Computer Networks and Communication

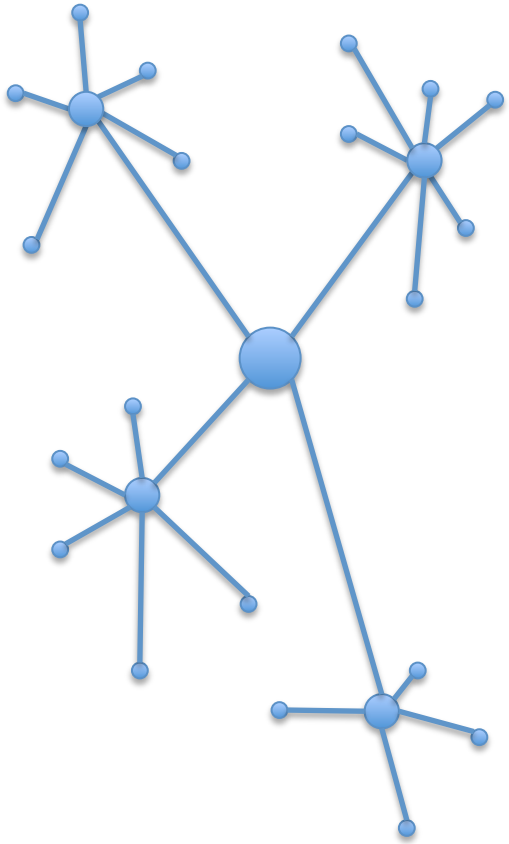
Lecture 3

The Internet,

Application Layer, DNS

History of The Internet

- It all began in the late 1950s
- The Department of Defense (DoD) wanted to make sure that their networks could survive nuclear war
- Their systems were running on hierarchical telephone networks which provide only limited redundancy



ARPANET

- In 1960, **Paul Baran** at RAND Corp designed a packet-switched network for the DoD but did not implement it
- Other researchers such as **Leonard Kleinrock** at MIT and **Donald Davies** at National Physical Lab were also exploring the idea of packet switching
- The first *actual* packet-switching network is done by **Lawrence Roberts at** DoD's Advanced Research Projects Agency (ARPA)
- This network is called the **ARPANET**
- The first packet-switching device is called **Interface Message Processors (IMP)**

Birth of TCP/IP and DNS

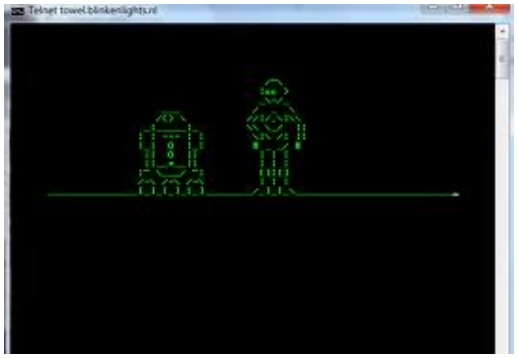
- ARPANET was growing rapidly serving numerous organizations and universities
- Two problems were faced:
  - Communication between internetworks was difficult using ARPANET original protocols
    - **TCP/IP** was introduced
  - As the ARPANET grew, it was also getting harder to find hosts using only IP addresses
    - **Domain Name System (DNS)** was invented

NSFNET

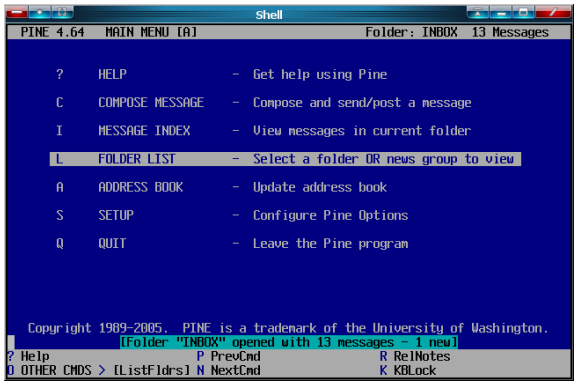
- ARPANET was accessible only for the organizations who have contracts with the DoD
- Therefore, National Science Foundation (NSF) funded another packet-switched network called NSFNET to connect their supercomputers
- Later ARPANET and NSFNET are connected with TCP/IP as standard protocols
- The backbones and connections are later privatized and no longer supported by the funding original organizations

