

---

# Network Applications: DNS; Network Programming: UDP, TCP

**Qiao Xiang, Congming Gao**

<https://sngroup.org.cn/courses/cnns-xmuf23/index.shtml>

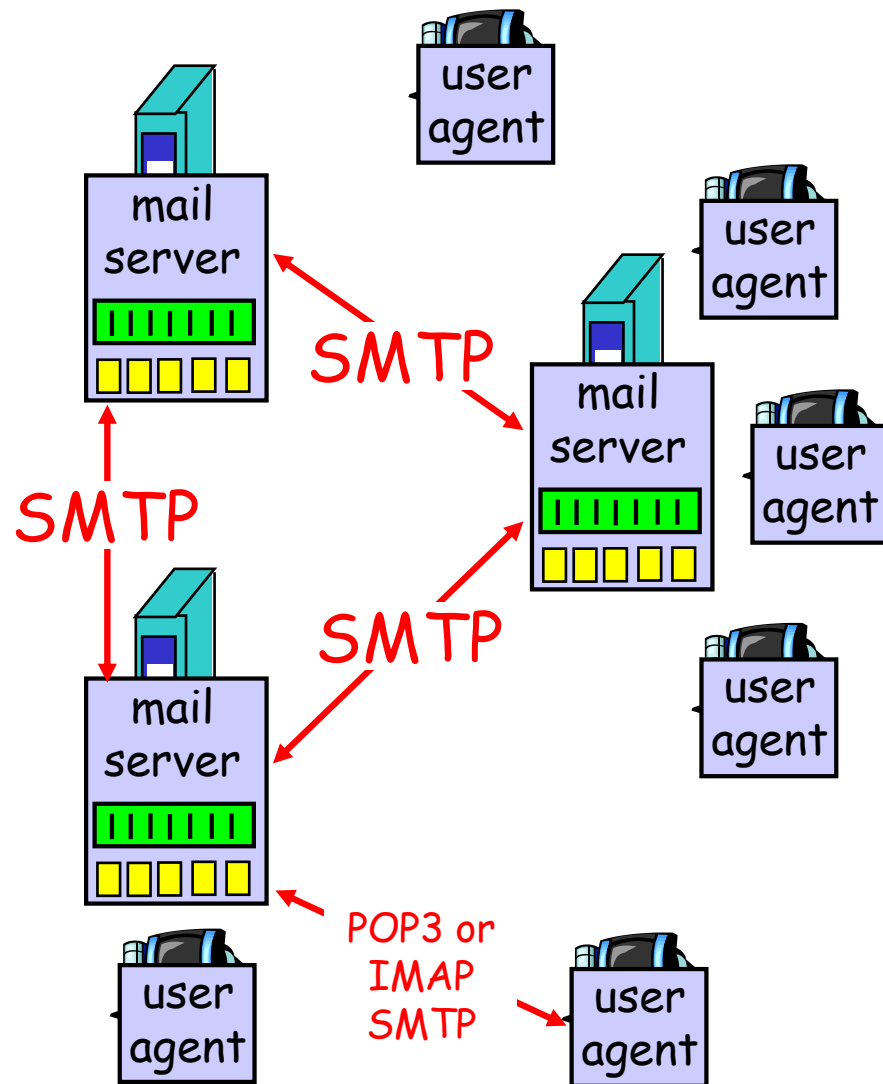
09/28/2023

# Outline

---

- ❑ Admin. and recap
- ❑ Basic network applications
  - DNS
- ❑ Java in a Nutshell
- ❑ Network application programming

# Recap: Email App



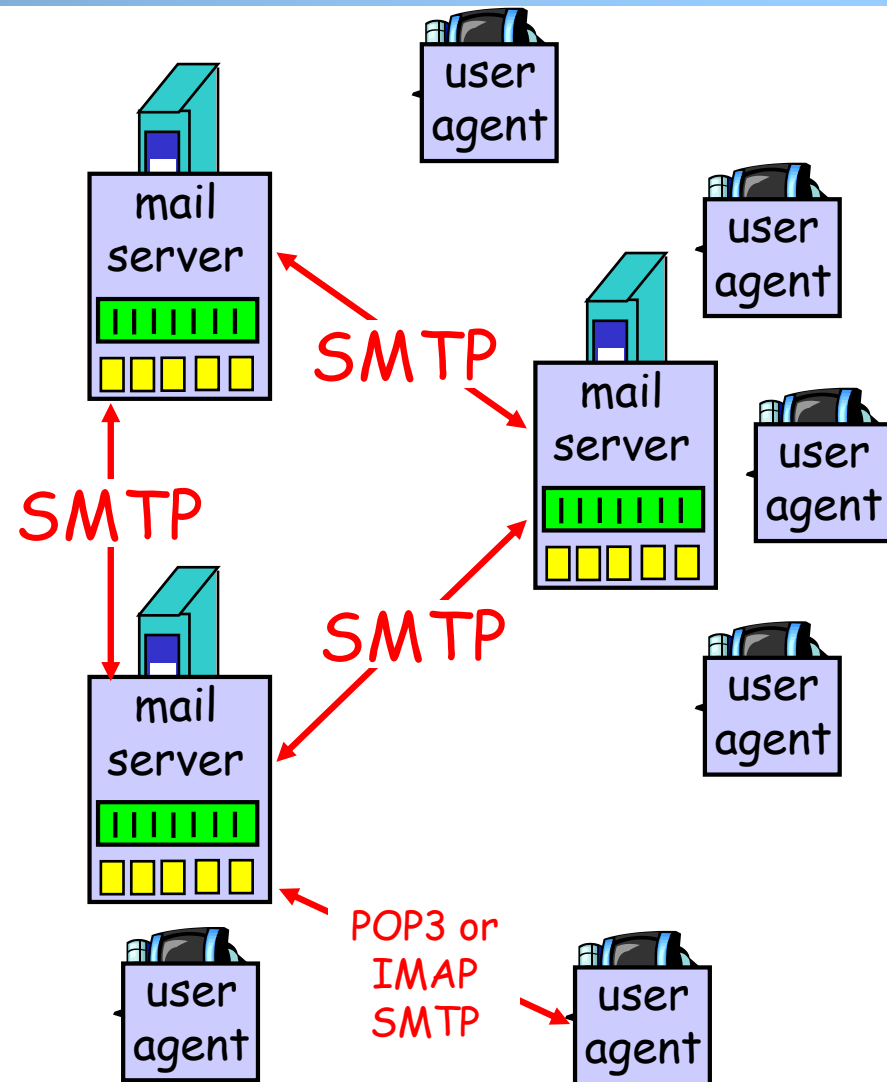
Some key design features of Email

- **Separate protocols for different functions**
  - email access (e.g., POP3, IMAP)
  - email transport (SMTP)
- **Separation of envelop and message body (end-to-end arguments)**
  - envelop: simple/basic requests to implement transport control;
  - message body: fine-grain control through ASCII header and message body
    - MIME type as self-describing data type
- **Status code** in response makes message easy to parse

# Evaluation of SMTP/POP/IMAP

Key questions to ask about a C-S application

- extensible?
- scalable?
- robust?
- security?



