Network Applications: Email Security, DNS

Y. Richard Yang

http://zoo.cs.yale.edu/classes/cs433/

9/18/2018

Outline

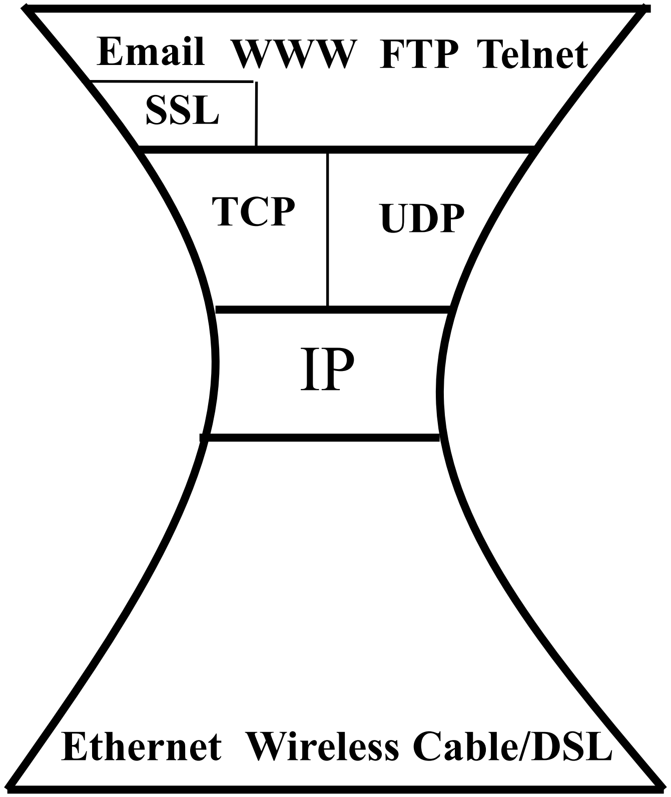
- Admin and recap
- Email
  - Basic email systems design
  - Email security
- DNS

Admin

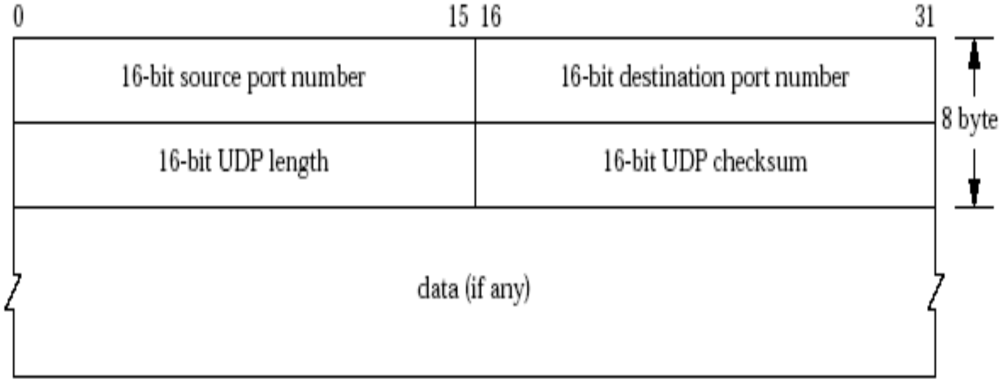
- 72 discretionary late hours for assignments across the semester

Recap: The Big Picture of the Internet

- Hosts and routers:
  - ~ 1 bill. hosts
  - organized into ~50K networks
  - backbone links 100 Gbps
- Software:
  - datagram switching with virtual circuit support
  - layered network architecture
    - use end-to-end arguments to determine the services provided by each layer
    - the 5-layer hourglass architecture of the Internet



Formats of main protocols



0 15 16 31

16-bit source port number

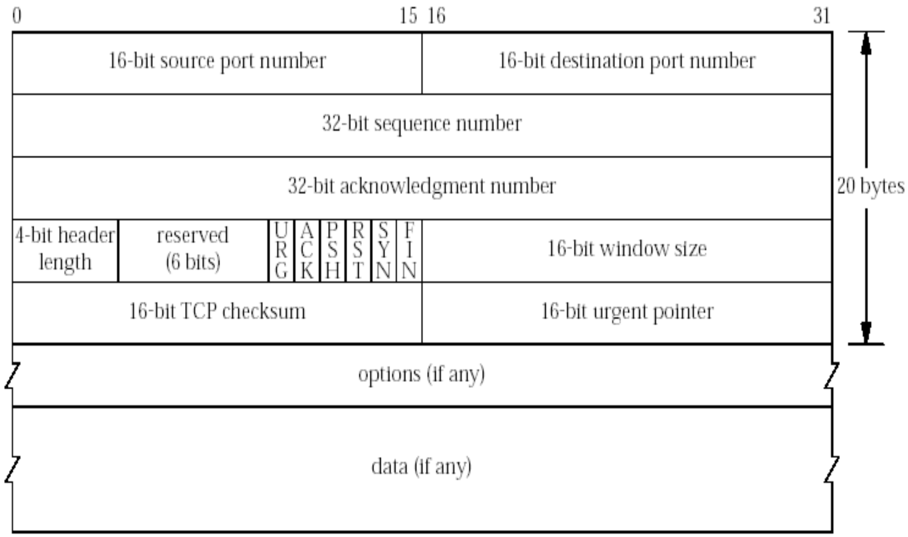
16-bit destination port number

16-bit UDP length

16-bit UDP checksum

data (if any)

8 byte



0 15 16 31

16-bit source port number

16-bit destination port number

32-bit sequence number

32-bit acknowledgment number

4-bit header length

reserved (6 bits)

URG

ACK

PSH

RST

SYN

FIN

16-bit window size

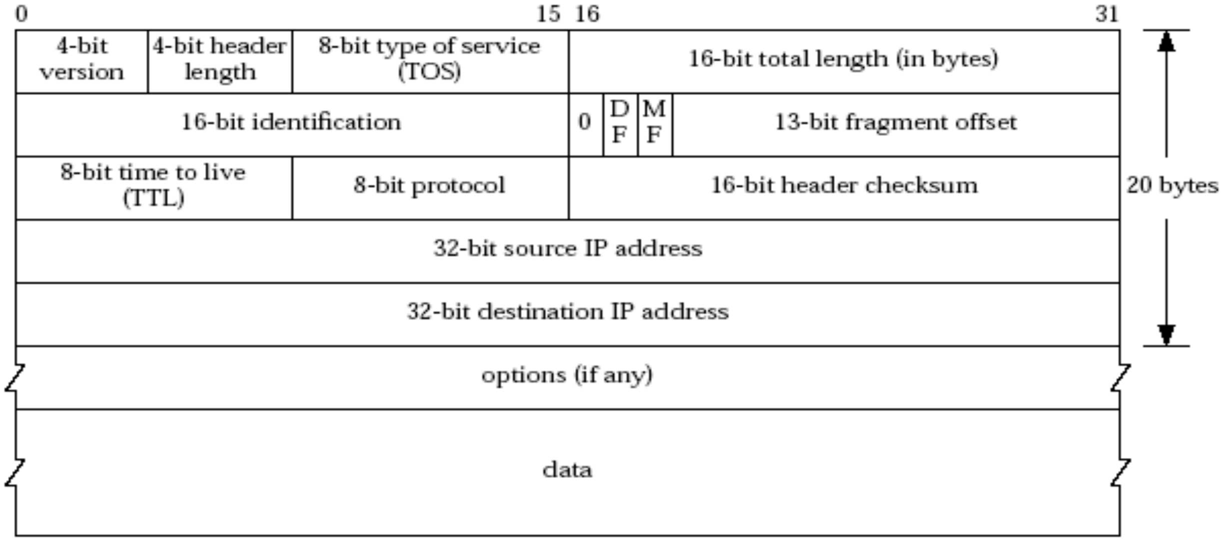
16-bit TCP checksum

16-bit urgent pointer

options (if any)

data (if any)

20 bytes



0 15 16 31

4-bit version

4-bit header length

8-bit type of service (TOS)

16-bit total length (in bytes)

16-bit identification

0

DF

MF

13-bit fragment offset

8-bit time to live (TTL)

8-bit protocol

16-bit header checksum

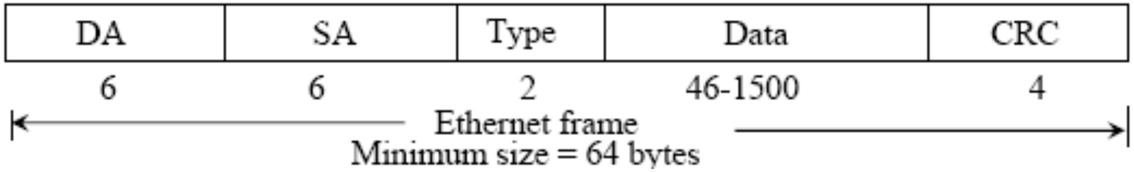
32-bit source IP address

32-bit destination IP address

options (if any)

data

20 bytes



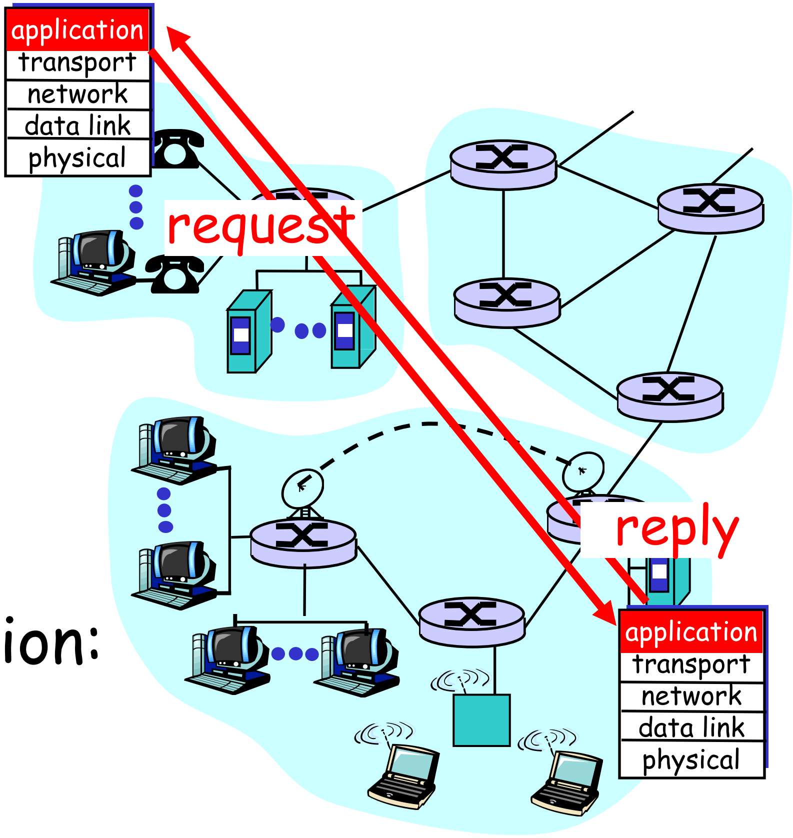
DA	SA	Type	Data	CRC
6	6	2	46-1500	4

Ethernet frame
Minimum size = 64 bytes

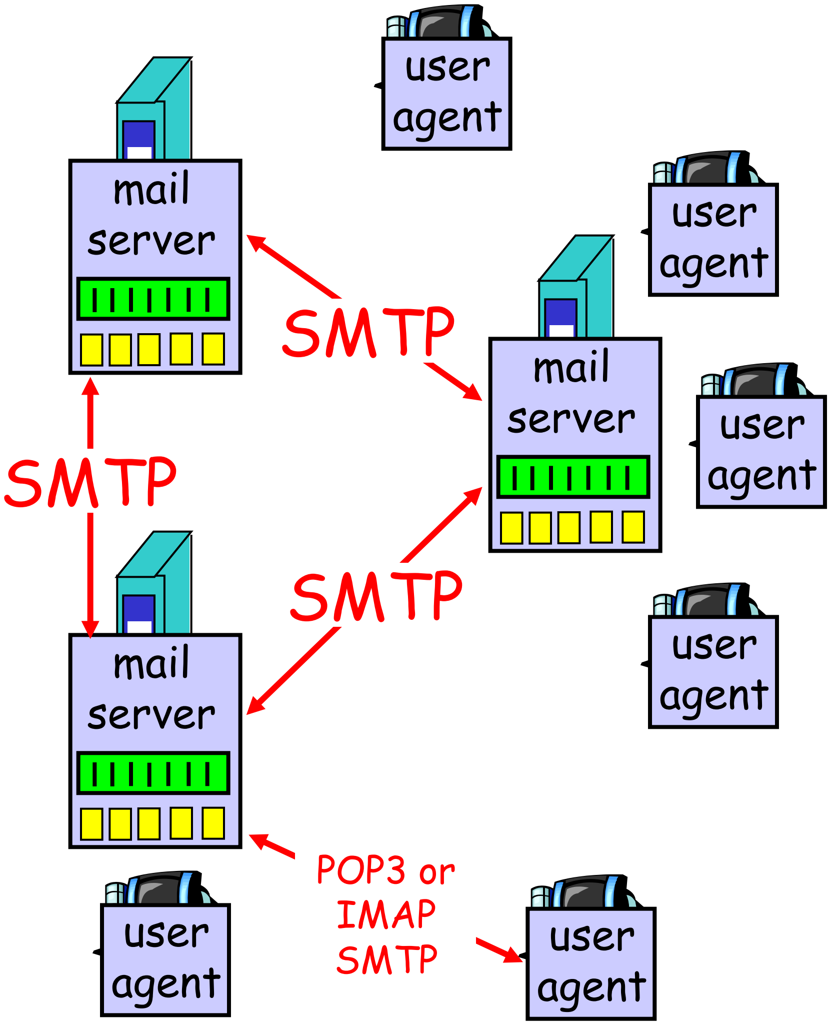
Recap: Client-Server Paradigm

- The basic paradigm of network applications is the client-server (C-S) paradigm

- Some key design questions to ask about a C-S application:
  - extensibility
  - scalability
  - robustness
  - security



