Class Notes

Exclusions (What will NOT be in sessional exams & final):

The following topics are excluded from sessional exams and final exams **BUT** they are included for other instruments such as quiz. I will try to cover them all until unless we are pressed for time at some point in time.

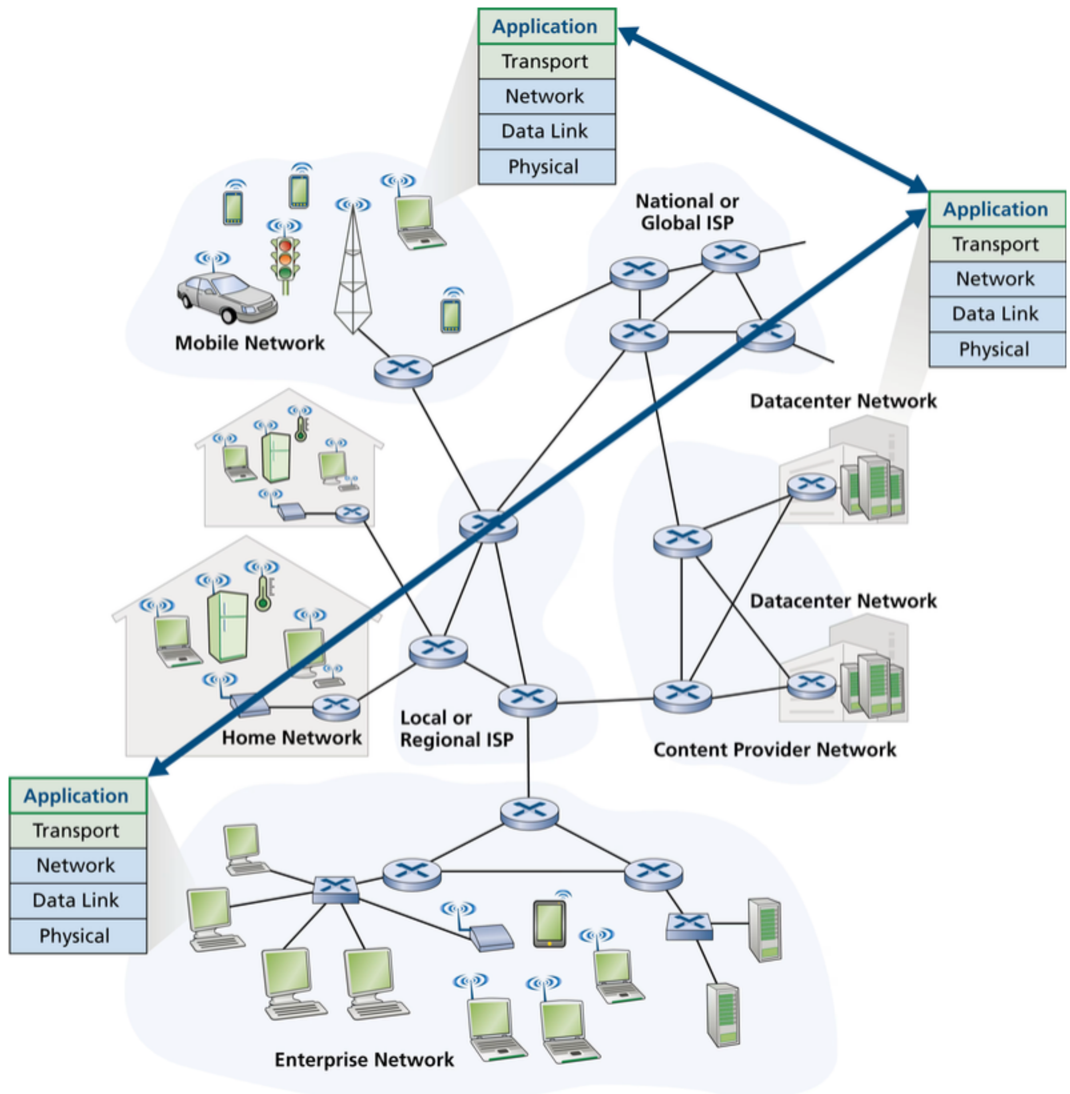
Fall 2025 (Edition 8)		
Chapter	#	**Excluded Topics**
Chapter 1	1.6	Networks Under Attack: Security
	1.7	History
Chapter 2	2.5	Peer-to-Peer File Distribution
	2.6	Video Streaming and CDN
	2.7	Socket Programming: Creating Network Applications
Chapter 3	3.6.1	The Causes and the Cost of Congestion (Congestion Scenarios)
	3.7.2	ECN, Delay based CC
	3.7.3	TCP Fairness
	3.8	Evolution of transport layer functionality (QUIC)
	Others	TCP CUBIC,Macroscopic Description of TCP Reno Throughput, Congested Bottleneck link,
Chapter 4	4.2	What's Inside a Router
Chapter 5	5.4.4	IP-Anycast
	5.4.6	Putting the Pieces togethre
	5.5.2, 5.5.3 & 5.5.4	OpneFlow Protocol, Data and Control plan Interaction, SDN Past and Future: ODL controller, ONOS Controller
	5.7	Network Management and SNMP, NETCONF/YANG
Chapter 6	6.3.4	DOCSIS
	6.4.4	VLAN
	6.5	Link Virtualization: A Network as a Link Layer: MPLS
	6.6.2	Trends in Data Center Networking