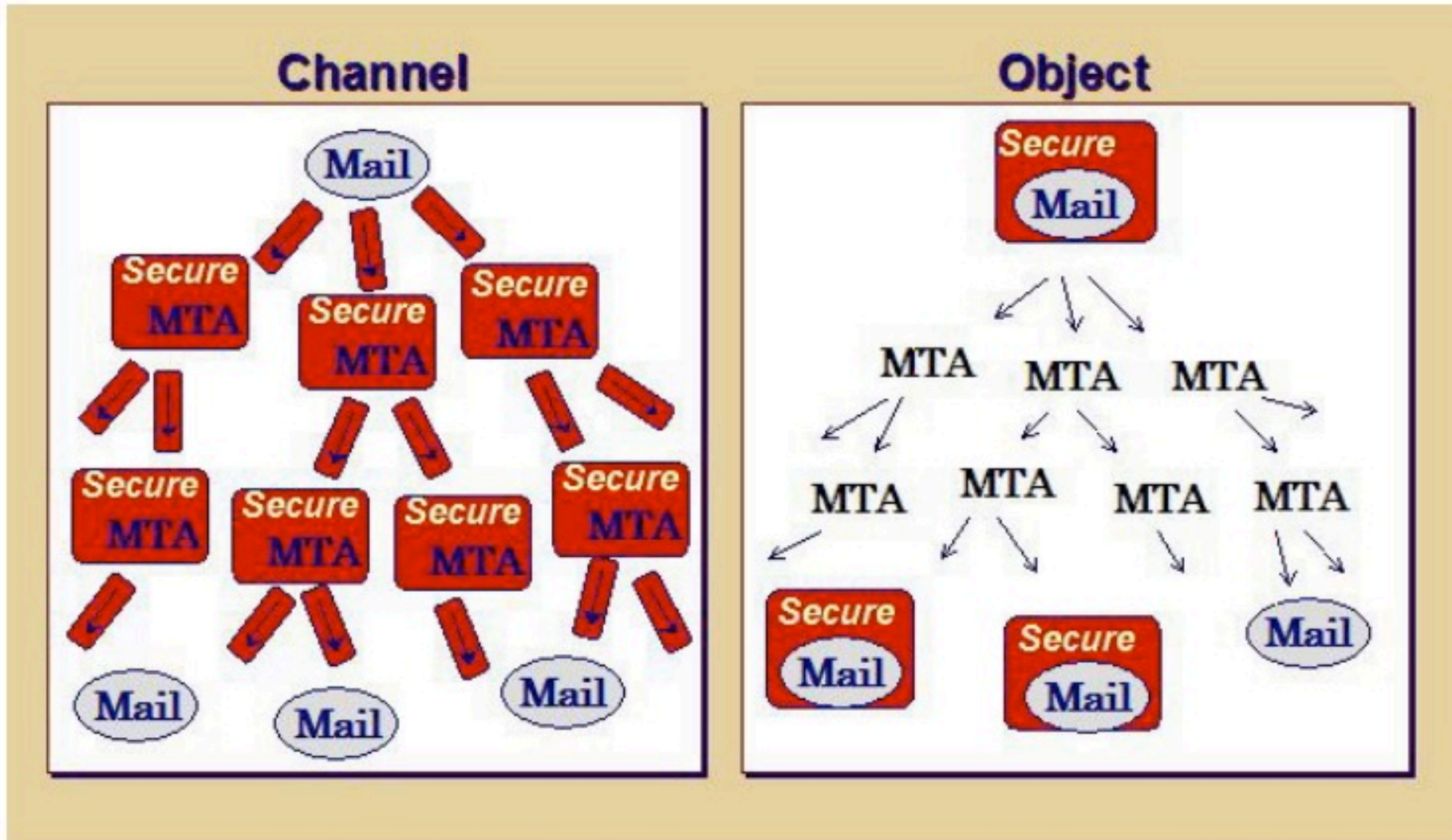
# Detection Methods Used by GMail

---

- ❑ Known phishing scams
- ❑ Message from unconfirmed sender identity
- ❑ Message you sent to Spam/similarity to suspicious messages
- ❑ Administrator-set policies