Recap: Email Design Features



Some key design features of Email

- Separate protocols for different functions
  - email access (e.g., POP3, IMAP)
  - email transport (SMTP)
- A SMTP transaction consists of an envelope and a message body
  - separation of envelope and message body (end-to-end arguments)
    - envelope: 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

Recap: Evaluation of SMTP

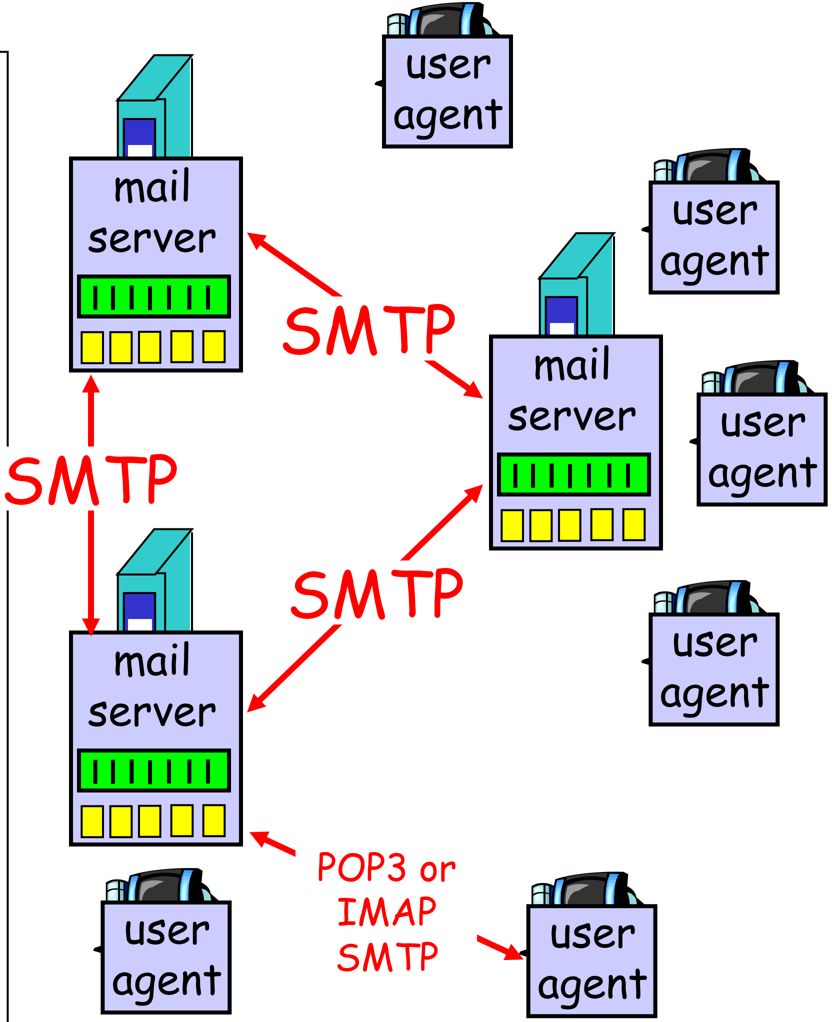
Key questions to ask about a C-S application

- **extensible?**
  separate envelope and msg;
  self-describing message;
  ehlo negotiation

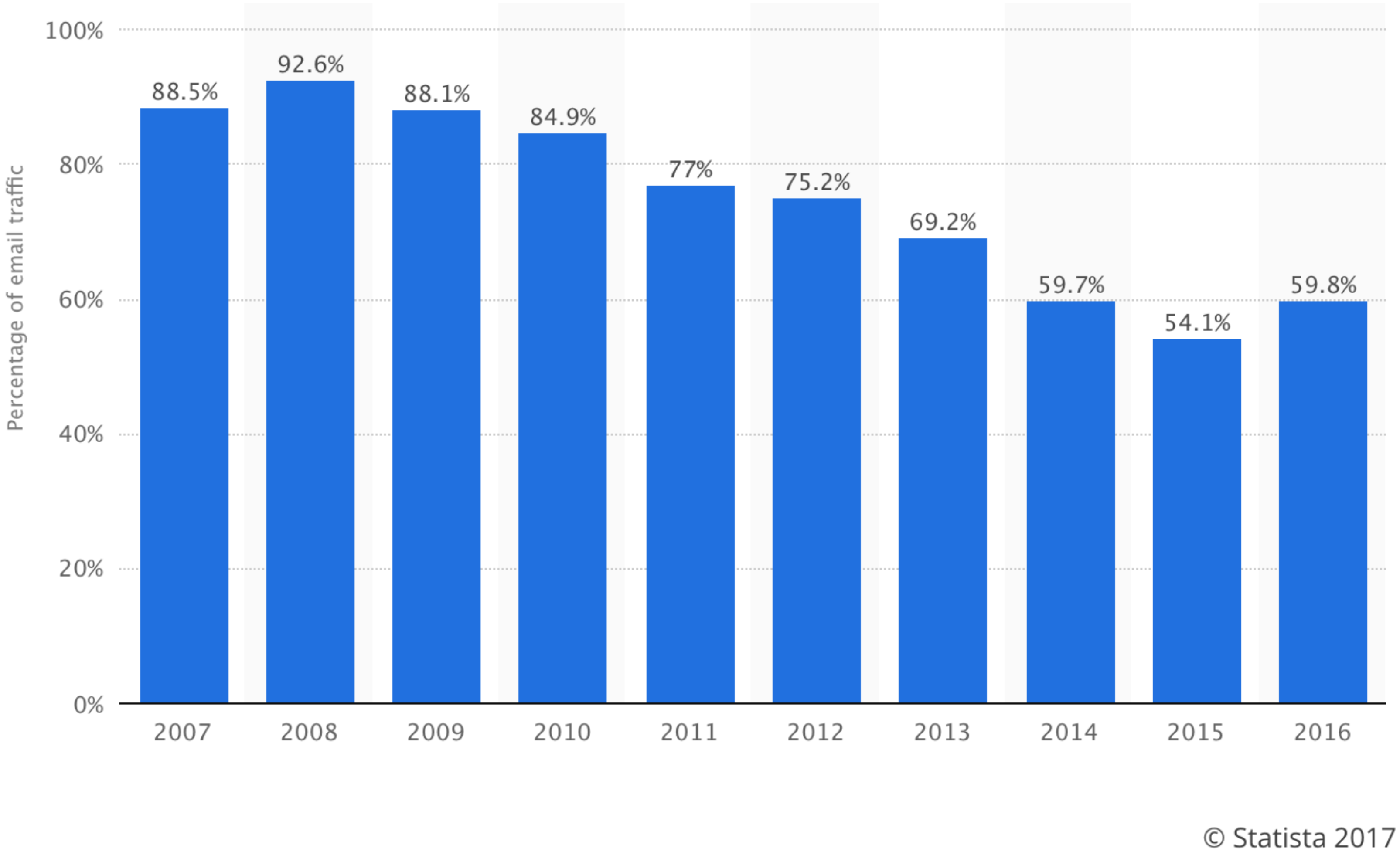
- **scalable?**
  have not seen mechanism yet

- **robust?**
  have not seen mechanism yet

- **security?**
  authentication/authorization (spoof, spam) are major issues of mail transport

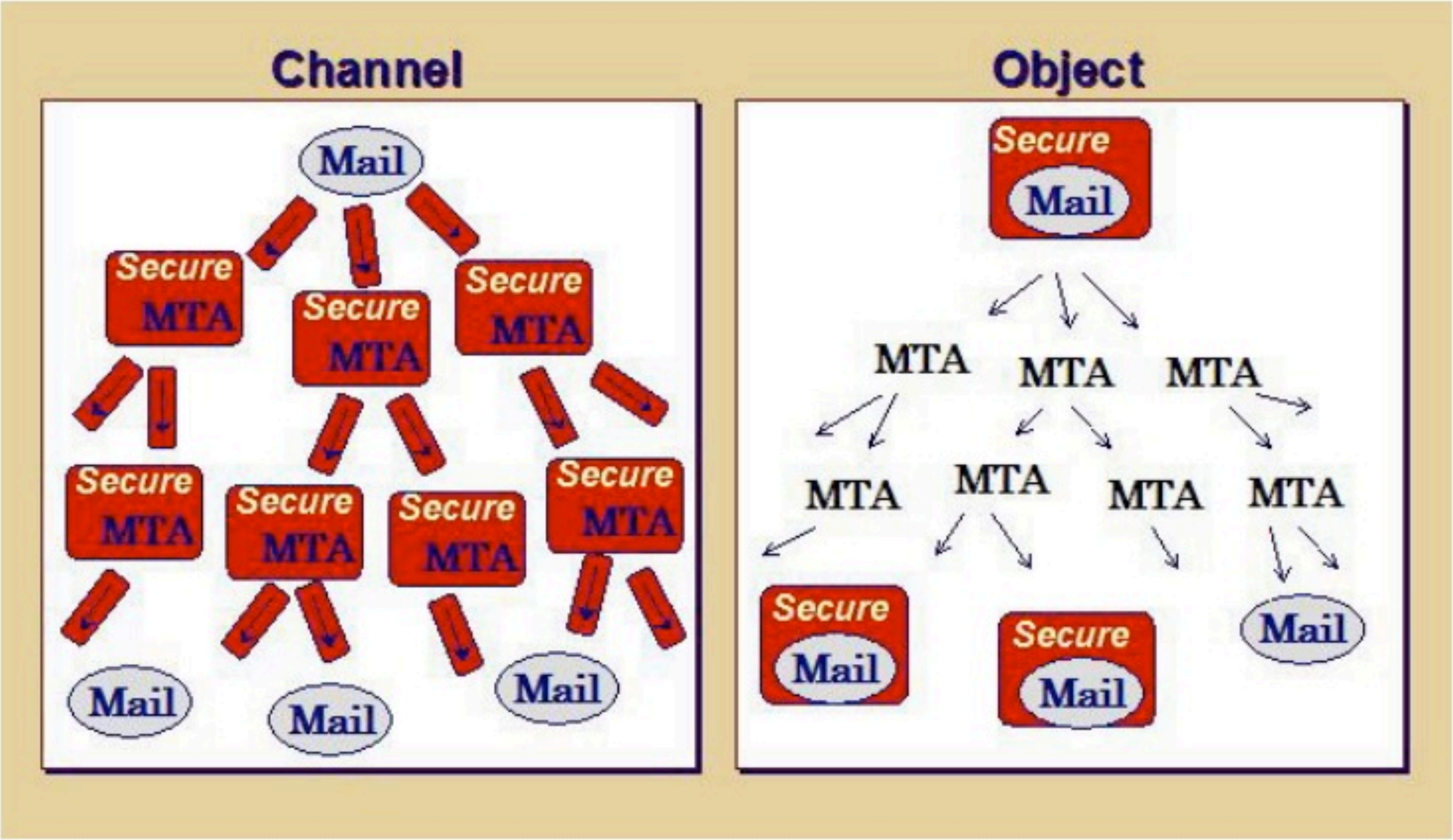


Spam Trend



Source: https://www.statista.com/statistics/420400/spam-email-traffic-share-annual/