(Classic) Internet Applications

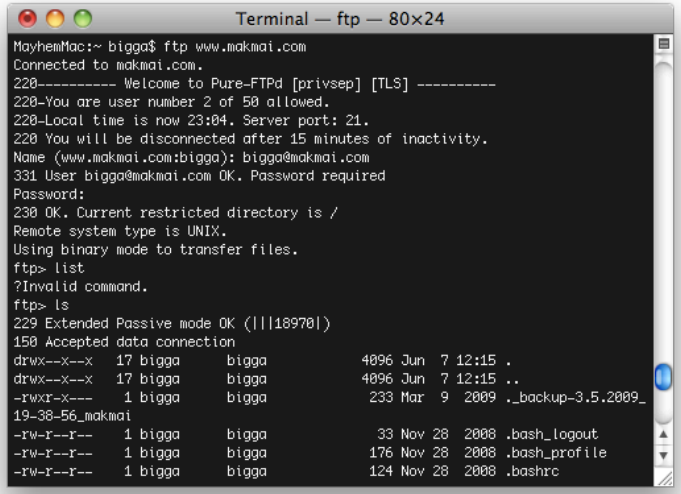
- Almost all of them are text-based
- Email: SMTP, POP and IMAP
- News: NNTP
- Remote login: Telnet, rlogin and SSH
- File transfer: FTP and sFTP



Telnet: Showing Star Wars



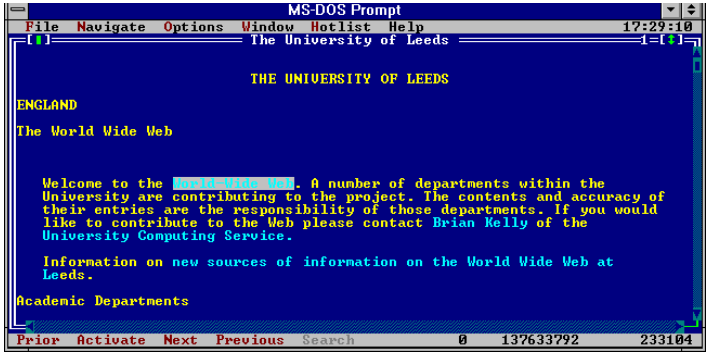
Pine: Text-based email client



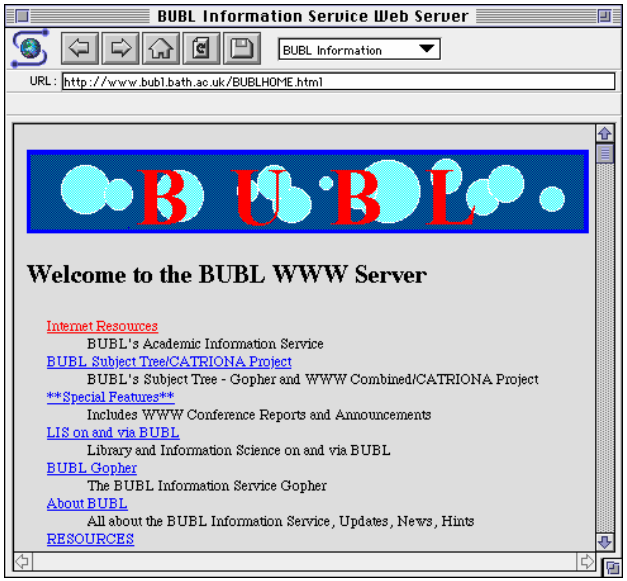
FTP

The Web

- The Web or World-Wide-Web was created by Sir Tim Berners-Lee at CERN in 1985
  - He defines HTML and HTTP
  - Developed on the NeXTSTEP
- First graphical web browser, Mosaic was created in 1991
- You know the rest



