ELEC-4120
Tutorial - 5

Manohar Kuse
mpkuse@ust.hk
http://ihome.ust.hk/~mpkuse

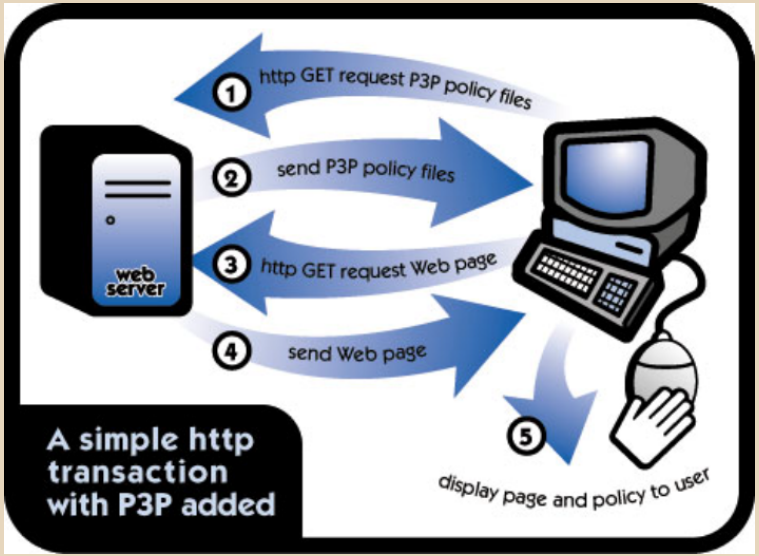
Topics to Cover

1. Web Caches
2. DNS

Part - I

Accessing Web Pages on the Internet

HTTP Protocol - Hyper Text Transfer Protocol



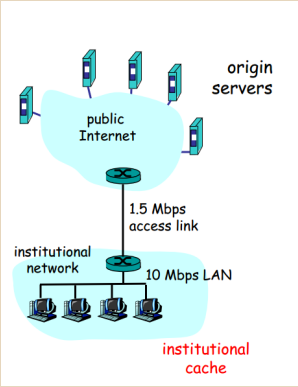
Self Reading : http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Solved Example - 1

Given:

- Avg Packet Size : 1500 Bits
- 1000 requests per sec
- If I do a ping from institute router to a web-server, I get an RTT of 1 sec
- Ignore any LAN delays
- Transmission time (delay) from institute router to the internet depends on transmission time of 1 packet (**h**) and number of messages per sec (**L**)

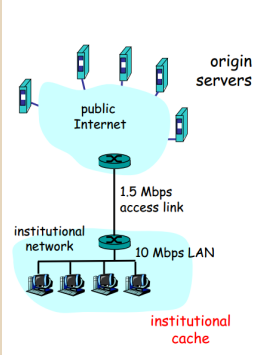
Transmission delay = **h** / (1 - **Lh**)



Solved Example - 1 (Cont…)

Calculate :

How much time does it take to access a web-page?



How to think?

-A- We already know, it takes 1 sec to get the webpage once it is passed the institute router.

-B- It takes some time to put the data on the link (transmission time). Need to calculate this

-- Ans : A+B

Solved Example - 1 (Cont…)

Transmission delay = **h** / (1 - **Lh**)

h => Transmission time for 1 packet
L => packets per sec, which is 1000 in this case

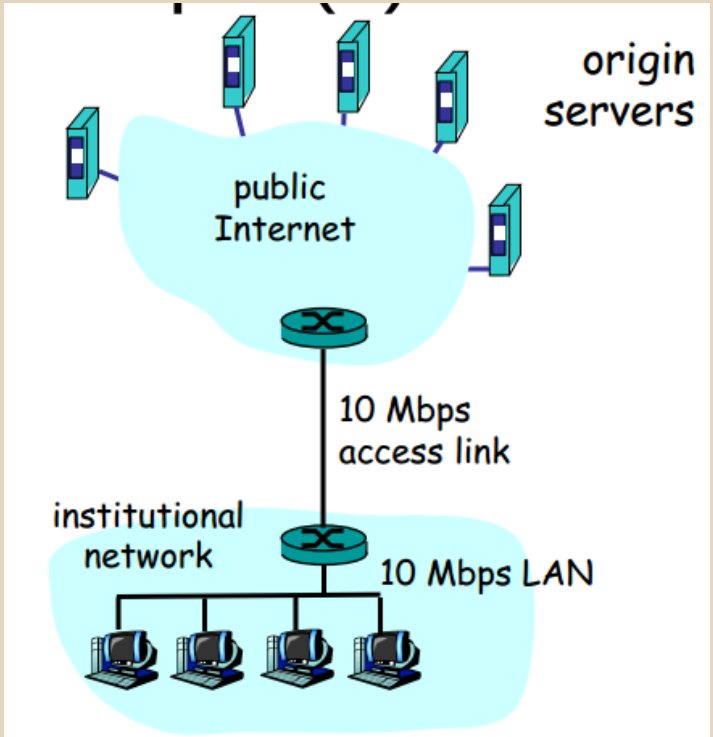
h = 1500 bits / 1.5 Mbps
= 10^(-3)