Current SMTP Authentication Approaches

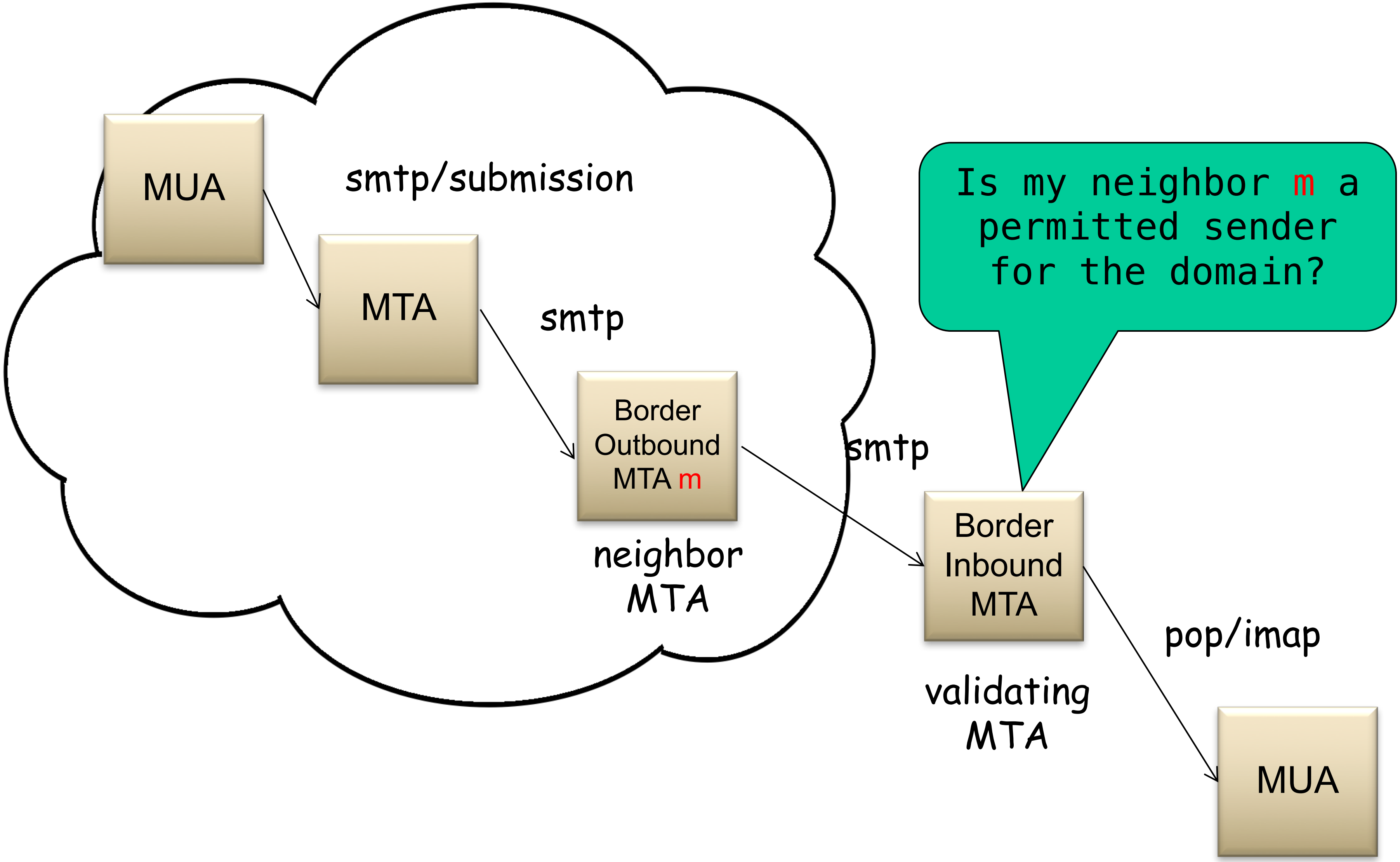


Sender Policy Frame (SPF)

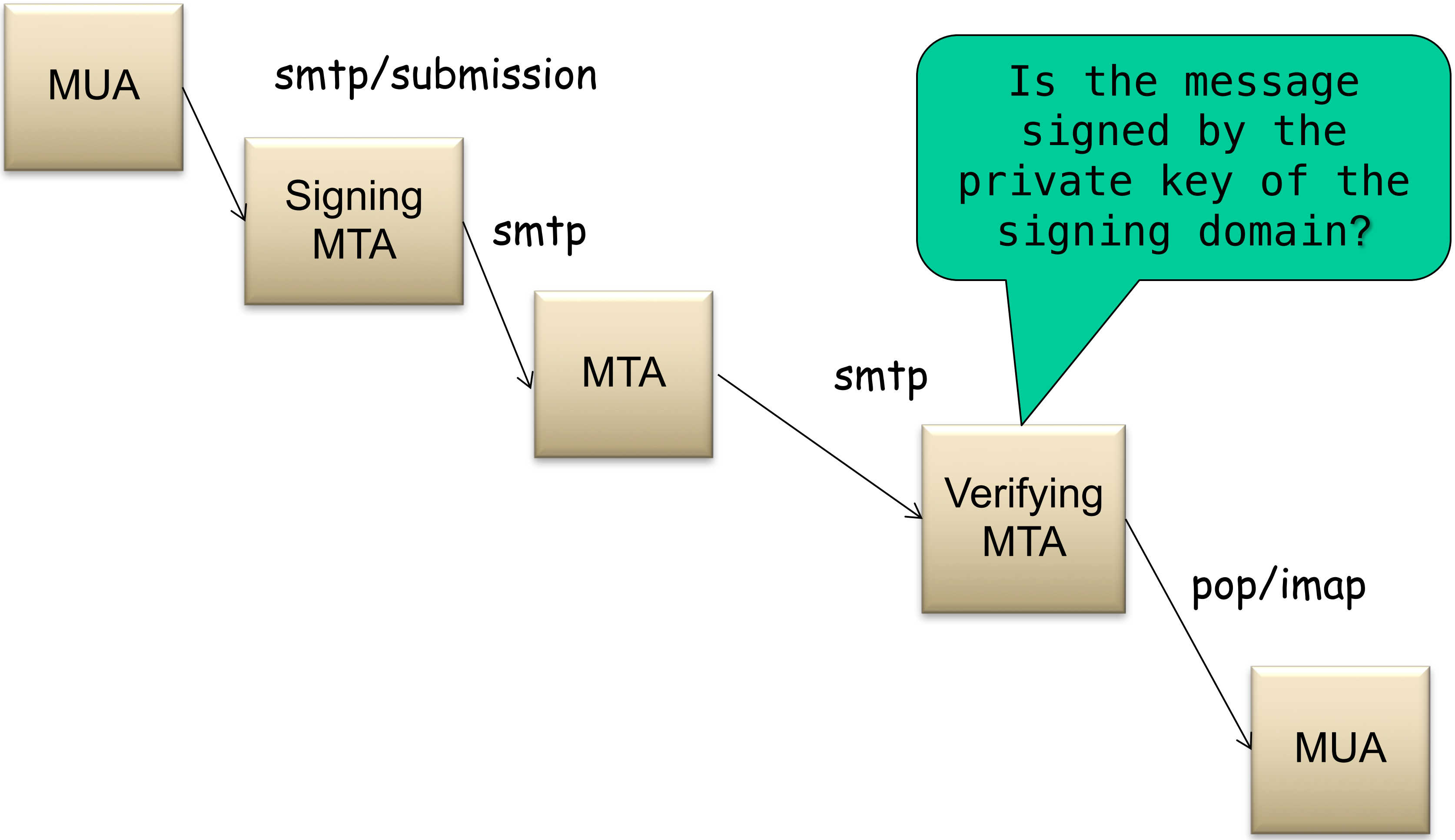
DomainKeys Identified Mail (DKIM)

https://tools.ietf.org/html/rfc7208

Sender Policy Framework (SPF RFC7208)



DomainKeys Identified Mail (DKIM)



Exercise

- Capture and look at SFP and DKIM in email messages

See pop3-trace.txt

DomainKeys Identified Mail (DKIM; RFC 5585; RFC6376)

- A domain-level digital signature authentication framework for email, using public key crypto, typically RSA
- Basic idea of RSA type public key signature
  - Owner has both public and private keys
  - Owner uses private key to sign a message to generate a signature
  - Others with public key can verify signature
  - Assumption: difficult to get private key even w/ public key distributed

Example: RSA

1. Choose two large prime numbers *p, q.* (e.g., 1024 bits each)

2. Compute $n = pq$, $z = (p-1)(q-1)$

3. Choose *e* (with $e < n$) that has no common factors with z. (*e, z* are "relatively prime").

4. Choose *d* such that *ed-1* is exactly divisible by *z*. (in other words: *ed* mod $z = 1$).

5. *Public* key is *(n,e)*. *Private* key is *(n,d)*.

RSA: Signing/Verification

0. Given (n,e) and (n,d) as computed above

1. To sign message, m, compute h = hash(m), then sign with private key

   $s = h^d \bmod n$ (i.e., remainder when h^d is divided by n)

2. To verify signature s, compute

   $h' = s^e \bmod n$ (i.e., remainder when s^e is divided by n)

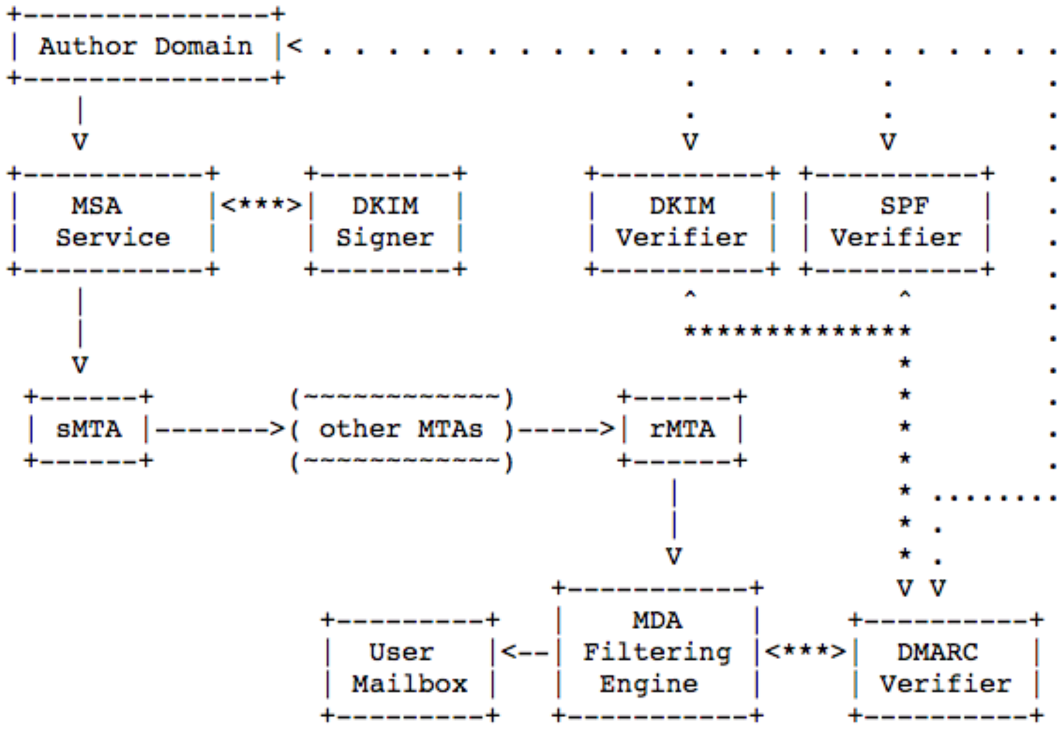
Magic happens!

$$h = (h^d \bmod n)^e \bmod n$$

The magic is a simple application of Euler's generalization of Fermat's little theorem

Domain-based Message Authentication, Reporting, and Conformance (DMARC) [RFC7489]

- Remaining issue: How to handle unauthenticated messages?



```
 +---------------+
 | Author Domain |< . . . . . . . . . . . . . . . . . . . . . . .
 +---------------+                        .           .         .
     |                                    .           .         .
     V                                    V           V         .
 +-----------+     +--------+       +----------+ +----------+   .
 |   MSA     |<***>|  DKIM  |       |   DKIM   | |    SPF   |   .
 |  Service  |     | Signer |       | Verifier | | Verifier |   .
 +-----------+     +--------+       +----------+ +----------+   .
     |                                    ^            ^        .
     |                                    **************        .
     V                                                 *        .
  +------+        (~~~~~~~~~~~~)      +------+         *        .
  | sMTA |------->( other MTAs )----->| rMTA |         *        .
  +------+        (~~~~~~~~~~~~)      +------+         *        .
                                         |             * ........
                                         |             * .
                                         V             * .
                                  +-----------+        V V
                    +---------+   |    MDA    |     +----------+
                    |  User   |<--| Filtering |<***>|  DMARC   |
                    | Mailbox |   |  Engine   |     | Verifier |
                    +---------+   +-----------+     +----------+
```

```
MSA = Mail Submission Agent
MDA = Mail Delivery Agent
```

