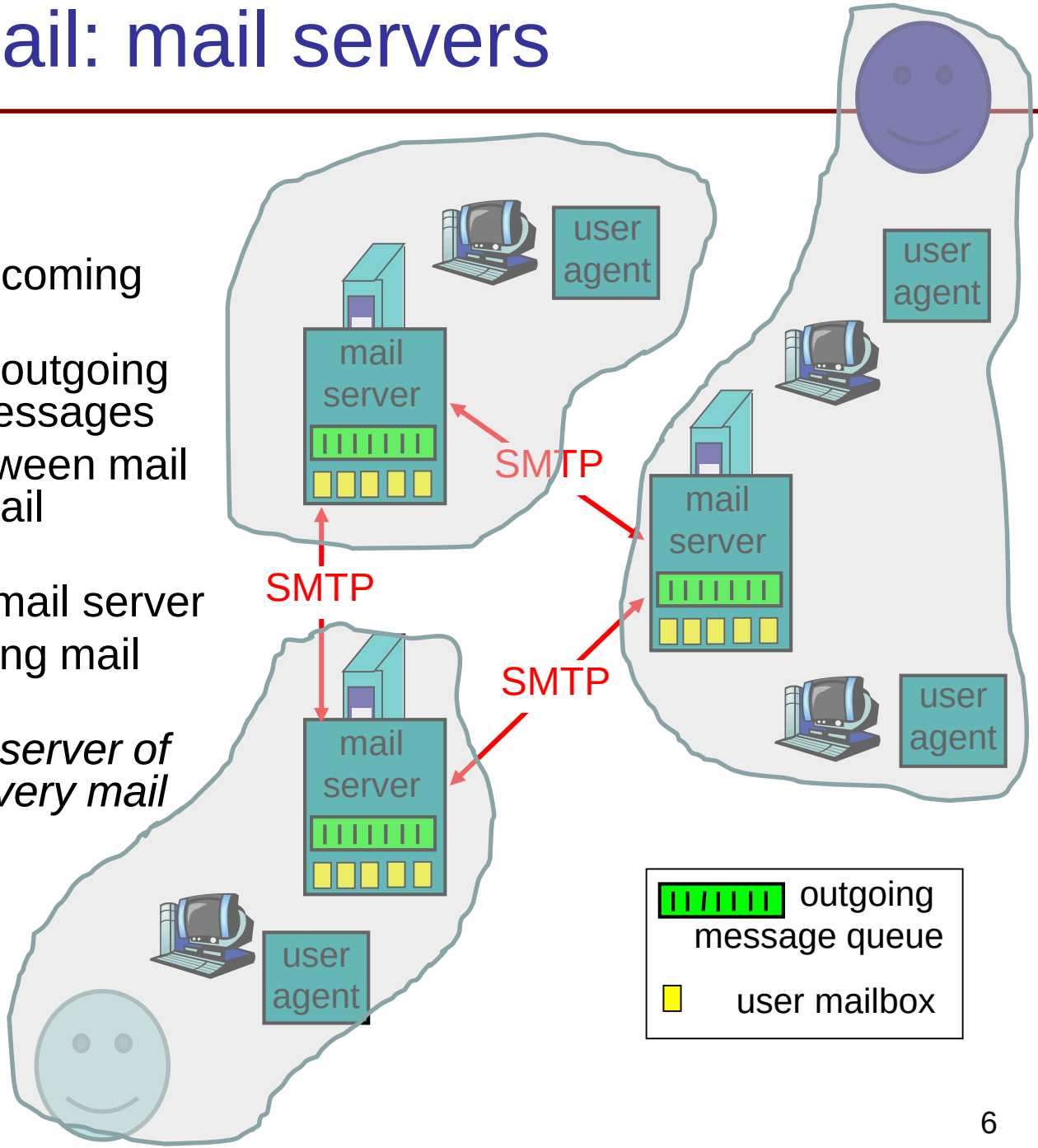
- a.k.a. “mail reader”
- composing, editing, reading mail messages
- e.g., Outlook
- outgoing, incoming messages stored on server



Email: mail servers

Mail Servers

- mailbox contains incoming messages for user
- message queue of outgoing (to be sent) mail messages
- SMTP protocol between mail servers to send email messages
  - client: sending mail server
  - "server": receiving mail server
  - *Both client and server of SMTP run on every mail server.*



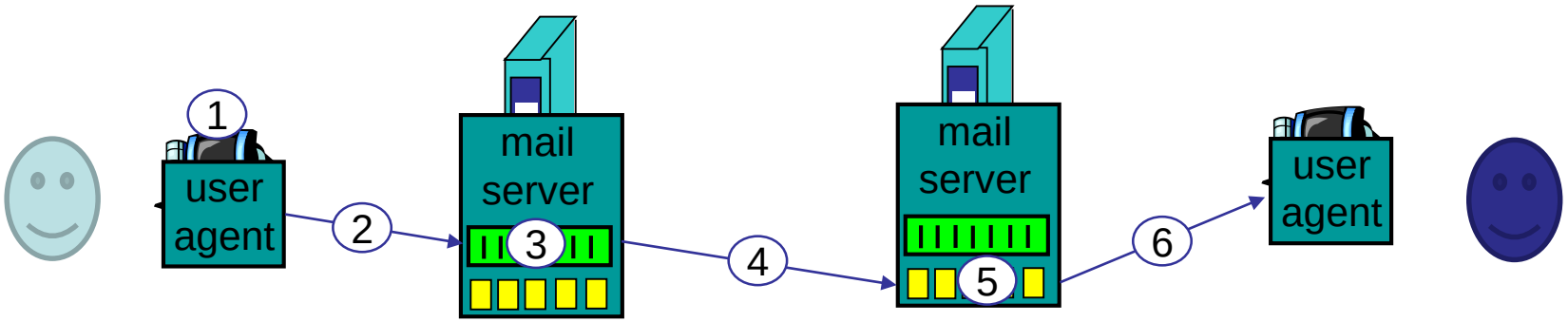
SMTP

- SMTP is at the heart of Internet email;
- First defined by RFC 821 in 1982;
- Updated in 2008 by RFC 5321;
- Non-standard protocols may be used internally by mail service providers;

Basic operation of SMTP

1) Alice uses UA (user agent) to compose message and "to" `bob@someschool.edu`
2) Alice's UA sends message to her mail server; message placed in message queue
3) Client side of SMTP opens TCP connection with Bob's mail server
4) SMTP client sends Alice's message over the TCP connection

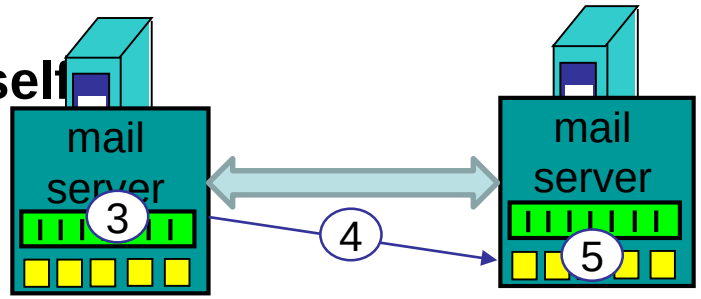
   -> if there are more messages – they are sent via a persistent TCP connection
5) Bob's mail server places the message in Bob's mailbox
6) Bob invokes his user agent to read message



Sample SMTP interaction

After TCP connection is established;

S: 220 server_host_name
C: HELO client_host_name
S: 250 Hello client_host_name, pleased to meet you
C: MAIL FROM: <alice@client_host_name>
S: 250 alice@client_host_name... Sender ok
C: RCPT TO: <bob@server_host_name>
S: 250 bob@server_host_name ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Hello Bob,
C: how are you doing?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 server_host_name closing connection



Comparison with HTTP

- both use persistent TCP connections
- both have ASCII command/response interaction, status codes
- HTTP: pull protocol; SMTP: push protocol
- SMTP typically restricts the body of all messages to simple 7-bit ASCII;

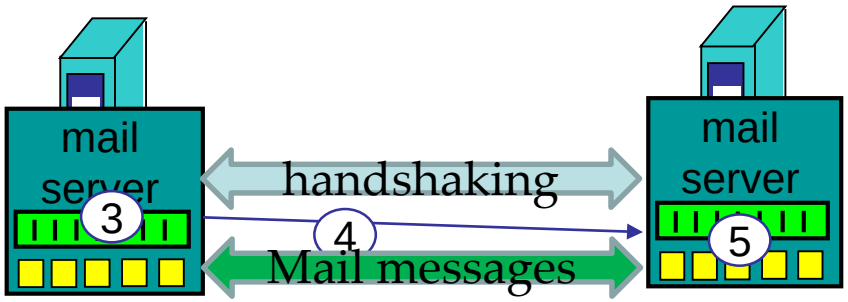
Handling documents with text and images:

- HTTP: each object encapsulated in its own response msg;
- SMTP: all objects placed into one message;

Mail message formats and MIME

- The base standard is defined in RFC822 →RFC 2822 → RFC5322;

From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Searching for the meaning of life.
[*a blank line*]
[*message body in ASCII*]



MIME
Multipurpose Internet Mail Extension
RFC 1341 → RFC 2045 ~ 2049;

The receiving server appends a `Received:` header line to the top of the message;

The MIME extension for non-ASCII data

From: alice@crepes.fr

To: bob@hamburger.edu

Subject: Picture of yummy crepe.