<https://support.google.com/mail/answer/1366858?hl=en>

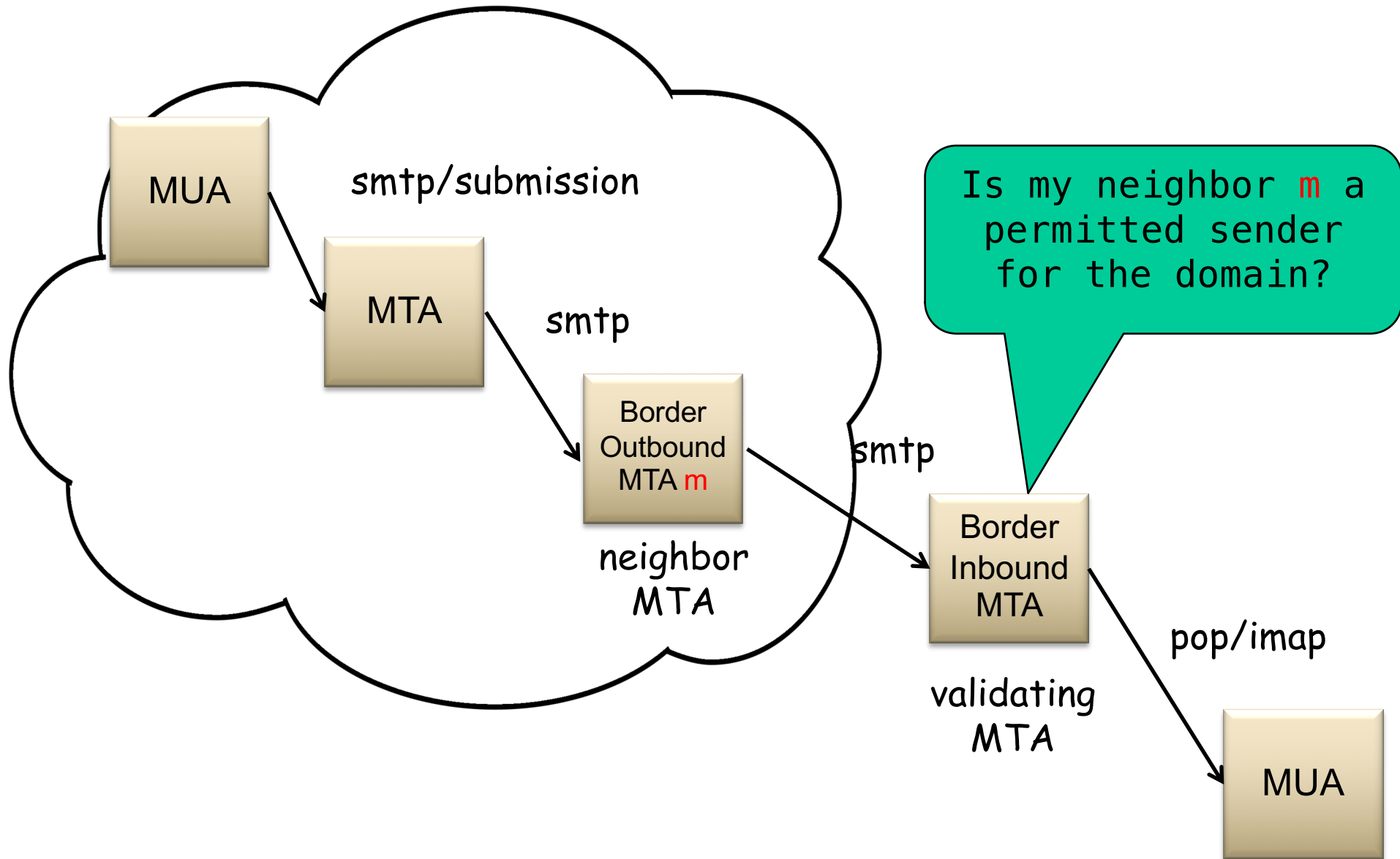
# Email Authentication Approaches



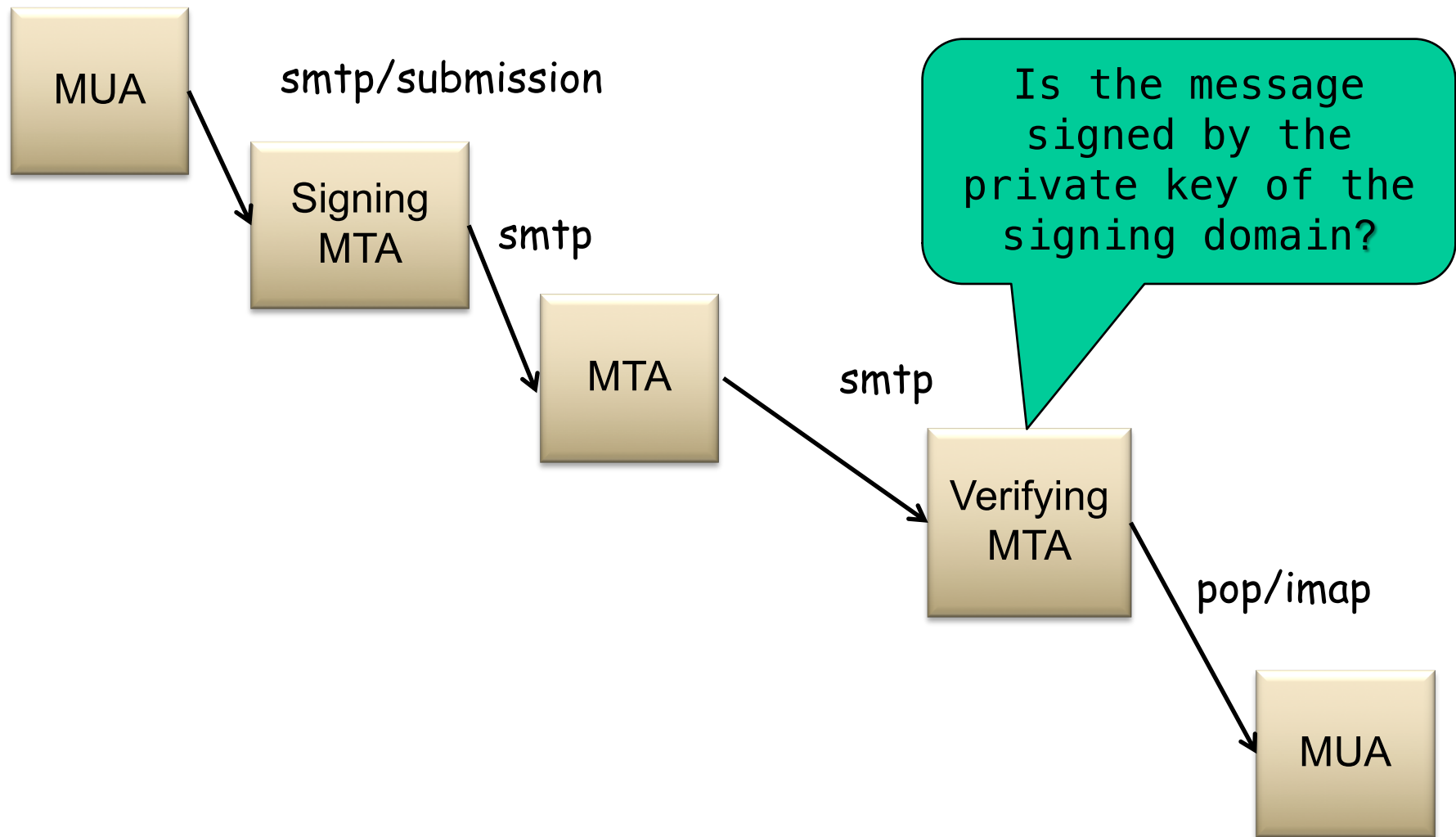
Sender Policy Frame (SPF)

DomainKeys Identified Mail (DKIM)  
Authenticated Results Chain (ARC)

# Sender Policy Framework (SPF RFC7208)



# DomainKeys Identified Mail (DKIM)





# Summary: Some Key Remaining Issues about Email

---

- ❑ Basic: How to find the email server of a domain?
- ❑ Scalability/robustness: how to find multiple servers for the email domain?
- ❑ Security
  - SPF: How does SPF know if its neighbor MTA is a permitted sender of the domain?
  - DKIM: How does DKIM retrieve the public key of the author domain?

# Outline

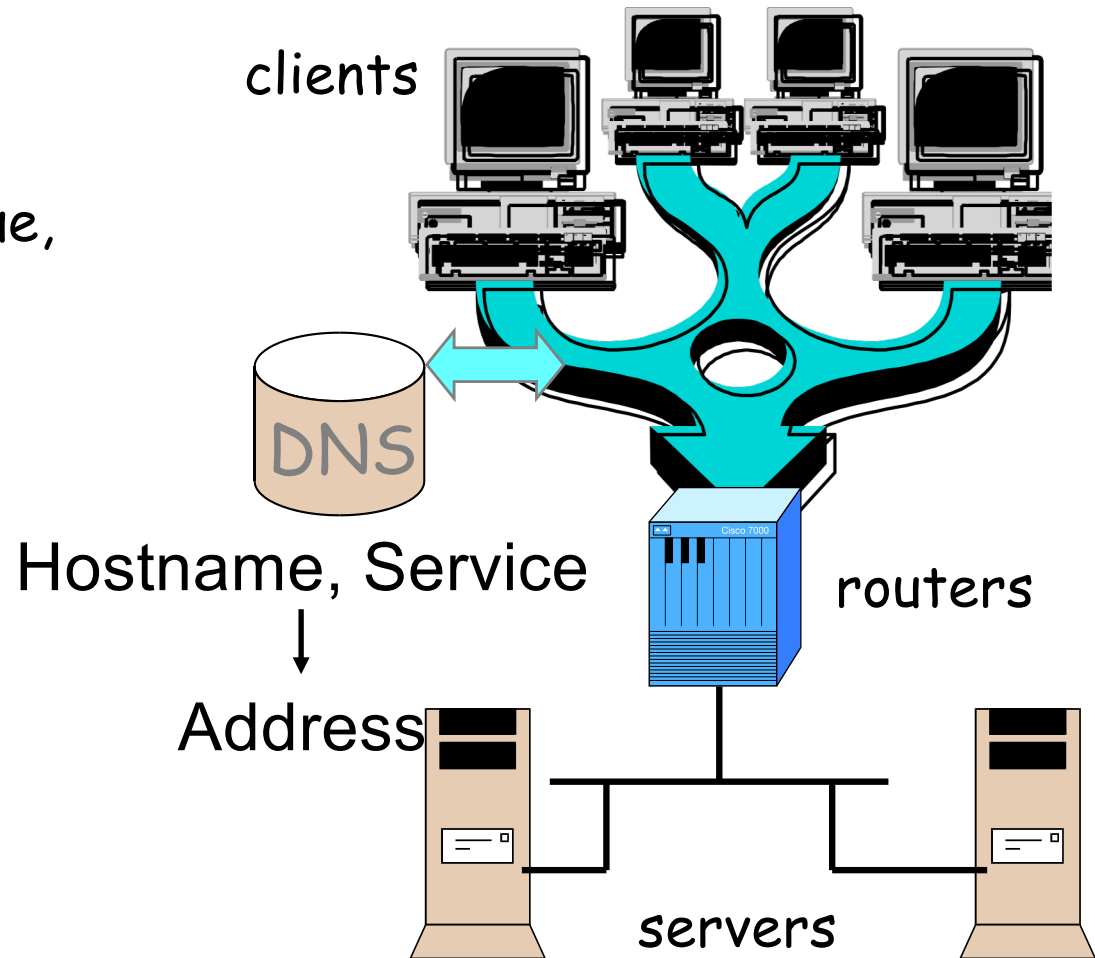
---

- Admin. and recap
- Network applications
  - Email
  - *DNS*

# DNS: Domain Name System

## □ Function

- map between (domain name, service) to value, e.g.,
  - (xmu.edu.cn, addr)  
-> 210.34.0.35
  - (xmu.edu.cn, email)  
-> cmsn1.xmu.edu.cn



# DNS Records

DNS: stores resource records (RR)

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

## □ Type=A

- name is hostname
- value is IP address

## □ Type=NS

- name is domain (e.g. xmu.edu.cn)
- value is the name of the authoritative name server for this domain

## □ Type=TXT

- general txt

## □ Type=CNAME

- name is an alias of a “canonical” (real) name
- value is canonical name

## □ Type=MX

- value is hostname of mail server associated with name

## □ Type=SRV

- general extension for services

## □ Type=PTR

- a pointer to another name 12

# Discussion

---

- Can DNS handle multiple values for the same (name, service)?