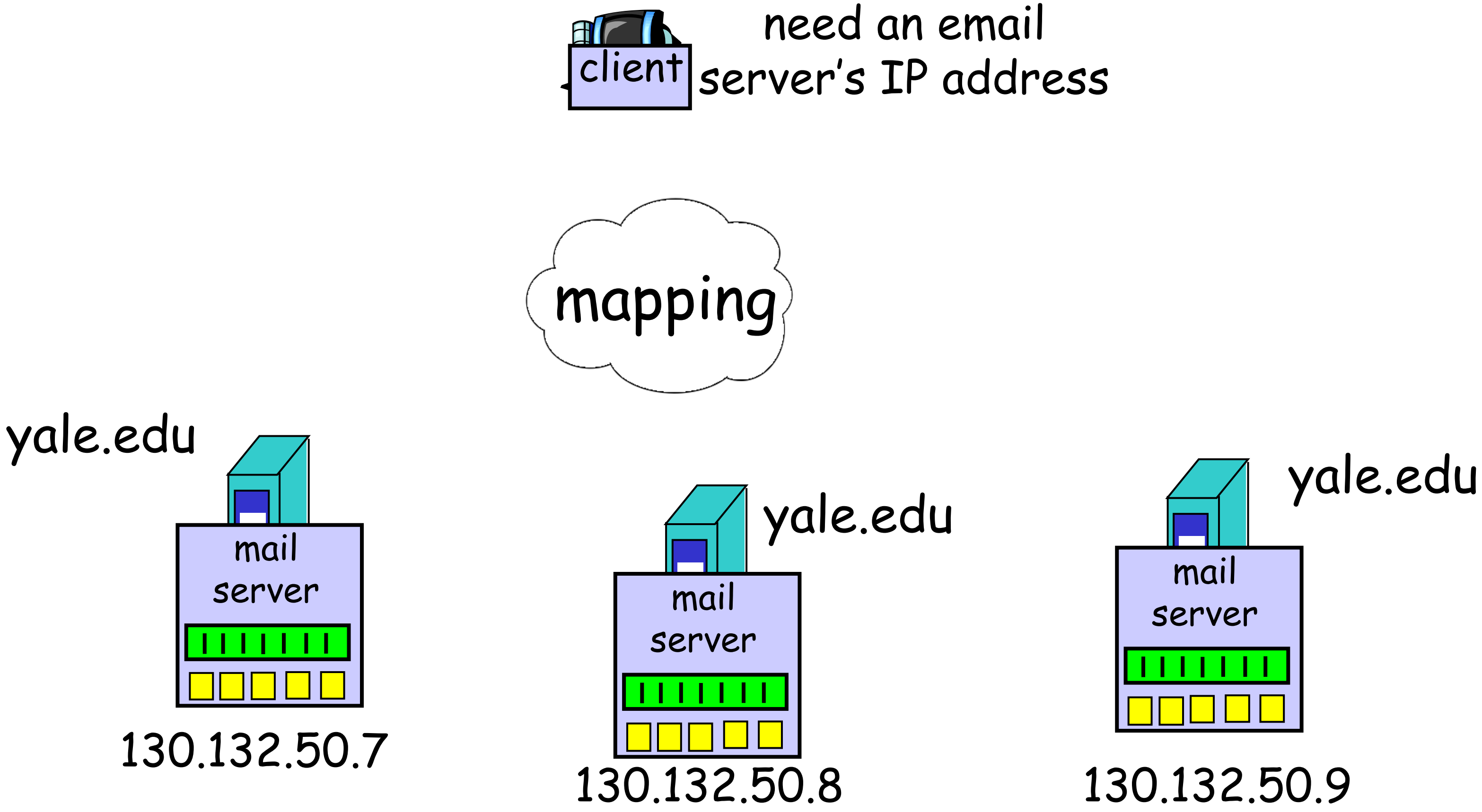
See pop3-trace.txt

Summary: Some Key Remaining Issues about Email

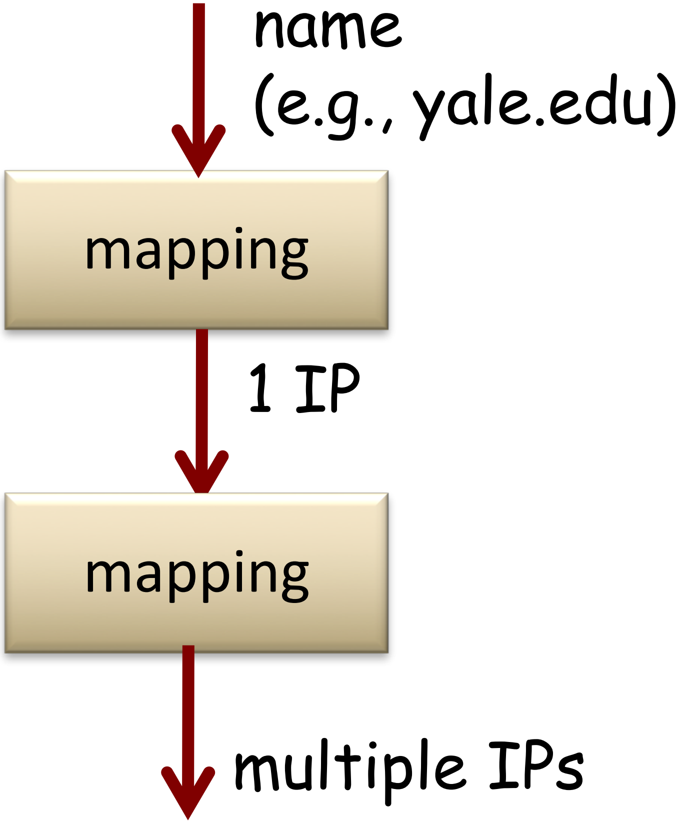
- Basic: How to find the email server of a domain?
- Scalability/robustness: how to find multiple servers for the email domain?
- Security
  - SPF: How does SPF know if its neighbor MTA is a permitted sender of the domain?
  - DKIM: How does DKIM retrieve the public key of the author domain?
  - DMARC: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

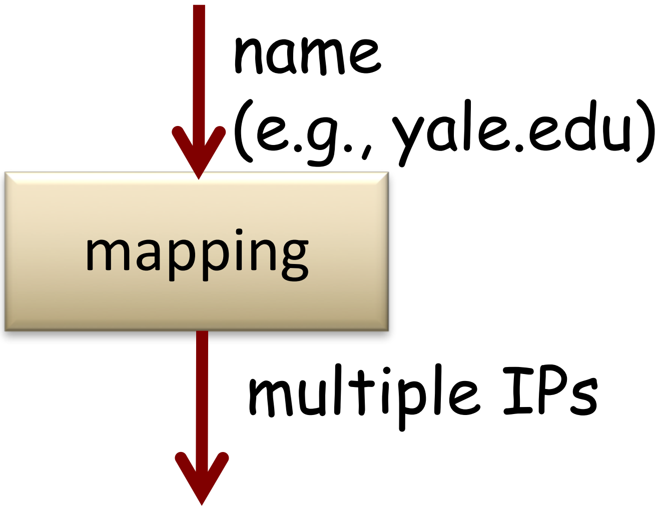
Scalability/Robustness

- Both scalability and robustness require that multiple email servers serve the same email address

