Application Layer Protocols

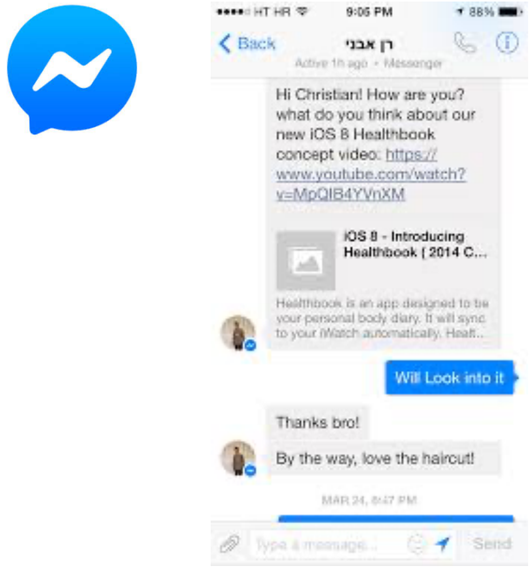
Peerapon S.

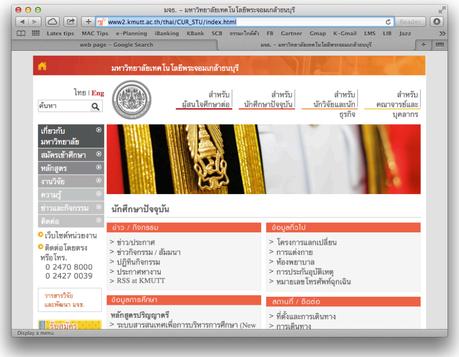
CPE 314 Computer Networks

Topics

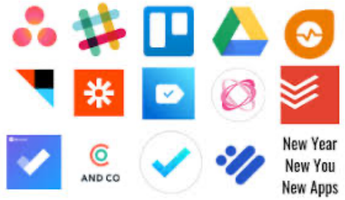
Network applications

Computer processes exchanging
(user/system) data to achieve certain goals.





TCP/IP Protocol Suite



Application layer

Name & Dir system: DNS, LDAP, RADIUS

Host Config: BOOTP, DHCP

Email: SMTP, POP, IMAP

File transfer: FTP, TFTP

Web: HTTP

Multi media: RTP, RTSP, RTCP

Signaling: SIP, SDP, H.323

IoT: CoAP, MQTT, DDS

TLS/SSL

Transport layer

UDP, TCP, SCTP

Internet layer

IP support: ICMP, NAT

IPv4, IPv6

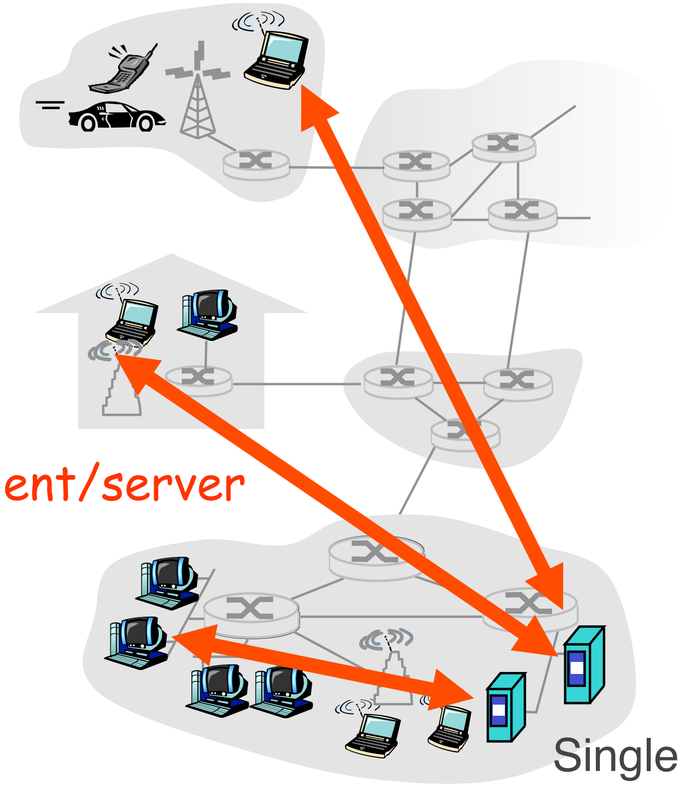
Routing: OSPF

ARP, 6LoWPAN

Network access layer

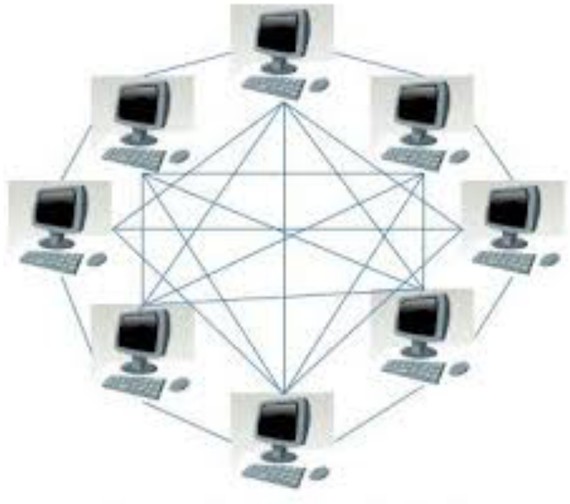
Ethernet, Wi-Fi, BLE, 802.15.4, ADSL, DOCSIS, 4G/5G, NB-IOT

Network Application Models

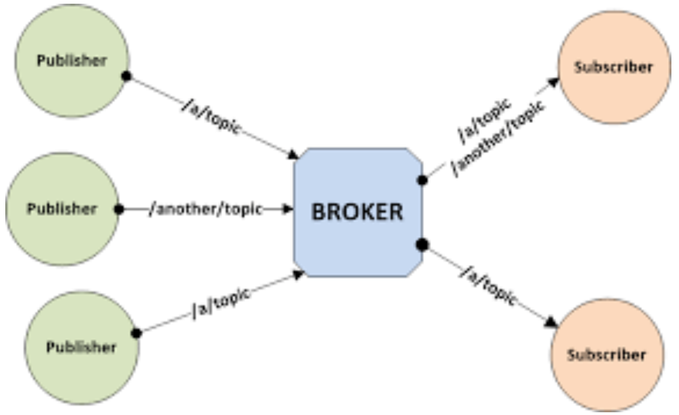


client/server

Single servers
or Data center



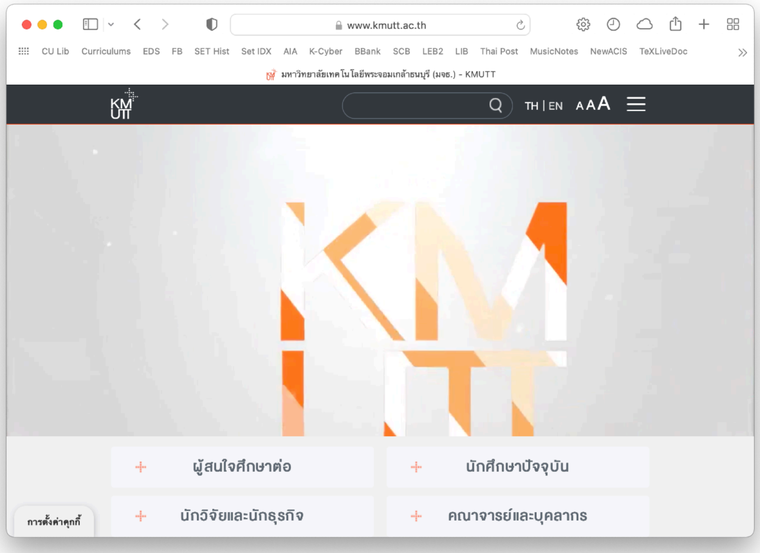
peer-to-peer



Broker-based

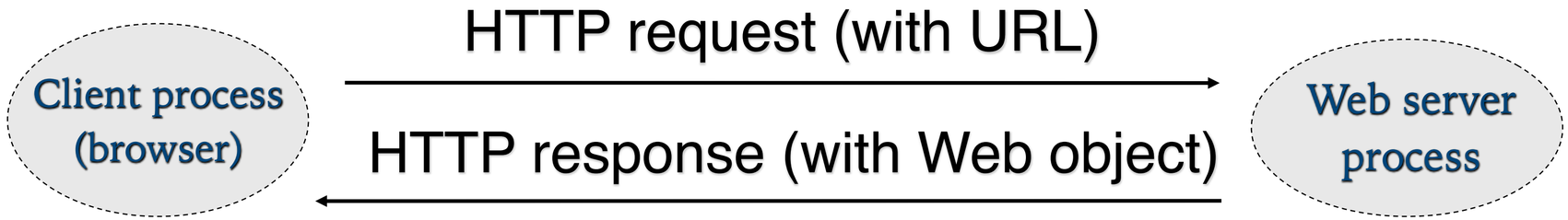
Web Application

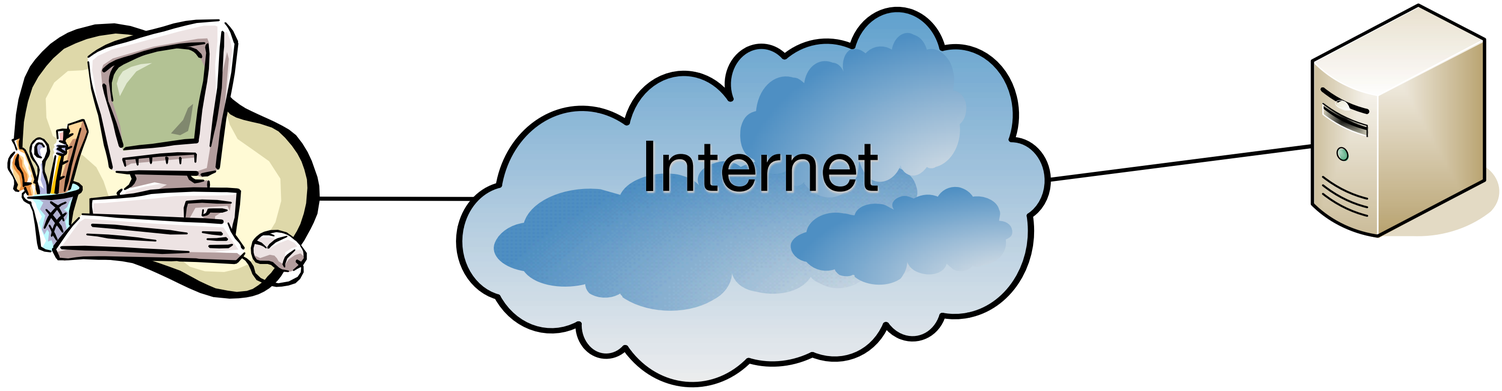
www.kmutt.ac.th

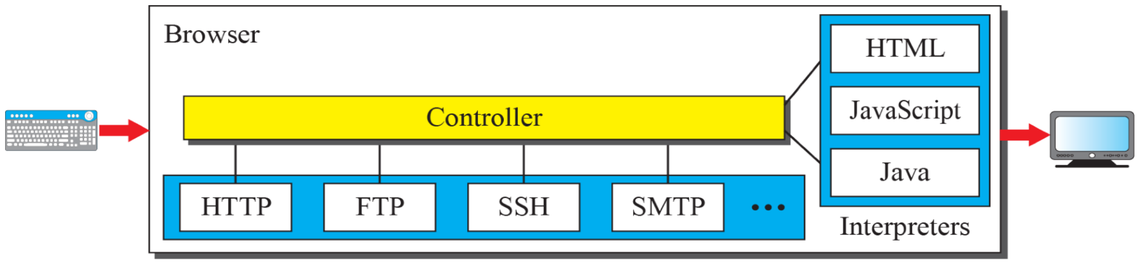


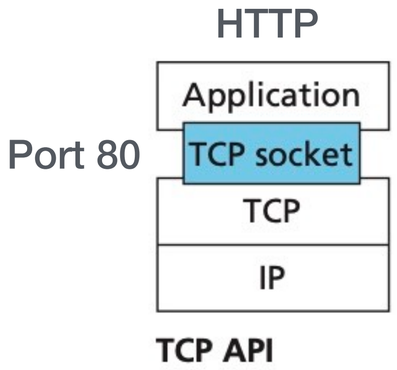
Web objects (HTML/JS contents, css files, images, video, scripts, etc.) addressable via **Uniform Resource Locator (URL)**

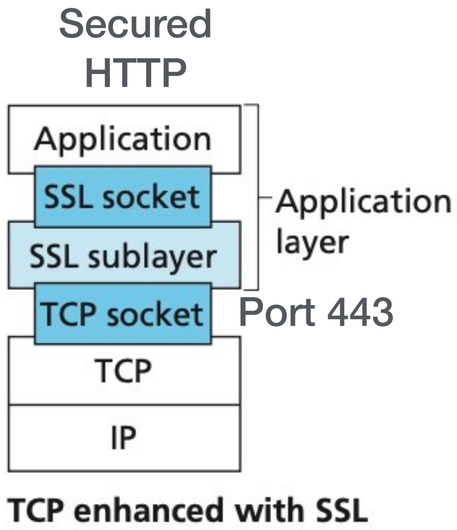
Hyper Text Transfer Protocol (HTTP)