Lynx: Text-base browser



Mosaic

Standard Organizations

- Internet Engineering Task Force (IETF)
  - RFCs (Request for Comments)
  - Mostly Internet protocol standards
- Institute of Electrical and Electronics Engineers (IEEE)
  - LANs and Wireless LANs
  - Publish most respectable scientific journals and articles
- International Telecommunication Union (ITU)
  - Recommendations

Application Layer Outline

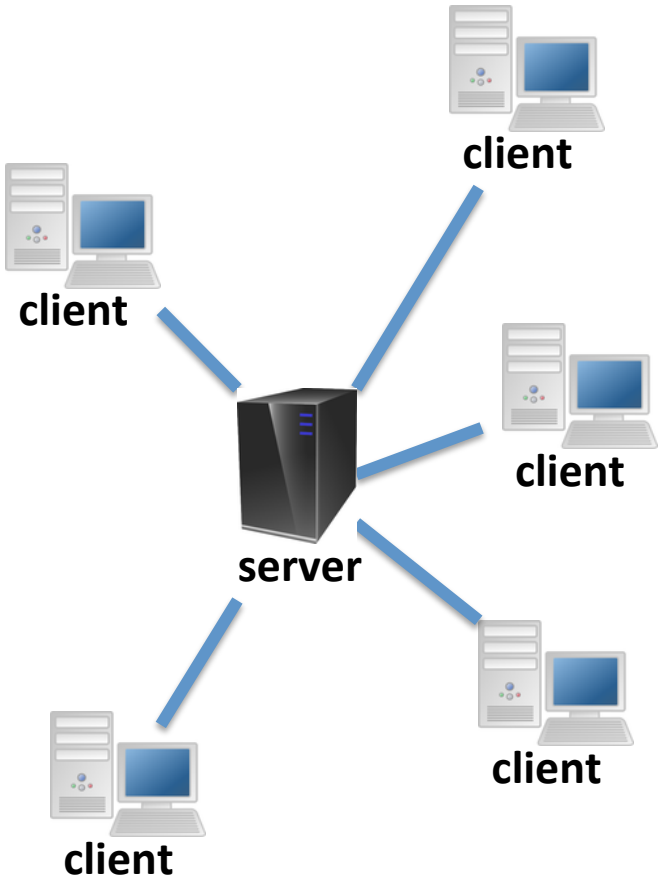
- Application / Service Architectures
- Socket and port numbers
- Connection-oriented and connectionless services
- DNS
- Email
- The Web (HTTP)

Application Architectures

- **Network applications** are applications that run on different hosts and want to communicate to each other
- Application architectures can be categorized into three groups:
  - **Client-server**: Many **clients** connect to one **server**
  - **Peer-to-peer**: No server. A number of **peers** are connected to each other
  - **Hybrid**: Mixture of both