Asking questions on Piazza:

(1) Some students are asking me course material related questions via email. I will prefer that students use the Piazza forum. There are two major reasons:
   a. If you have confusions / questions, others likely will have them too. So it will help everyone to learn together.
   b. If some other student knows the answer to your question s/he can contribute as well. Teaching assistants can probably reply faster than myself (especially over the weekend).

Section 2.1 Principles of network applications

Figure 2.1: Communication for a network application takes place between end systems at the application layer



(1) Importantly, you do not need to write software that runs on network-core devices, such as routers or link-layer switches.

(2) This basic design—namely, confining application software to the end systems—as shown in the above picture has facilitated the rapid development and deployment of a vast array of network applications.

Section 2.4: Domain Name System (DNS)--- The Internet's Directory Service

The global Internet is a packet-switched network. To router and deliver IP packets in this network, a network address is required. We call it the **IP address**. As an analogy from the telephone network, you need your intended receiver's phone number.

Example IP addresses:

www.google.com has IPv4 (version 4 of length 32 bits) 172.217.19.196

www.google.com has IPv6 (version 6 of length 128 bits) 2a00:1450:4019:80e::2004 (Each number is a hex digit, means 4 bits called nibble. So there can be 8 such groups separated by ":")

Remembering human-friendly names (like www.google.com) is much easier than remembering IP addresses! DNS provides this mapping.

(Actually, Google has many IPv4 addresses. Different regions might get a different IP address from DNS for load balancing purposes.)

```
C:\Users\abdul>ping www.google.com

Pinging www.google.com [2a00:1450:4019:80e::2004] with 32 bytes of data:
Reply from 2a00:1450:4019:80e::2004: time=88ms
Reply from 2a00:1450:4019:80e::2004: time=60ms
Reply from 2a00:1450:4019:80e::2004: time=76ms
Reply from 2a00:1450:4019:80e::2004: time=82ms

Ping statistics for 2a00:1450:4019:80e::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 60ms, Maximum = 88ms, Average = 76ms
```

```
C:\Users\abdul>ping -4 www.google.com

Pinging www.google.com [172.217.19.196] with 32 bytes of data:
Reply from 172.217.19.196: bytes=32 time=80ms TTL=54
Reply from 172.217.19.196: bytes=32 time=71ms TTL=54
Reply from 172.217.19.196: bytes=32 time=71ms TTL=54
Reply from 172.217.19.196: bytes=32 time=70ms TTL=54

Ping statistics for 172.217.19.196:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 70ms, Maximum = 80ms, Average = 73ms
```