# Try DNS: Examples

---

- ❑ `dig <name> <type>`
  - Try `xmu.edu.cn` / others and various types
  
- ❑ `dig <domain> txt` to retrieve spf

<http://www.zytrax.com/books/dns/ch9/spf.html>

# Observations

---

- ❑ MX can return multiple servers
- ❑ DNS may rotate the servers in answer
- ❑ Address can also return multiple addresses
- ❑ SPF is encoded as the txt type

# Outline

---

- Admin. and recap
- DNS
  - High-level design
  - *Details*



# DKIM Example

---

- Send email from hotmail and check message

S: +OK sina pop3 server ready

C: user xmucnns

S: +OK welcome to sina mail

C: pass 334f5605df1504f9

S: +OK 4 messages (32377 octets)

# DKIM Example

---

- DKIM / ARC:

Msg: ARC-Message-Signature: i=1; a=rsa-sha256;  
c=relaxed/relaxed; d=microsoft.com; s=arcselector9901;  
h=From:Date:Subject:Message-ID:Content-Type:MIME-  
Version;  
bh=bO91TxHI+4MjgAusrfg0EWGiDmvQ5hZRZ/aqb1MKLY8  
=; ...

DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;  
d=hotmail.com; s=selector1; h=From:Date:Subject:Message-  
ID:Content-Type:MIME-Version:X-MS-Exchange-  
SenderADCheck;...

- Query: dig arcselector9901.\_domainkey.microsoft.com txt

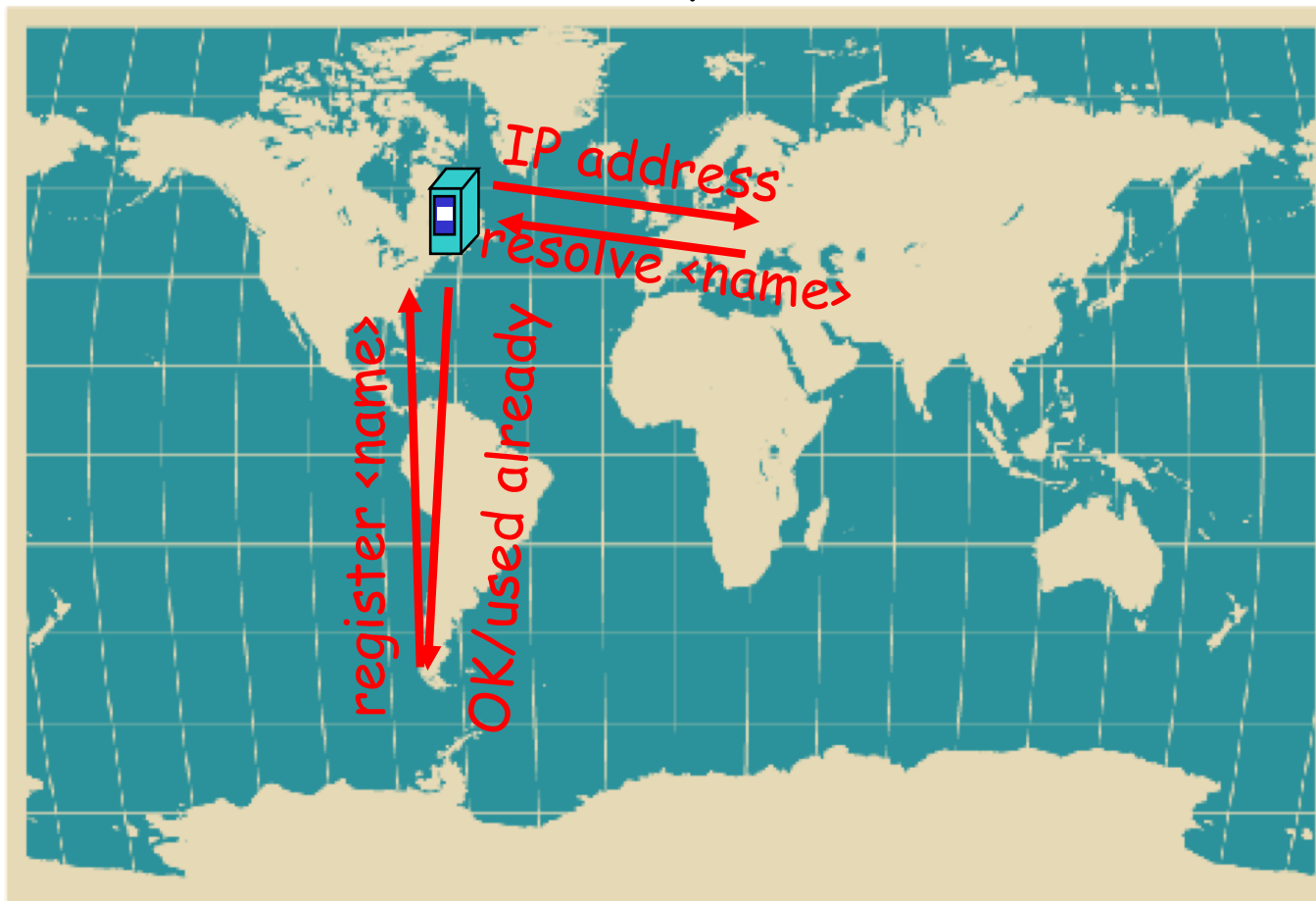
- DKIM introduces a session key to allow multiple public keys

- <session>.\_domainkey.<domain>

# DNS Design: Dummy Design

- DNS itself can be considered as a client-server system as well
- How about a dummy design: introducing one super Internet DNS server?