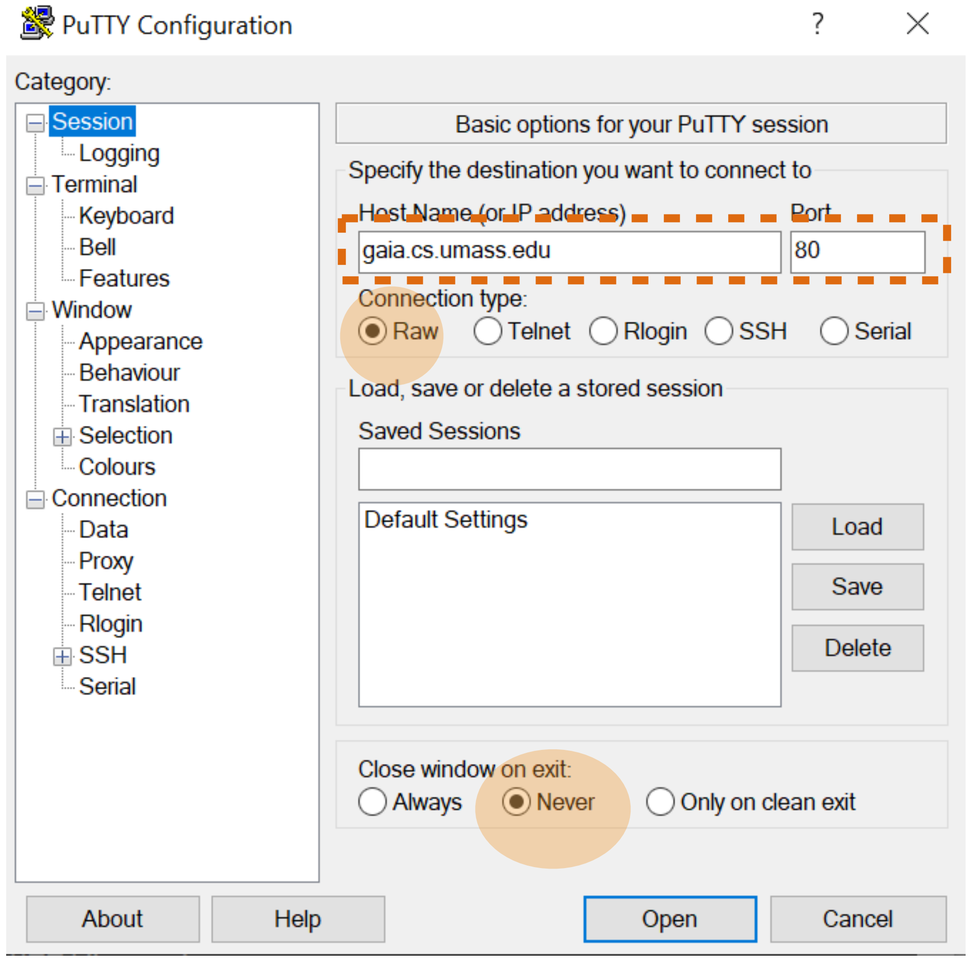






Trying Out HTTP Manually (MS Windows Client)

`https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html`



Trying out HTTP Manually (MacOS Client)

☐ At a terminal command prompt, type

```
$ nc -v gaia.cs.umass.edu 80
```

☐ After the connection is established, type

```
GET / HTTP/1.1
Host: gaia.cs.umass.edu
```

Press Ctrl-V, Enter, Enter after in the first line
Press Ctrl-V, Enter, Enter, Ctrl-V, Enter, Enter after in the 2nd line
and wait a few seconds

☐ What information do you see in the response ?

HTTP Request (GET method)

HTTP Response Header

HTTP Response Body

```
$ nc -v gaia.cs.umass.edu 80
Connection to gaia.cs.umass.edu port 80 [tcp/http] succeeded!
GET / HTTP/1.1^M
Host: gaia.cs.umass.edu^M
^M
HTTP/1.1 200 OK
Date: Wed, 20 Jan 2021 12:34:29 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.14 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 01 Mar 2016 18:57:50 GMT
ETag: "a5b-52d015789ee9e"
Accept-Ranges: bytes
Content-Length: 2651
Content-Type: text/html; charset=UTF-8

<html>
<head>
<title>Computer Network Research Group - UMass Amherst
</title>
</head>
<body bgcolor="#ffffff">
<center>
<p><img
src="cnrg_imap.jpg"
border="0" usemap="#cnrg_imapMAP">
<map name="cnrg_imapMAP">
<area coords="290,177,407,205" shape="rect" href="/networks/resources/index.html">
<area coords="163,178,275,205" shape="rect" href="/networks/education/index.html">
<area coords="62,165,145,191" shape="rect" href="/search.html">
<area coords="6,63,157,90" shape="rect" href="/networks/collaborations.html">
<area coords="64,7,146,34" shape="rect" href="/networks/people.html">
<area coords="163,7,270,33" shape="rect" href="/networks/research.html">
<area coords="288,6,417,33" shape="rect"
href="/networks/publications.html">
</map>
<P>
<BR>
<BR>
<P>
<table width=100% border=0 cellpadding=0 cellspacing=0>
  <tr>
```

Basic HTTP Methods

Method	Action
GET	Request a resource from the server
HEAD	Request metadata of resource
POST	Create a new resource at the server (Non-idempotent)
PUT	Update an existing resource at the server (Idempotent)
DELETE	Delete a source at the server

```
from urllib.request import urlopen
u = urlopen("http://gaia.cs.umass.edu/")
data = u.read()
print(data)
```

Ex: HTTP Response Status Codes

200 OK

- Request succeeded, requested object later in this message

301 Moved Permanently

- Requested object moved, new location specified later in this message (Location:)

400 Bad Request

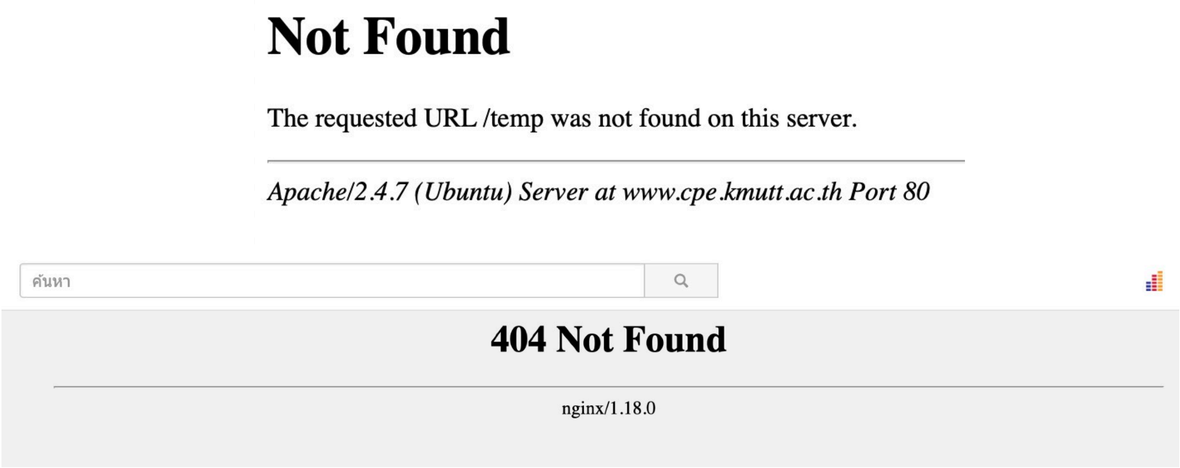
- Request message not understood by server

404 Not Found

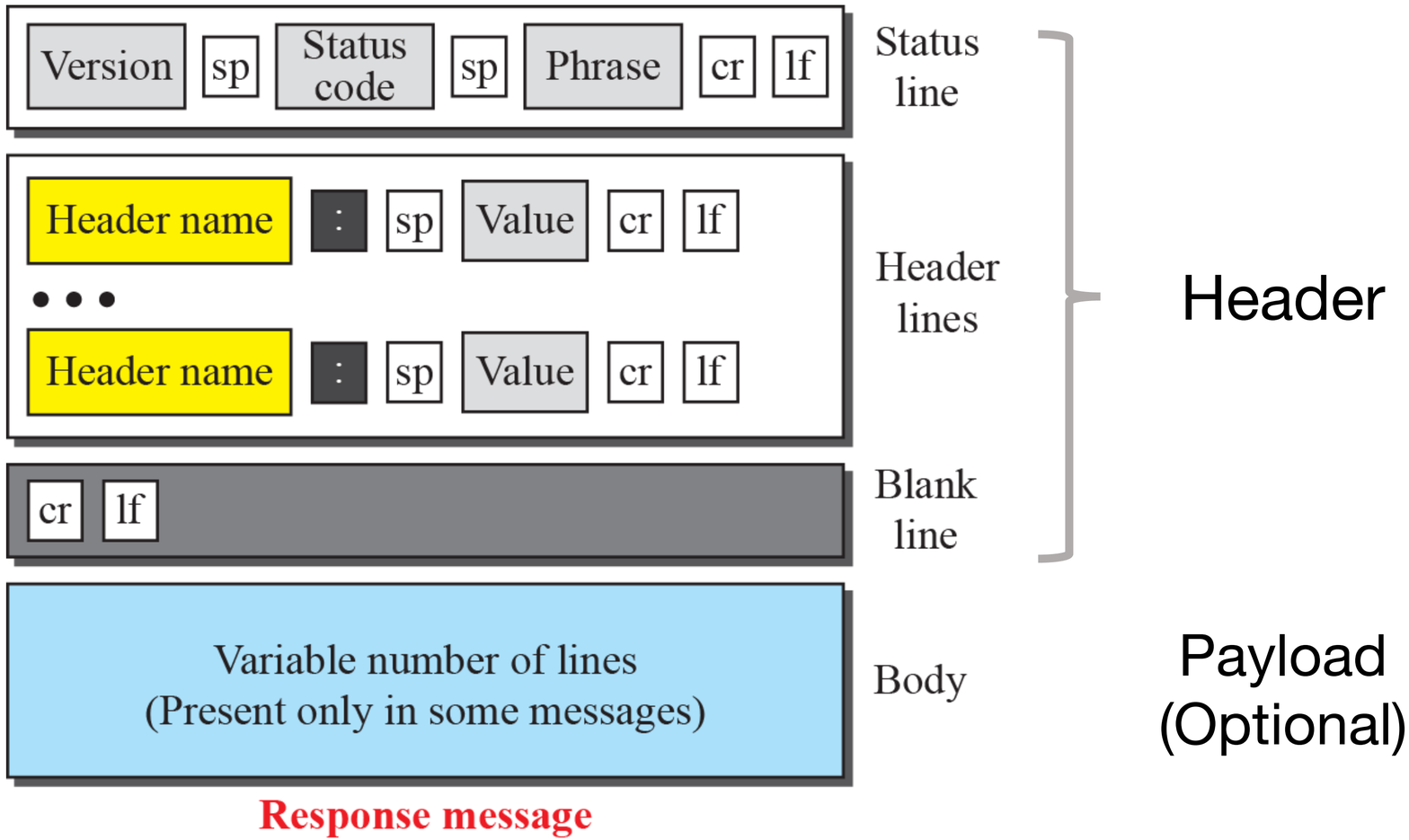
- Requested document not found on this server

502 Bad Gateway

505 HTTP Version Not Supported

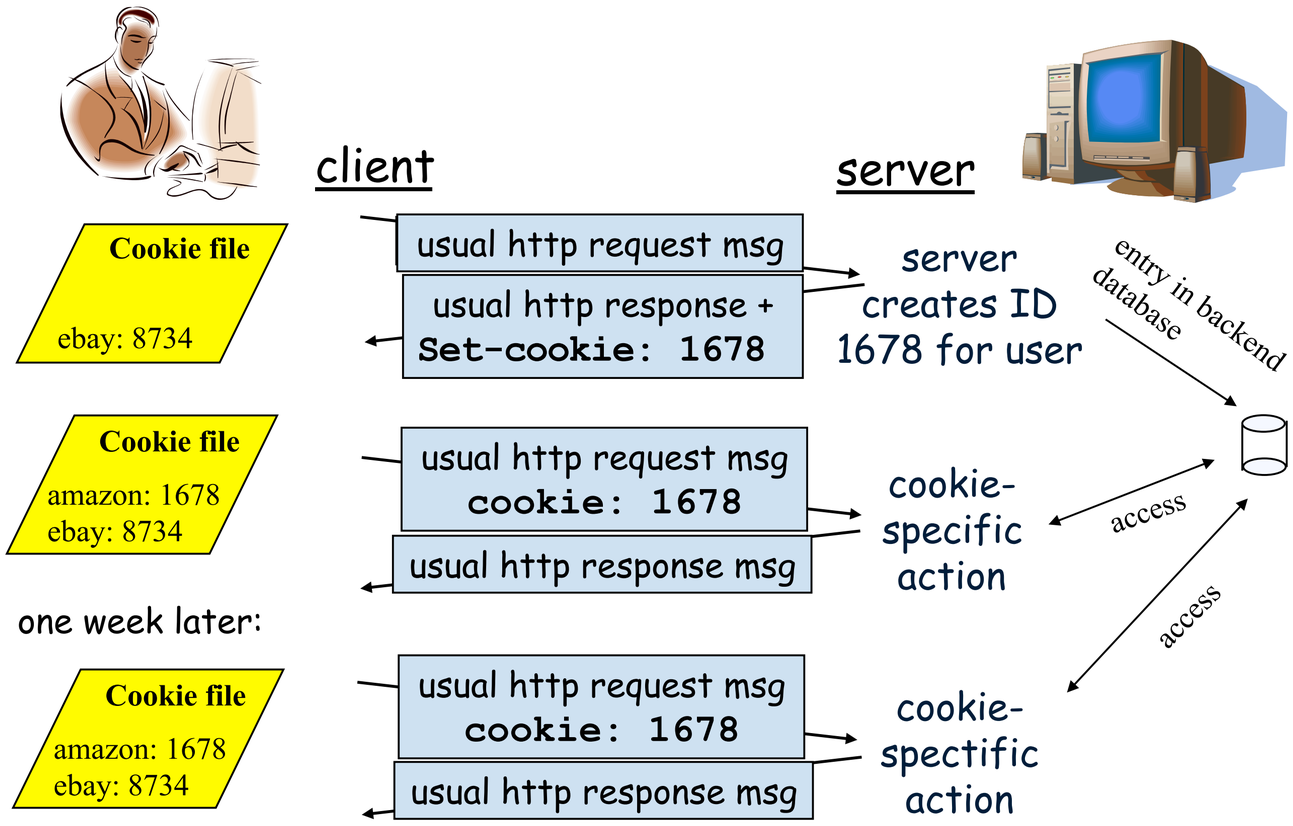


HTTP Response Message



Cookies

- HTTP is ***stateless*** by default.
- Cookies enables HTTP from being ***stateless*** to ***stateful***.