Some organizations try to be creative with their IPv6 addresses:

```
C:\Users\abdul>ping -6 www.facebook.com

Pinging star-mini.c10r.facebook.com [2a03:2880:f167:81:face:b00c:0:25de] with 32 bytes of data:
Reply from 2a03:2880:f167:81:face:b00c:0:25de: time=219ms
Reply from 2a03:2880:f167:81:face:b00c:0:25de: time=80ms
Reply from 2a03:2880:f167:81:face:b00c:0:25de: time=69ms
Reply from 2a03:2880:f167:81:face:b00c:0:25de: time=79ms

Ping statistics for 2a03:2880:f167:81:face:b00c:0:25de:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 69ms, Maximum = 219ms, Average = 111ms
```

Same goes for the email. If you need to send an email to somone@lhr.nu.edu.pk, the client side of Simple Main Transfer Protocol (SMTP) needs to know where is the SMTP mail server that is responsible for receiving "someone"'s emails. Again DNS provides that mapping.

DNS is a critical infrastructure application on the Internet. If DNS is not available to you, you might not visit any website, or could not send any email, and most of your work will be disruptive. DNS's availability is critical. That is why special engineering effort is devoted to the DNS, with thousands of servers, spread across the globe to make sure DNS is always available. Sharding and replication (two key concepts in distributed systems) is heavily used for DNS engineering.

> One machine will not be enough to server DNS queries, that could be trillions of queries per day. For one, one machine is a single-point-of-failure. If that server crashes, the whole DNS goes away.
>
> In the early days of the Arpa net, there was this hosts file that maintained host name to IP address mapping. Different folks will download that file after a while (say every week).

What is DNS:

(1) DNS is a distributed database implemented in a hierarchy of DNS servers
(2) An application layer protocol that allows hosts to query the distributed database

DNS resolution delay:

DNS resolution can add substantial delays for the end user (for example when we type www.nu.edu.pk and the browser needs the IP address of the host nu.edu.pk). We will see later, that DNS responses can be cached locally or "near-by" so that future requests for the same name could be fast.

Other auxiliary services of DNS:

(1) Host aliasing.
   a. Canonical name could be: relay1.west-coast.enterprise.com
   b. Alias could be: www.enterpirse.com
   c. DNS can give the canonical name and IP of the host given the alias.

(2) Mail server aliasing.

a. Canonical name of the mail server could be relay1.west-coast.yahoo.com
b. Alias could be you@yahoo.com
c. Using a special DNS record type (called MX record), both host names and mail server name could be the same (aliases).

(3) Load distribution

a. Large sites (such as cnn.com or google.com) have excessive user load (for example billions of search queries coming to the web server at Google.) Such sites therefore have many replicated services on many servers, each server having a different IP address. DNS might rotate available IPs for a name for different queries, say in round-robin fashion.
b. At times DNS server return multiple IP addresses to the client. Other times, server picks one IP of its choice and provides that one. DNS might provide an IP address that is "near" to the client. What constitutes "near" varies. It could be geographical distance or network distance (say in terms of number of hops from the client to the service.)

DNS Data sharded into zones:

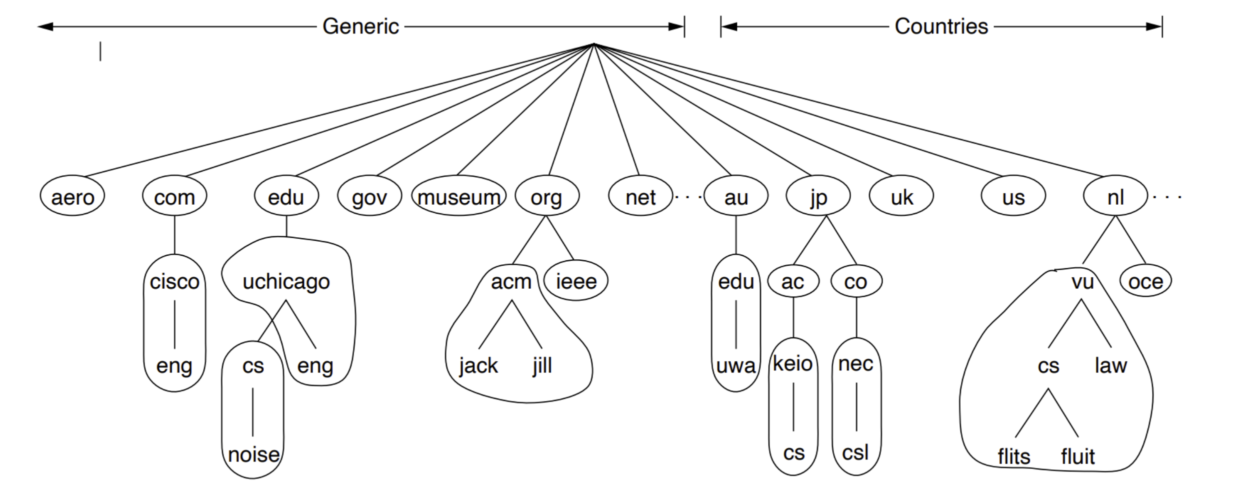


Figure 7-6. Part of the DNS name space divided into zones (which are circled).

Tuvalu is located in south Pacific Ocean, near Australia.

Tuvalu

اردو میں | In English

tv is the Internet country code top-level domain (ccTLD) for Tuvalu. Except for reserved names like com.tv, net.tv, org.tv and others, anyone may register second-level domains under . tv.

Wikipedia
https://en.wikipedia.org › wiki

.tv - Wikipedia

The Top 10 Most Expensive Domains Ever Reported

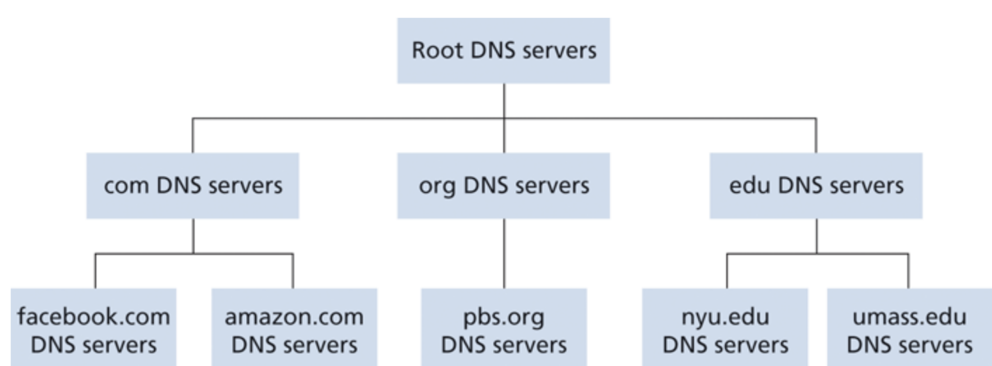
When it comes to memorable domain names that are short, to the point and can describe their intent in a matter of seconds, businesses and large corporations are willing to spend big bucks to secure their space on the Internet.

Here are some of the most valuable domain names ever reported:

1. Cars.com – $872 million
2. CarINsurance.com – $49.7 million
3. Insurance.com – $35.6 million
4. VacationRentals.com – $35 million
5. PrivateJet.com – $30.18 million
6. Voice.com – $30 million
7. Internet.com – $18 million
8. 360.com – $17 million
9. Insure.com – $16 million
10. Fund.com – $9.95 million