THE DNS server of the Internet



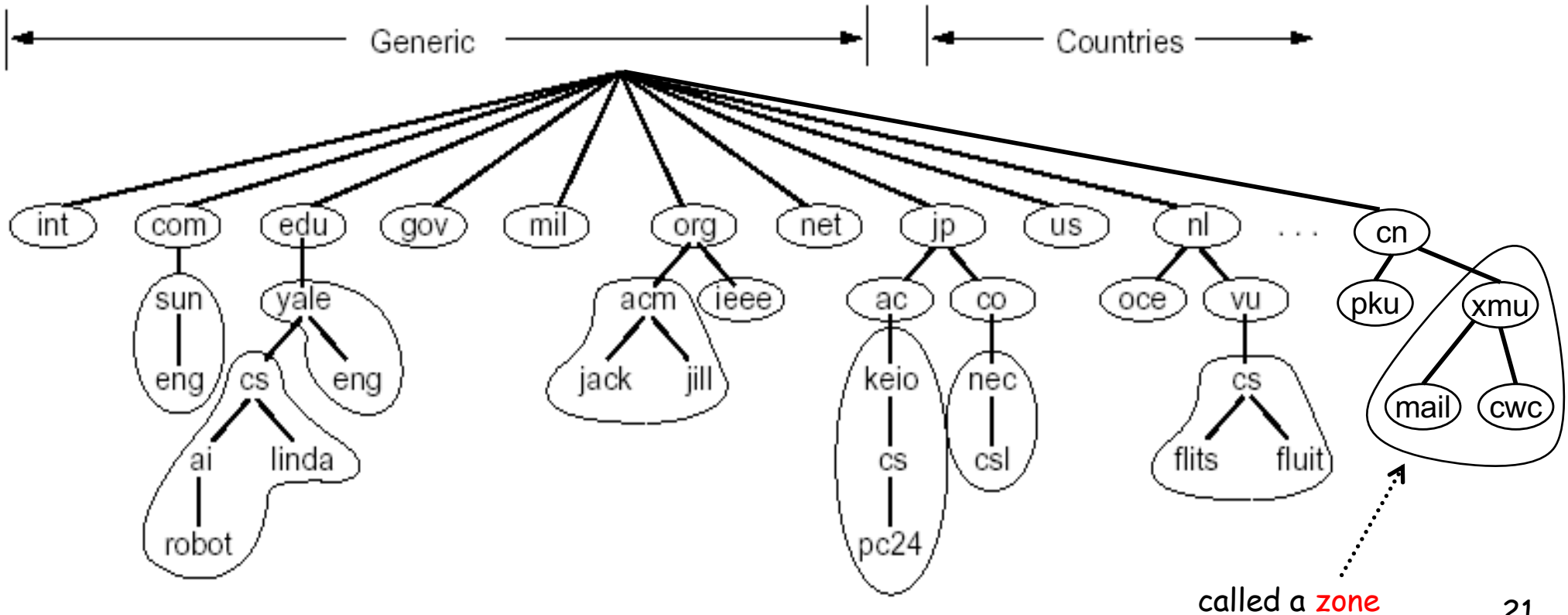
# Problems of a Single DNS Server

---

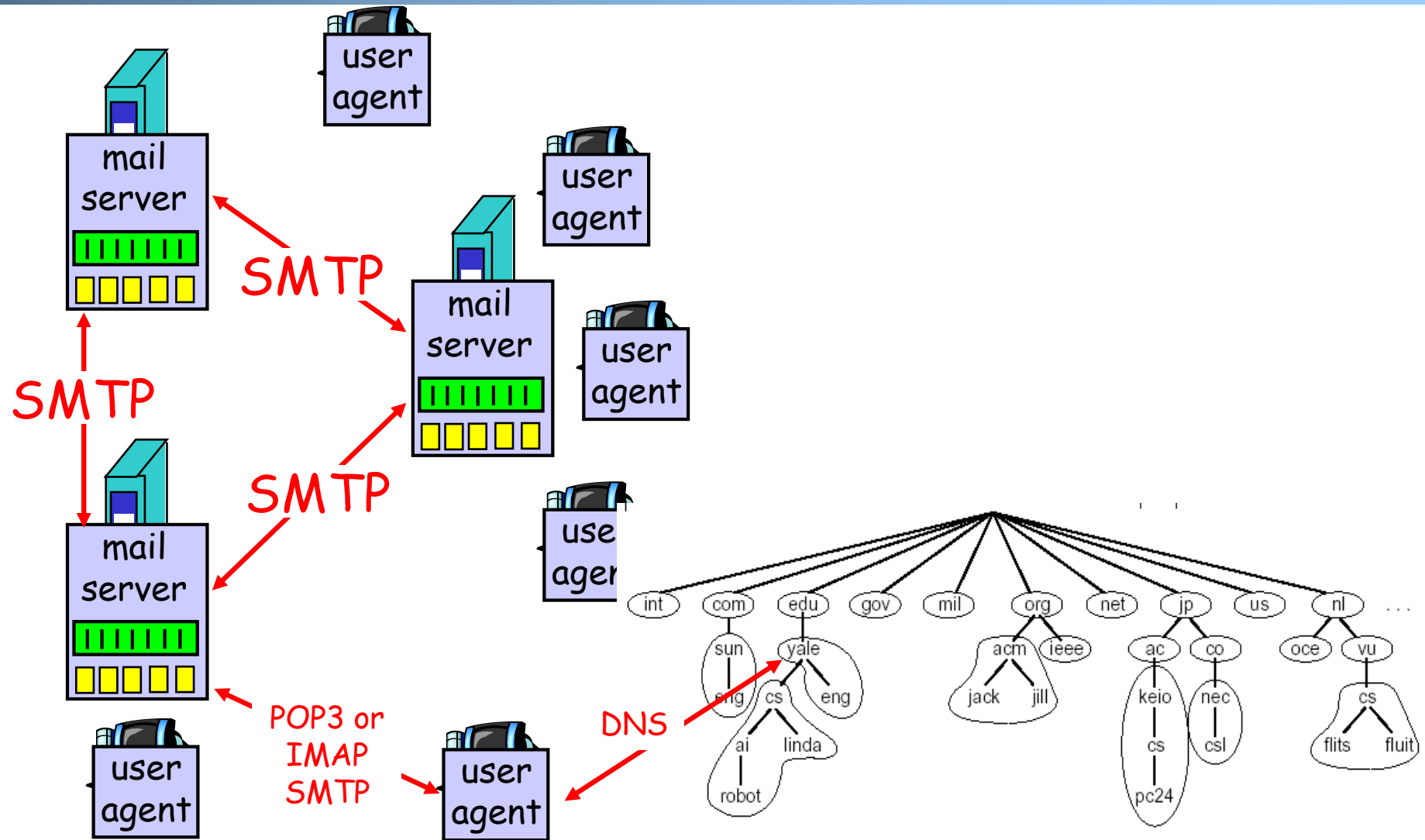
- ❑ Scalability and robustness bottleneck
- ❑ Administrative bottleneck

# DNS: Distributed Management of the Domain Name Space

- A distributed database managed by authoritative name servers
  - divided into zones, where each zone is a sub-tree of the global tree
  - each zone has its own **authoritative name servers**
  - an authoritative name server of a zone may **delegate** a subset (i.e. a sub-tree) of its zone to another name server



# Email Architecture + DNS



# Root Zone and Root Servers

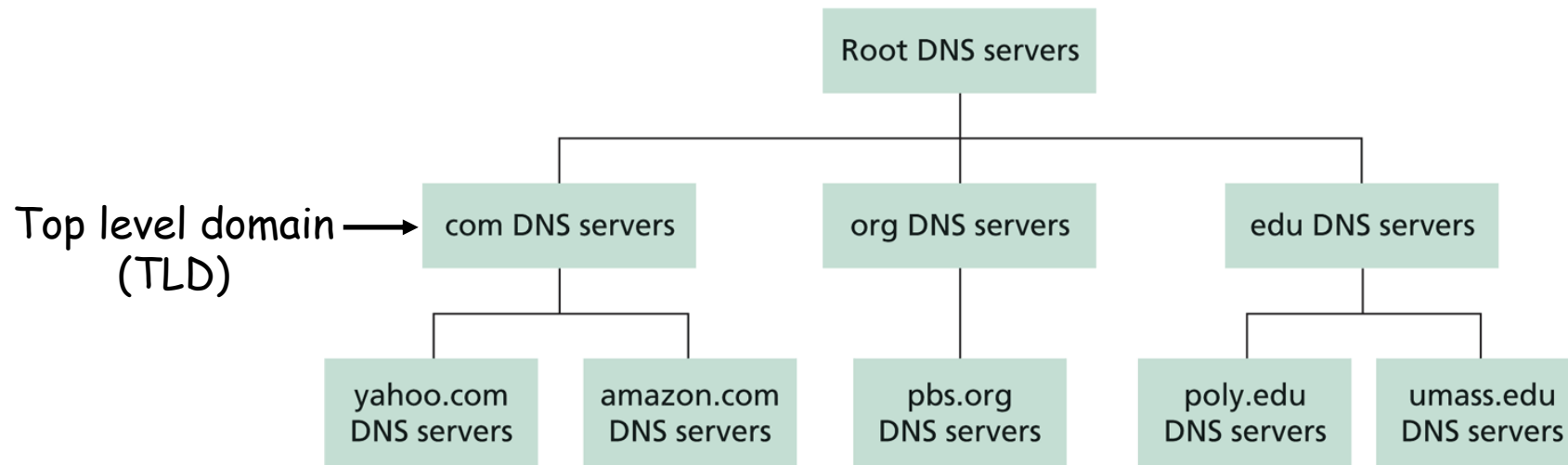
- The root zone is managed by the root name servers
  - 13 root name servers worldwide



See <http://root-servers.org/> for more details

# Linking the Name Servers

- ❑ Each name server knows the addresses of the root servers
- ❑ Each name server knows the addresses of its immediate children (i.e., those it delegates)



Q: how to query a hierarchy?