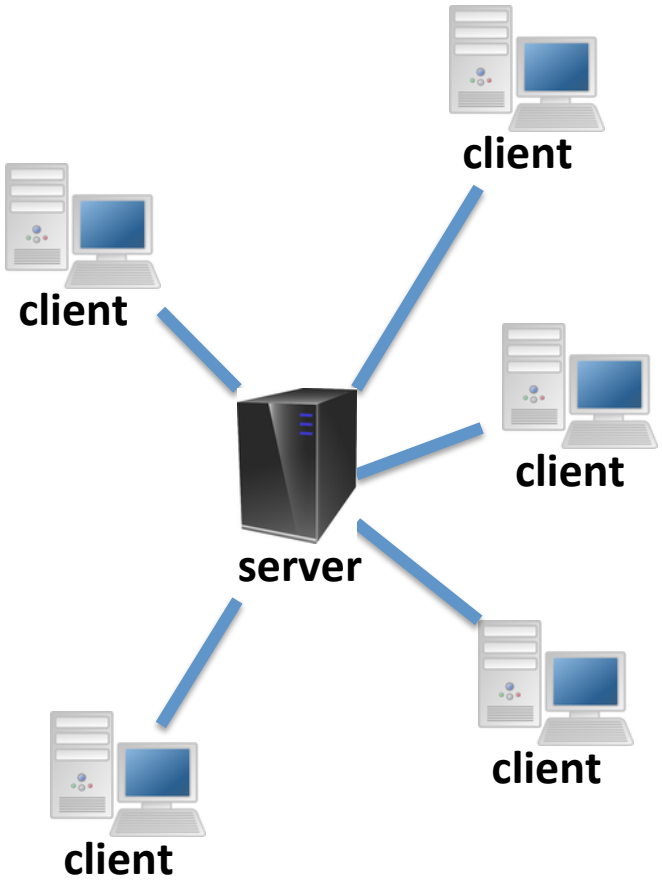
Client-Server Architecture

- A server **waits** to provide services to clients
- Clients connect to and **request** a service from the server
- Client is the one who initiates the communication
- Clients do not communicate to each other



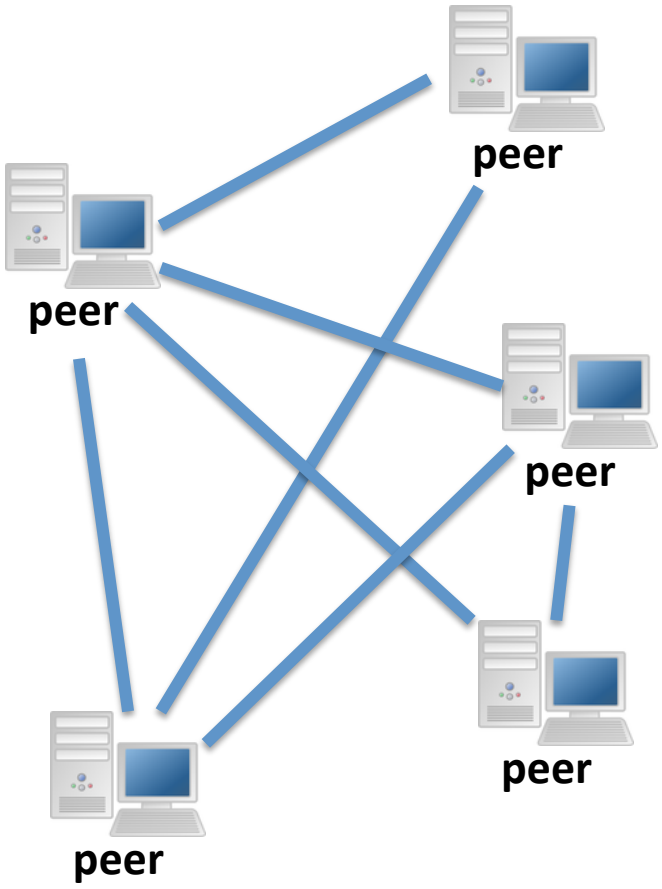
Client-Server Architecture (2)

- Infrastructure-intensive:
  - Servers and network infrastructures must always be maintained
- Expensive
- Example: WWW, FTP, Email, etc.