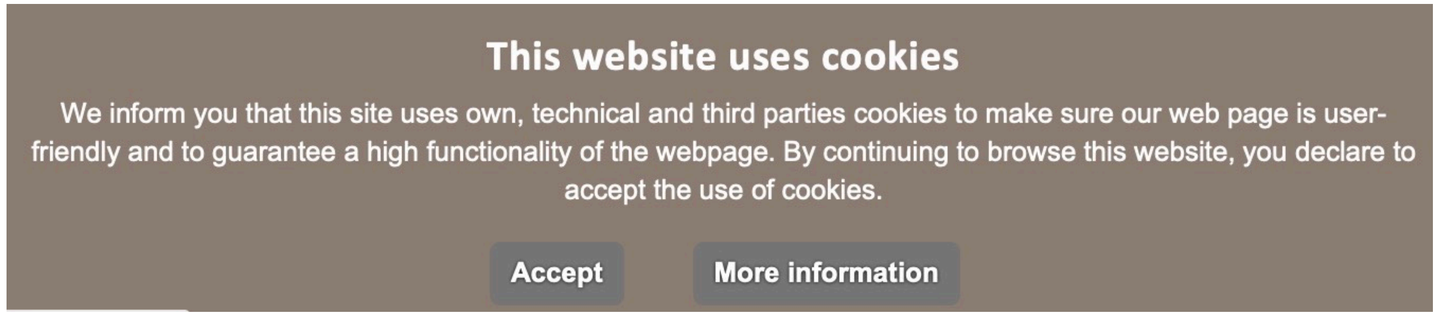


☐ What cookies can bring:

- Authorization
- Shopping carts
- Recommendations
- User session state (Web e-mail)

Cookies and privacy:

- Cookies permit sites to learn a lot about you
- You may supply name and e-mail to sites
- Search engines use redirection & cookies to learn yet more
- Advertising companies obtain info across sites



Domain Name System (DNS)

www.google.co.th
www.facebook.com
www.kmutt.ac.th

www2.kmutt.ac.th

540141@st.kmutt.ac.th
peerapon@gmail.com

(Humans preferred)

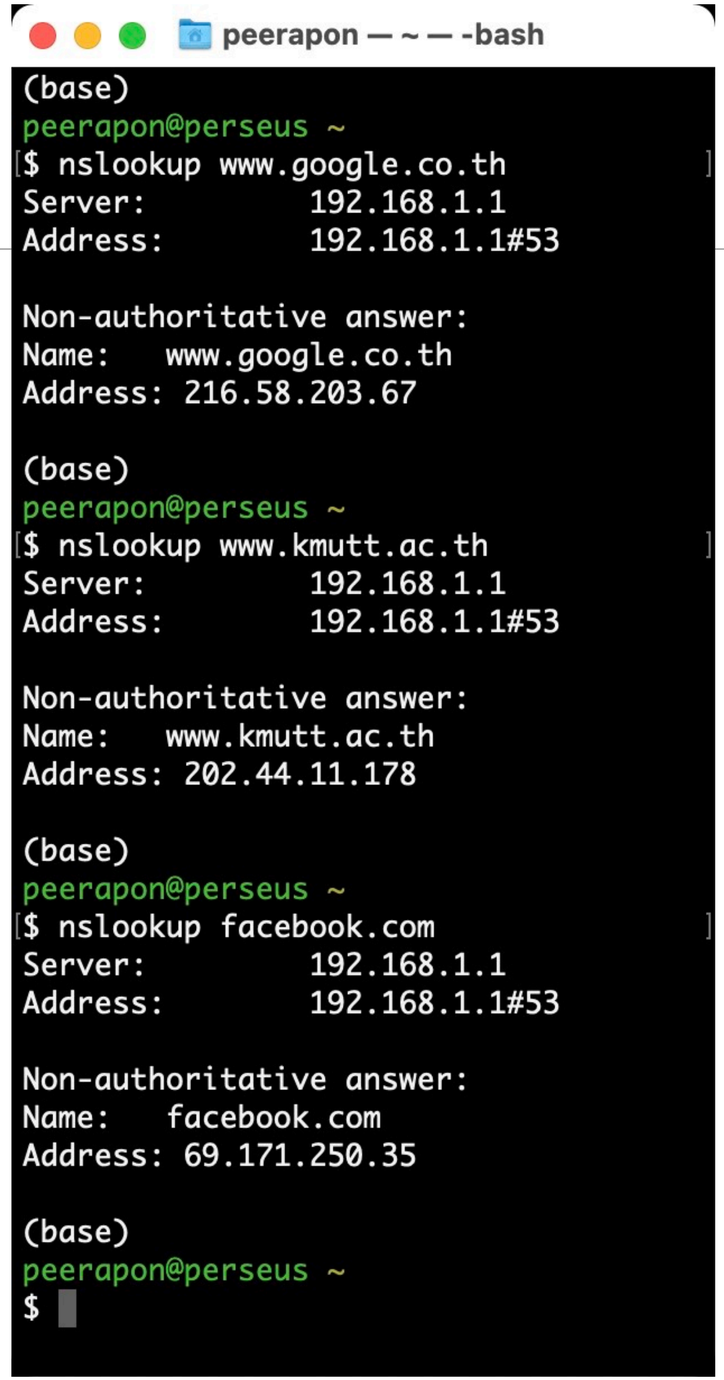
Name translation
Host aliasing
Mail server aliasing
Server load distribution

64.233.81.19
157.240.10.35
202.44.88.5
.....

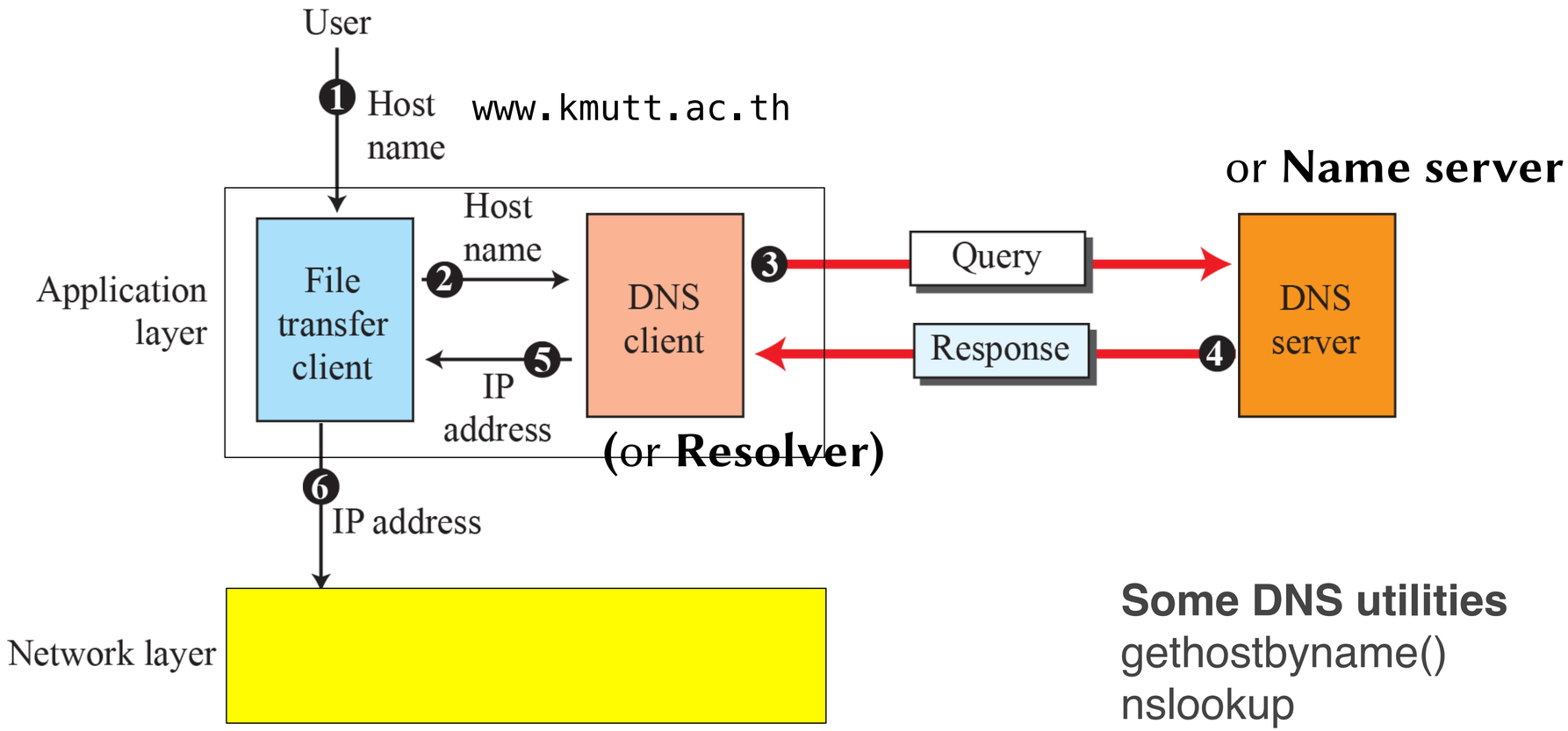
DNS is a critical service in the Internet !

DNS Lookup Demo

- Clearing DNS cache
  - MS Windows: `ipconfig /flushdns`
  - Mac OSX: `sudo dscacheutil –flushcache`



Name Resolution Steps



Some DNS utilities

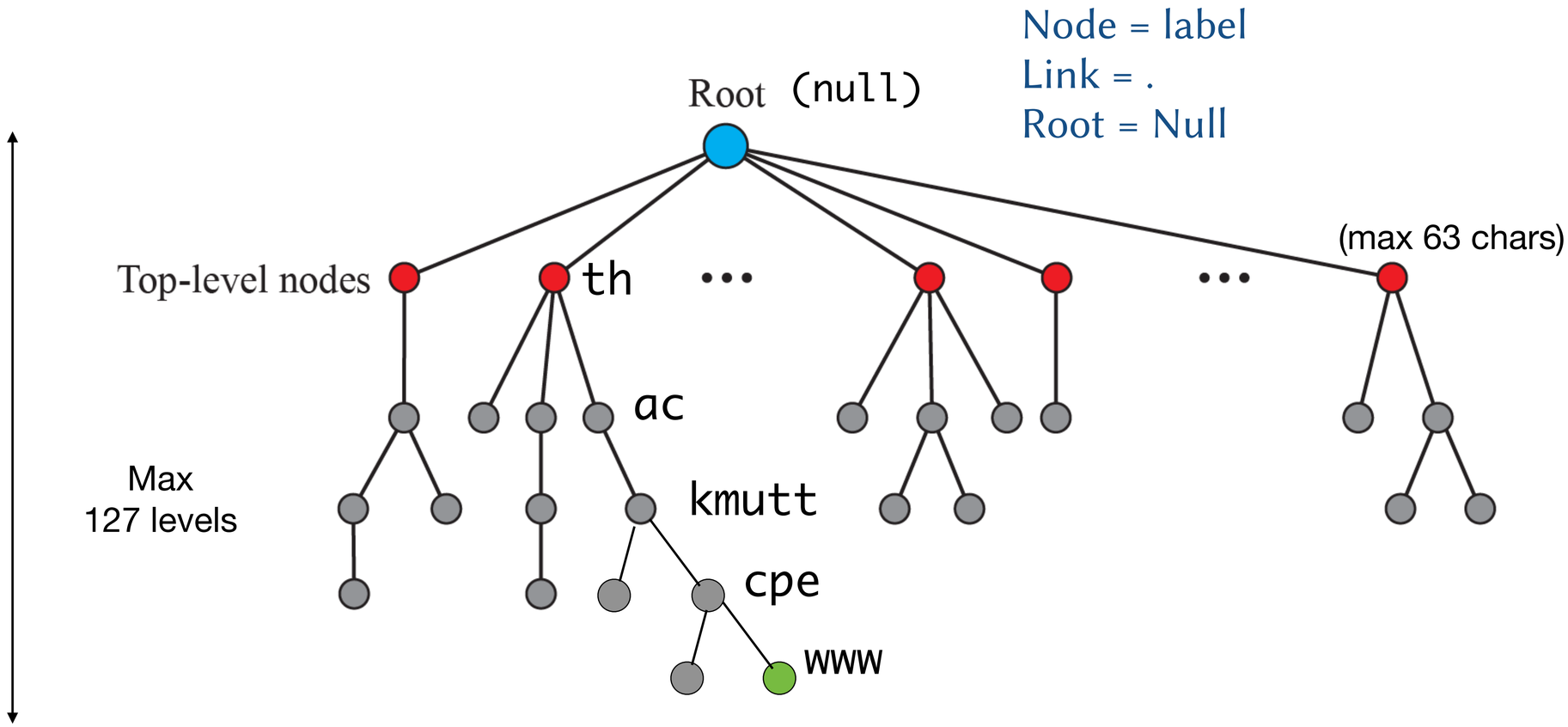
gethostbyname()