# DNS Message Flow: Two Types of Queries

---

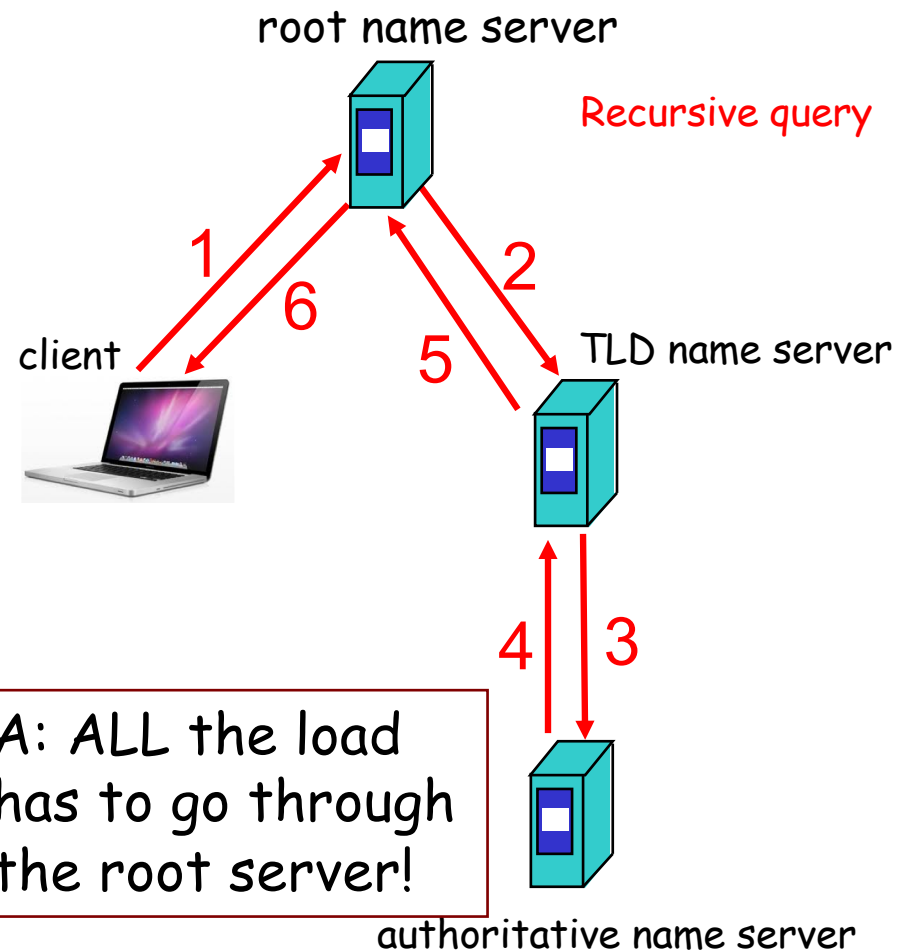
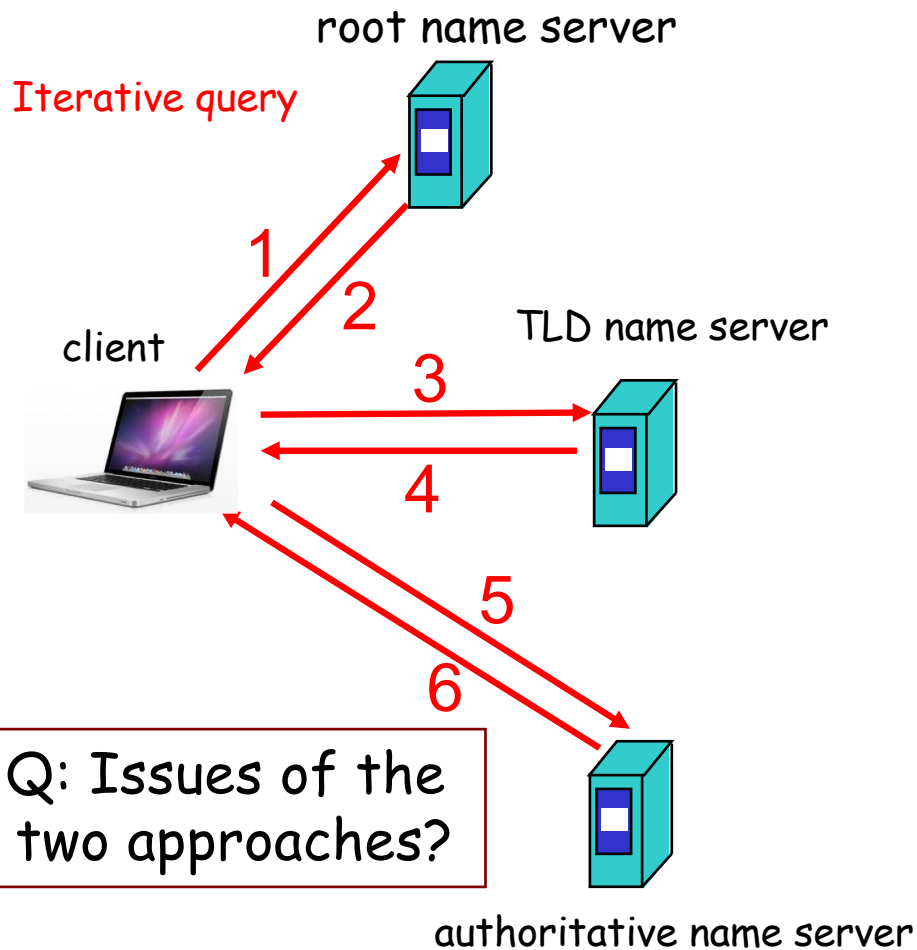
## Recursive query:

- ❑ The contacted name server resolves the name completely

## Iterated query:

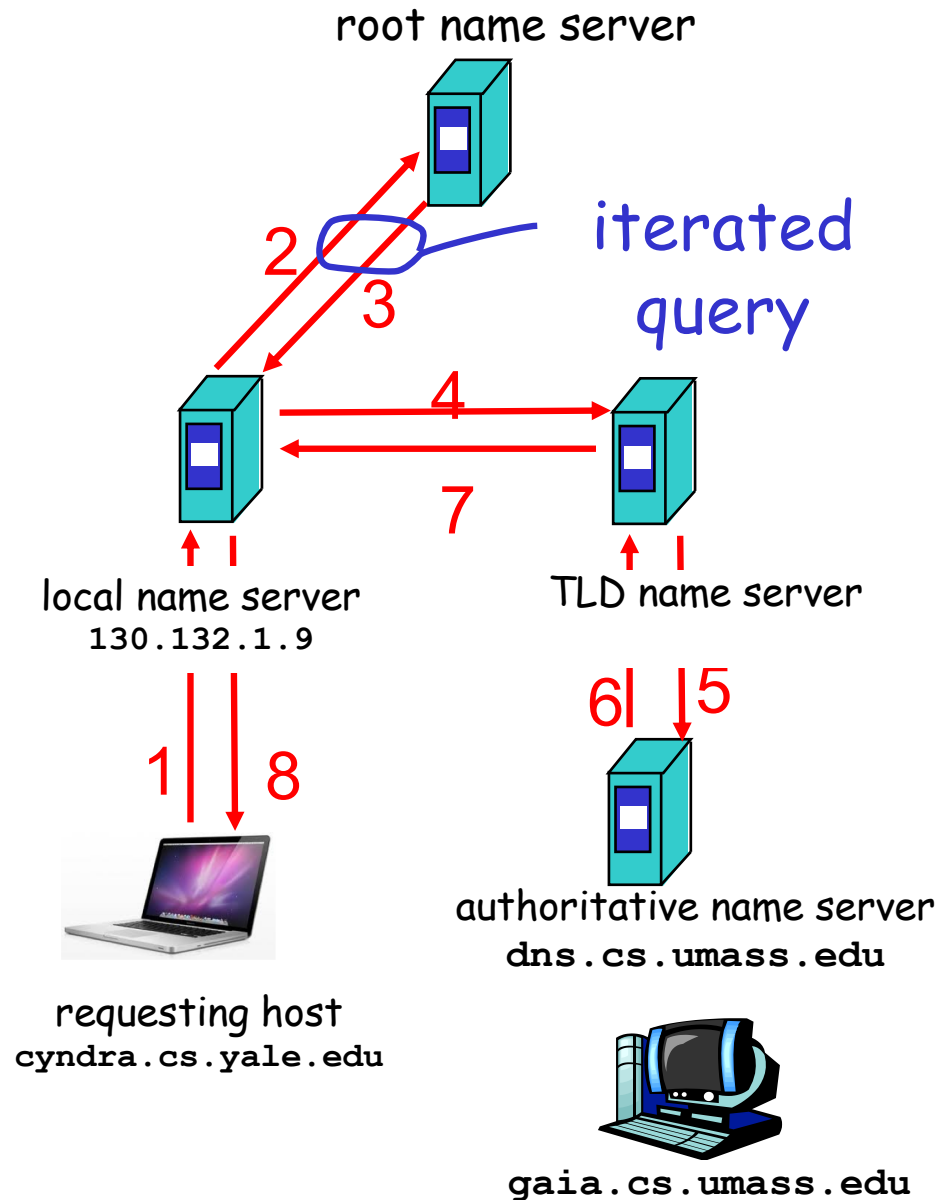
- ❑ Contacted server replies with name of server to contact
  - “I don’t know this name, but ask this server”