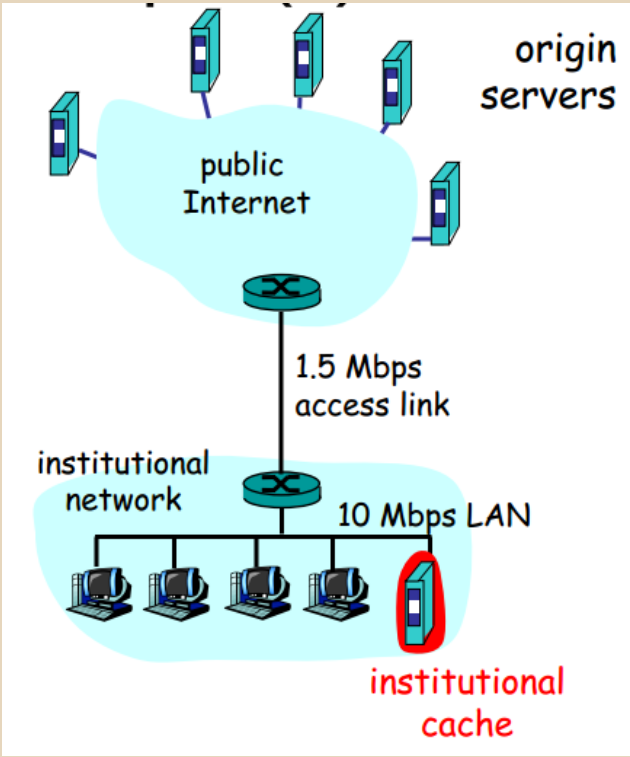
So, transmission delay = infinite

This implies, the packet reaches its destination in a very long time.

How do we deal with this

Web Cache


origin
servers
public
Internet
10 Mbps
access link
institutional
network
10 Mbps LAN


origin
servers
public
Internet
1.5 Mbps
access link
institutional
network
10 Mbps LAN
institutional
cache

Solved Example - 2

Important Terms to understand

- Hit rate (ratio of requests served by the cache)
- Miss rate (ratio of request for which data has to be actually fetched from remote web-server)

Given

Hit rate is 5%.
Other conditions same as problem-1

What is the average time to retrieve a web-page?

Solved Example -2 (Cont …)

How to think

- There are 2 cases here,
    1. When request is served from cache
    2. Data is actually fetched

What is time required in case 1?

What is time required in case 2?

Remember that, we are interested in finding the average time to retrieve the web-page

Solved Example - 2 (Cont …)

Case 1: Request served from cache

Time to fetch : 0 sec

Case 2: Data is fetched from remote server

RTT (from inst. router) + Transmission delay

= 1 sec + **h** / (1 - **Lh**)

recall h=10^{-3} (from problem-1)

= 1 + { 10^-3 / (1 - .95*1000 * 10^-3) }

= 1 + 10^-3 / 0.05

= 1.02 sec

Solved Example - 2 (Cont …)

(Using total conditional probability rule)

Average time = .95 * 1.02 + .05 * 0

= .96 sec

Part - II

Domain Name System (DNS)

- Routers only understand IP address (172.22.16.20)
- Humans find it difficult
- Instead we human like something like ust.hk or google.com or hsbc.com.hk or booking.hkexpress.com

Thus DNS, (sometimes also referred as ‘name server’ (NS)

DNS is a phonebook for translating domain names (example.com) to their IP addresses

Ref:
http://en.wikipedia.org/wiki/Domain_Name_System

DNS is like a lookup table

Will have billions of entries

Domain Name	IP Address	Type
google.com	202.40.221.16	A
mail.google.com	202.40.220.112	MX
ust.hk	143.89.13.187	A
www.ust.hk	143.89.13.187	CNAME
lmes.ust.hk	143.89.15.205	A
ns3.yahoo.com	203.84.221.53	NS

‘Type’ - DNS Entry Types

1. A → Ip address of a host server
2. MX → Ip address of a mail server
3. NS → Ip address of another name-server
4. CNAME → Canonical (Alias) name

Tools

1. ping
2. traceroute
3. nslookup, nslookup -query=mx yahoo.com

How to store all this data?