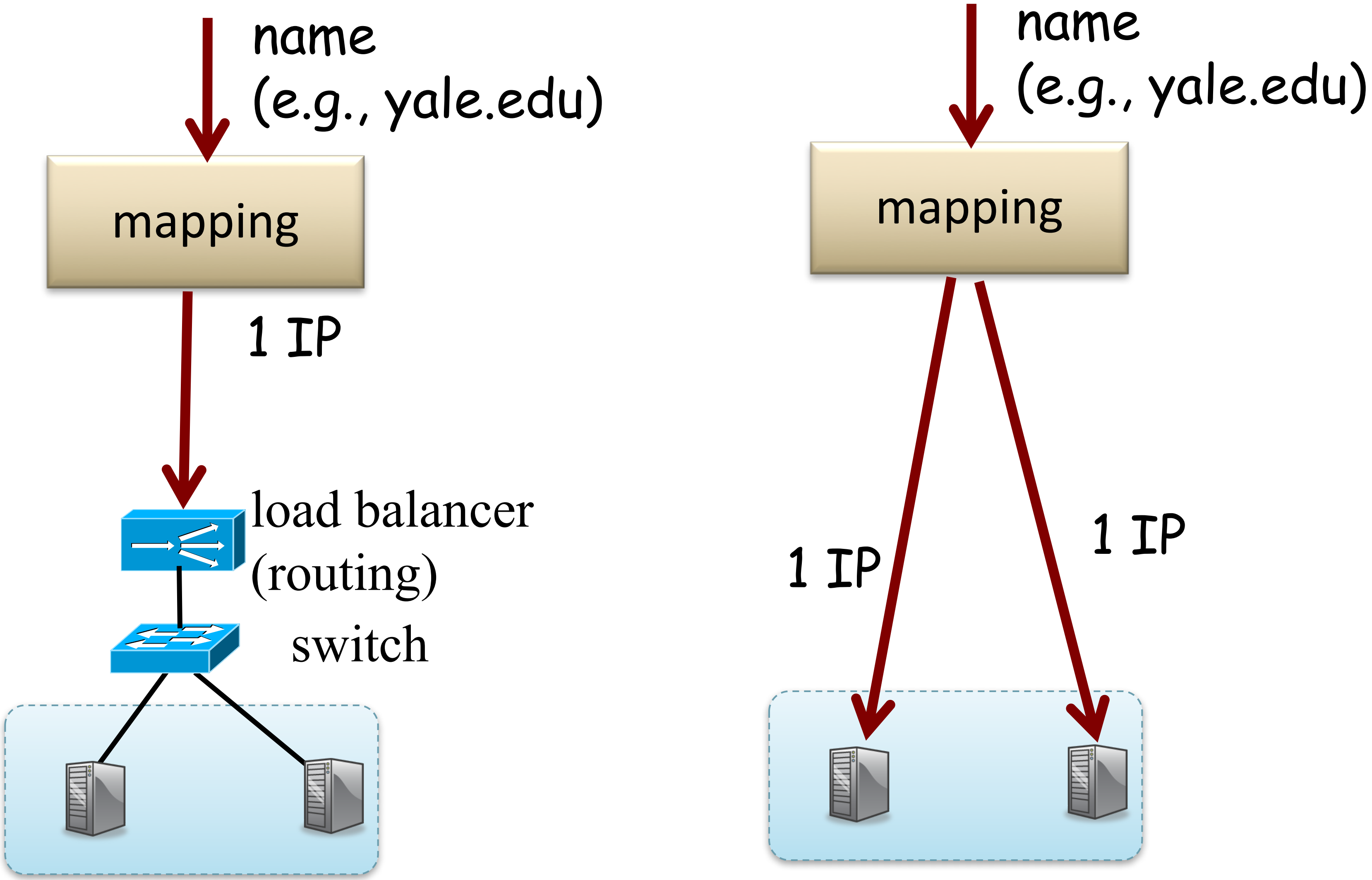


Mapping Functions Design Alternatives





Mapping Functions Design Alternatives

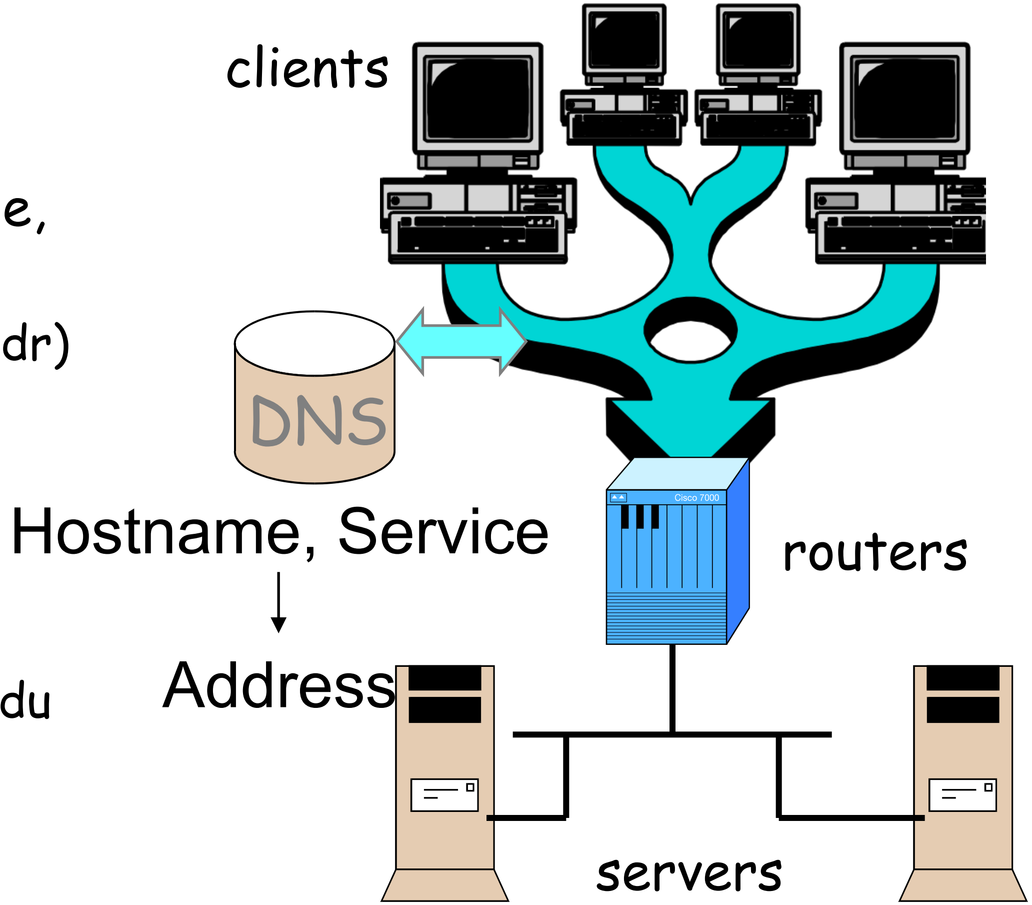


Outline

- Recap
- Email security (authentication)
- ➢ DNS

DNS: Domain Name System

- Function
  - map between (domain name, service) to value, e.g.,
    - (www.cs.yale.edu, addr) -> 128.36.229.30
    - (yale.edu, email) -> chai.mail.yale.edu rosehip.mail.yale.edu



Hostname, Service

↓

Address

DNS Records

http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml

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. yale.edu)
  - **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

Try DNS: Examples

- dig [@dnsserver] <name> <type>
  - try yale.edu and various types
  - dig www.yale.edu ANY
  - dig -x IP
  - try www.yale.edu

Observations

- A name/type can return multiple answers
- DNS may rotate the answered servers
- …

SPF Exercise

- telnet to netra.cs.yale.edu smtp
- Some test cases
  - From: yry@yale.edu
  - From: yry@harvard.edu
- dig <domain> txt to retrieve spf

DKIM Exercise

- Send email from gmail and check message

DKIM Example

- DKIM:
  Msg: DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=accounts.google.com; s=20161025; h=mime-version:date:feedback-id:message-id:subject:from:to; …
  Query: 20161025._domainkey.accounts.google.com txt
- DKIM introduces a session key to allow multiple public keys
  - <session>._domainkey.<domain>

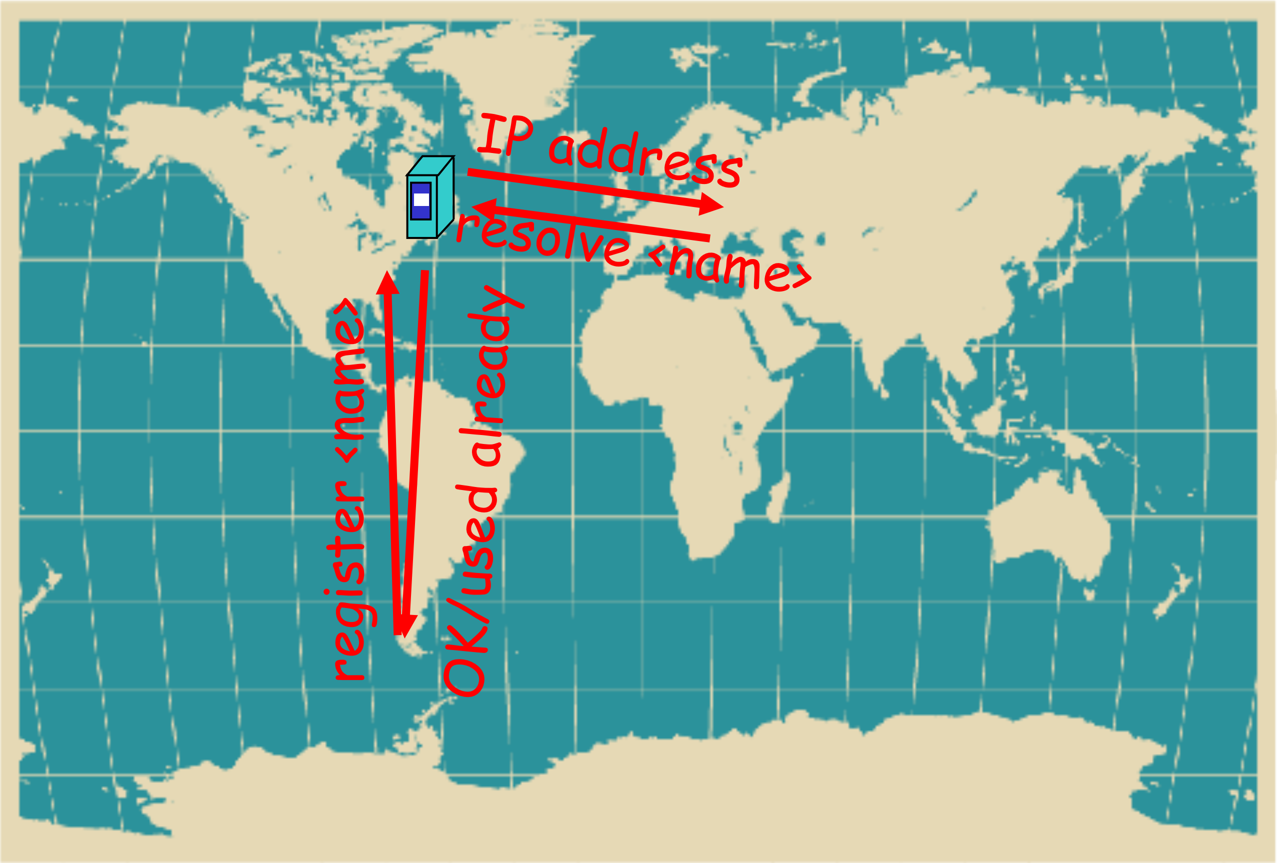
Outline

- Recap
- Email security (authentication)
- DNS
  - Interface
  - Architecture design

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

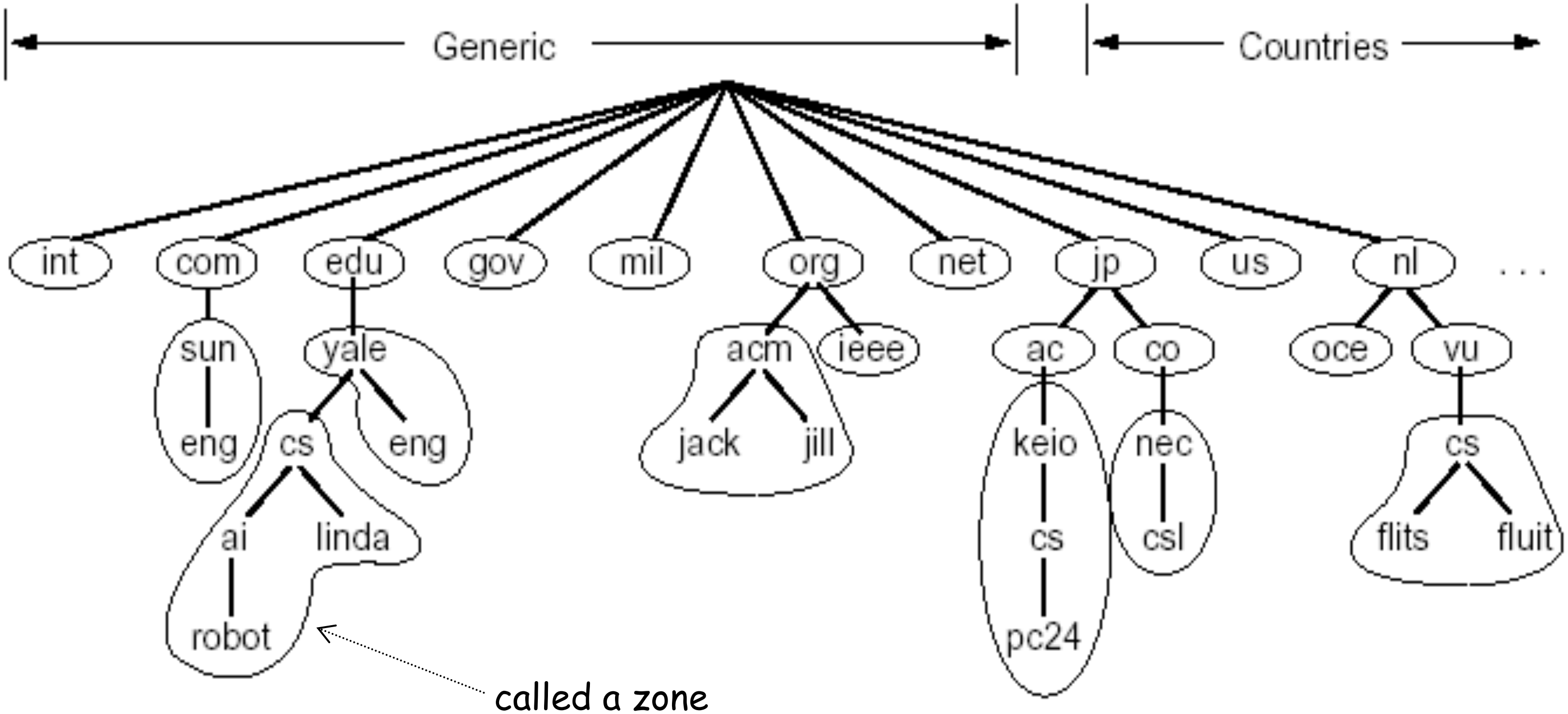


Problems of a Single DNS Server

- Scalability and robustness bottleneck
- Administrative bottleneck

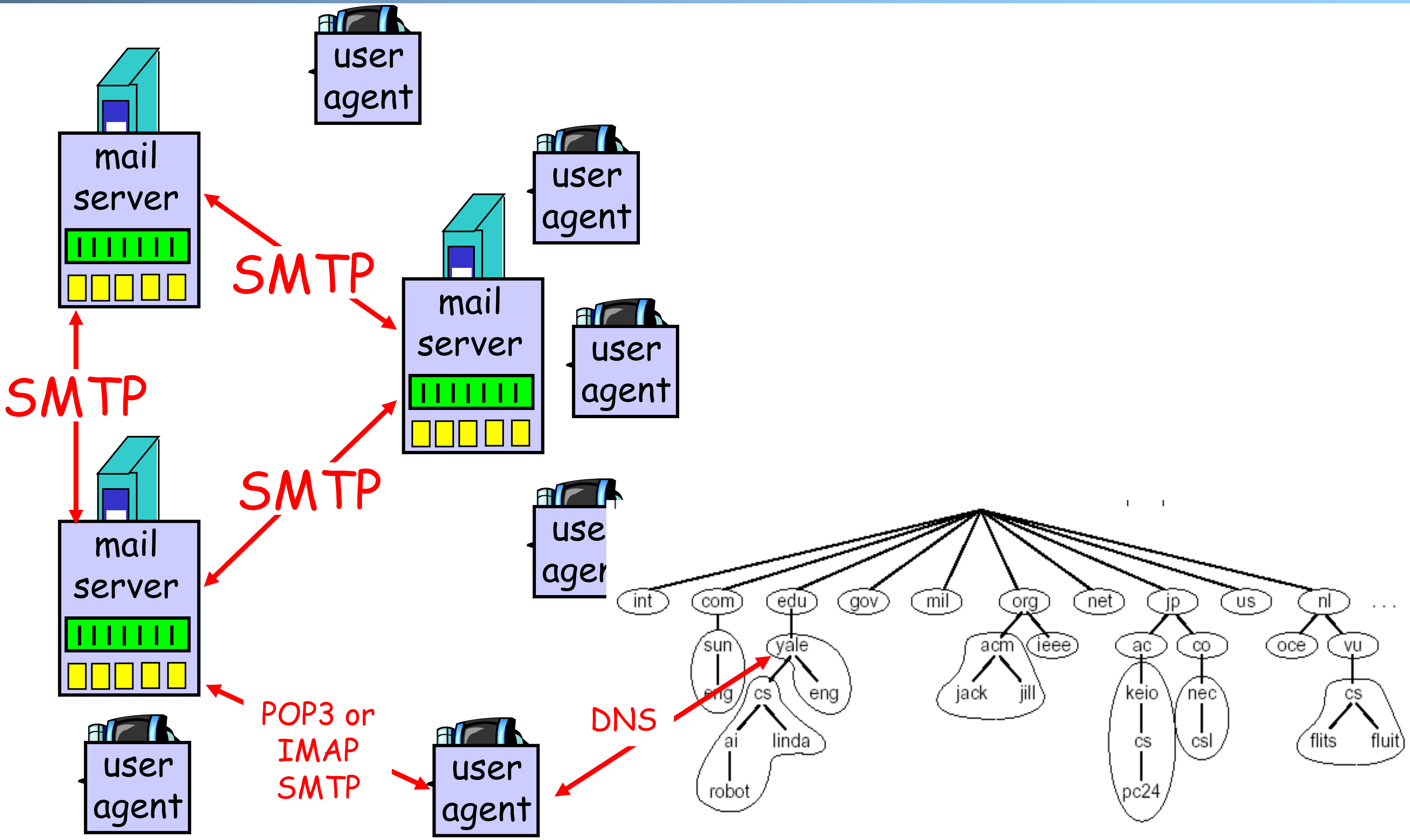
DNS: Distributed Management of the Domain Name Space