Peer-to-Peer Architecture

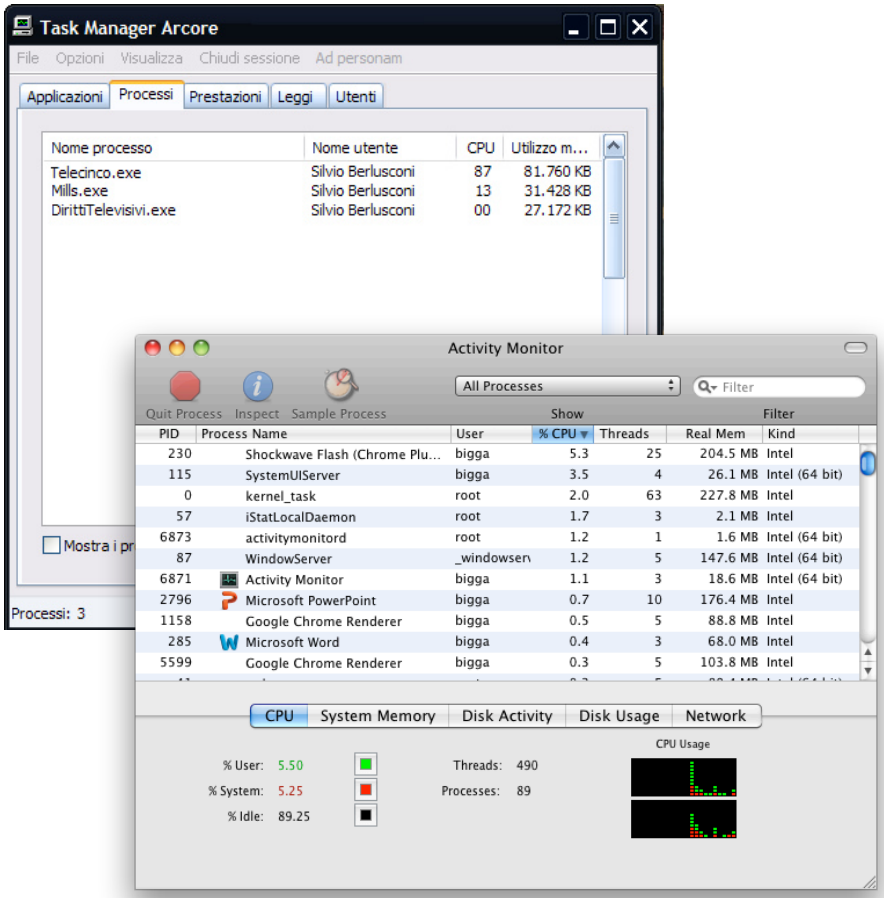
- Server-less
- Direct connections among **peers**
- Good for traffic-intensive applications
- Cost-effective: No server, infrastructure and bandwidth costs
- Scalable: The more peers, the more capability
- Example: Bittorrent, eMule, Skype



Hybrid Architecture

- Server manages some part of communication and the rest is done by the clients
- Example:
  - **MSN Messenger**: The server maintains the IP and statuses of the clients, but the clients communicate directly to each other
  - **Bittorrent**:
    - The server distributes the **torrent description files**.
    - The files to be downloaded are distributed through **seeds**. In this sense, the seeds are file servers
    - After the peers downloaded part of the files, they become one of the file distributors

Processes Communication



- We refer to an instance of an application or program as a **process**
- Processes on different machines are communicating by exchanging **messages**
- Messages are sent and received through a software interface called a **socket**

Image: http://www.flickr.com/photos/pablomoroe/2731541455/

Socket

- A socket can be compared to a service elevator (**dumbwaiter**)
- You put something in and take something out
- You do not have to care how things are done behind the scene
- Still, you have to know what kind of the elevator you should use

Image: http://www.flickr.com/photos/iamthebestartist/5559787935/

Socket (2)

- When a process wants to open a socket, it has to specify
  - Transport-layer protocol
    - TCP: **Connection-oriented**, reliable transport service
    - UDP: **Connectionless**, unreliable transport service
  - **Transport port number**: Identifying the **target process** at the host
  - **IP address**: Identifying the destination host

Connection-Oriented Service

- Modeled after telephone system
- A (virtual) **connection** has to be established **before** the application-layer messages (i.e., the data) can be sent
- Provided by TCP
  - Transport-layer control information has to be exchanged between both communicating parties first
  - This info-exchange process is called **handshaking**
  - After handshaking is success, process can communicating with each other

Connection-Oriented Service

- With handshaking, reliable, synchronized session is possible
- When the process finishes sending messages, the connection must be closed
- **Note**:
  - A TCP connection is only a logical connection
  - It identifies information of communication as a whole e.g., sessions, packet orders and acknowledgements
  - It does not establish any physical connection
  - Packets are transferred individually without resource reservation

