CSE 677: Introduction to Computer Networking

Application Layer

Presentation H

Kannan Srinivasan

Study: 22.1, 23.1

11-01-2012

Domain Name System – DNS

- To identify a remote computer, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to and address (or an address to a name)
- One solution would be to store all Name/IPaddress pairs in a single computer and allow access to this centralized information to every computer that needs mapping. But
  - this would create huge amount on the Internet
  - the whole system will stop if the central computer is down
- DNS uses approach where this huge amount of information is divided into smaller parts and each part is stored in a different computer. The host that needs mapping can contact its closest computer holding the needed information.

Mapping the Host Name to an IP Address

- Assume that a user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The following steps map the remote host name to an IP address:
  1. The user passes the host name to the file transfer client,
  2. The file transfer client passes the host name to its name resolver, i.e. its (local) DNS client,
  3. Each computer has to learn (usually at its booting time) the IP address of one DNS server. The DNS client sends a message to a DNS server with the remote host name,
  4. The DNS server responds with IP address of the desired file transfer server,
  5. The DNS client passes the IP address to the file transfer client,
  6. The file transfer client now can use the received IP address to access the file transfer server.

DNS in the Internet

- To be unambiguous, the names assigned to the computers must be carefully selected form a name space with complete control over the binding between names and IP addresses.
- The Internet uses a hierarchical name space, where each name is made of several parts. The first part can define the nature of organization, the second part can define the name of organization, the third part can define departments in the organization, and so on.
- In the Internet, the domain name space is divided into:
  1. Generic domains define registered hosts according to their generic behavior, e.g. com, edu, gov, org.
  2. Country domains use two-character country abbreviations, e.g. us for United States.
  3. Inverse domains are used to map an address to a name.

Portion of Internet Domain Tree

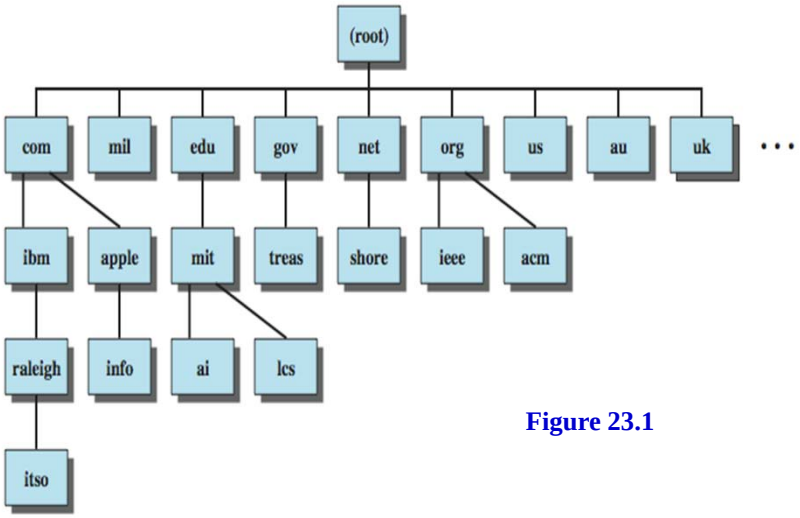


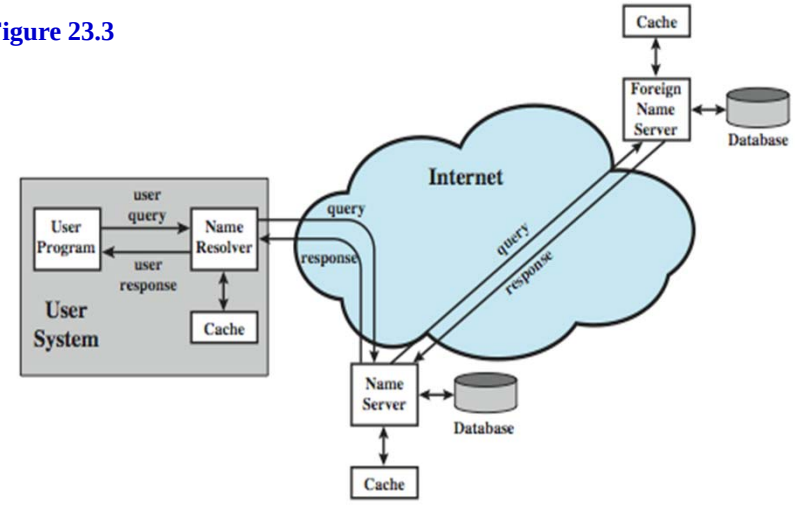
Figure 23.1

Recursive and Iterative Resolution

- Recursive Resolution: The client (resolver) can ask for recursive answer from a server, meaning the resolver expects the server to supply the final answer. If the server is the authority for the name, it checks database and responds. If the server is not the authority, it sends the request to another server and waits for the response. If this server is authority, it response; otherwise, it sends the query to yet another server.
- Interactive Resolution: If the client does not ask for a recursive answer, the mapping can be done iteratively. If the server is an authority for the name, it sends the answer. If it is not, it returns to the client the IP address of the server that it thinks can resolve the query. The client is responsible for repeating the query to this second server. If the newly addressed server can resolve the problem, it answers the query with the IP address; otherwise, it returns the IP address of a new server to the client and now the client must repeat the query to the third server, and so on.