Source: https://www.name.com/blog/the-top-10-most-expensive-domains-ever-sold

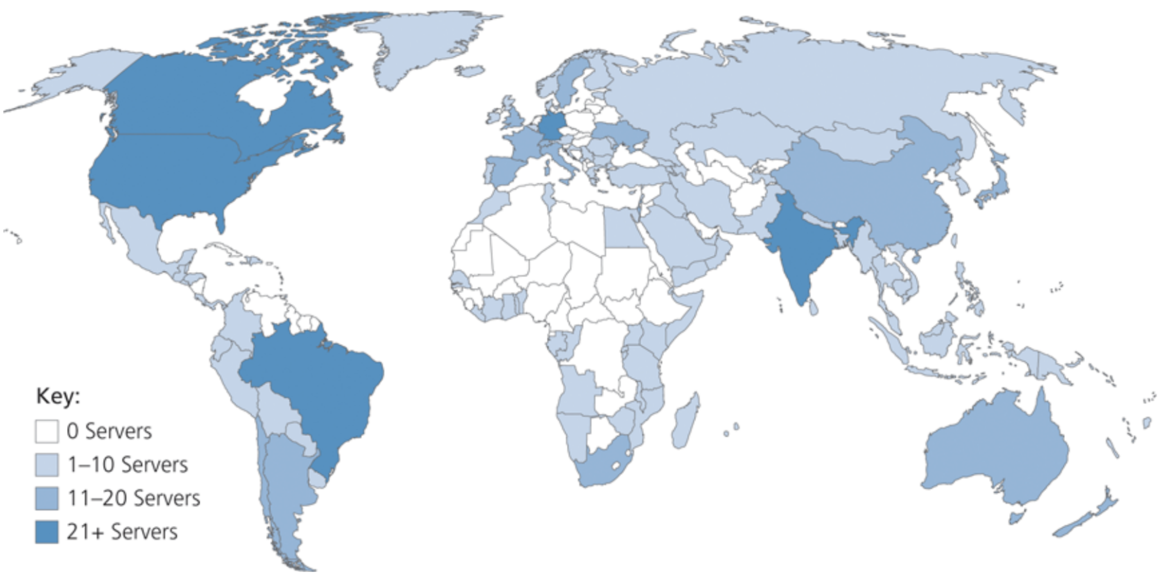
FIGURE 2.17 PORTION OF THE HIERARCHY OF DNS SERVERS



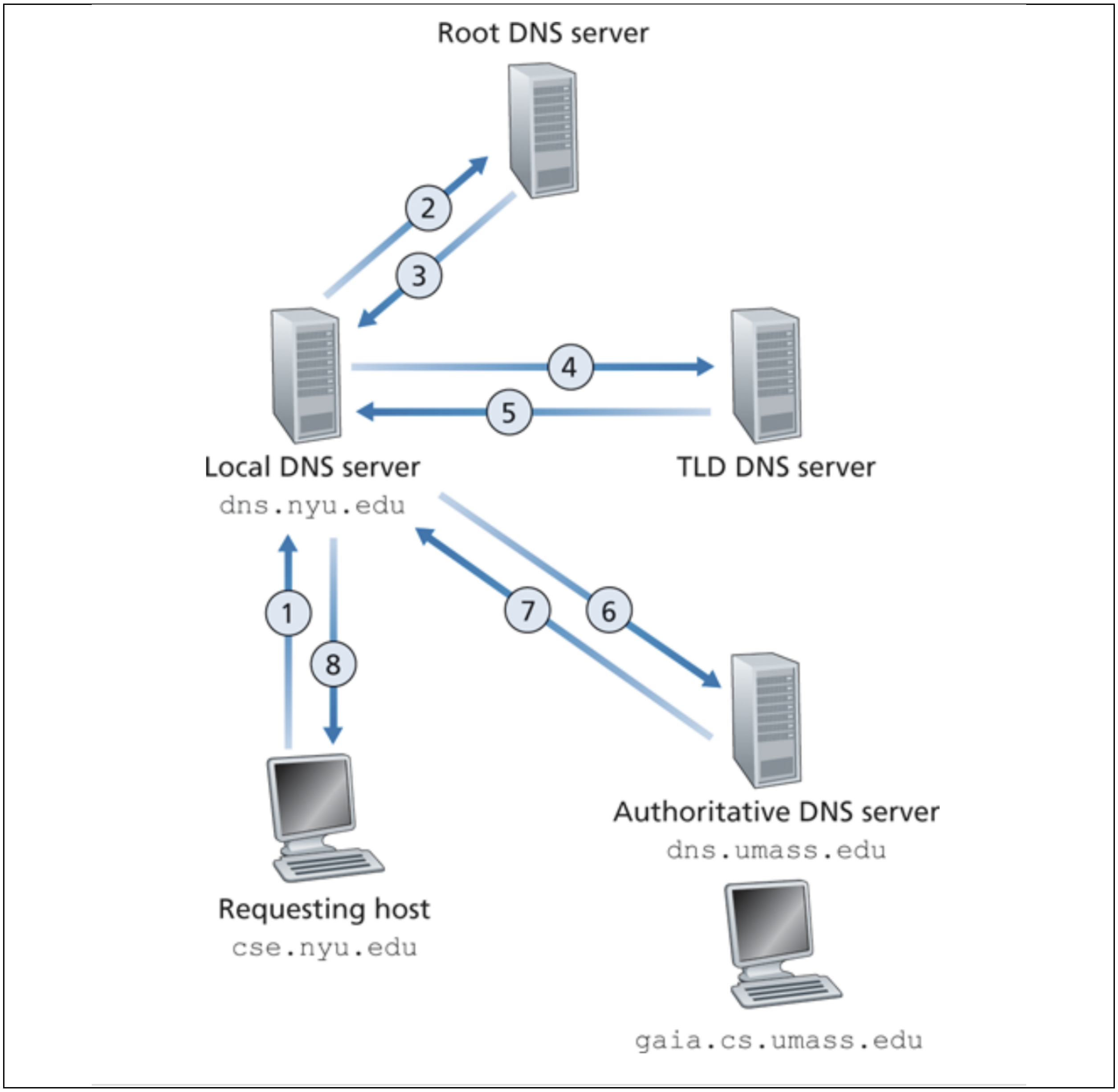
13 replicated root servers across the globe:

Logically 13, but physically many.

FIGURE 2.18 DNS ROOT SERVERS IN 2020



Typical DNS query resolution:



Two helpful program you should learn about:

(1) nslookup
(2) dig

Examples:

```
[aqadeer@AQWM ~]$ nslookup www.nu.edu.pk
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
www.nu.edu.pk   canonical name = nu.edu.pk.
Name:   nu.edu.pk
Address: 203.124.44.78

[aqadeer@AQWM ~]$ nslookup www.google.com
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.19.196
Name:   www.google.com
Address: 2a00:1450:4019:80e::2004
```

```
[aqadeer@AQWM ~]$ dig www.nu.edu.pk

; <<>> DiG 9.18.26 <<>> www.nu.edu.pk
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42741
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.nu.edu.pk.                 IN      A

;; ANSWER SECTION:
www.nu.edu.pk.          9230    IN      CNAME   nu.edu.pk.
nu.edu.pk.              11014   IN      A       203.124.44.78

;; Query time: 6 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Mon Sep 02 13:01:16 PKT 2024
;; MSG SIZE  rcvd: 81
```

```
[aqadeer@AQWM ~]$ dig -t AAAA www.google.com

; <<>> DiG 9.18.26 <<>> -t AAAA www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17344
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.google.com.                        IN      AAAA

;; ANSWER SECTION:
www.google.com.         74      IN      AAAA    2a00:1450:4019:80b::2004

;; Query time: 6 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Mon Sep 02 13:06:18 PKT 2024
;; MSG SIZE  rcvd: 71
```

Lets do all the steps ourselves:

Step 1: List all the root servers

```
[aqadeer@AQWM ~]$ dig

; <<>> DiG 9.18.26 <<>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53399
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1220
; COOKIE: e9dd36b3e4fc123d1c42836866d5724be1e176bab9243c22 (good)
;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       50482   IN      NS      l.root-servers.net.
.                       50482   IN      NS      a.root-servers.net.
.                       50482   IN      NS      b.root-servers.net.
.                       50482   IN      NS      m.root-servers.net.
.                       50482   IN      NS      g.root-servers.net.
.                       50482   IN      NS      k.root-servers.net.
.                       50482   IN      NS      e.root-servers.net.
.                       50482   IN      NS      c.root-servers.net.
.                       50482   IN      NS      i.root-servers.net.
.                       50482   IN      NS      d.root-servers.net.
.                       50482   IN      NS      h.root-servers.net.
.                       50482   IN      NS      j.root-servers.net.
.                       50482   IN      NS      f.root-servers.net.

;; Query time: 69 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Mon Sep 02 13:07:39 PKT 2024
;; MSG SIZE  rcvd: 267
```