Too much data

Not very secure if one institute holds all this data

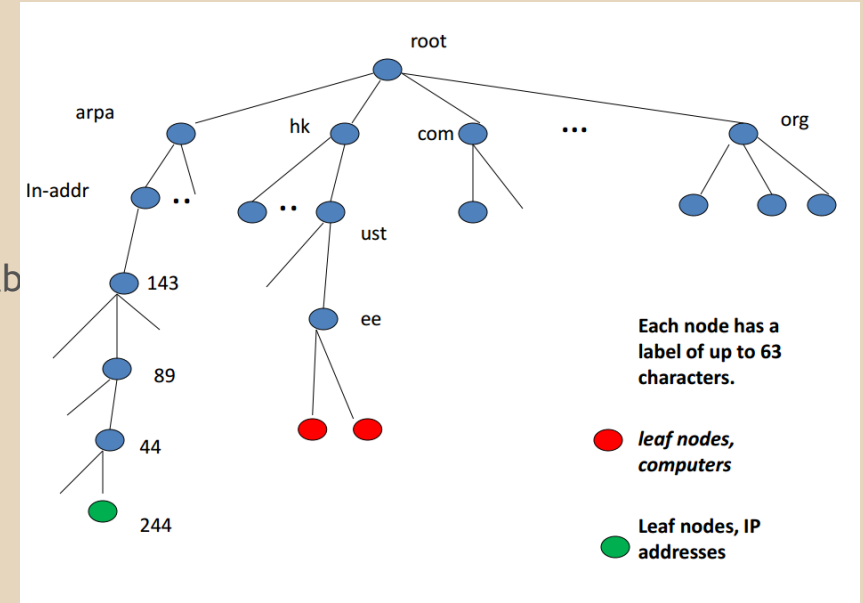
$\Rightarrow$ Distributed database

How DNS database is distributed

ee.ust.hk

Root NS ⇒ .hk NS ⇒ ust.hk NS ⇒

Rule :
Store only your own IP-Domain name translation tab

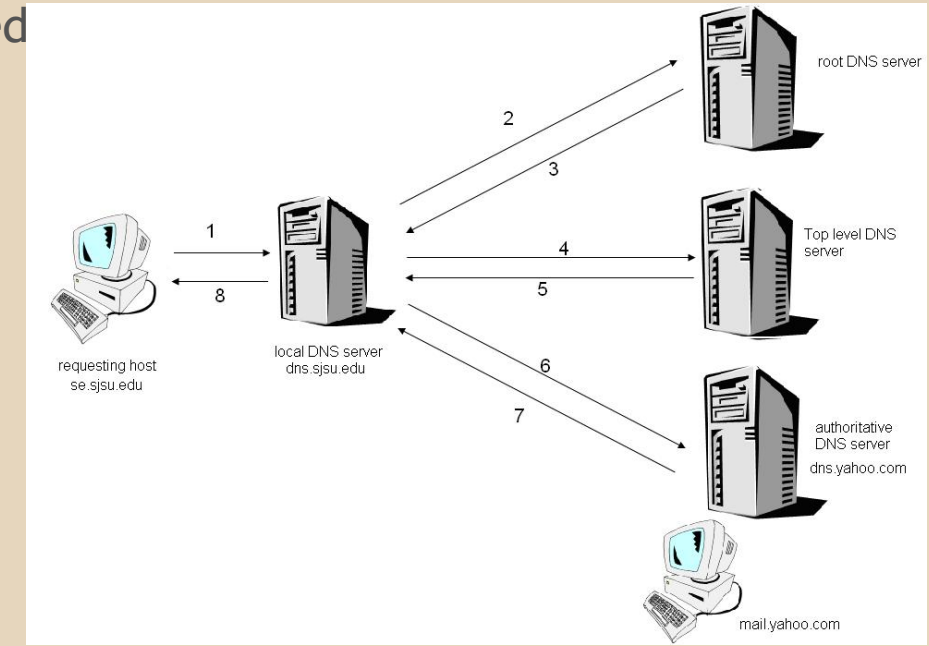


References :
http://en.wikipedia.org/wiki/Domain_Name_System
http://en.wikipedia.org/wiki/Root_name_server
http://en.wikipedia.org/wiki/Top-level_domain
http://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