- 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



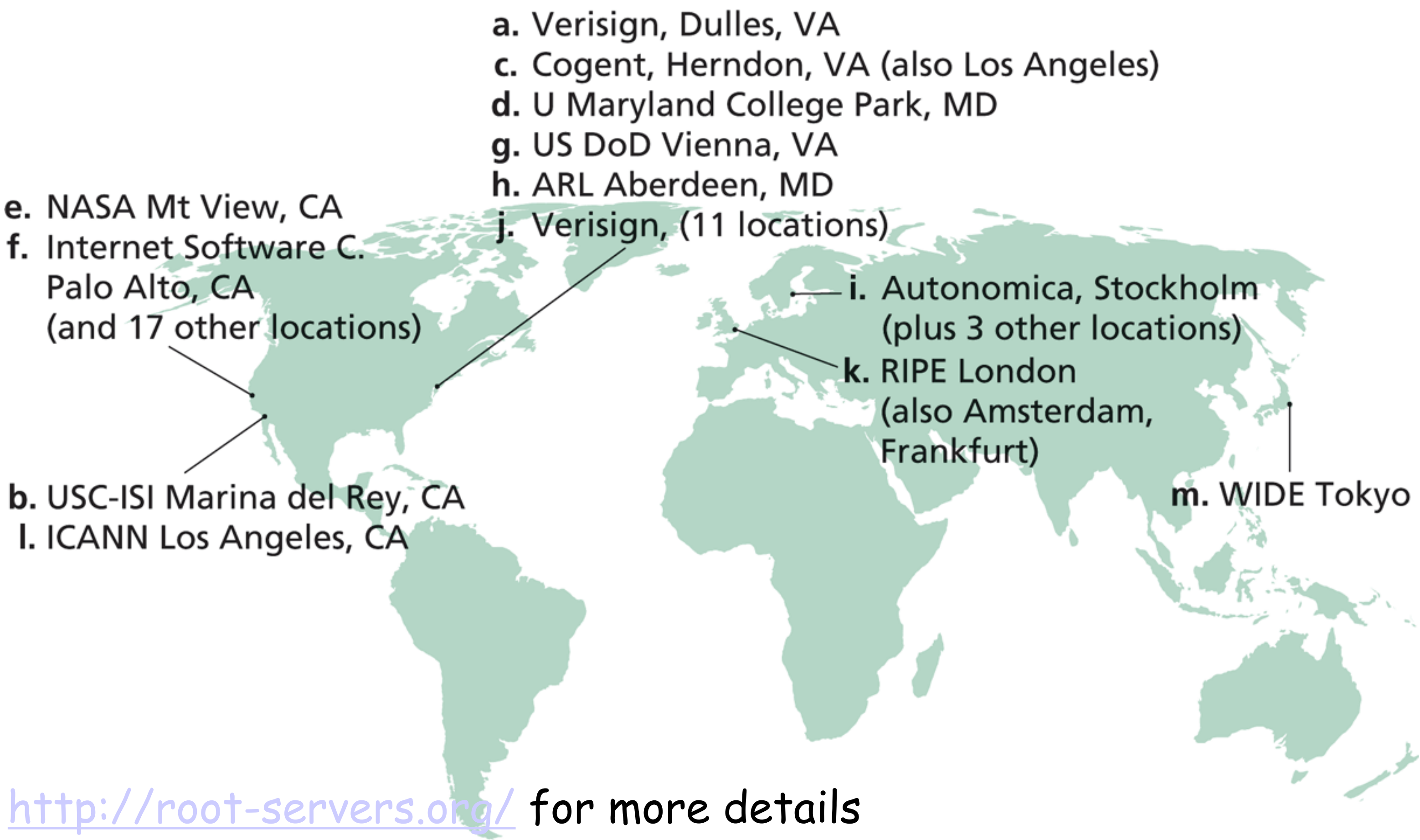
called a zone

Email Architecture + DNS



Root Zone and Root Servers

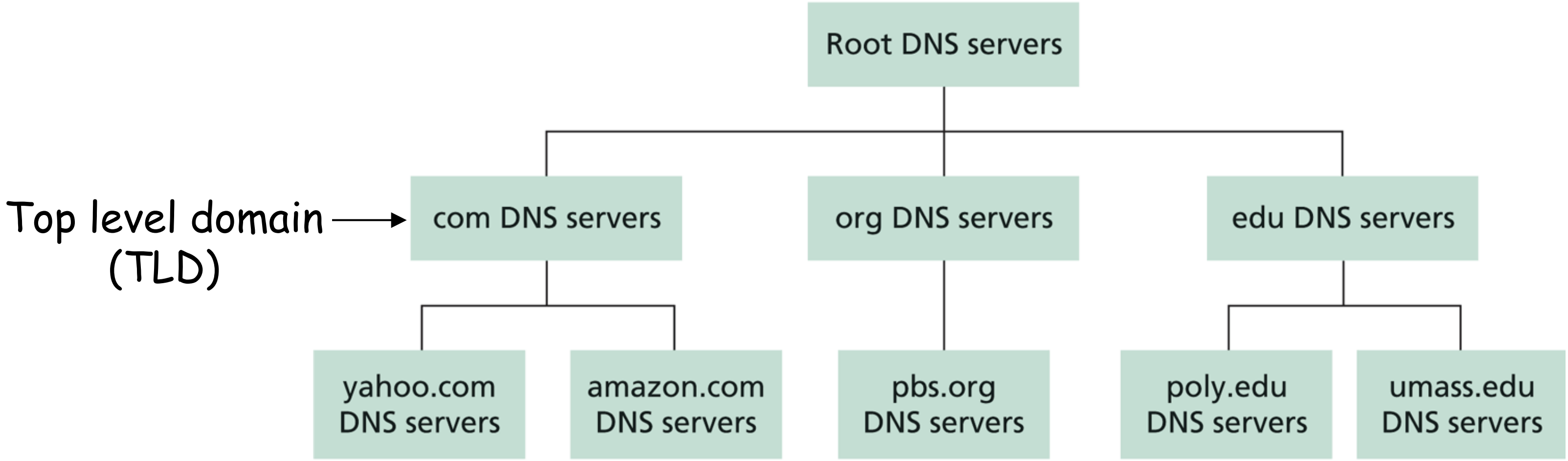
- The root zone is managed by the root name servers
  - 13 root name servers worldwide



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)



Root DNS servers

Top level domain (TLD)

com DNS servers

org DNS servers

edu DNS servers

yahoo.com DNS servers

amazon.com DNS servers

pbs.org DNS servers

poly.edu DNS servers

umass.edu DNS servers

Q: how to query a hierarchy?

DNS Message Flow: Two Types of Queries

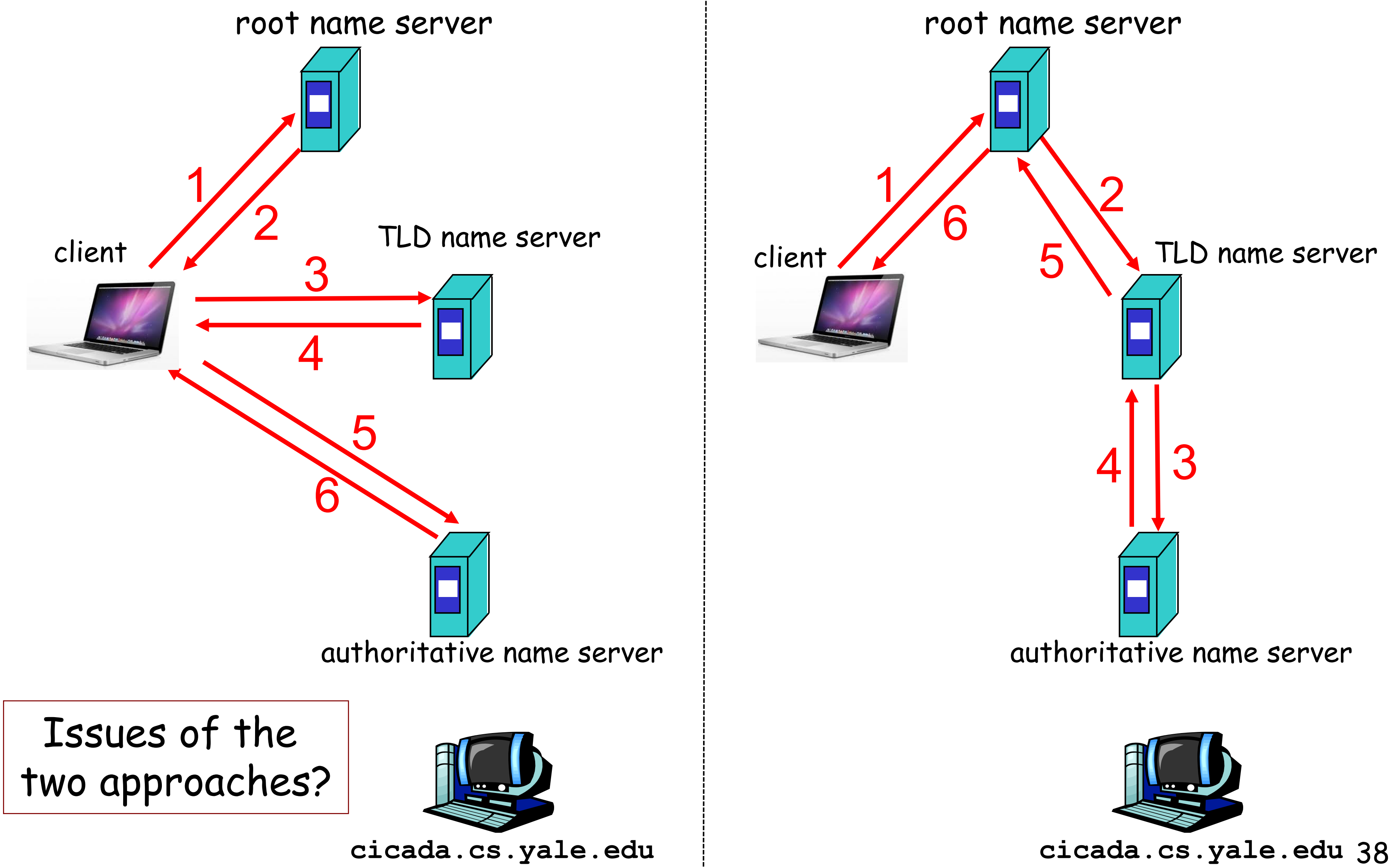
Recursive query:

- 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”