nslookup

whois

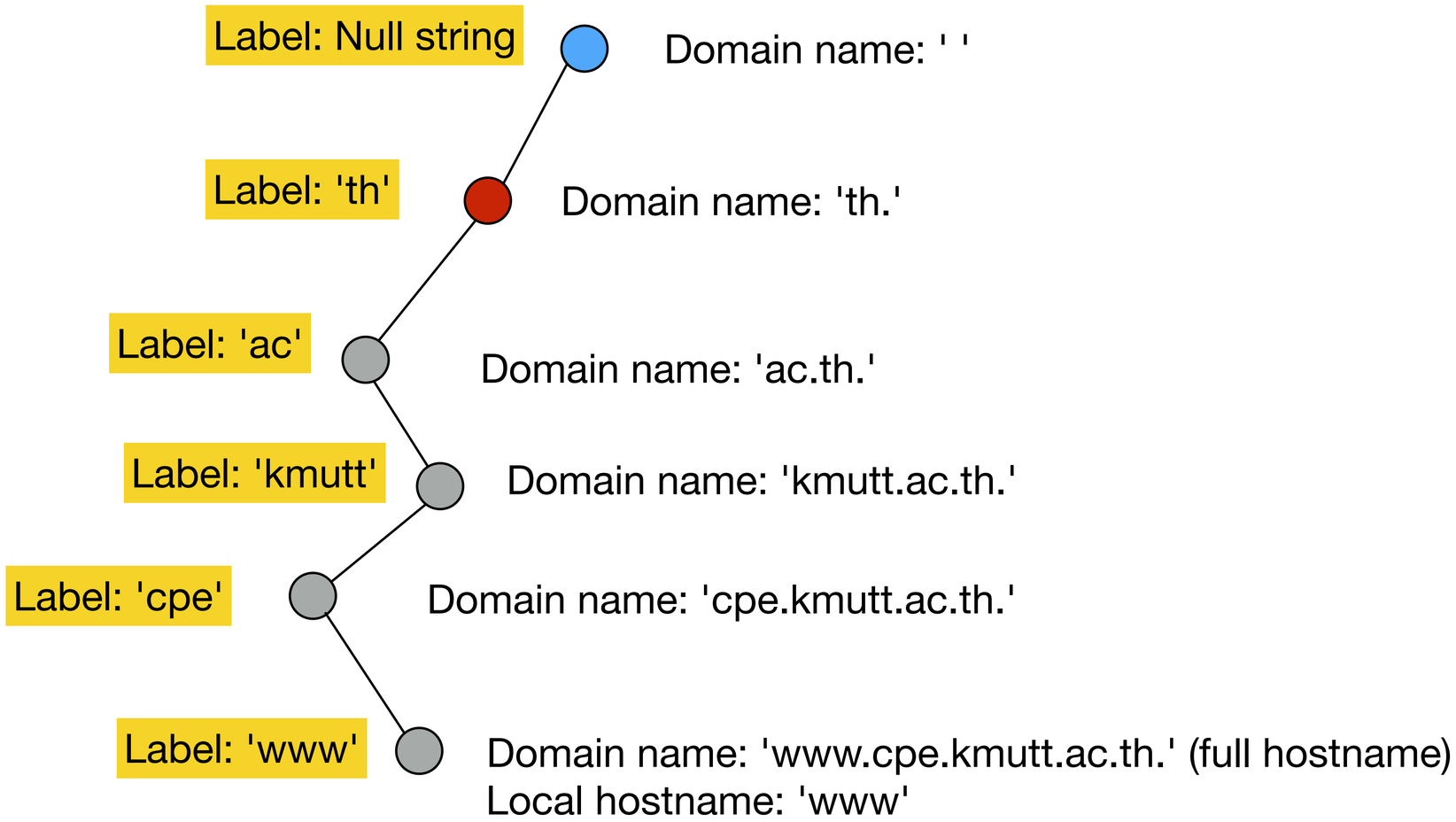
Domain Name Space and DNS Tree

- Dotted separated labels organized in hierarchical structure.
- Ex: www.cpe.kmutt.ac.th



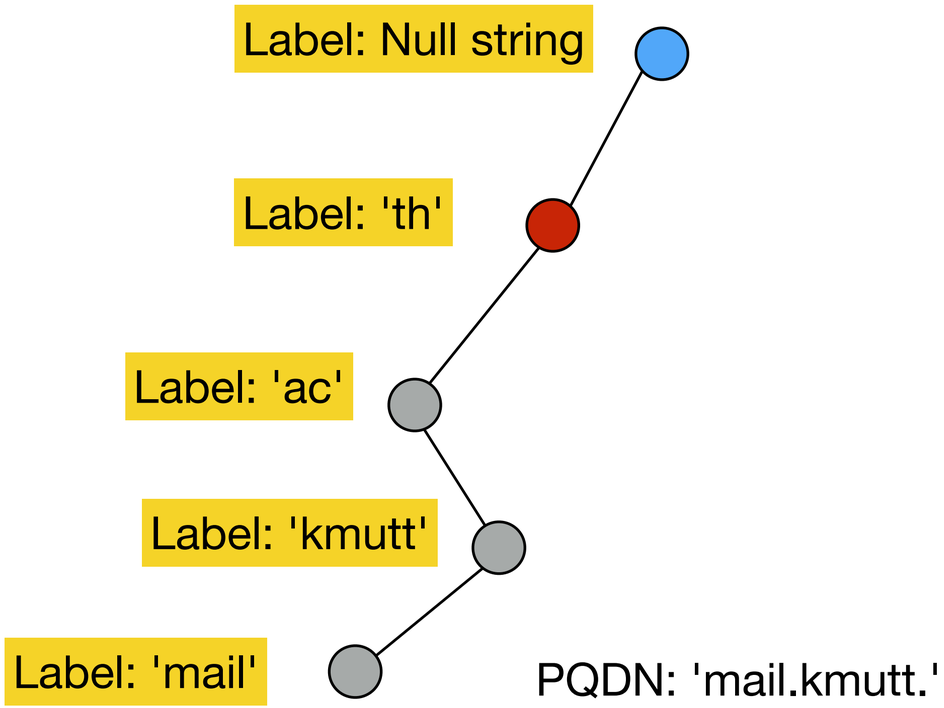
Full/Absolute Domain Name or Fully Qualified Domain Name (FQDN)

- Join labels separated by 'dot' upward to root.



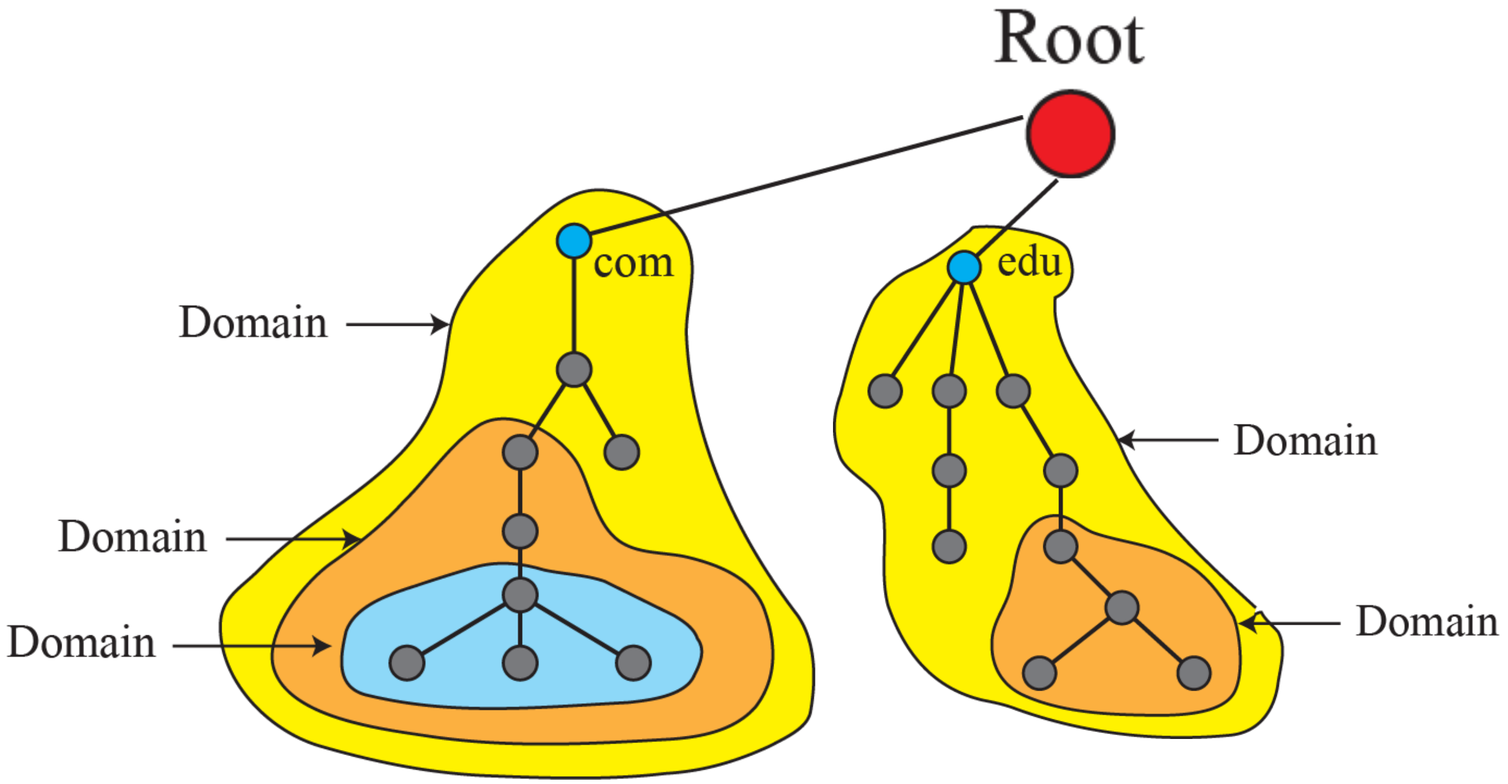
Relative Domain Name or Partially Qualified Domain Name (PQDN)

- Join labels separated by 'dot' but not reaching the root.
- Resolver applies suffix (the missing part) to create an FQDN.



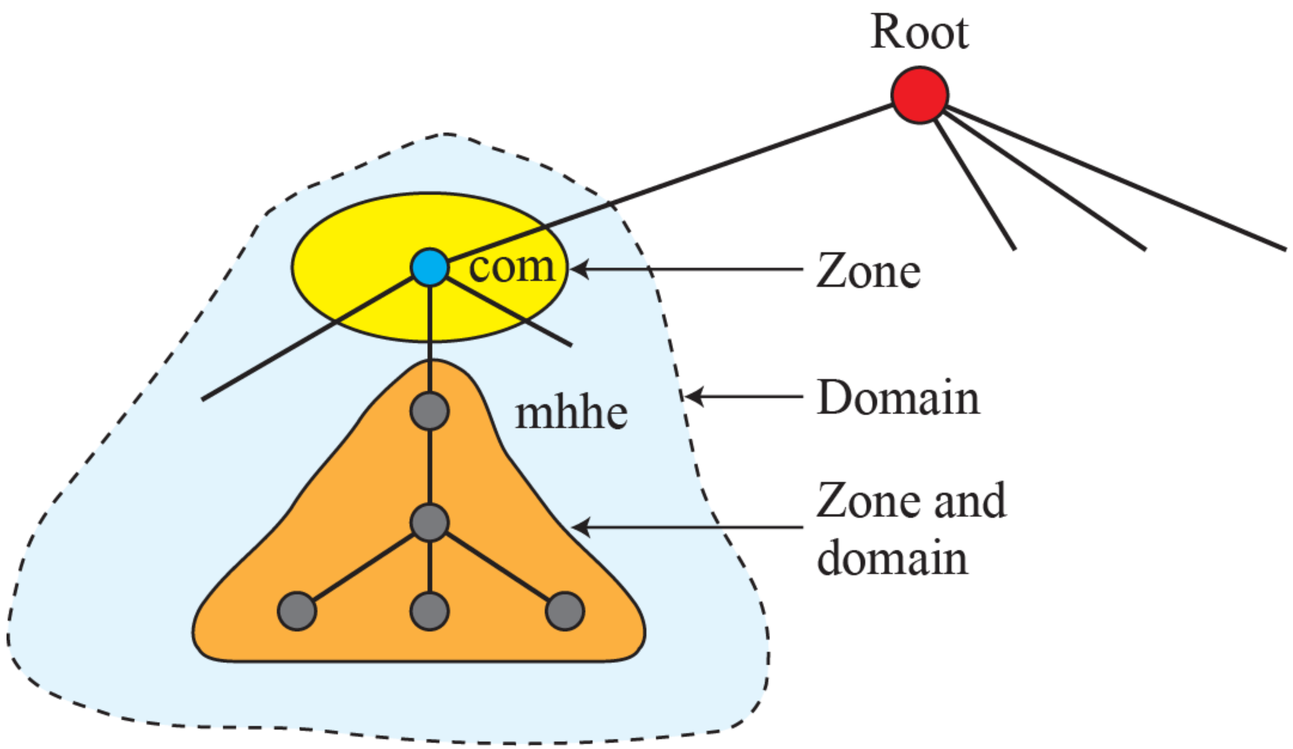
Domain

☐ Subtree of the domain name space rooted at a given node.