# Two Extreme DNS Message Flows



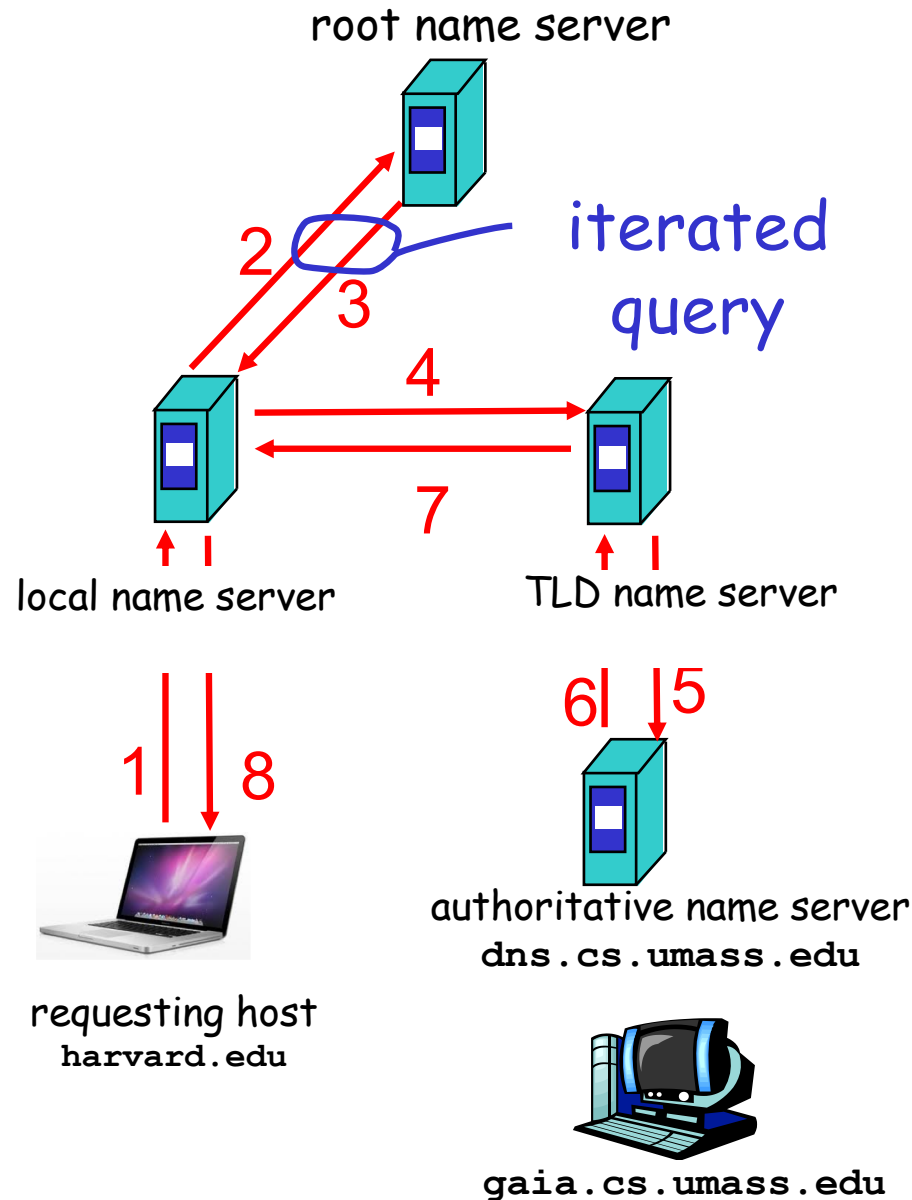
# Typical DNS Message Flow: The Hybrid Case

- Host knows only local name server
- Local name server is learned from DHCP, or configured, e.g. /etc/resolv.conf
- Local DNS server helps clients resolve DNS names



# Typical DNS Message Flow: The Hybrid Case

- Host knows only local name server
- Local name server is learned from DHCP, or configured, e.g. /etc/resolv.conf
- Local DNS server helps clients resolve DNS names
- Benefits of local name servers (often called **resolvers**)
  - simplifies client
  - caches/reuses results



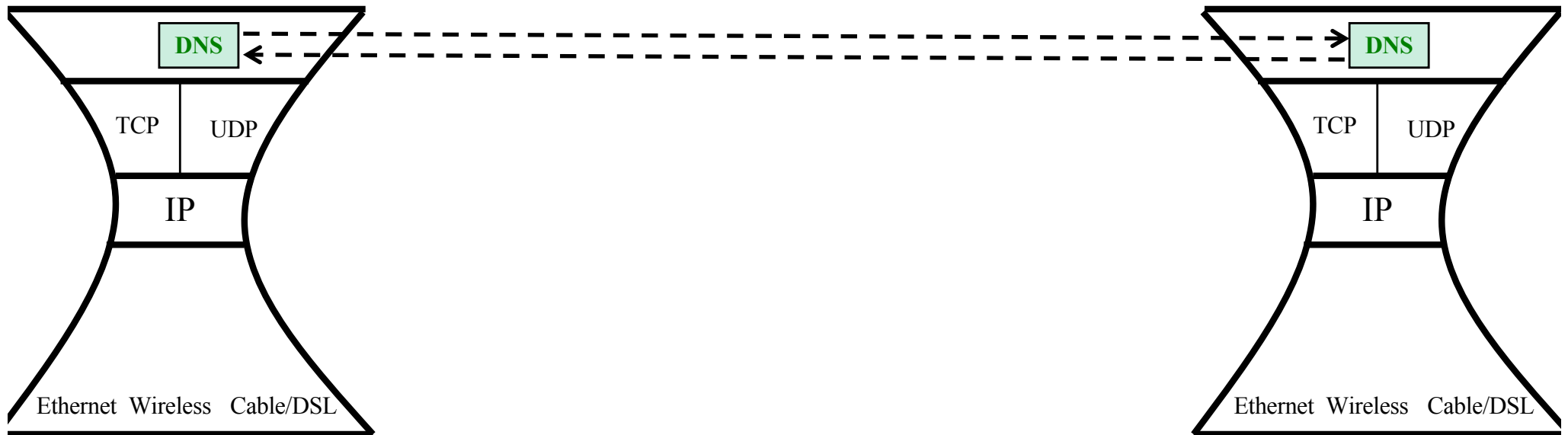
# Outline

---

- Admin. and recap
- *DNS*
  - High-level design
  - *Details*

# DNS Message Format?

Basic encoding decisions: UDP/TCP,  
how to encode domain name, how to  
encode answers...