Step 2: Lets pick any one of them to move forward in resolution.

```
[aqadeer@AQWM ~]$ dig www.nu.edu.pk @b.root-servers.net

; <<>> DiG 9.18.26 <<>> www.nu.edu.pk @b.root-servers.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34243
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 7
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.nu.edu.pk.                 IN      A

;; AUTHORITY SECTION:
pk.                     172800  IN      NS      root-e.pknic.pk.
pk.                     172800  IN      NS      root-s.pknic.pk.
pk.                     172800  IN      NS      root-c1.pknic.pk.
pk.                     172800  IN      NS      root-c2.pknic.pk.

;; ADDITIONAL SECTION:
root-e.pknic.pk.        172800  IN      A       107.6.178.178
root-s.pknic.pk.        172800  IN      A       119.81.34.90
root-c1.pknic.pk.       172800  IN      A       185.159.197.160
root-c1.pknic.pk.       172800  IN      AAAA    2620:10a:80aa::160
root-c2.pknic.pk.       172800  IN      A       185.159.198.160
root-c2.pknic.pk.       172800  IN      AAAA    2620:10a:80ab::160

;; Query time: 213 msec
;; SERVER: 2801:1b8:10::b#53(b.root-servers.net) (UDP)
;; WHEN: Mon Sep 02 13:16:48 PKT 2024
;; MSG SIZE  rcvd: 254
```

Step 3:

```
[aqadeer@AQWM ~]$ dig www.nu.edu.pk @root-c1.pknic.pk.

; <<>> DiG 9.18.26 <<>> www.nu.edu.pk @root-c1.pknic.pk.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38778
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.nu.edu.pk.                 IN      A

;; AUTHORITY SECTION:
nu.edu.PK.              86400   IN      NS      n1.comsats.net.pk.
nu.edu.PK.              86400   IN      NS      n2.comsats.net.pk.

;; ADDITIONAL SECTION:
n2.comsats.net.PK.      86400   IN      A       203.124.45.92
n1.comsats.net.PK.      86400   IN      A       210.56.11.130

;; Query time: 180 msec
;; SERVER: 2620:10a:80aa::160#53(root-c1.pknic.pk.) (UDP)
;; WHEN: Mon Sep 02 13:19:35 PKT 2024
;; MSG SIZE  rcvd: 151
```

Step 4:

```
[aqadeer@AQWM ~]$ dig www.nu.edu.pk @n1.comsats.net.pk.

; <<>> DiG 9.18.26 <<>> www.nu.edu.pk @n1.comsats.net.pk.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61437
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d8a4a46233579285aaa2587566d57498a382807b126c310d (good)
;; QUESTION SECTION:
;www.nu.edu.pk.                 IN      A

;; ANSWER SECTION:
www.nu.edu.pk.          14400   IN      CNAME   nu.edu.pk.
nu.edu.pk.              14400   IN      A       203.124.44.78

;; AUTHORITY SECTION:
nu.edu.pk.              86400   IN      NS      n2.comsats.net.pk.
nu.edu.pk.              86400   IN      NS      n1.comsats.net.pk.

;; Query time: 55 msec
;; SERVER: 210.56.11.130#53(n1.comsats.net.pk.) (UDP)
;; WHEN: Mon Sep 02 13:20:43 PKT 2024
;; MSG SIZE  rcvd: 146
```

Similar for the email:

```
[aqadeer@AQWM ~]$ dig lhr.nu.edu.pk MX

; <<>> DiG 9.18.26 <<>> lhr.nu.edu.pk MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42030
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1220
; COOKIE: 497453a70c42c56dd89b7b4466d57749af7a0748547db1bb (good)
;; QUESTION SECTION:
;lhr.nu.edu.pk.                 IN      MX

;; ANSWER SECTION:
lhr.nu.edu.pk.          3484    IN      MX      1 ASPMX.L.GOOGLE.COM.
lhr.nu.edu.pk.          3484    IN      MX      10 ALT4.ASPMX.L.GOOGLE.COM.
lhr.nu.edu.pk.          3484    IN      MX      10 ALT3.ASPMX.L.GOOGLE.COM.
lhr.nu.edu.pk.          3484    IN      MX      5 ALT1.ASPMX.L.GOOGLE.COM.
lhr.nu.edu.pk.          3484    IN      MX      5 ALT2.ASPMX.L.GOOGLE.COM.

;; Query time: 48 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Mon Sep 02 13:28:57 PKT 2024
;; MSG SIZE  rcvd: 188
```

DNS record types:

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

Figure 7-4. The principal DNS resource record types.