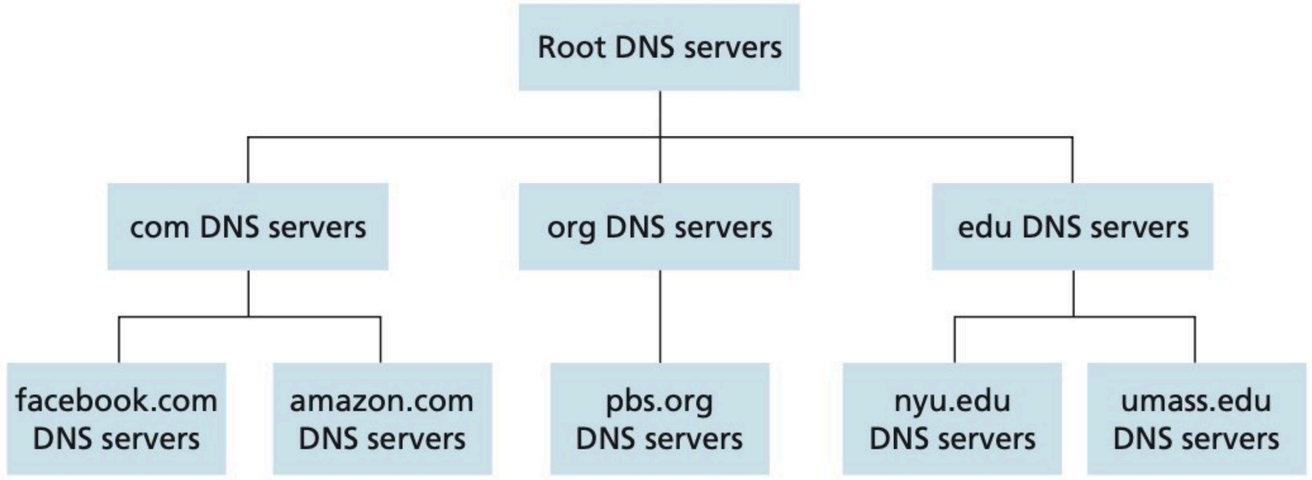
Zone

☐ Contiguous part of subtree, delegated to a legal entity typically for administrative purposes.

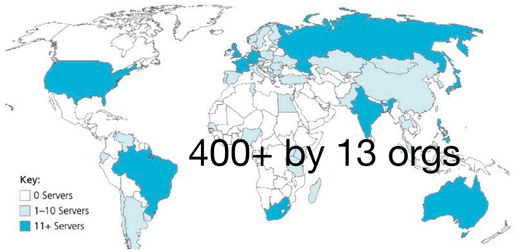


Distributed Name Server Implementation

- Hierarchical name structure allows distributed, hierarchical DNS database implementation through DNS/Name servers.
- Each server is responsible for a certain domain or zone.
- Name resolution done by asking DNS/Name servers from root downward the hierarchy.



Root DNS servers



TLD DNS servers

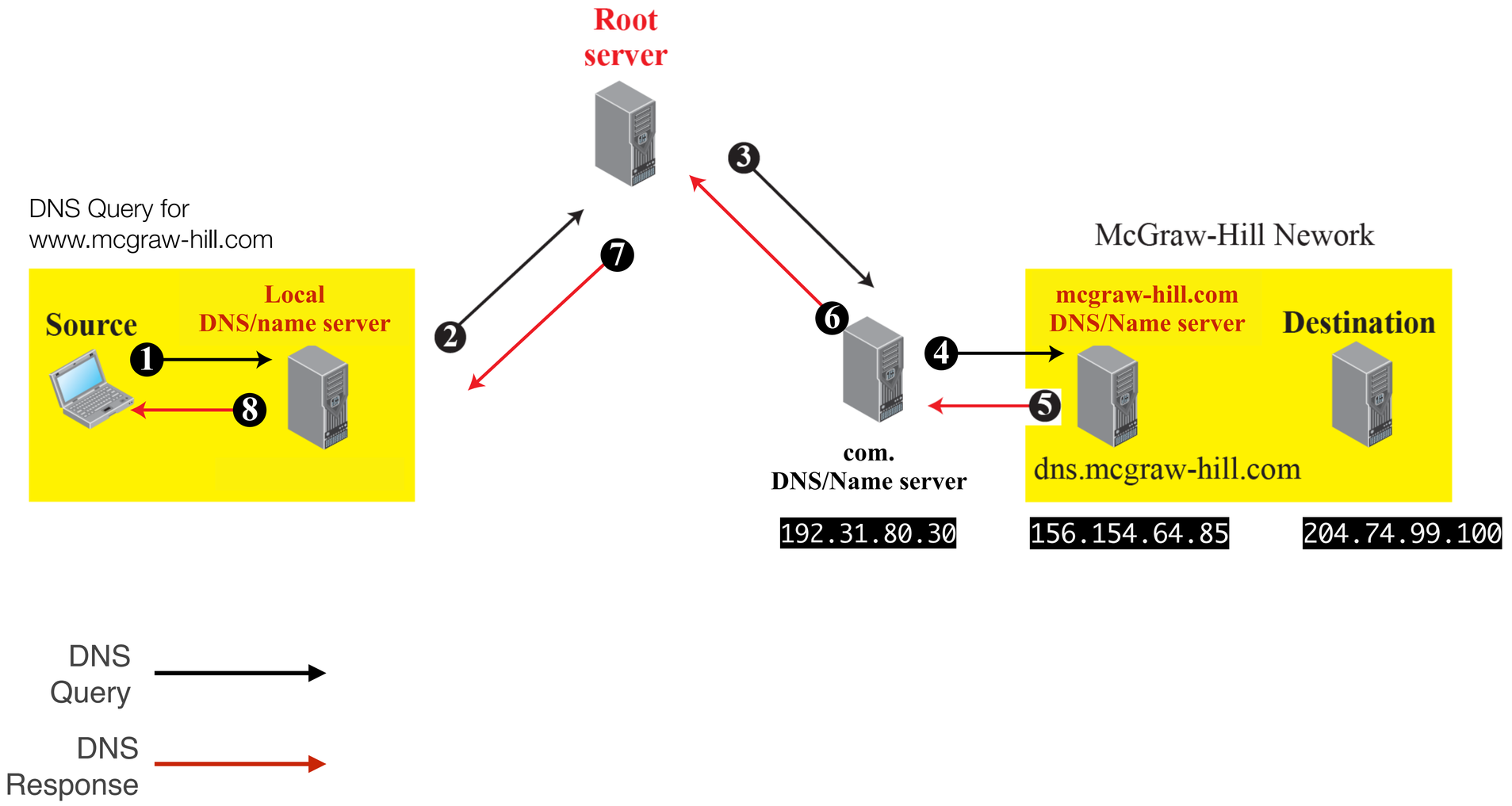
Managed by different companies and countries

Authoritative DNS servers
(Primary and secondary)

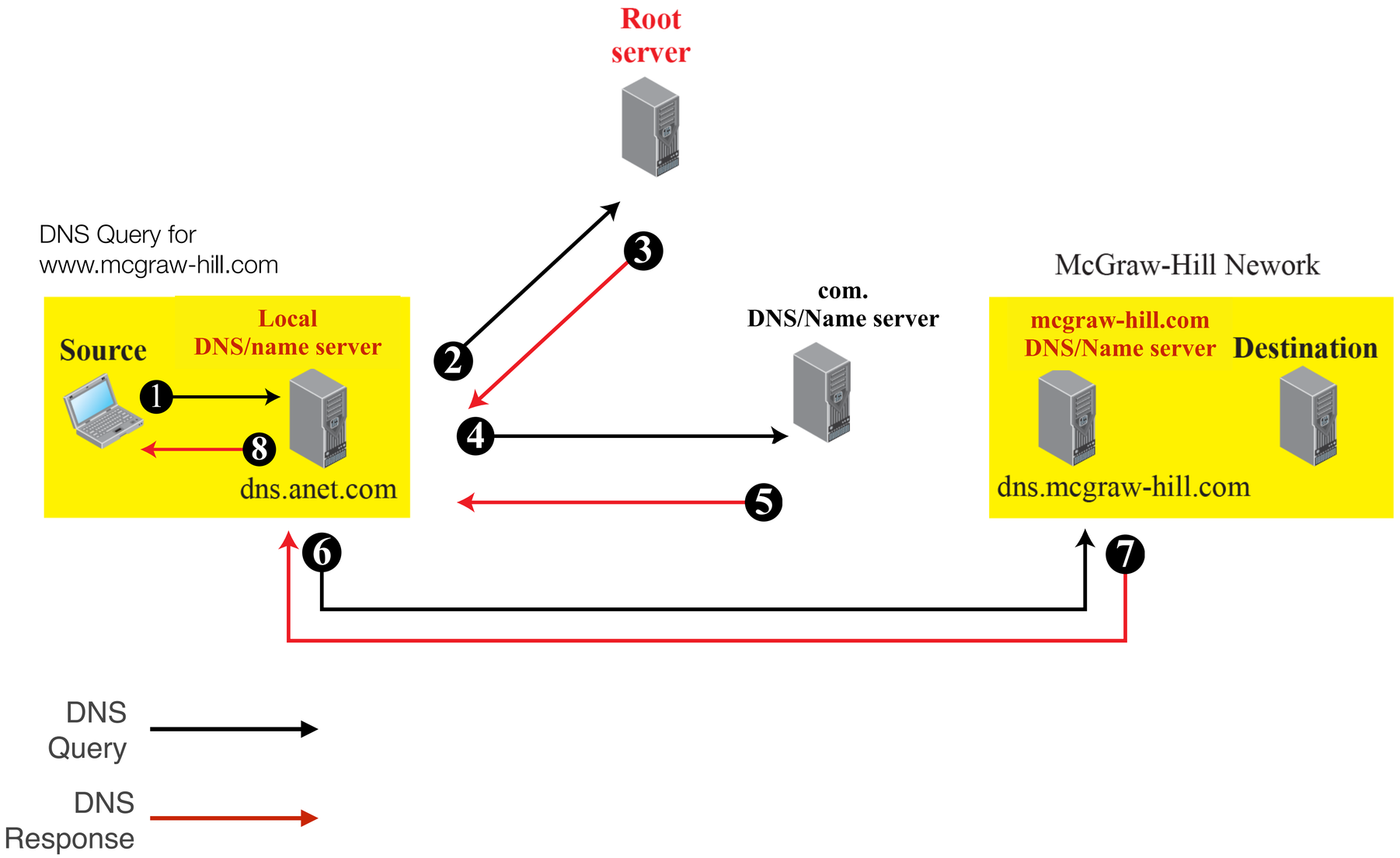
Managed by organizations

Local DNS servers

Recursive Resolution



Iterative Resolution

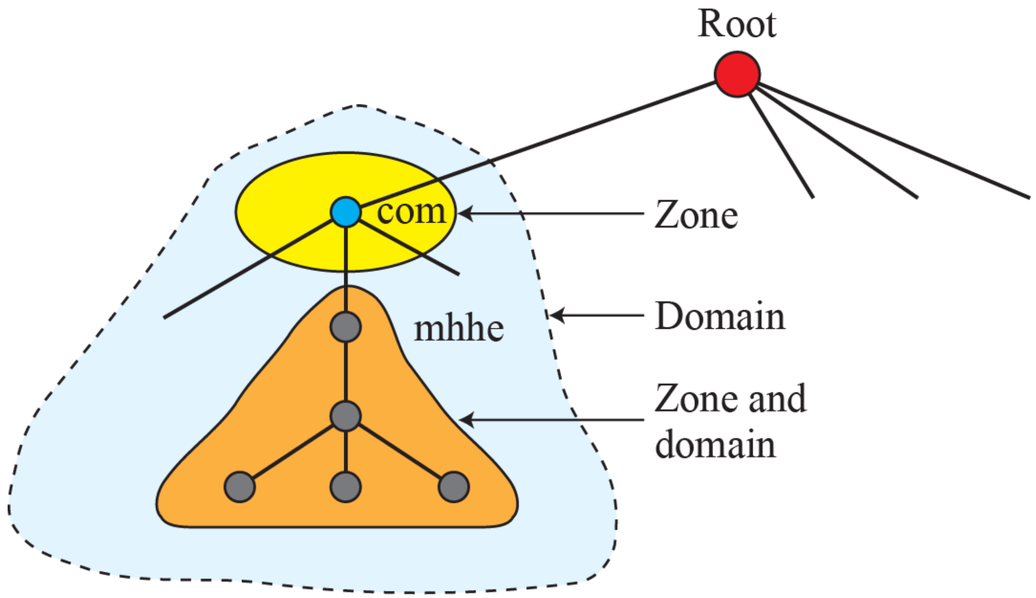


Authoritative DNS/Name server

☐ Each DNS/Name server has ***authority*** over a given ***zone***.

- Create and store original mapping information about the zone or hostnames.
- Ex: Domain name to IP address, Domain name to IP address of name servers.

☐ Two types: ***Primary*** and ***Secondary*** server.