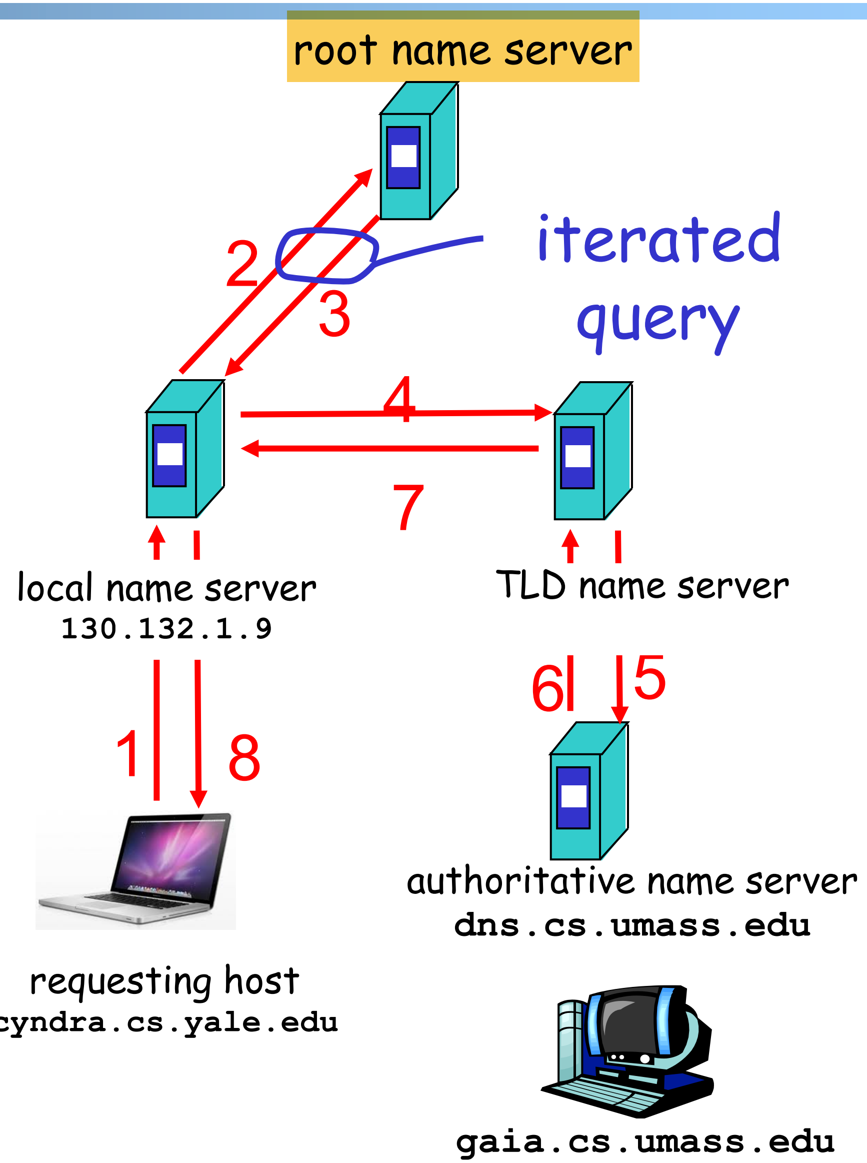
Two Extreme DNS Message Flows



Issues of the two approaches?

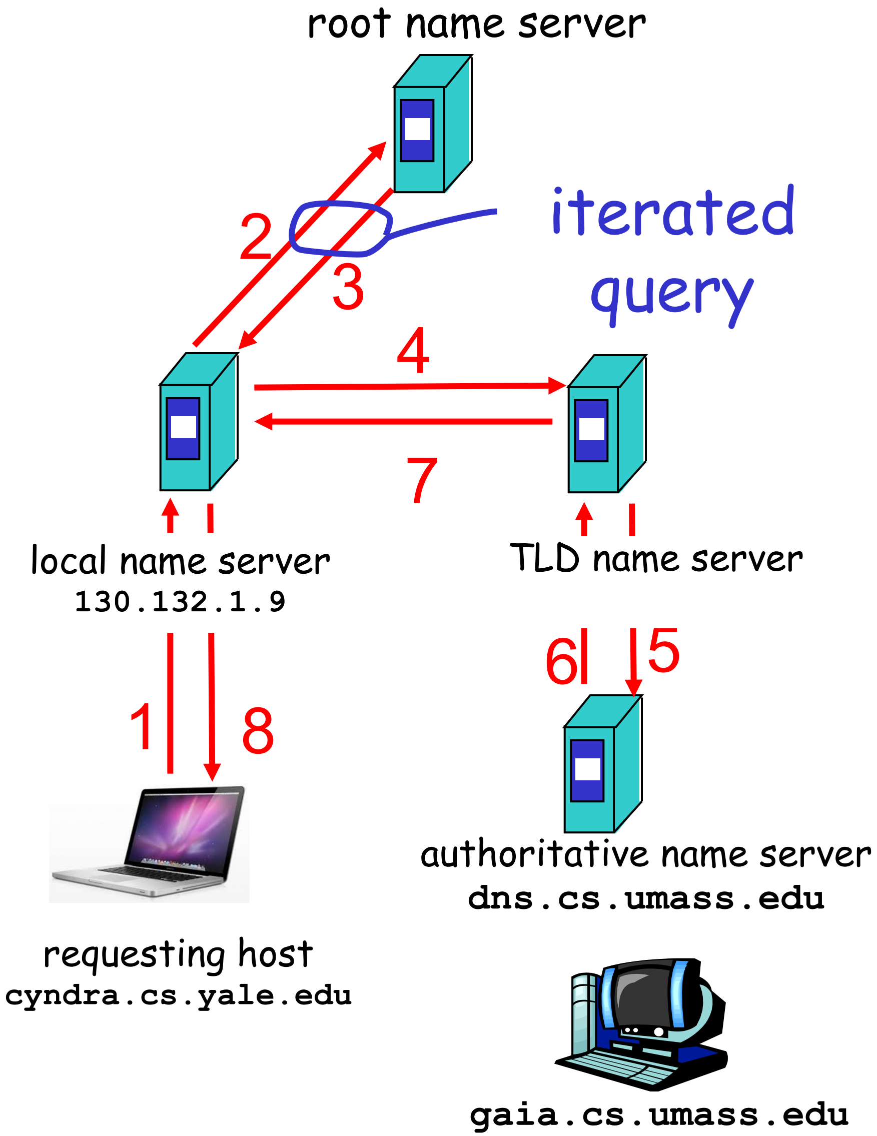
Typical DNS Message Flow: The Hybrid Case

- 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

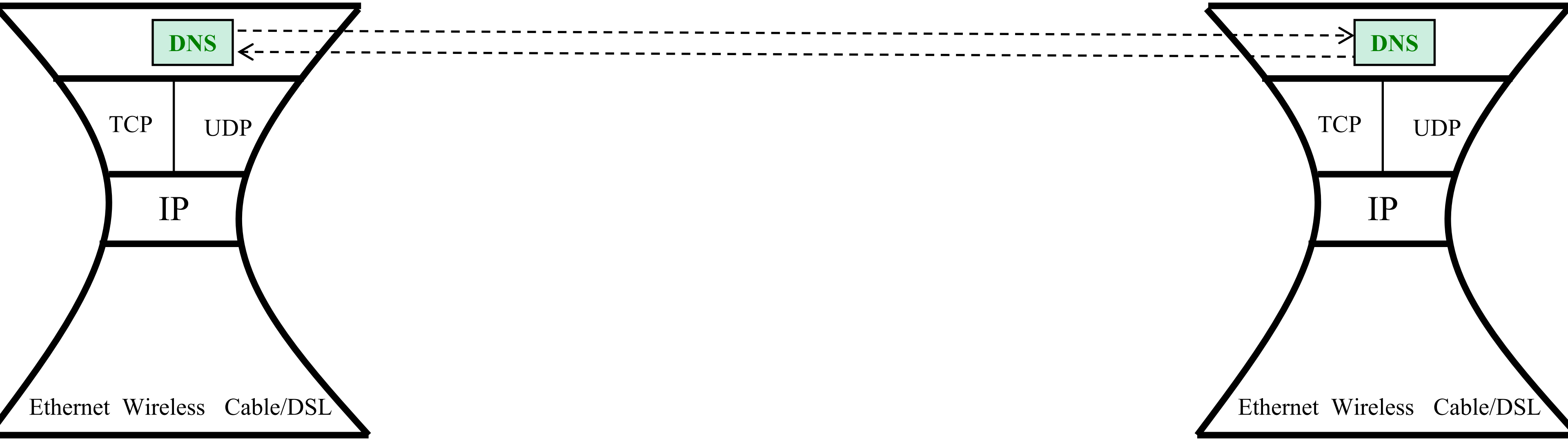


Outline

- Recap
- Email security (authentication)
- DNS
  - Interface
  - Architecture design
  - Message design

DNS Message Format?

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



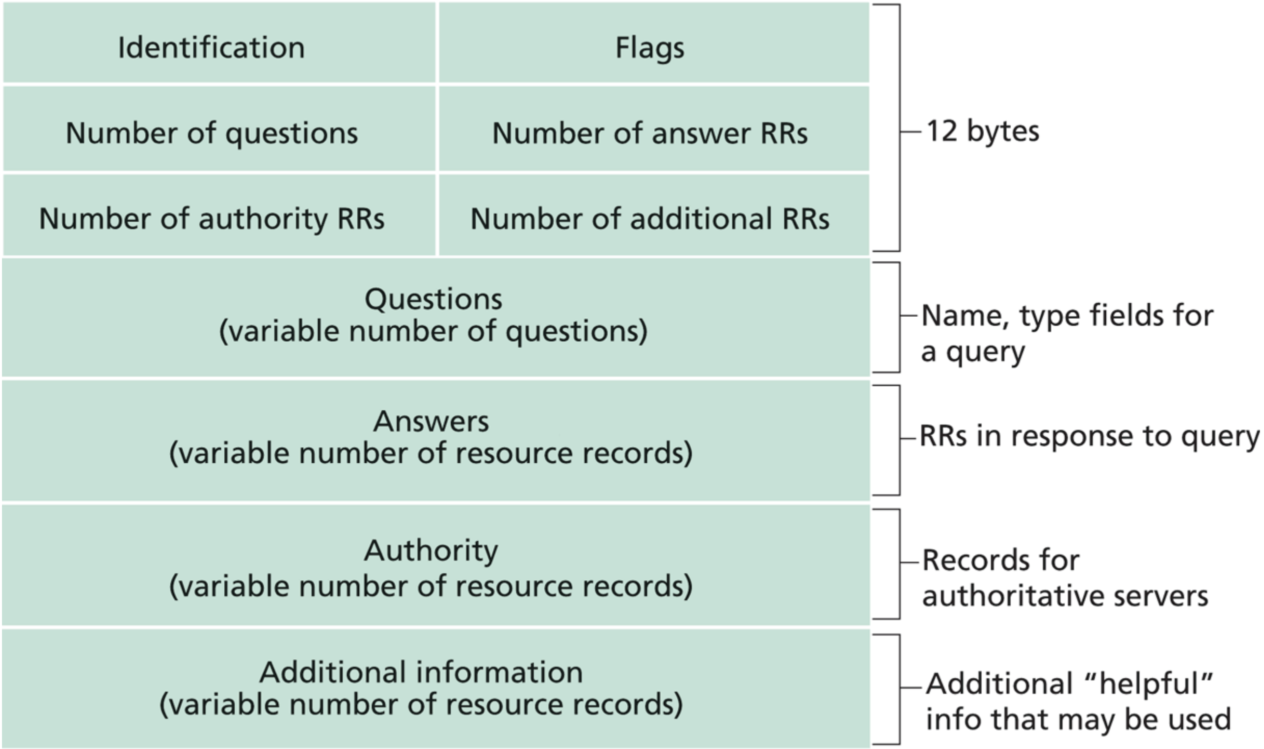
Observing DNS Messages

- Capture the messages
  - DNS server is at port 53
    - Display and clear DNS cache
      - https://support.apple.com/en-us/HT202516 (e.g., MAC sudo killall -HUP mDNSResponder)
  - visit gmail.com
  - dig +tcp to see TCP mode
  - Try to load the dns-capture file from class Schedule page, if you do not want live capture

https://www.ietf.org/rfc/rfc1035.txt

DNS Protocol, Messages

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



Identification | Flags

Number of questions | Number of answer RRs

Number of authority RRs | Number of additional RRs

12 bytes

Questions
(variable number of questions)

Name, type fields for a query

Answers
(variable number of resource records)

RRs in response to query

Authority
(variable number of resource records)