Connectionless Service

- Modeled after postal system
- Provided by UDP
- No connection is established. No handshaking
- Each packet is transferred individually
- No ordering, no acknowledgement

Process and Service

- A port number specifies application
- That means: a port number specifies the target **service** at a host
- That means: If we try to *connect* to

  `www.facebook.com:80`,

  we are trying to talk to service that is waiting at port 80 at `www.facebook.com`
- That service is typically HTTP service

IP Address

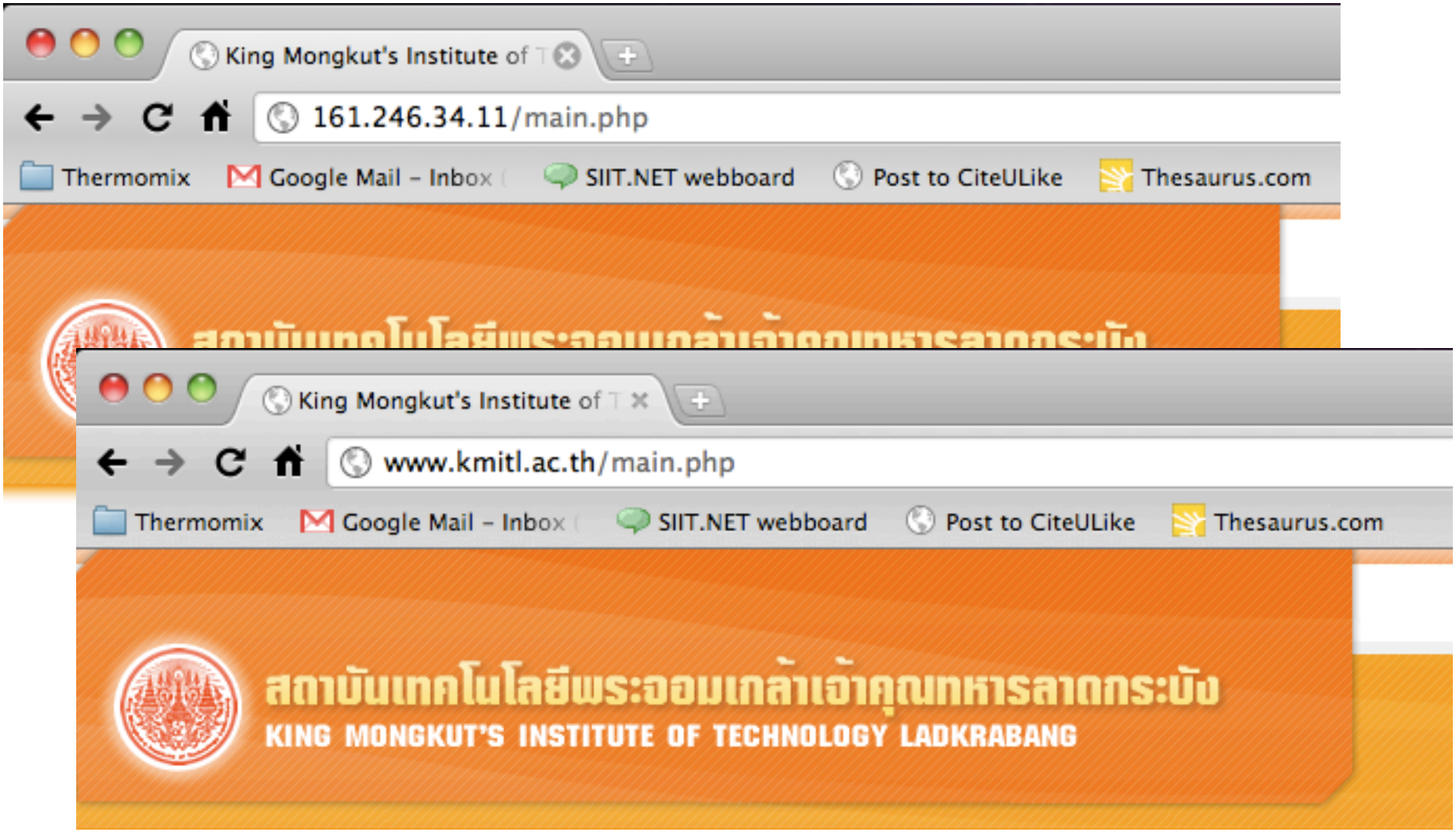
- Computer does not refer to other hosts with names (e.g. `facebook.com`)
- In IP network, a host is identified by an IP address. (Recall IPv4 and IPv6 addresses)
- An IPv4 address consists of 4 sets of numbers xxx.xxx.xxx.xxx : Each set ranges from 0 to 255
- IP protocol routes the packet based on numbers in the address
- It is similar to referring to a student using only student ID