MIME-Version: 1.0

Content-Transfer-Encoding: base64

Content-Type: image/jpeg

(base64 encoded data …

……base64 encoded data)

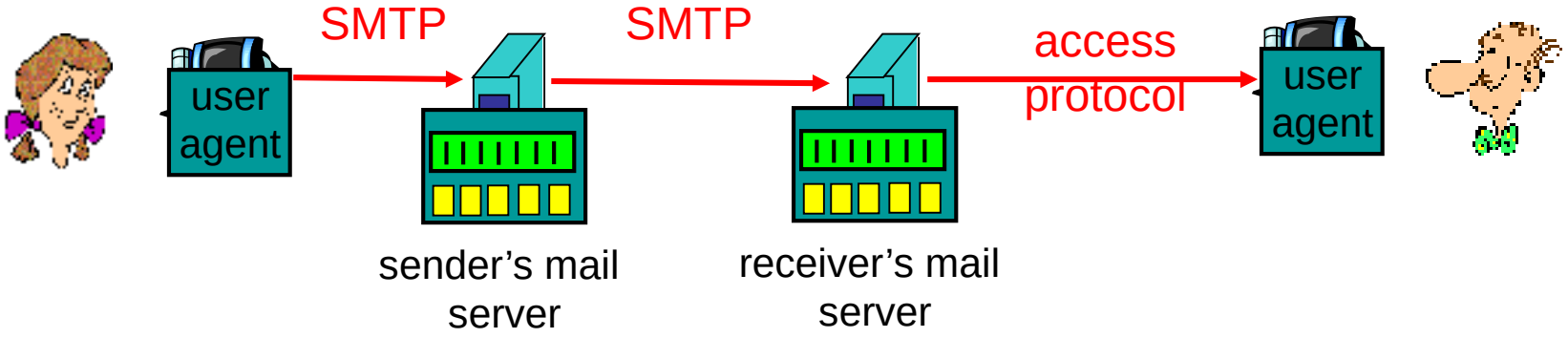
Converting data back to original non-ASCII data

For the user agent to decide its next action, e.g., JPEG decompression

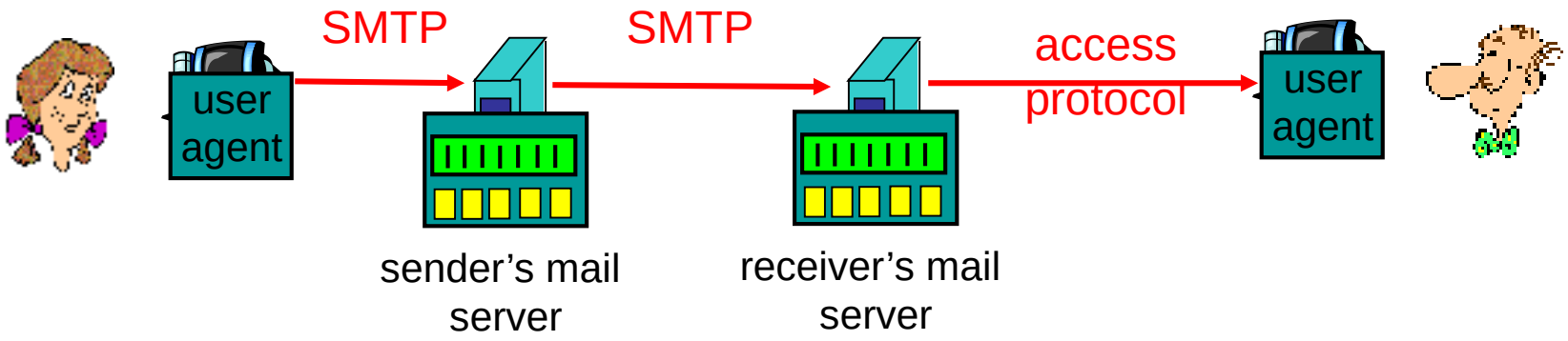
```
Content-Type: image/png; name="4000.png"
Content-Disposition: inline; filename="4000.png"
Content-Transfer-Encoding: base64
Content-ID: <ii_k1n9enb40>
X-Attachment-Id: ii_k1n9enb40

iVBORw0KGgoAAAANSUhEUgAAAJgAAAAWCAIAAACzGsLeAAAEfUlEQVRoBe1YPXbiMBAWOQum4HEC
5wSQJhVtOruEJh1lOhpT4o42VZq1TwAn4FFg34WdGf2OLLKJYZMHzyqCNNJ88z+S0zudTqIbt++B
h9s3obMAPdAF8k7yoAtkF8g78cCdmPEgyrTHRlrWq0dG6T2u6kvN9aU8rlZpLy0RFsXJ2aVCGrZw
M2j1b1vKVJ4BvexhTbxYxf8GAK9WGkUikkIvzlDY9ncXRZZVkgdExTCvspiswvn1B6Ar4KZln0kr
EmAjFokAf8EvVZY4WqLmVmtcCek84HOHdCgnKj7J5B7WGFY5OmPlwIaFYmRk+bE7so7EIYIMhzyf
7LPNrC/6sw24ICm2s4qawlWq0lR+NN/t5hEW4SQX+UTVpi0xz4dyCcq9DVE3b9R/3nMJ1uhOZRrN
BQR5PZY8roc1zUa9ysQ8QkP7s62NmAxRnL0qDIlULuc7Vw8wDTxHSW9gnH0NR0moF/TbpLDtNgvM
KNdSlWNeK2iDrHmM0pDOSlKApE97v0UCmsDxOJa9QrkpKahMUVkJiaVCM6oZR3vkdSpXwnMirPzW
R17w+fAc6mHoHAYlO4J/riLxIqTiEFAnmJKqdKBchCqYq1SkQQtUJJCcDA5Mx+t1tHrLk8WWlcta
fKCWoj7uGZMqRlN3bPMbi5pksj5ApGIztSgl6DAamF7RH4xE/kFvDHnmh1pruXyfqpuQshiFq1yD
```

SMTP and Mail access protocols



Mail access protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP3: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
  - HTTP: gmail, outlook, etc.

POP3 protocol

C: telnet mailserver 110 (*opens a TCP con. to the server on port 110*)

authorization phase

- client commands:
  - **`user:`** declare username
  - **`pass:`** password
- server responses
  - **`+OK`**
  - **`-ERR`**

transaction phase, client:

- **`list:`** list message numbers
- **`retr:`** retrieve message by number
- **`dele:`** delete
- **`Quit (`***after clients quits, server start deleting*`)`**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

```
duser@192.168.11.128:~$ telnet pop.163.com 110
Trying 123.126.97.79...
Connected to pop3.163.idns.yeah.net.
Escape character is '^]'.
+OK Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b52404dcf
5cs])
user barrywuh
+OK core mail
pass
+OK 689 message(s) [259644431 byte(s)]
list
```

```
retr 686
+OK 4053 octets
Received: from ucpmdf1p.canterbury.ac.nz (unknown [132.181.2.27])
        by mx26 (Coremail) with SMTP id TMCowAA3kbDGj1BdSrEhDw--.35971S3;
        Mon, 12 Aug 2019 05:59:39 +0800 (CST)
Received: from CONVERSION-DAEMON.it.canterbury.ac.nz by it.canterbury.ac.nz
 (PMDF V6.7-x02 #2267) id <0PW300R01EF8GM@it.canterbury.ac.nz> for
 barrywuh@163.com; Mon, 12 Aug 2019 09:59:34 +1200 (NZST)
Received: from UCEXHubCAS04-I.canterbury.ac.nz ([132.181.7.54])
 by it.canterbury.ac.nz (PMDF V6.7-x02 #2267)
 with ESMTPS id <0PW300CJBEF9EW@it.canterbury.ac.nz> for barrywuh@163.com; Mon,
 12 Aug 2019 09:59:33 +1200 (NZST)
Received: from UCEXMBX03-I.canterbury.ac.nz ([fe80::49ae:76ca:7b42:d8e2])
 by UCEXHubCAS04-I.canterbury.ac.nz ([fe80::a9a9:df3d:7009:e5de%13])
 with mapi id 14.03.0439.000; Mon, 12 Aug 2019 09:59:33 +1200
Date: Sun, 11 Aug 2019 21:59:33 +0000
From: Barry Wu <barry.wu@canterbury.ac.nz>
Subject: Test for email for COSC264
X-Originating-IP: [10.34.40.169]
To: "barrywuh@163.com" <barrywuh@163.com>
Message-id:
 <3ED135E755A27346BAF2B655127B8502B05ACF@UCEXMBX03-I.canterbury.ac.nz>
MIME-version: 1.0
```