# Observing DNS Messages

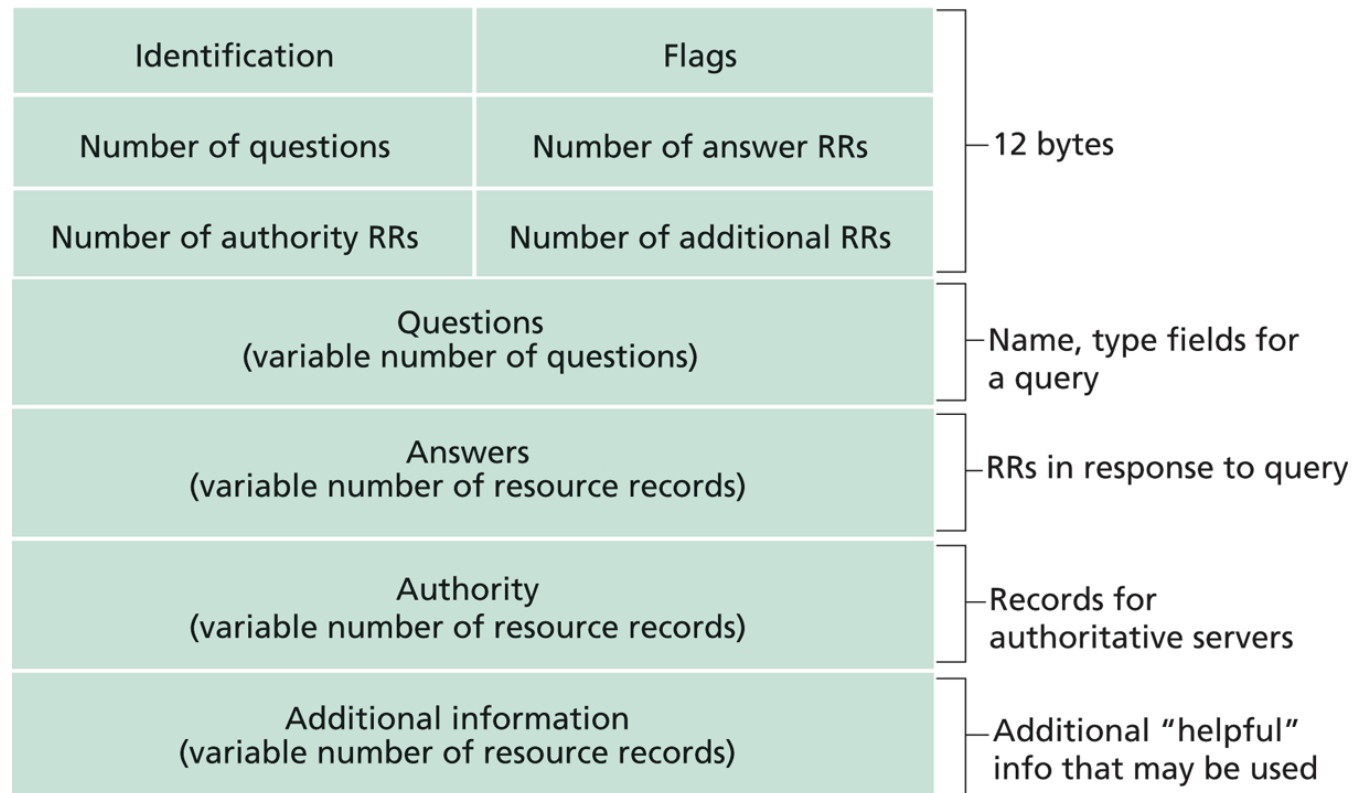
---

## □ Capture the messages

- DNS server is at port 53
  - Display and clear DNS cache
    - MacOS: <https://support.apple.com/en-us/HT202516>  
sudo killall -HUP mDNSResponder
    - Ubuntu:  
sudo systemd-resolve --flush-caches  
sudo systemd-resolve --statistics
- Try to load the dns-capture file from class Schedule page, if you do not want live capture

# DNS Protocol, Messages

DNS protocol : typically over UDP (can use TCP);  
*query* and *reply* messages, both with the **same**  
*message format*





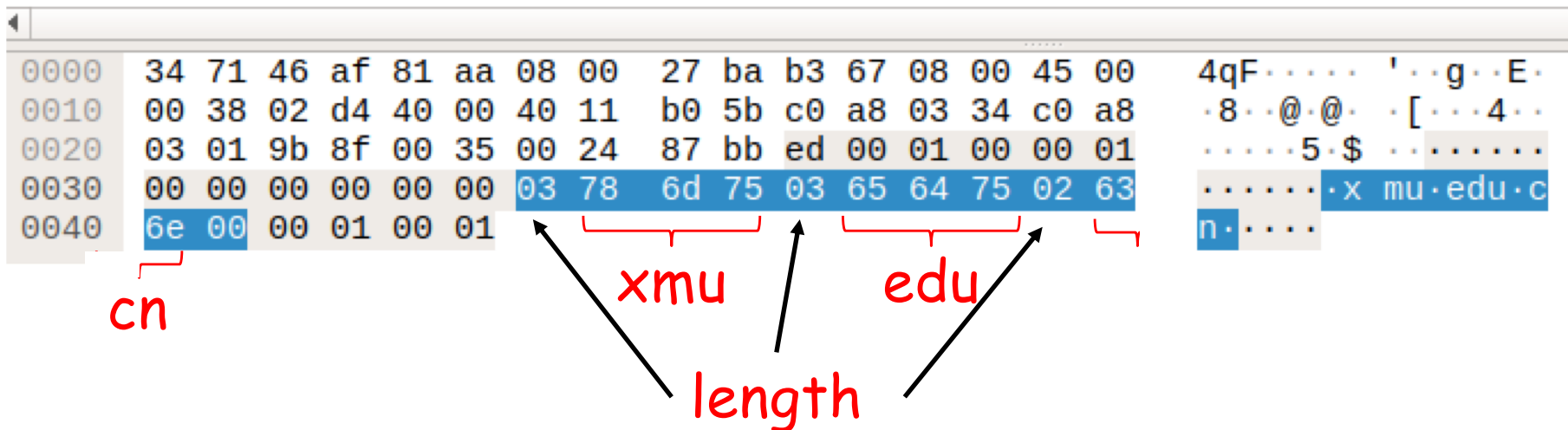
# DNS Details

---

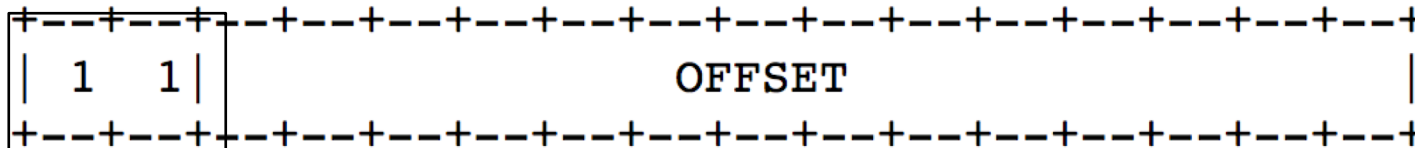
- ❑ Header (Sec. 4.1.1 of <https://www.ietf.org/rfc/rfc1035.txt>)
- ❑ Encoding of questions (Sec. 4.1.2):
  - [Label-length label-chars]
- ❑ Encoding of answers (Sec. 4.1.3)
  - Pointer format  
(<http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml>)
  
- ❑ See example DNS packets

# Name Encoding

- ▼ Queries
    - ▼ xmu.edu.cn: type A, class IN
- Name: xmu.edu.cn  
[Name Length: 10]  
[Label Count: 3]  
Type: A (Host Address) (1)  
Class: IN (0x0001)  
[\[Response In: 3\]](#)



# Message Compression (Label Pointer)



```

Domain Name System (response)
  Transaction ID: 0xed00
  ▶ Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  ▶ Queries
  ▼ Answers
    ▼ xmu.edu.cn: type A, class IN, addr 210.34.0.35
  
```

```

Name: xmu.edu.cn
Type: A (Host Address) (1)
Class: IN (0x0001)
  
```

0000	08 00 27 ba b3 67 34 71 46 af 81 aa 08 00 45 00	..'.g4q F.....E.
0010	00 48 7f 0e 40 00 40 11 34 11 c0 a8 03 01 c0 a8	.H..@.@.4.....
0020	03 34 00 35 9b 8f 00 34 91 ee ed 00 81 80 00 01	.4.5...4.....
0030	00 01 00 00 00 00 03 78 6d 75 03 65 64 75 02 63	.....x mu.edu.c
0040	6e 00 00 01 00 01 c0 0c 00 01 00 01 00 00 00 44	n.....D
0050	00 04 d2 22 00 23	...".#

DNS  
start

question

Answer:  
offset 12

# Recap: DNS Protocol, Messages

Many features: typically over **UDP** (can use TCP); *query* and *reply* messages with the **same message format**; *length/content encoding of names*; simple *compression*; *additional info as server push*