Domain Name System (DNS)

- The IP addresses are not suitable for human. Thus, we usually assign a human-friendly **hostname** to a host
- Therefore, we need a system that can map between IP addresses and hostnames
- **Domain Name System (DNS) server** is used to **resolve** a hostname to the actual IP address

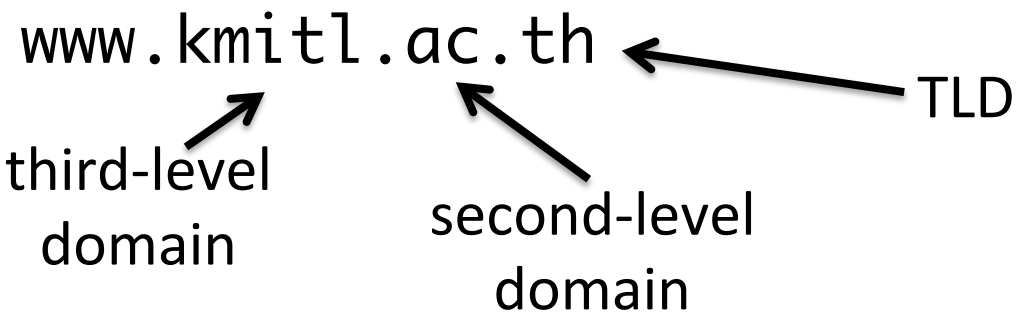
  Example: `www.kmitl.ac.th` to `161.246.34.11`
- Another advantage: IP can be changed anytime

DNS (2)



Domain Names

- It is not possible to maintain direct mapping between hostnames and IP address
  - There are just too many hostnames out there!
- Thus, the DNS divides hostnames into **domains**
- Each domain consists of many **subdomains**
- Each subdomain can further be partitioned into smaller subdomains
- Top-most (first-level) domain is called **top-level domain (TLD)**