IMAP

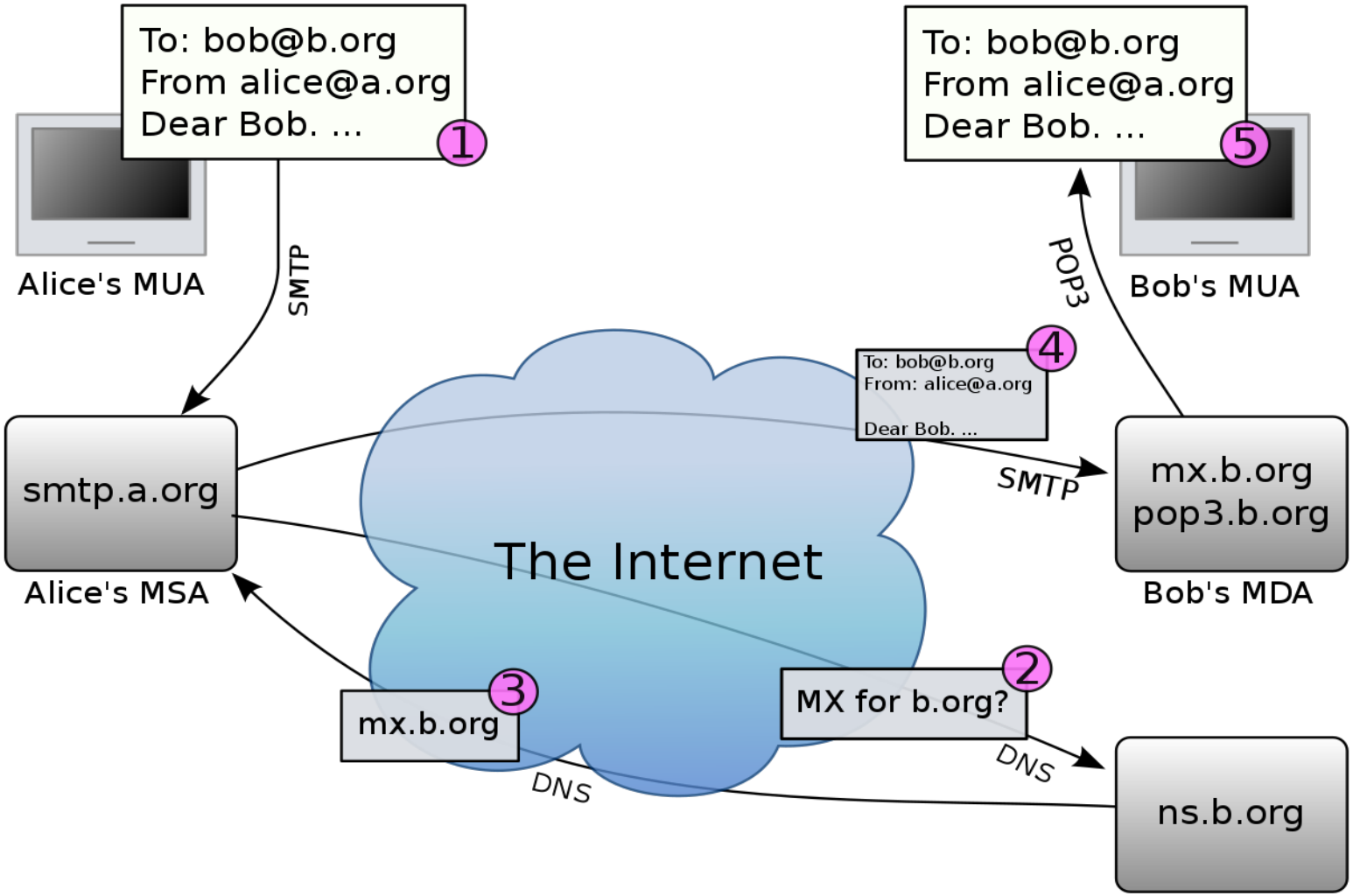
POP3 does not provide any means for a user to create remote folders and assign messages to folders;

IMAP

- 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
- Permits user agent to obtain components of messages, e.g., message header only;

Web-based email

- Hotmail (*a web-based email application*) appeared in mid 1990s;
- User communicates with its remote mailbox via HTTP.


To: bob@b.org
From alice@a.org
Dear Bob. ...
1
Alice's MUA
SMTP
smtp.a.org
Alice's MSA
The Internet
3
mx.b.org
DNS
2
MX for b.org?
DNS
ns.b.org
To: bob@b.org
From: alice@a.org
Dear Bob. ...
4
SMTP
mx.b.org
pop3.b.org
Bob's MDA
POP3
To: bob@b.org
From alice@a.org
Dear Bob. ...
5
Bob's MUA

[wiki: email]

Outline

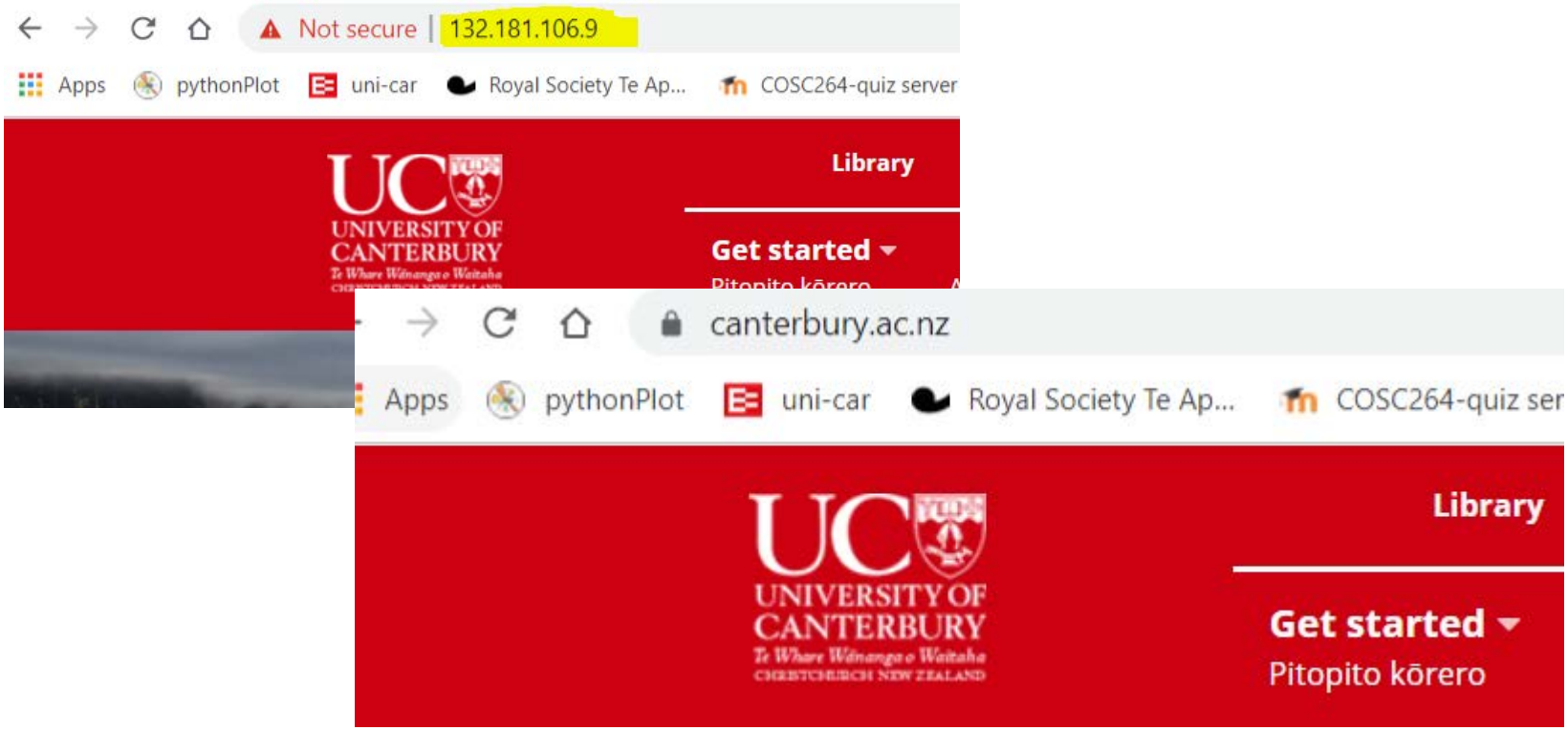
- Electronic Mail
  - SMTP, POP3, IMAP
- DNS

DNS: Domain Name System

Hosts can be identified by their IP addresses;

Routers prefer IP addresses;

People prefer the more mnemonic hostname identifier;