Resource Records (RRs)

- Name server stores a database of resource records (RR) in its ***zone file***.
- Different structures based on RR types
- Ex: 5-tuple structure: {**Domain name**, **Class** , **Type**, *TTL*, **Value**}
  - Type defines how Domain name and Value are interpreted.
  - Ex: {**www.cpe.kmutt.ac.th**, **IN**, **A**, *86400*, **202.44.12,85**}

"Basic" RR types

Type	*Interpretation of value*
A	A 32-bit IPv4 address (see Chapter 4)
NS	Identifies the authoritative servers for a zone
CNAME	Defines an alias for the official name of a host
SOA	Marks the beginning of a zone
MX	Redirects mail to a mail server
AAAA	An IPv6 address (see Chapter 4)

/etc/bind/named.conf.local

Multiple domains in one zone file (same server)

```
zone "cpe.net" IN {
    type master;
    file "/etc/bind/zones/cpe.net.db";
};

zone "reslabs.cpe.net" IN {
    type master;
    file "/etc/bind/zones/cpe.net.db";
};
```

/etc/bind/named.conf.local

One domain in one zone file

```
zone "cpe.net" IN {
    type master;
    file "/etc/bind/zones/cpe.net.db";
};

zone "reslabs.cpe.net" IN {
    type master;
    file "/etc/bind/zones/reslabs.cpe.net.db";
};
```

/etc/bind/zones/cpe.net.db

```
 1: $TTL 86400    ; Default TTL for a DNS record cache in seconds (1 day)
 2: @  IN SOA ns1.cpe.net. root ( ; Must be canonical name, not alias name
 3:           1      ; serial
 4:           28800  ; refresh every 8 hours
 5:           14400   ; retry every 4 hours
 6:           3600000 ; expire
 7:           86400  ; TTL
 8:         )
 9:    IN NS ns2.cpe.net.
10:    IN MX mail.cpe.net.
11: www IN CNAME @
12: ns1  IN A  202.0.1.1
13: ns2  IN A  202.0.1.2
14: mail IN A  202.0.1.2
15: pc2  IN A  202.0.1.10
16: dns-server IN CNAME pc2
```

SOA RR
and configurable
parameters

Resource records

Authoritative vs. Non-authoritative Answers

```
$ nslookup -type=NS apple.com
Server:         202.44.8.66
Address:        202.44.8.66#53

Non-authoritative answer:
apple.com       nameserver = b.ns.apple.com.
apple.com       nameserver = c.ns.apple.com.
apple.com       nameserver = a.ns.apple.com.
apple.com       nameserver = d.ns.apple.com.

Authoritative answers can be found from:
a.ns.apple.com  internet address = 17.253.200.1
b.ns.apple.com  internet address = 17.253.207.1
c.ns.apple.com  internet address = 204.19.119.1
c.ns.apple.com  has AAAA address 2620:171:800:714::1
d.ns.apple.com  internet address = 204.26.57.1
d.ns.apple.com  has AAAA address 2620:171:801:714::1
```

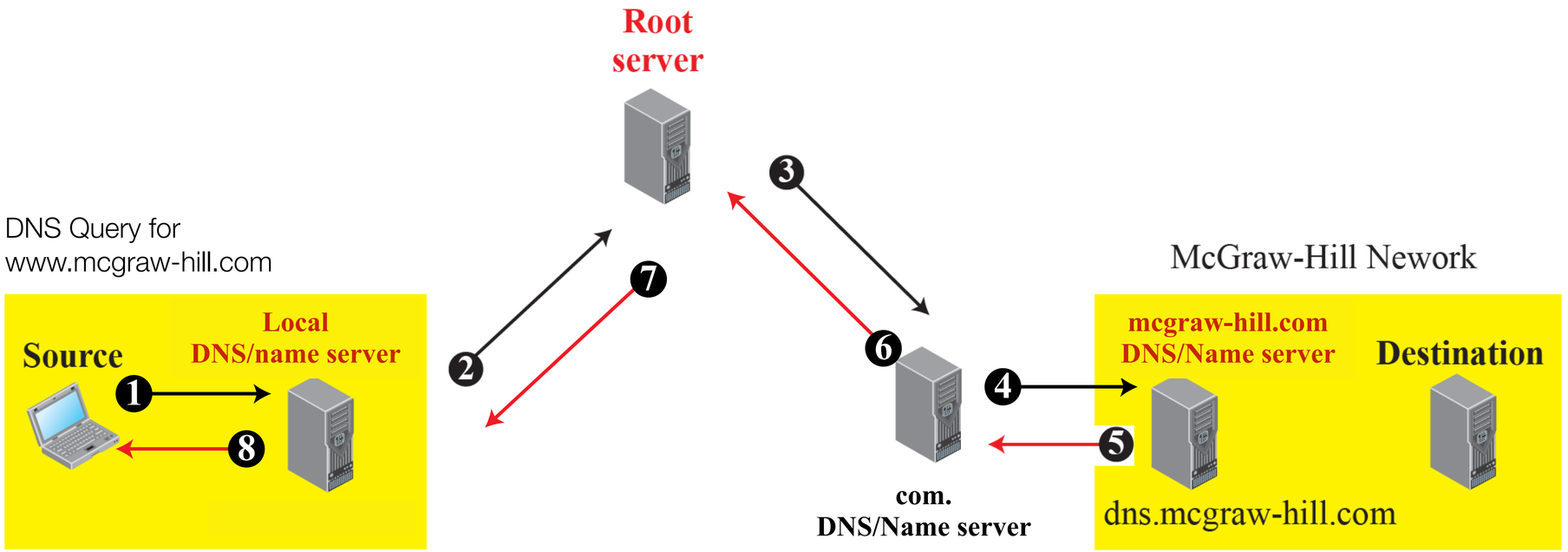
Organization DNS server

```
$ nslookup -type=NS useridbangkokbank.net
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
useridbangkokbank.net   nameserver = sima.ns.cloudflare.com.
useridbangkokbank.net   nameserver = valentin.ns.cloudflare.com.

Authoritative answers can be found from:
```

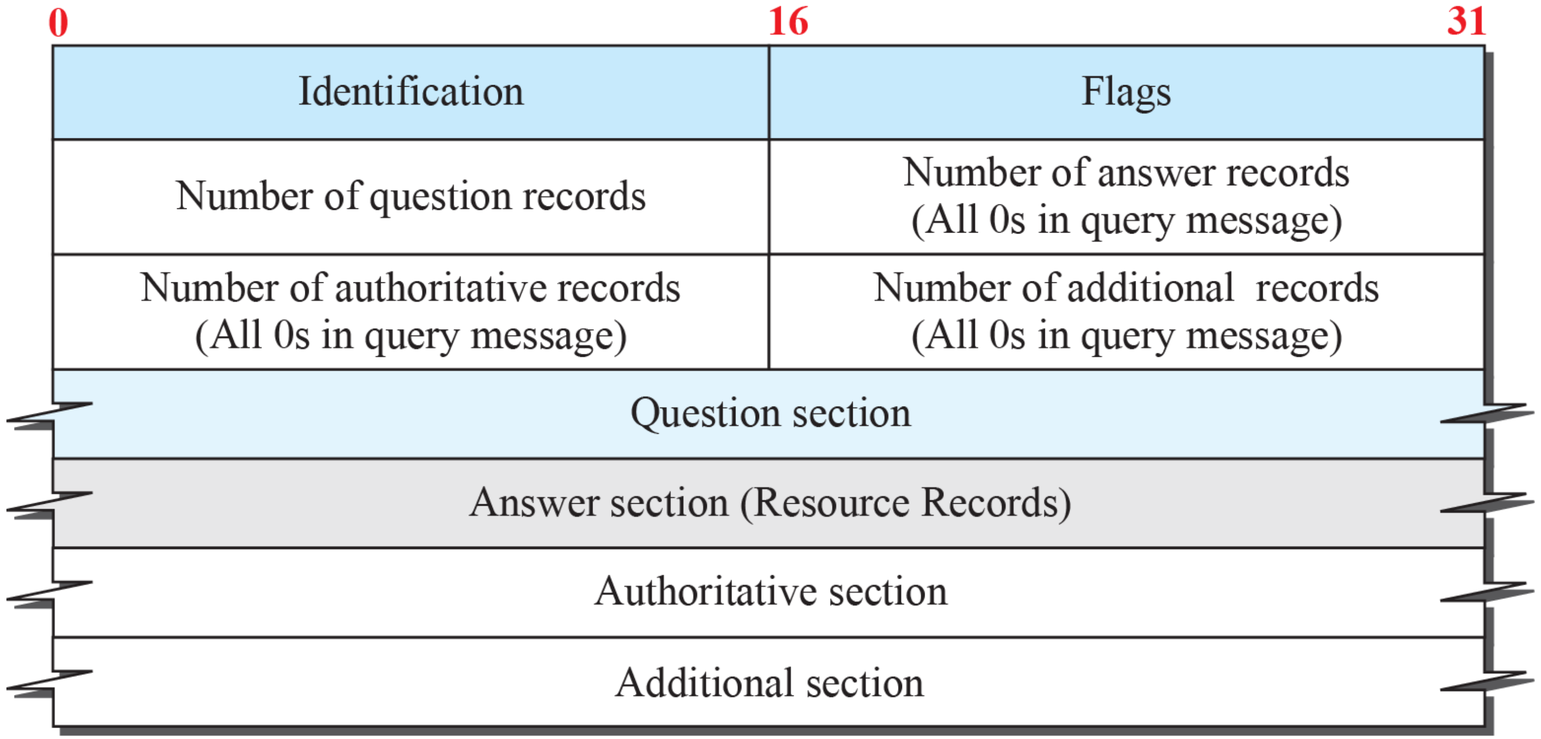
3rd party DNS server

In-Class Activity: Resource Records in DNS server

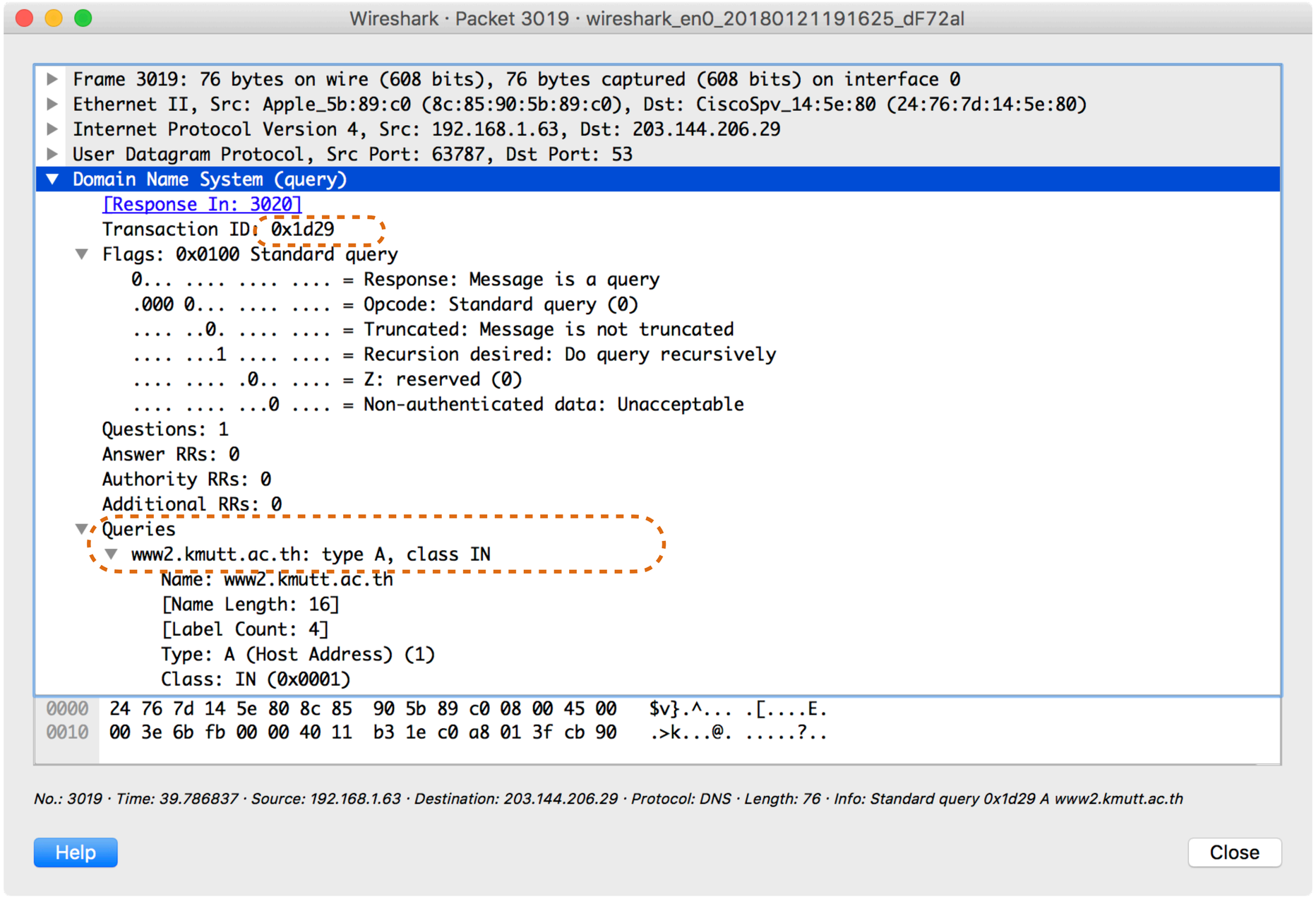
- From the network below, use the `nslookup` command to determine resource records <Domain name, Type, Class, Value> that must present at the root server so that the query in Step 3 can be carried out.