Source: Computer Networks by Tanenbaum et al.

Top Level Domains:

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

Figure 7-2. The original generic TLDs, as of 2010. As of 2020, there are more than 1,200 gTLDs.

Source: Computer networks by Tanenbaum

ICANN and Others:

From Wikipedia: “The **Internet Corporation for Assigned Names and Numbers** (**ICANN** /ˈaɪkæn/ *EYE-kan*) is a global multistakeholder group and nonprofit organization headquartered in the United States responsible for coordinating the maintenance and procedures of several databases related to the namespaces and numerical spaces of the Internet, ensuring the Internet's stable and secure operation.”

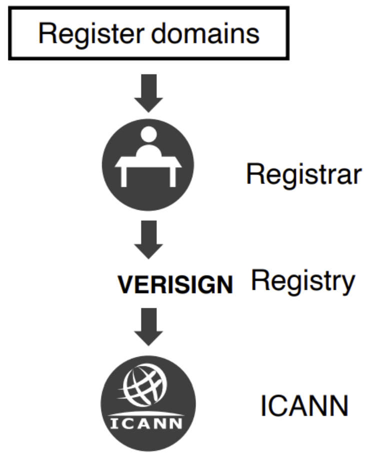


Figure 7-3. The relationship between registries and registrars.

DNS packet format:

1. **Header**: Contains metadata about the DNS query or response, including flags, operation codes, and counts of various sections.

2. **Question**: Specifies the domain name being queried, along with the type of query (e.g., A, MX) and the class (usually IN for internet).

3. **Answer**: Holds the resource records (RRs) that directly answer the question, containing the resolved data like IP addresses or mail server names.

4. **Authority**: Lists the resource records pointing to authoritative name servers for the domain, used to help resolve the query further if needed.

5. **Additional**: Provides extra information that may be useful for resolving the query, such as additional RRs not directly answering the question but related to the authority section.

And further details in the header:

DNS Header

<table>
<tr><th colspan="2">Offsets</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="16" rowspan="2">Transaction ID</td><td colspan="16">Flags</td></tr>
<tr><td>QR</td><td colspan="4">OPCODE</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td colspan="3">Z</td><td colspan="4">RCODE</td></tr>
<tr><td>4</td><td>32</td><td colspan="16">Number of questions</td><td colspan="16">Number of answers</td></tr>
<tr><td>8</td><td>64</td><td colspan="16">Number of authority RRs</td><td colspan="16">Number of additional RRs</td></tr>
</table>

Source: Wikipedia

Field	Description
ID	A unique identifier for matching responses with queries.
QR (Query/Response)	Indicates if the packet is a query (0) or a response (1).
Opcode	Specifies the type of query (e.g., standard, inverse, or server status).
AA (Authoritative Answer)	Indicates if the response is from an authoritative name server.
TC (Truncation)	Indicates if the message was truncated due to size limits.
RD (Recursion Desired)	Requests that the server perform recursion to resolve the query.
RA (Recursion Available)	Indicates if the server supports recursion in the response.
Z	Reserved for future use, must be set to 0.
AD (Authenticated Data)	Indicates if the data has been authenticated by the server (DNSSEC).
CD (Checking Disabled)	Instructs the server to disable DNSSEC validation.
RCODE (Response Code)	Indicates the status of the response (e.g., no error, format error).
QDCOUNT (Question Count)	Number of entries in the Question section.
ANCOUNT (Answer Count)	Number of resource records in the Answer section.
NSCOUNT (Authority Count)	Number of resource records in the Authority section.
ARCOUNT (Additional Count)	Number of resource records in the Additional section.

Security aspects:

Any compromise to the DNS resolution system can badly impact security of the user. Example: If yourbank.com could be mapped to a malicious server, that impersonate your bank's usual look and feel, malicious actors might be able to deceive to give up your login and password. Which then they can use on the real bank website to transfer funds!

Homework:

(1) Do some DNS queryies and capture their request and response packets using Wireshark. Match what you see in Wireshark to the output in nslookup/dig.

(2) Read section 2.4 on DNS from the book. There are many details we didn't get to in the class. For example recursive and iterative query resolution.