DNS

Domain Name System is:

- *a directory service;*
- *distributed database* implemented in hierarchy of DNS servers;
- *application-layer protocol* (*over UDP on port 53*) allowing hosts to query the distributed database;
- RFC 882/883/1034/1035, …

DNS

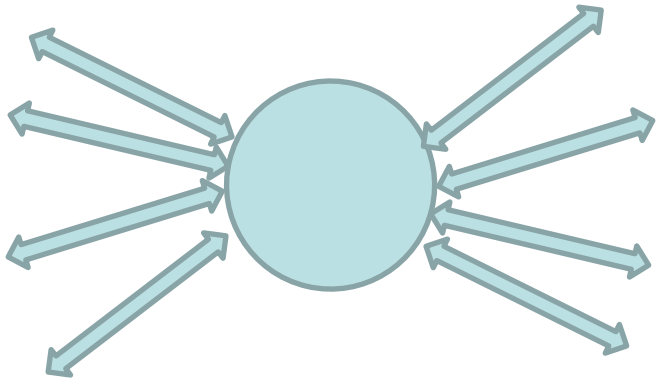
DNS services

- Hostname to IP address translation
  - commonly employed by HTTP, SMTP, etc.
  - E.g., www.northwestern.edu
- Host aliasing
  - Canonical and alias names
  - E.g., two aliases: dell.com, www.dell.com – canonical name is relay1.west.dell.com;
- Mail server aliasing
  - E.g., bob@hotmail.com, alias name of relay1.mailserv.Hotmail.com;
- ***Load distribution***
  - Replicated Web servers: set of IP addresses for one canonical name
  - E.g., cnn.com may have multiple replicated servers with different IP addresses;
  - The reply for “cnn.com” query might include a set of IP addresses but with different order (DNS rotation);
  - (ip1, ip2, ip3); (ip2, ip3, ip1); (ip3, ip1, ip2);
  - An HTTP client typically chooses the first IP addr.

Why not centralise DNS?

- single point of failure
- traffic volume
- distant centralised database
  - Cannot be close to ALL hosts;
- Maintenance
  - To be updated frequently;

A centralised DNS doesn't *scale!*



Some history

2.1. The history of domain names

The impetus for the development of the domain system was growth in the Internet:

- Host name to address mappings were maintained by the Network Information Center (NIC) in a single file (HOSTS.TXT) which was FTPed by all hosts [RFC-952, RFC-953]. The total network

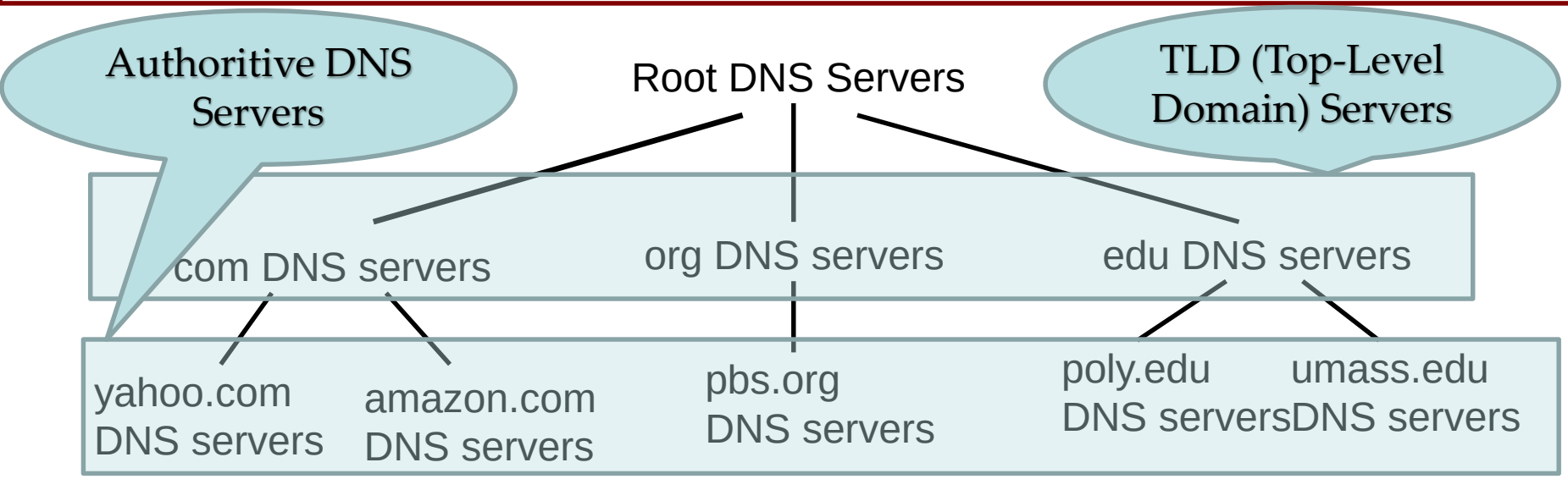
bandwidth consumed in distributing a new version by this scheme is proportional to the square of the number of hosts in the network, and even when multiple levels of FTP are used, the outgoing FTP load on the NIC host is considerable. Explosive growth in the number of hosts didn't bode well for the future.

- The network population was also changing in character. The timeshared hosts that made up the original ARPANET were being replaced with local networks of workstations. Local organizations were administering their own names and addresses, but had to wait for the NIC to change HOSTS.TXT to make changes visible to the Internet at large. Organizations also wanted some local structure on the name space.

- The applications on the Internet were getting more sophisticated and creating a need for general purpose name service.

The result was several ideas about name spaces and their management [IEN-116, RFC-799, RFC-819, RFC-830]. The proposals varied, but a common thread was the idea of a hierarchical name space, with the hierarchy roughly corresponding to organizational structure, and names using "." as the character to mark the boundary between hierarchy levels. A design using a distributed database and generalized resources was described in [RFC-882, RFC-883]. Based on experience with several implementations, the system evolved into the scheme described in this memo.

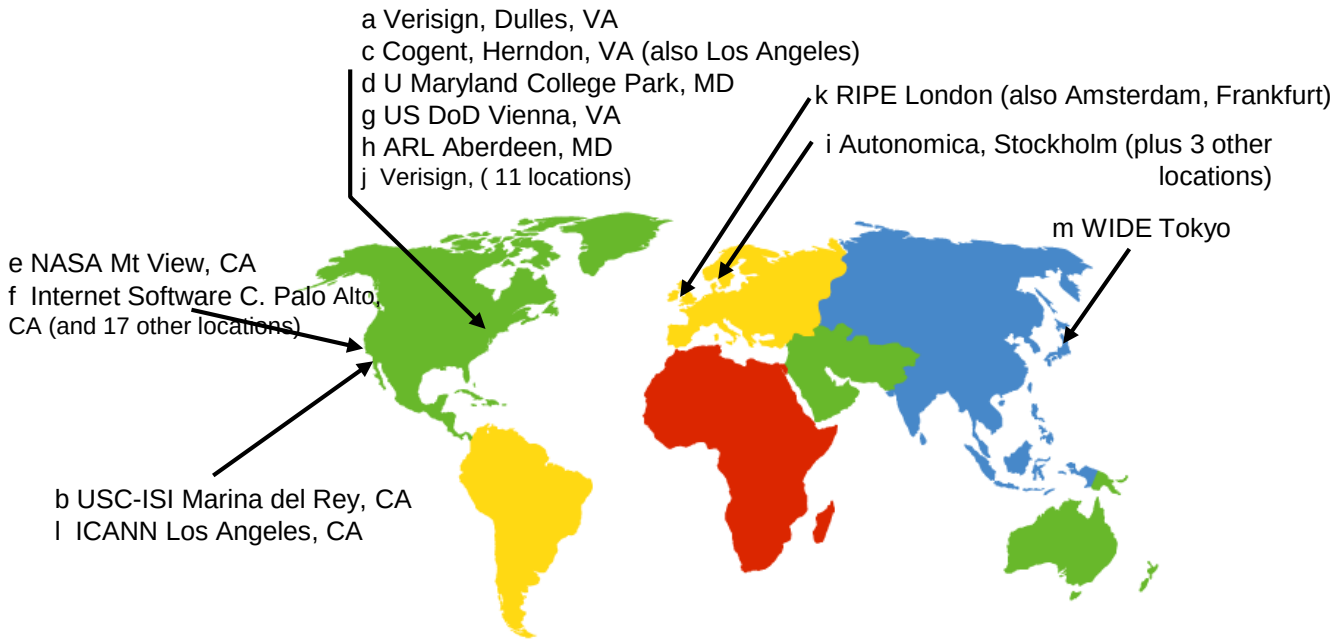
Distributed, Hierarchical Database



Client wants IP for www.amazon.com; 1st approx:

- Client queries a root server to find *com DNS server*
- Client queries *com DNS server* to get *amazon.com DNS server*
- Client queries *amazon.com DNS server* to get IP address for www.amazon.com