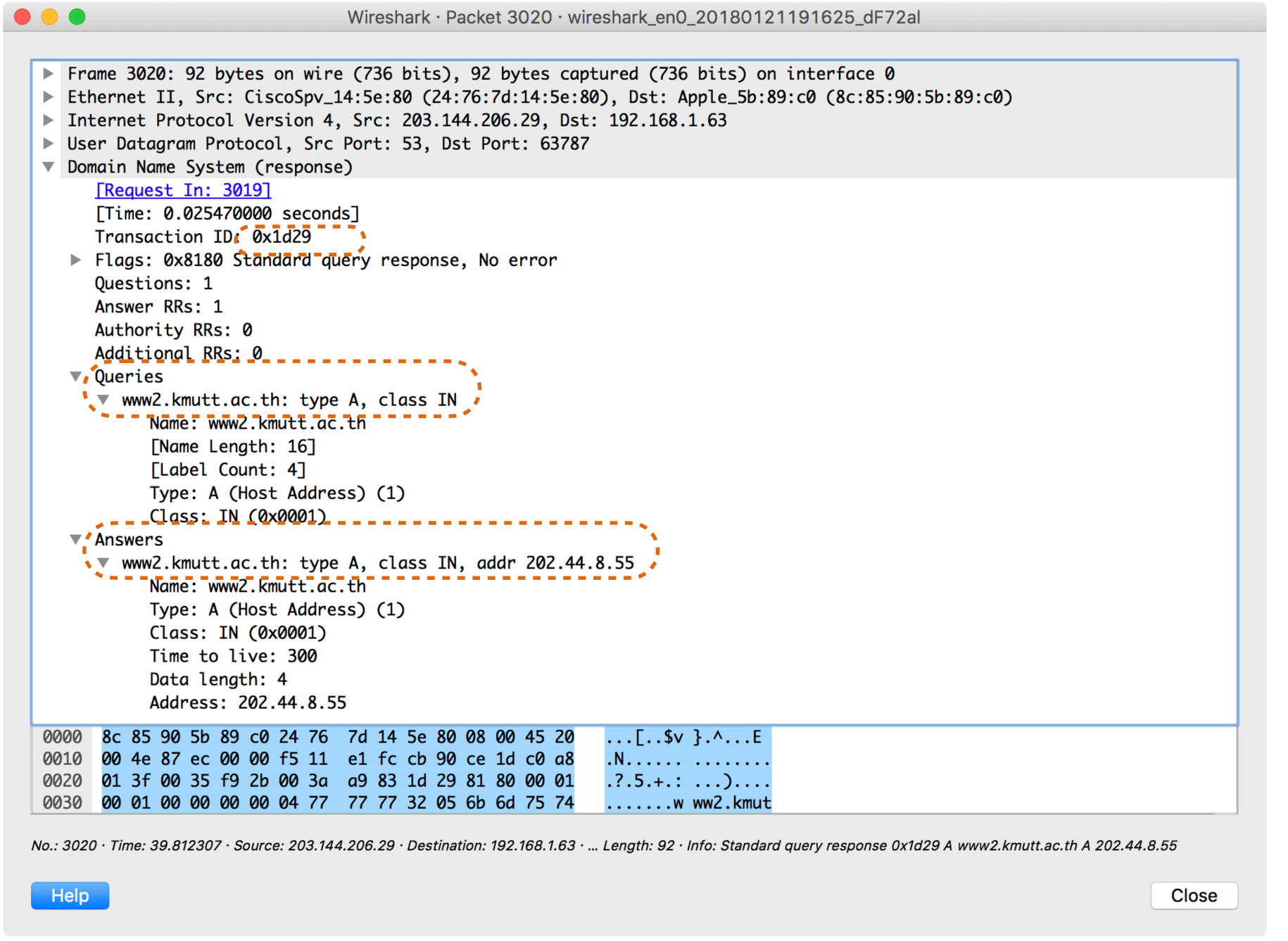


DNS Messages (Query, Response)




```
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
```


Wireshark · Packet 3019 · wireshark_en0_20180121191625_dF72al
Frame 3019: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
Ethernet II, Src: Apple_5b:89:c0 (8c:85:90:5b:89:c0), Dst: CiscoSpv_14:5e:80 (24:76:7d:14:5e:80)
Internet Protocol Version 4, Src: 192.168.1.63, Dst: 203.144.206.29
User Datagram Protocol, Src Port: 63787, Dst Port: 53
Domain Name System (query)
[Response In: 3020]
Transaction ID: 0x1d29
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
www2.kmutt.ac.th: type A, class IN
Name: www2.kmutt.ac.th
[Name Length: 16]
[Label Count: 4]
Type: A (Host Address) (1)
Class: IN (0x0001)
0000 24 76 7d 14 5e 80 8c 85 90 5b 89 c0 08 00 45 00 $v}.^... .[....E.
0010 00 3e 6b fb 00 00 40 11 b3 1e c0 a8 01 3f cb 90 .>k...@. .....?..
No.: 3019 · Time: 39.786837 · Source: 192.168.1.63 · Destination: 203.144.206.29 · Protocol: DNS · Length: 76 · Info: Standard query 0x1d29 A www2.kmutt.ac.th
Help
Close


Wireshark · Packet 3020 · wireshark_en0_20180121191625_dF72al
Frame 3020: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
Ethernet II, Src: CiscoSpv_14:5e:80 (24:76:7d:14:5e:80), Dst: Apple_5b:89:c0 (8c:85:90:5b:89:c0)
Internet Protocol Version 4, Src: 203.144.206.29, Dst: 192.168.1.63
User Datagram Protocol, Src Port: 53, Dst Port: 63787
Domain Name System (response)
[Request In: 3019]
[Time: 0.025470000 seconds]
Transaction ID: 0x1d29
Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
www2.kmutt.ac.th: type A, class IN
Name: www2.kmutt.ac.th
[Name Length: 16]
[Label Count: 4]
Type: A (Host Address) (1)
Class: IN (0x0001)
Answers
www2.kmutt.ac.th: type A, class IN, addr 202.44.8.55
Name: www2.kmutt.ac.th
Type: A (Host Address) (1)
Class: IN (0x0001)
Time to live: 300
Data length: 4
Address: 202.44.8.55
0000 8c 85 90 5b 89 c0 24 76 7d 14 5e 80 08 00 45 20 ...[..$v }.^...E
0010 00 4e 87 ec 00 00 f5 11 e1 fc cb 90 ce 1d c0 a8 .N..... ....
0020 01 3f 00 35 f9 2b 00 3a a9 83 1d 29 81 80 00 01 .?.5.+.: ...)....
0030 00 01 00 00 00 00 04 77 77 77 32 05 6b 6d 75 74 ....w ww2.kmut
No.: 3020 · Time: 39.812307 · Source: 203.144.206.29 · Destination: 192.168.1.63 · ... Length: 92 · Info: Standard query response 0x1d29 A www2.kmutt.ac.th A 202.44.8.55
Help
Close

Making your Website known to the world

DNS Registra

2 Register 'www.networkutopia.com'

4 Insert
<www.networkutopia.com, NS, dns1.webhost.com>
<dns1.webhost.com, A, 9.9.9.9 >

TLD com server
(Hosting company)

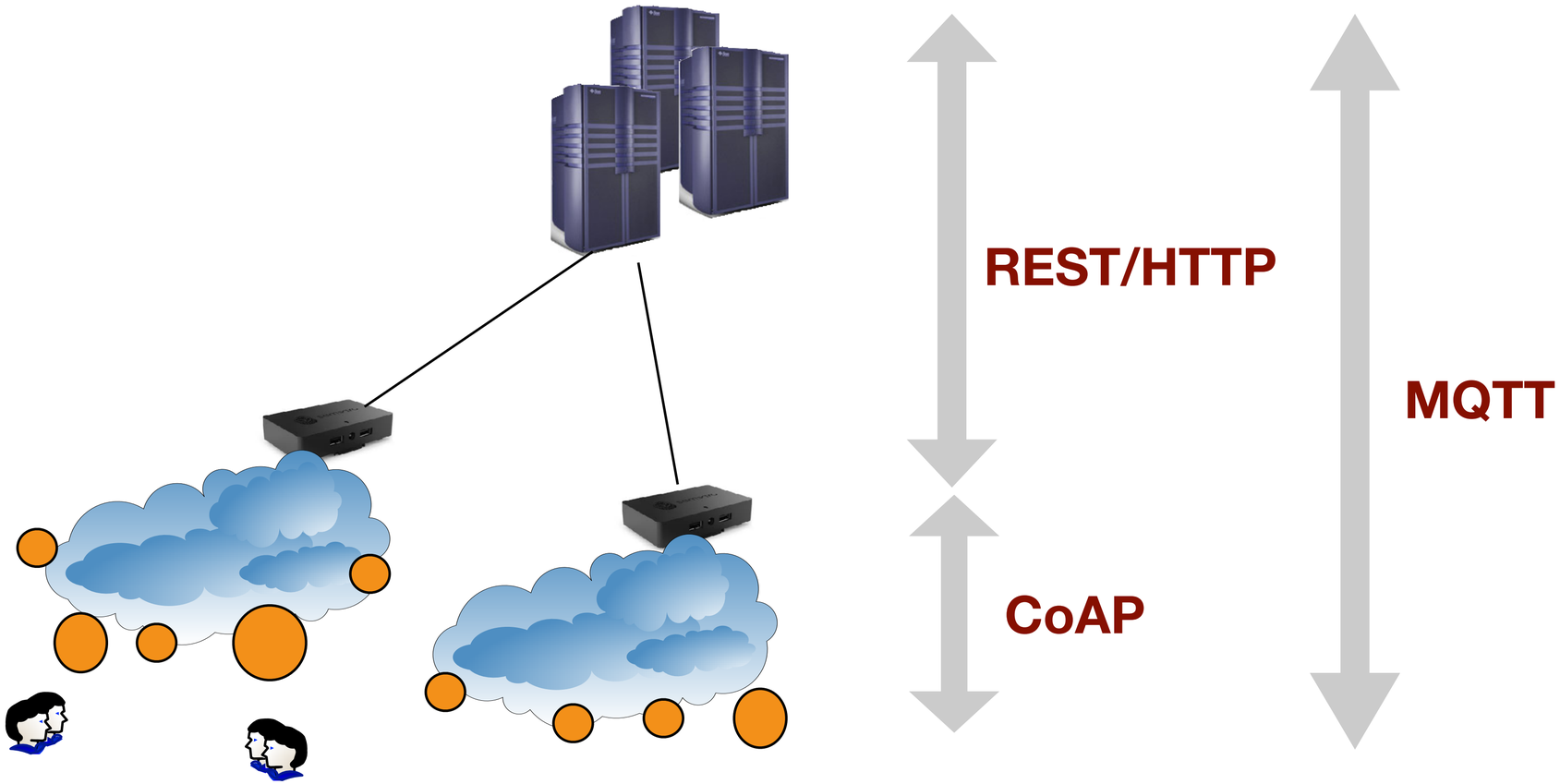
Auth. DNS server
(Hosting company)
dns1.webhost.com
9.9.9.9

You

3 Insert
<www.networkutopia.com, A, 212.44.71.4>

1 Rent www.networkutopia.com
(212.44.71.4) from DNS registra.

IoT Data Transport Protocols



Application data transfer (Request/response, Publish/Subscribe)
Service and device discoveries

RESTful HTTP

Representation

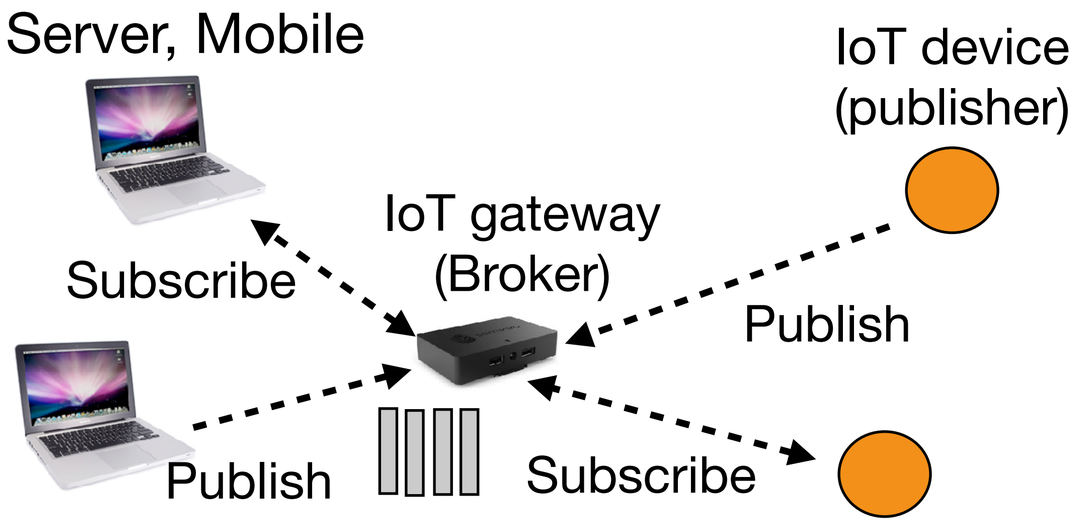
```
GET /sensors/temperature HTTP/1.1
Content-type: application/json
```

```
HTTP/1.1 200 OK
Content-type: application/json

{"sensors":[{"name": "Temperature", "value": 26.1}]}
```