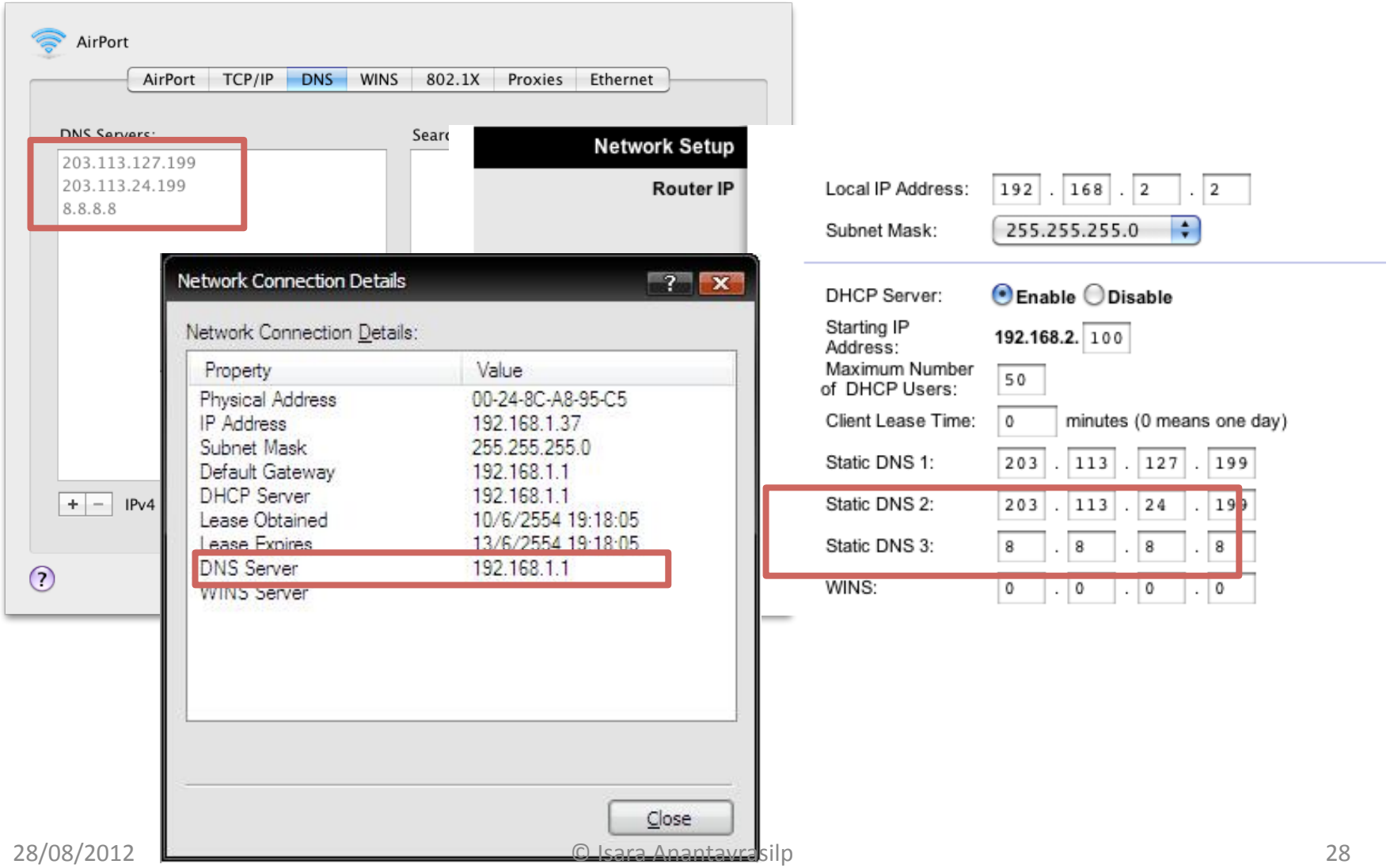
DNS Server

- Upon resolving a domain name, say `www.kmitl.ac.th`, the client asks a **root DNS server** for the IP address
- The root server returns the address of a **TLD DNS server** which is corresponding to `.th` domain
- The TLD server gives the client the address of **authoritative DNS server** corresponding to `.ac.th`
- Then the servers returns the address of another server corresponding to `kmitl.ac.th`
- The last DNS server at KMITL returns the IP address of `www.kmitl.ac.th`

Local DNS Server

- Normally, an end system (e.g., mobile phone, desktop computer, laptop) does not directly resolve an IP address through a root DNS server
- It asks it's local DNS server to do the resolving job for it
- The local DNS servers are generally provided to the end systems by the ISPs
- With local DNS servers
  - The end-systems do not have to know where the root or TLD DNS servers are
  - The resolved IP addresses can be **cached** within the local server

Local DNS Server (2)



DNS Resource Records

- DNS is based on exchanging **resource records**
- Resource records are information that are associated to a domain

Type	Description
A (Address)	IP address of a hostname
NS (Name Server)	Hostname of the DNS server that knows how to obtain IP address for hosts in the domain
CNAME (Canonical Name)	Alias of a hostname or domain name
MX (Mail Exchange)	Hostname of the mail server of the domain (CNAME of the mail server)
PTR (Pointer)	Hostname of an IP address (reverse lookup)
HINFO (Host Info)	Additional information (CPU, OS, …)