DNS: Root name servers



13 root name servers worldwide

each server is actually a cluster of replicated servers

TLD and Authoritative Servers

- Top-level domain (TLD) servers: responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp, nz.
- Authoritative DNS servers: organisation's DNS servers, providing authoritative hostname to IP mappings for organisation's servers (e.g., Web and mail).
  - Can be maintained by organisation or service provider

Local Name Server

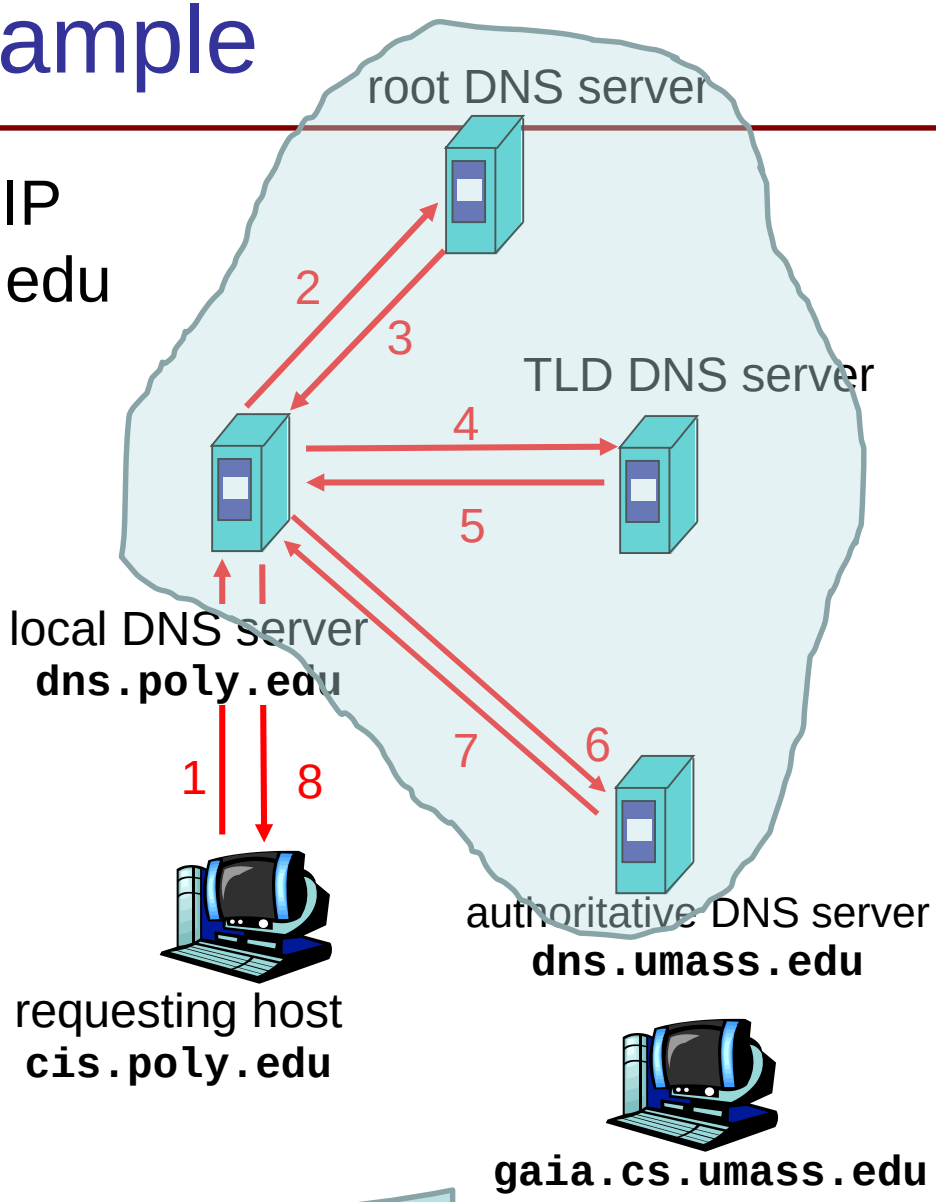
- Does not strictly belong to the hierarchy (*but is central to the DNS architecture*)
- Each ISP (residential ISP, company, etc) has one.
  - Also called "*default name server*"
- When a host makes a DNS query, query is sent to its local DNS server
  - Acts as a proxy, forwards query into the hierarchy.

Example

- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu

iterated query:
contacted server replies with name of server to contact

"*I don't know this name, but ask this server*"

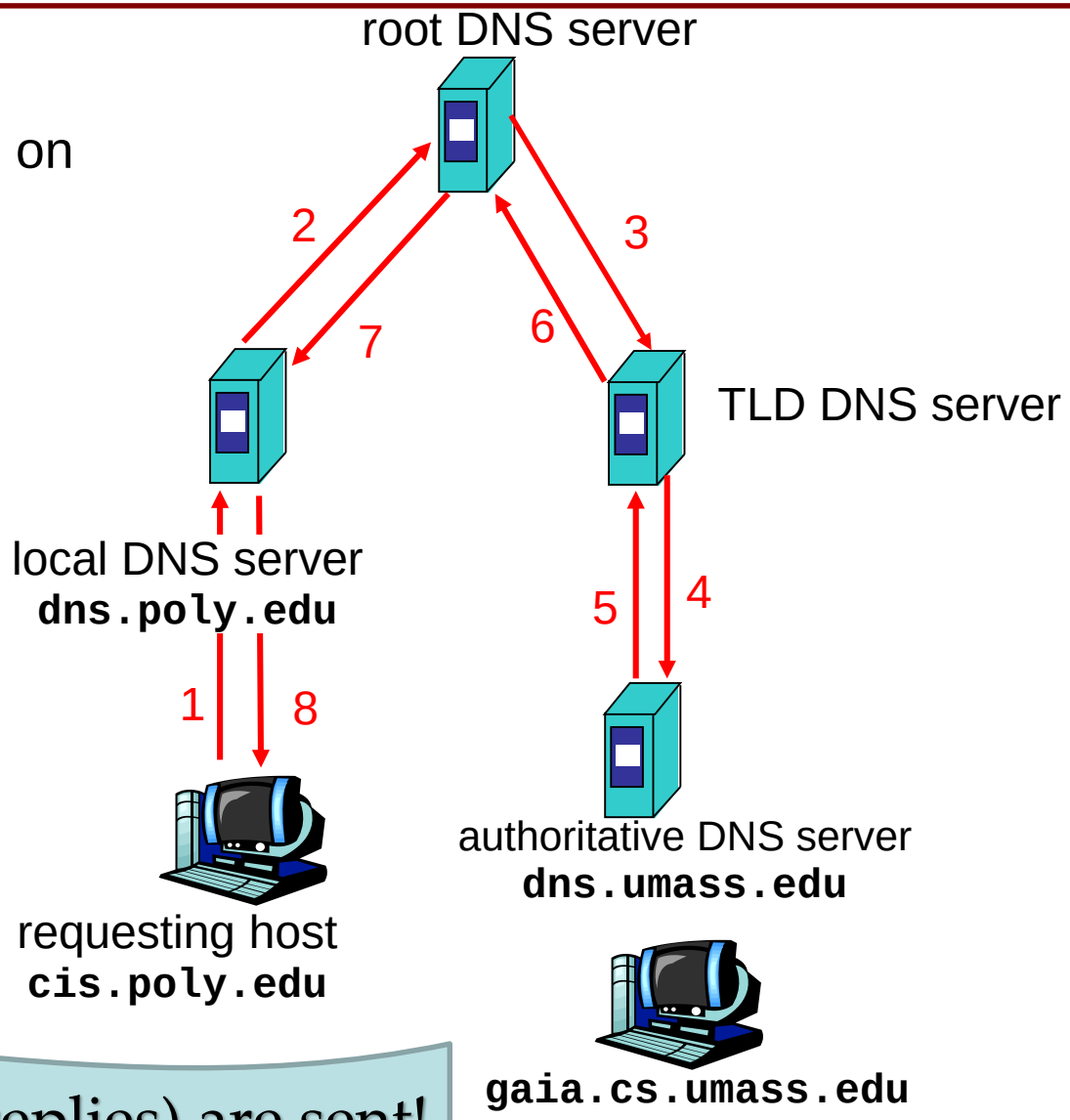


8 messages (4 queries + 4 replies) are sent!