State

Resources accessed by URI
CRUD operations by HTTP methods

MQTT: MQ Telemetry Transport



Typical scenarios

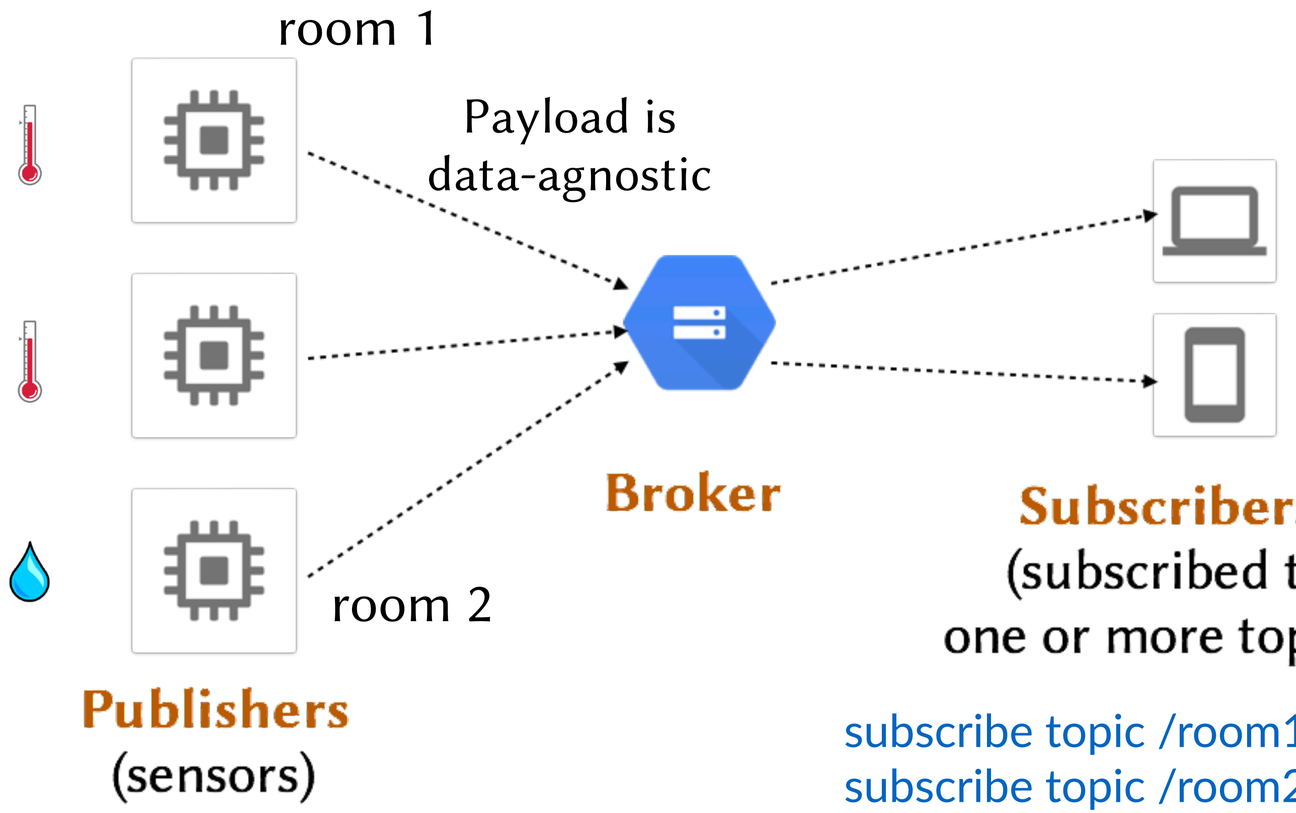
Resource-constrained environment

(Small message size, Stream-based)

Device-to-Server data collection

Remote monitoring for dumb devices with small messages on low bandwidth

MQTT App Model — Telemetry



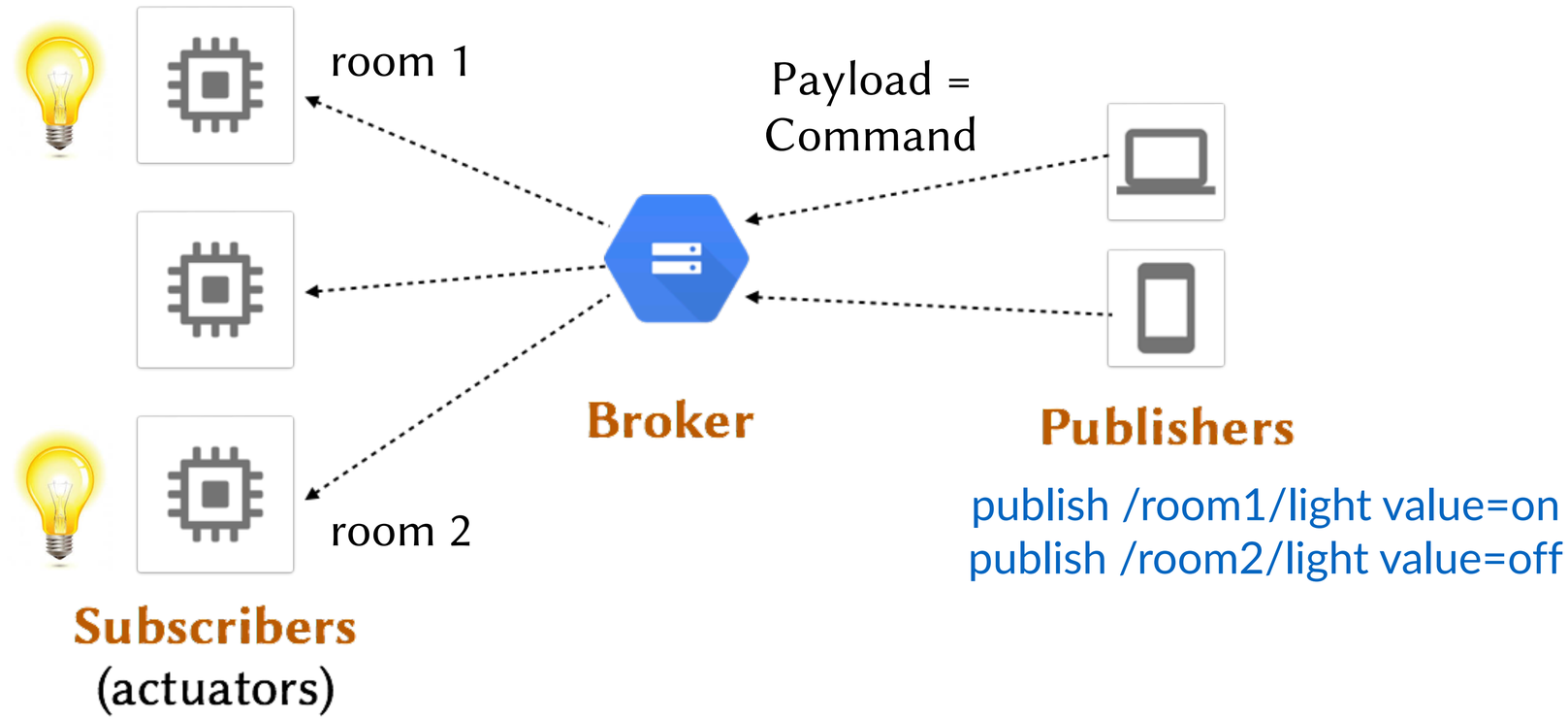
publish /room1/temperature value=25
publish /room2/humidity=40

(Converted to MQTT message format via API)

subscribe topic /room1/temperature
subscribe topic /room2/humidity

(Converted to MQTT message format via API)

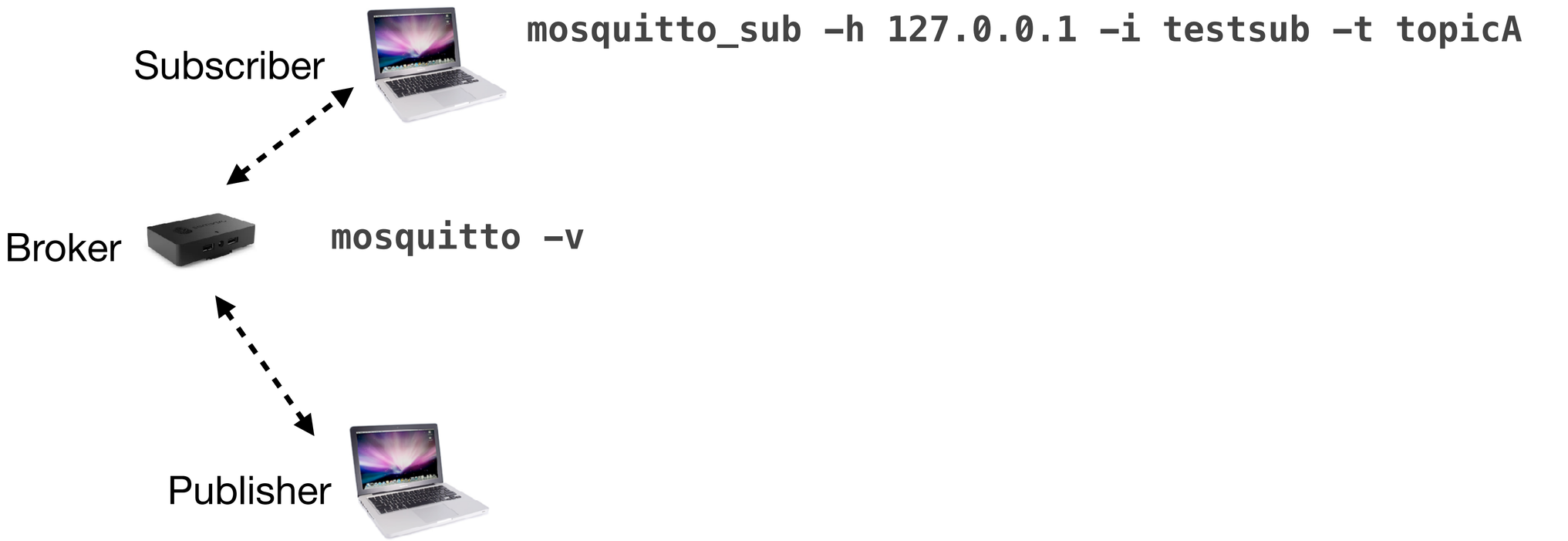
MQTT App Model — Remote Control



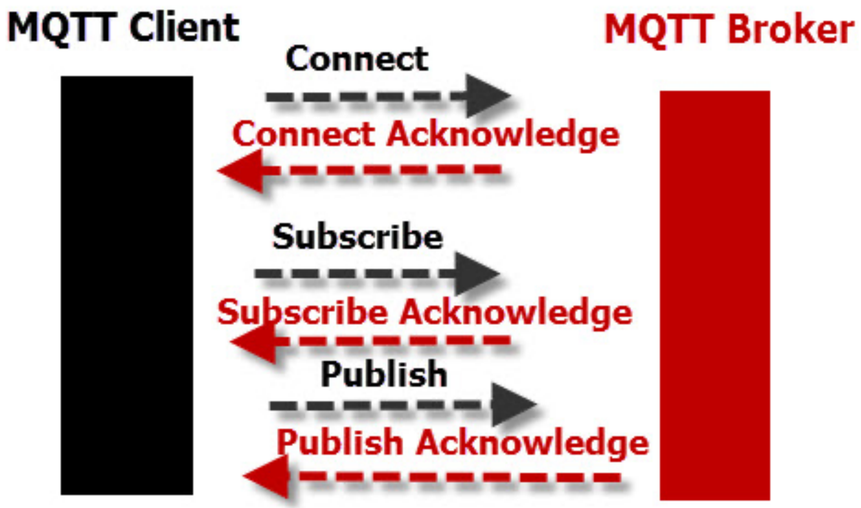
Trying MQTT Publisher and Subscriber

https://mosquitto.org/download/

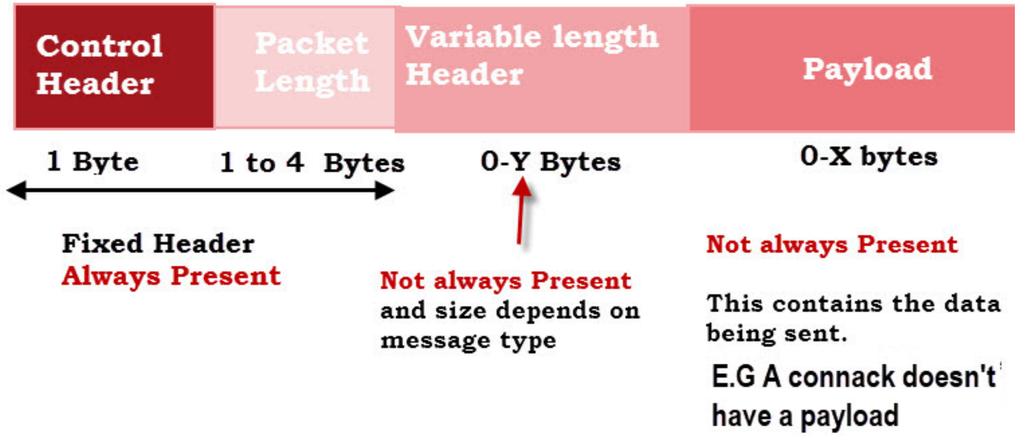
- Text messaging applications based on MQTT.
- Command line arguments are parsed and converted to appropriate fields in MQTT packets.



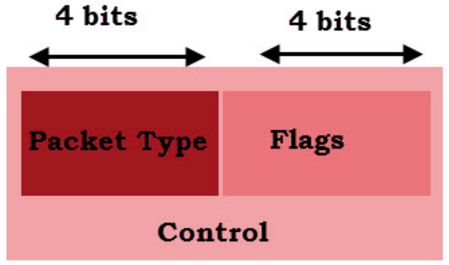
http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/



MQTT Client To Broker Protocol



MQTT Standard Packet Structure



Packet Type Examples:

Connect =0001 =1
Connack=0010 =2

Disconnect=1110=14

Control Packet	Fixed header flags	Bit 3	Bit 2	Bit 1	
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT 3.1.1	DUP[1]	QoS[2]	QoS[2]	R
PUBACK	Reserved	0		0	0
PUBREC	Reserved	0	0	0	0

Duplicate message

Quality of Service
00,01,10 =QOS 0,1,2

Ret

Conclusion

☐ Client-server model (Request/Command/Response) most widely deployed, and recently broker model.

☐ AL protocols to support transport of application data

☐ Header fields mostly text-based for ease of human readability.

☐ Some key AL protocol design principles

- Stateless vs. Stateful
- Text-based vs. Binary syntax
- Centralized vs. Distributed contents
- Flat vs. Hierarchical structure
- UDP vs TCP data transport
- One-to-one vs. Many-to-Many
- Polling vs. Pushing