How all this work together

I need to find :

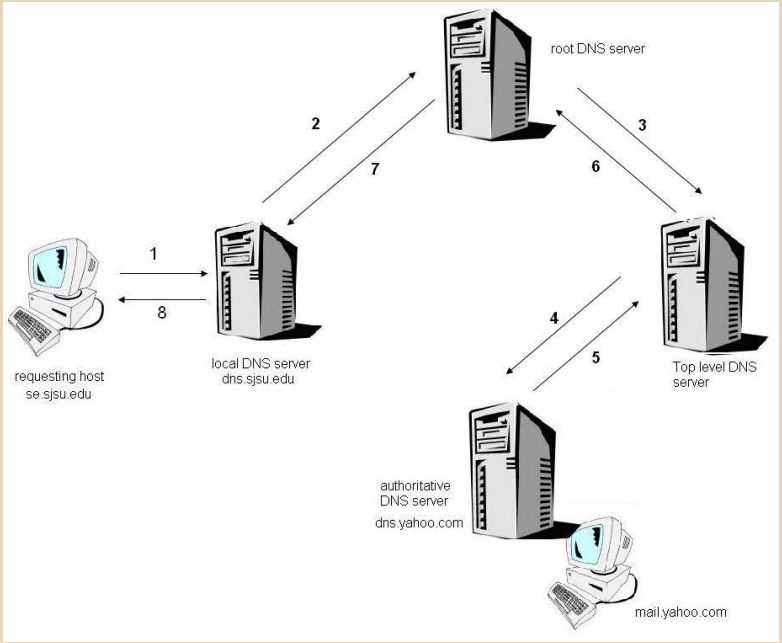
- IP address for `mail.yahoo.com`
- I am `se.sjsu.ed



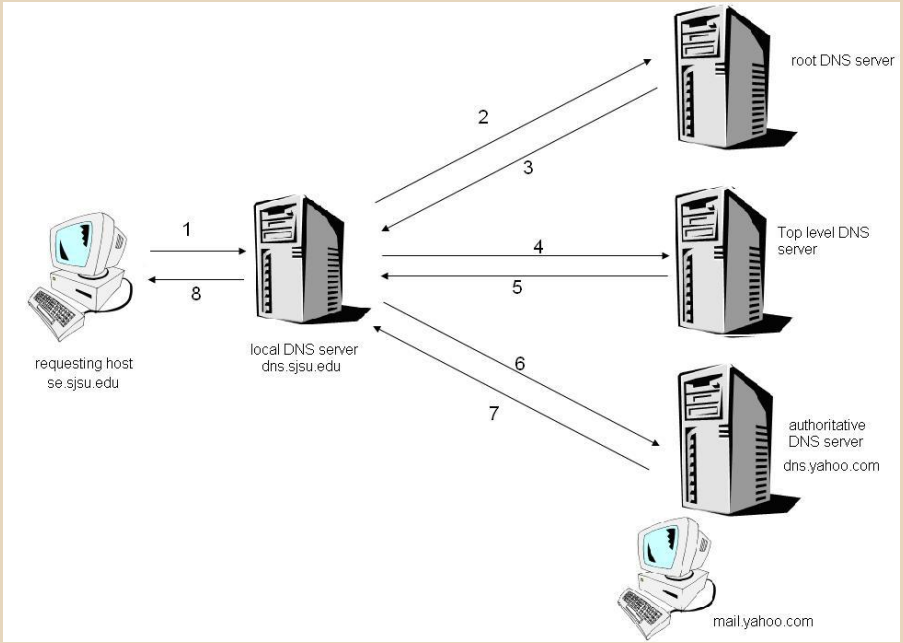
Actually there are 2 ways

1. Iterative DNS querying (just discussed)
2. Recursive DNS querying

Actually there are 2 ways



Recursive DNS Querying



Iterative DNS Querying