Recursive queries

recursive query:

puts burden of name resolution on contacted name server

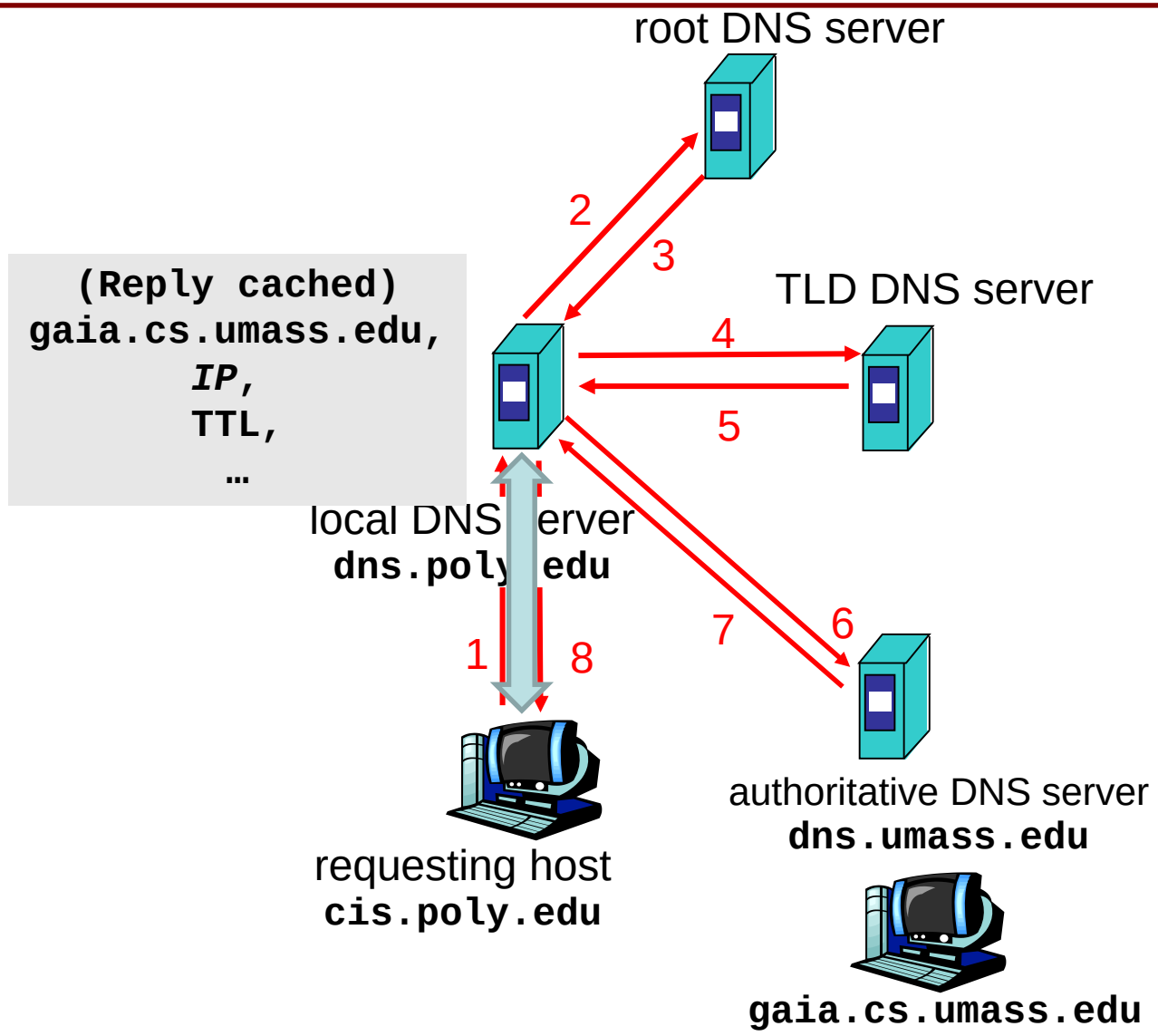


8 messages (4 queries + 4 replies) are sent!

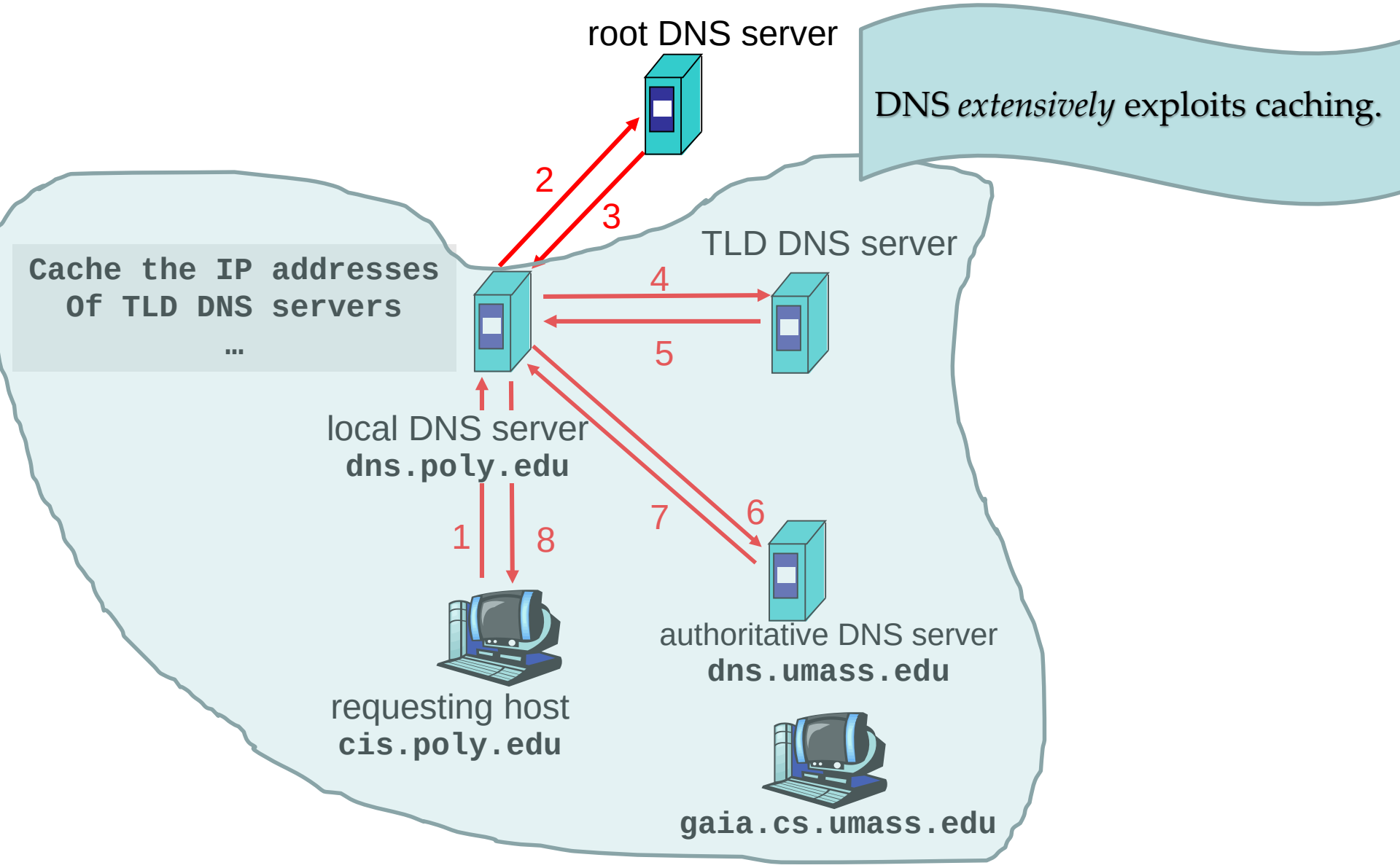
DNS: caching and updating records

- once (any) name server learns mapping, it *caches* mapping
  - the DNS server can provide the desired IP address even if it is not authoritative for that hostname;
- cache entries timeout (disappear) after some time
  - because hosts and mapping between host names and IP addresses *are by no means permanent;*
- TLD servers typically cached in local name servers
  - Thus root name servers not often visited

Get a reply from cached records



Bypass the root DNS



DNS records

DNS: distributed db storing resource records (RR)

RR format: **(name, value, type, ttl)**

Type=A

name is hostname
value is IP address

Type=NS

- **name** is domain (e.g. foo.com)
- **value** is hostname of authoritative name server for this domain

Type=CNAME

name is alias name for some "canonical" (the real) name
www.ibm.com is really servereast.backup2.ibm.com
value is canonical name

Type=MX

value is canonical name of mailserver associated with alias **name**

An example

name	value	type	TTL
canterbury.ac.nz	web.canterbury.ac.nz	CNAME	XXXX
www.canterbury.ac.nz	web.canterbury.ac.nz	CNAME	XXXX
web.canterbury.ac.nz	132.181.106.9	A	XXXX
canterbury.ac.nz	itdns.canterbury.ac.nz	NS	XXXX
canterbury.ac.nz	mail.canterbury.ac.nz	MX	XXXX
www.google.co.nz	172.217.167.99	A	86400
itdns.canterbury.ac.nz	132.181.2.225	A	XXXX
mail.canterbury.ac.nz	132.181.3.225	A	XXXX