Records for authoritative servers

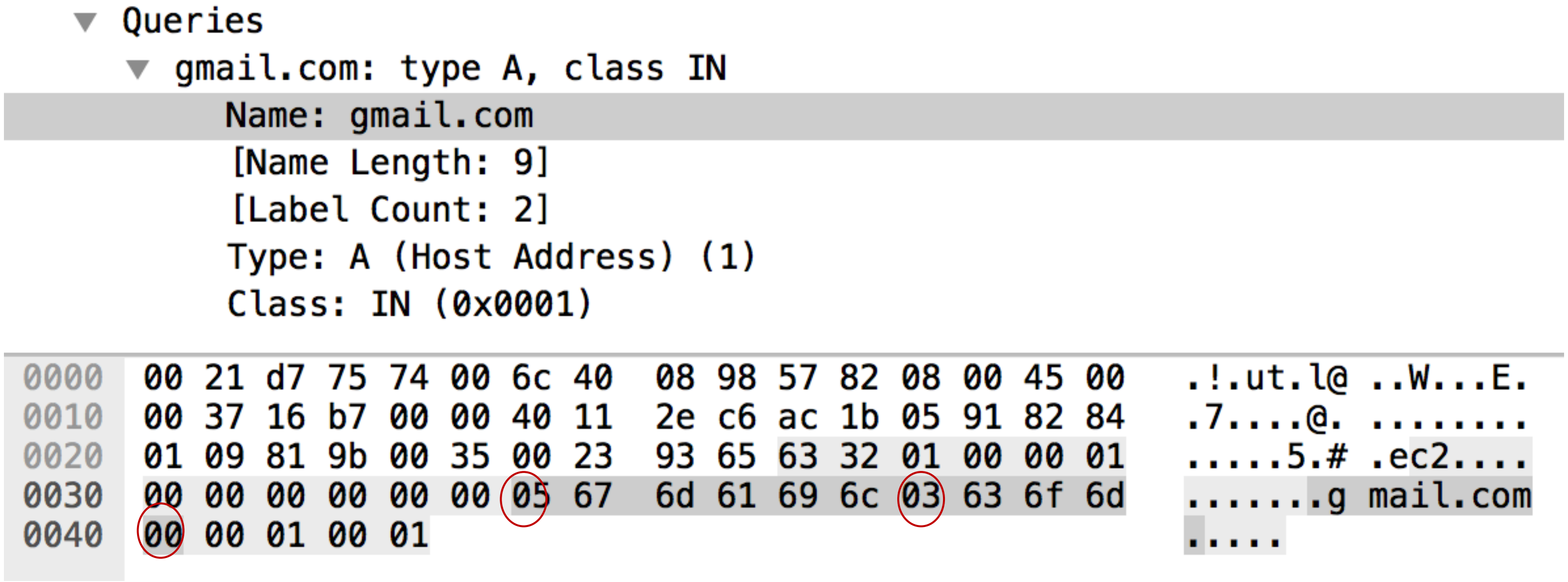
Additional information
(variable number of resource records)

Additional "helpful" info that may be used

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

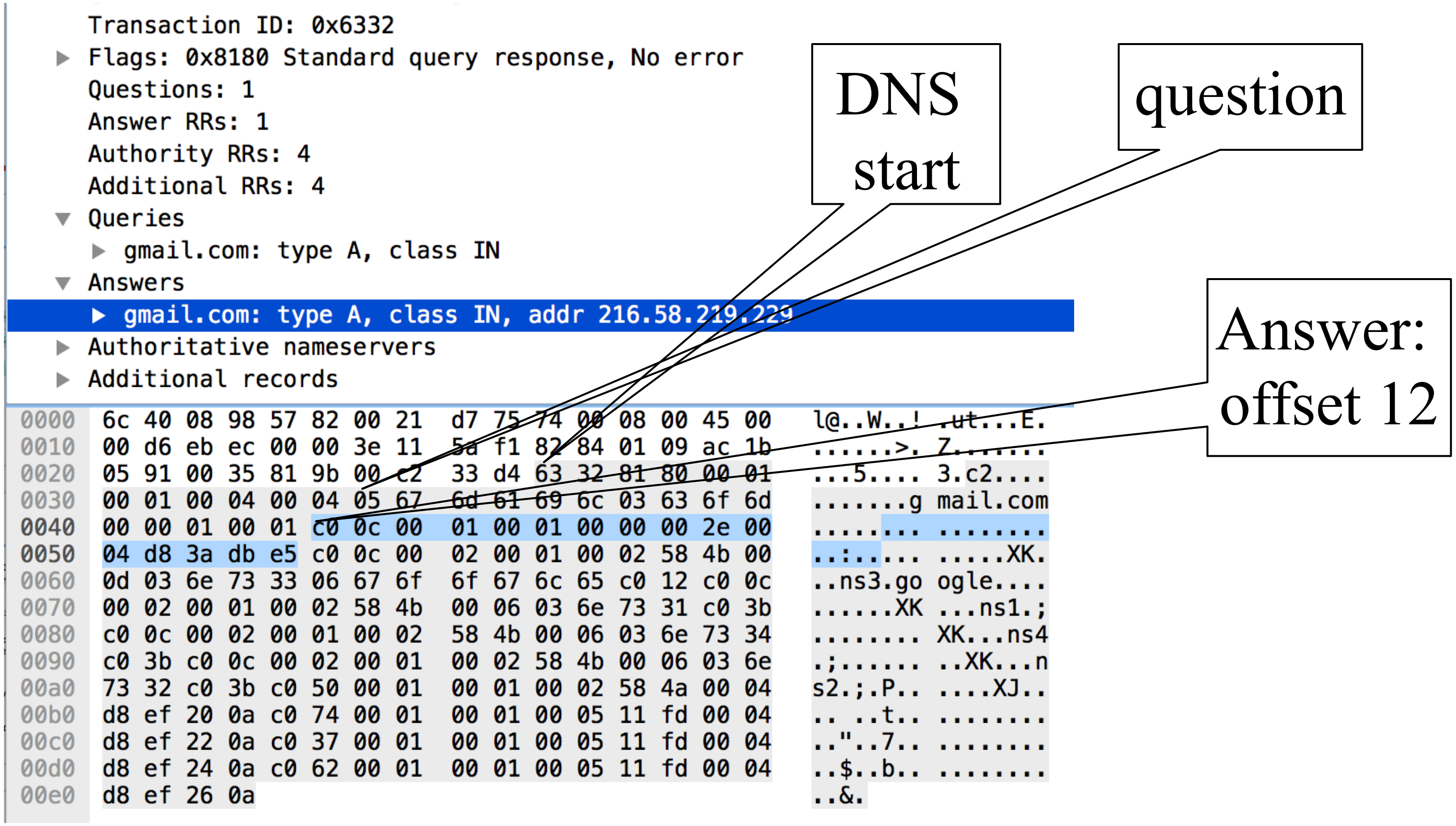


```
▼ Queries
  ▼ gmail.com: type A, class IN
       Name: gmail.com
       [Name Length: 9]
       [Label Count: 2]
       Type: A (Host Address) (1)
       Class: IN (0x0001)

0000  00 21 d7 75 74 00 6c 40  08 98 57 82 08 00 45 00   .!.ut.l@ ..W...E.
0010  00 37 16 b7 00 00 40 11  2e c6 ac 1b 05 91 82 84   .7....@. ........
0020  01 09 81 9b 00 35 00 23  93 65 63 32 01 00 00 01   .....5.# .ec2....
0030  00 00 00 00 00 00 05 67  6d 61 69 6c 03 63 6f 6d   .......g mail.com
0040  00 00 01 00 01                                      .....
```

Message Compression (Label Pointer)

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| 1  1|                OFFSET                   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```



```
Transaction ID: 0x6332
▶ Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 4
  Additional RRs: 4
▼ Queries
  ▶ gmail.com: type A, class IN
▼ Answers
  ▶ gmail.com: type A, class IN, addr 216.58.219.229
▶ Authoritative nameservers
▶ Additional records

0000  6c 40 08 98 57 82 00 21  d7 75 74 00 08 00 45 00   l@..W..! .ut...E.
0010  00 d6 eb ec 00 00 3e 11  5a f1 82 84 01 09 ac 1b   ......>. Z.......
0020  05 91 00 35 81 9b 00 c2  33 d4 63 32 81 80 00 01   ...5.... 3.c2....
0030  00 01 00 04 00 04 05 67  6d 61 69 6c 03 63 6f 6d   .......g mail.com
0040  00 00 01 00 01 c0 0c 00  01 00 01 00 00 00 2e 00   ........ ........
0050  04 d8 3a db e5 c0 0c 00  02 00 01 00 02 58 4b 00   ..:..... .....XK.
0060  0d 03 6e 73 33 06 67 6f  6f 67 6c 65 c0 12 c0 0c   ..ns3.go ogle....
0070  00 02 00 01 00 02 58 4b  00 06 03 6e 73 31 c0 3b   ......XK ...ns1.;
0080  c0 0c 00 02 00 01 00 02  58 4b 00 06 03 6e 73 34   ........ XK...ns4
0090  c0 3b c0 0c 00 02 00 01  00 02 58 4b 00 06 03 6e   .;...... ..XK...n
00a0  73 32 c0 3b c0 50 00 01  00 01 00 02 58 4a 00 04   s2.;.P.. ....XJ..
00b0  d8 ef 20 0a c0 74 00 01  00 01 00 05 11 fd 00 04   .. ..t.. ........
00c0  d8 ef 22 0a c0 37 00 01  00 01 00 05 11 fd 00 04   .."..7.. ........
00d0  d8 ef 24 0a c0 62 00 01  00 01 00 05 11 fd 00 04   ..$..b.. ........
00e0  d8 ef 26 0a                                        ..&.
```

DNS start

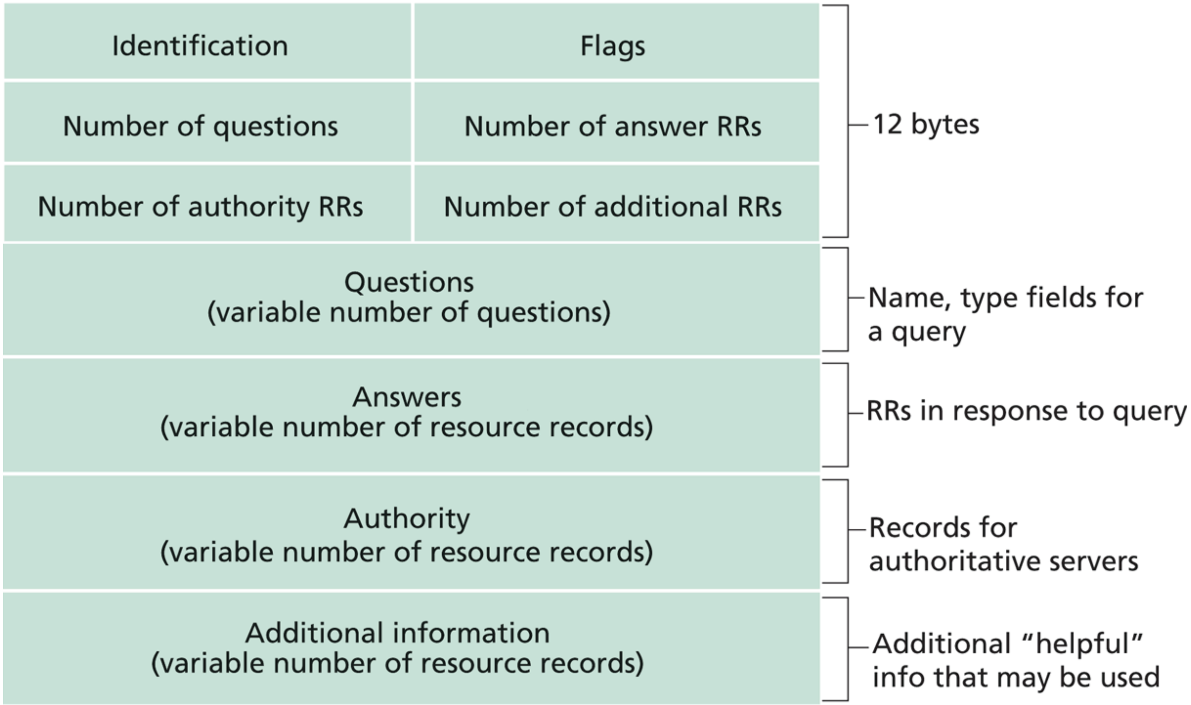
question

Answer: offset 12

https://www.ietf.org/rfc/rfc1035.txt

Summary: 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*



Identification	Flags
Number of questions	Number of answer RRs
Number of authority RRs	Number of additional RRs

12 bytes

Questions
(variable number of questions) — Name, type fields for a query

Answers
(variable number of resource records) — RRs in response to query

Authority
(variable number of resource records) — Records for authoritative servers

Additional information
(variable number of resource records) — Additional "helpful" info that may be used

Discussion: What DNS did Right