DNS protocol, messages

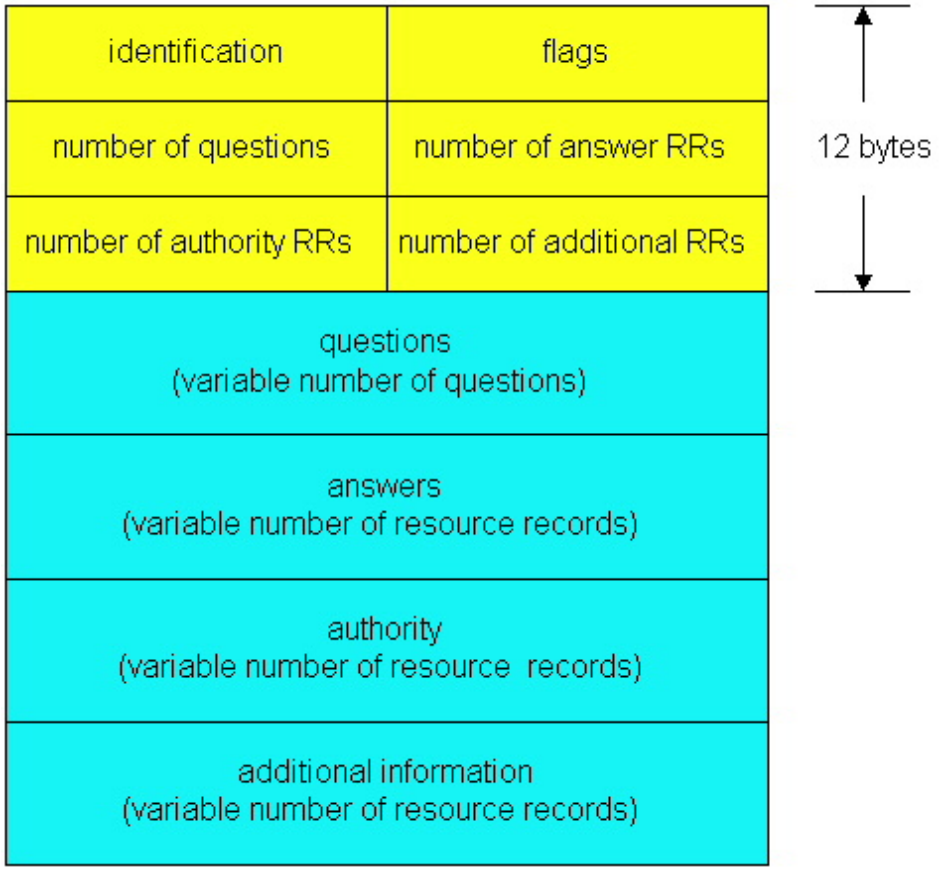
DNS protocol : *query* and *reply* messages, both with same *message format*

msg header

identification: 16 bit # for query, reply to query uses same #

flags:

- Query (0) or reply (1)
- recursion desired (in a query)
- recursion available (in a reply)
- reply is authoritative (in a reply)



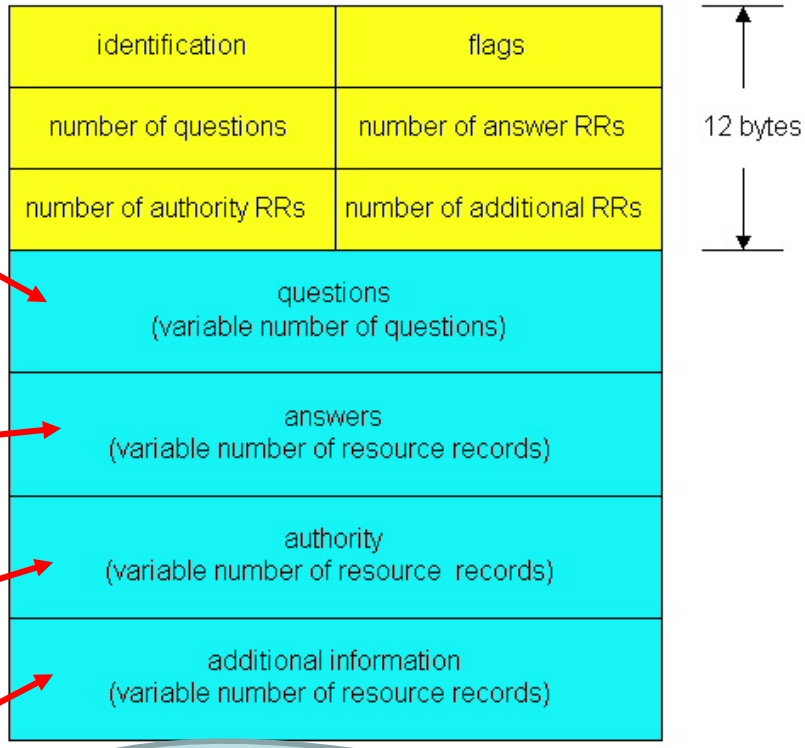
DNS protocol, messages

Name, type fields for a query
(Name, Type)
e.g., (ibm.com, CNAME)

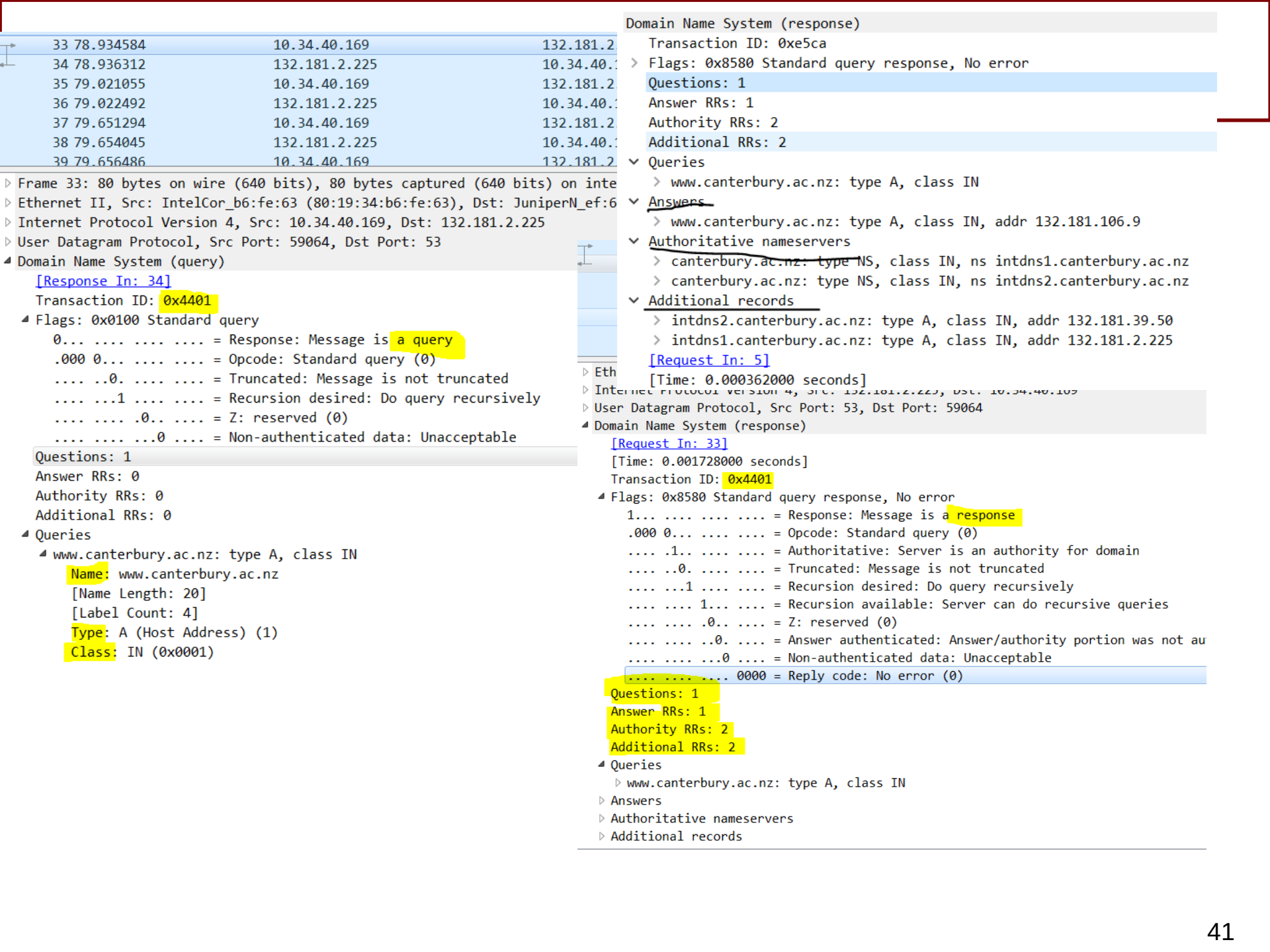
RRs in reponse to query
(Type, Value, TTL)
(CNAME, serv.bckup.ibm.com,5)

records for authoritative servers

additional "helpful" info that may be used
e.g., (serv.bckup.ibm.com, 254.24.54.42,A)



Multiple RRs possible for the same name!

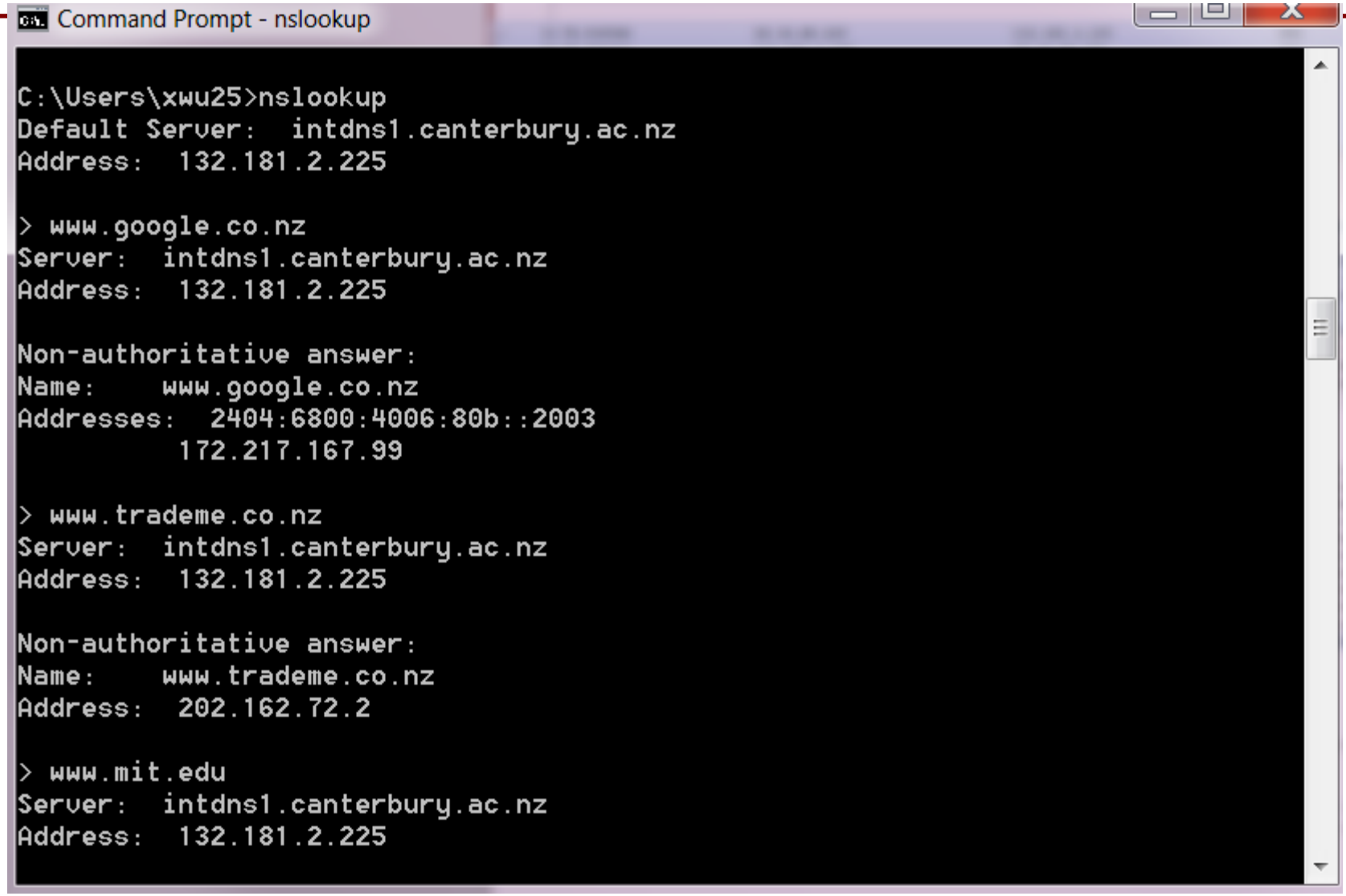

33 78.934584 10.34.40.169 132.181.2
34 78.936312 132.181.2.225 10.34.40.1
35 79.021055 10.34.40.169 132.181.2
36 79.022492 132.181.2.225 10.34.40.1
37 79.651294 10.34.40.169 132.181.2
38 79.654045 132.181.2.225 10.34.40.1
39 79.656486 10.34.40.169 132.181.2
Frame 33: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on inte
Ethernet II, Src: IntelCor_b6:fe:63 (80:19:34:b6:fe:63), Dst: JuniperN_ef:6
Internet Protocol Version 4, Src: 10.34.40.169, Dst: 132.181.2.225
User Datagram Protocol, Src Port: 59064, Dst Port: 53
Domain Name System (query)
[Response In: 34]
Transaction ID: 0x4401
Flags: 0x0100 Standard query
0... .... = Response: Message is a query
.000 0... .... = Opcode: Standard query (0)
.... ..0. .... = Truncated: Message is not truncated
....1 .... = Recursion desired: Do query recursively
.... .0.. .... = Z: reserved (0)
....0 .... = Non-authenticated data: Unacceptable
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
www.canterbury.ac.nz: type A, class IN
Name: www.canterbury.ac.nz
[Name Length: 20]
[Label Count: 4]
Type: A (Host Address) (1)
Class: IN (0x0001)
Domain Name System (response)
Transaction ID: 0xe5ca
Flags: 0x8580 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 2
Additional RRs: 2
Queries
www.canterbury.ac.nz: type A, class IN
Answers
www.canterbury.ac.nz: type A, class IN, addr 132.181.106.9
Authoritative nameservers
canterbury.ac.nz: type NS, class IN, ns intdns1.canterbury.ac.nz
canterbury.ac.nz: type NS, class IN, ns intdns2.canterbury.ac.nz
Additional records
intdns2.canterbury.ac.nz: type A, class IN, addr 132.181.39.50
intdns1.canterbury.ac.nz: type A, class IN, addr 132.181.2.225
[Request In: 5]
[Time: 0.000362000 seconds]
User Datagram Protocol, Src Port: 53, Dst Port: 59064
Domain Name System (response)
[Request In: 33]
[Time: 0.001728000 seconds]
Transaction ID: 0x4401
Flags: 0x8580 Standard query response, No error
1... .... = Response: Message is a response
.000 0... .... = Opcode: Standard query (0)
.... .1.. .... = Authoritative: Server is an authority for domain
.... ..0. .... = Truncated: Message is not truncated
....1 .... = Recursion desired: Do query recursively
.... 1... .... = Recursion available: Server can do recursive queries
.... .0.. .... = Z: reserved (0)
....0. .... = Answer authenticated: Answer/authority portion was not au
....0 .... = Non-authenticated data: Unacceptable
.... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 2
Additional RRs: 2
Queries
www.canterbury.ac.nz: type A, class IN
Answers
Authoritative nameservers
Additional records

Inserting records into DNS

- Example: just created startup “Network Utopia”
- Register name networkuptopia.com at a registrar (e.g., “Network Solutions”)
  - Need to provide registrar with names and IP addresses of your authoritative name server (primary and secondary)
  - Registrar inserts two RRs into the com TLD server for *each* authoritative name server:

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

- Put in *your* authoritative server Type A record for www.networkuptopia.com and Type MX record for networkutopia.com


Command Prompt - nslookup
C:\Users\xwu25>nslookup
Default Server: intdns1.canterbury.ac.nz
Address: 132.181.2.225
> www.google.co.nz
Server: intdns1.canterbury.ac.nz
Address: 132.181.2.225
Non-authoritative answer:
Name: www.google.co.nz
Addresses: 2404:6800:4006:80b::2003
          172.217.167.99
> www.trademe.co.nz
Server: intdns1.canterbury.ac.nz
Address: 132.181.2.225
Non-authoritative answer:
Name: www.trademe.co.nz
Address: 202.162.72.2
> www.mit.edu
Server: intdns1.canterbury.ac.nz
Address: 132.181.2.225

Summary

- Electronic Mail
  - SMTP, POP3, IMAP
- DNS

References

- [KR3] James F. Kurose, Keith W. Ross, *Computer networking: a top-down approach featuring the Internet*, 3rd edition.
- [PD5] Larry L. Peterson, Bruce S. Davie, *Computer networks: a systems approach*, 5th edition
- [TW5] Andrew S. Tanenbaum, David J. Wetherall, *Computer network*, 5th edition
- [LHBi]Y-D. Lin, R-H. Hwang, F. Baker, *Computer network: an open source approach*, International edition

Acknowledgements

- All slides are developed based on slides from the following two sources:
  - Dr DongSeong Kim's slides for COSC264, University of Canterbury;
  - Prof Aleksandar Kuzmanovic's lecture notes for CS340,Northwestern University, https://users.cs.northwestern.edu/~akuzma/classes/CS340-w05/lecture_notes.htm