Recursive Resolution

Figure 23.3



DNS Protocol

- DNS can use either UDP or TCP. In both cases the well known port used by the server is port 53. UDP is used when the size of the response message is less than 512 bytes because most UDP packages don't support more than 512 bytes (and this is a most usual case), while a TCP connection has to be used when response message is longer.
- In Unix and Windows, the *nslookup* utility can be used to retrieve IPaddress/Name mapping.
- E.g. the command:

  >nslookup cse.ohio-state.edu

  produces the following respose:

  Server: cse.ohio-state.edu

  Addresses: 164.107.112.224, 164.107.114.11, 164.107.114.70

File Transfer Protocol - FTP

- File Transfer Protocol (FTP) is the standard protocol provided by TCP/IP for copying a file from one host to another. Although it may seems simple, there are many issues to resolve, e.g. two systems may have different ways to represent text and data or two systems may have different directory structure.
- FTP establishes two TCP connections between hosts; one is used for data transfer, the other for control information (commands and responses). This separation makes FTP more efficient. The control connection uses very simple rules of communication, since we need only a transfer of a line of command or a line of response at a time. The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.
- Well known port 21 is used for the control connection and port 20 is used for the data connection.

The Control Connection

- The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transferred. It opens each time when commands that involve transferring files are used, and it closes when the file is transferred. Thus, while the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.
- The control connection is created in the same way as any other application program:
  1. The server issues a passive open on the well-known port 21 and waits for a client.
  2. The client uses an ephemeral port and issues an active open, i.e. initiates TCP connection with the server.
- For control connection, FTP uses simple 7-bit NVT ASCII character set (NVT – Network Virtual Terminal).

The Control Connection: Commands

- One command is sent at a time and response follows. Each line is terminated with a two-character sequence (carriage return and line feed). Examples of commands:
  - RETR *file_name(s)* – retrieve file(s); files are transferred from server to client
  - STOR *file_name(s)* – store file(s); file(s) are transferred from client to server
  - CWD *directory_name* – change to another directory
  - LIST *directory_name* – list subdirectories and files
  - PASS *user_password* – password
  - USER *user_id* – user information
  - PORT *port_ID* – client chooses port
  - TYPE *x* – defines file type
  - STRU *x* – defines data organization
  - QUIT – logout of the system

The Control Connection: Responses

- Every FTP command generates at least one response. A response has two parts: a three digit number followed by text. The number part defines the code, while text part defines needed parameters or extra explanations.
- Examples of response codes with description:
  - 220: Service ready
  - 225: Data connection open
  - 250: Requested file action OK
  - 200: Command OK
  - 331: User name OK; password needed
  - 230: User login OK
  - 226: Closing data connection
  - 500: Syntax error
  - 425: Cannot open data connection
  - 221: Service closing

The Data Connection

- The following steps show how FTP creates a data connection:
  1. The client, not the server, issues a passive open using an ephemeral port. This has to be done by client because it is the client that issues commands for transferring files
  2. The client sends this port number to the server using PORT command
  3. The server receives the port number and issues an active open using the well-known port 20 and the received ephemeral port number
- Before file transfer starts over the data connection, the client must define:
  - file type: A (ASCII file), E (EBCDIC) file or I (image file)
  - data structure: F (no structure; default), R (record structure) or P (page structure)
  - transmission mode: stream mode, block mode, or compressed mode

Example of Using FTP for Retrieving File

Client	Server
	← 220 (service ready) After TCP conn. estab.
USER kannan→	
	← 331 (user name OK)
PASS xxxxx →	
	← 230 (user login OK)
PORT 1333 →	
	← 225 (data connection open)
TYPE F →	
	← 200 (command OK)
STRU I →	
	← 200 (command OK)
RETR /usr/class/cis677/Lab3new/Lab3.demo →	
	← 250 (requested file action OK)
data transfer from server; messages sent through data connection	
	← 226 (closing data connection)
QUIT →	
	← 221 (service closing)

Electronic Mail

- Electronic mail allows users to exchange messages, but its nature is quite different from other applications. In application such as FTP, the server program is running all the time waiting for request from a client and when the request arrives, the server provides the services. Thus, there is a request and a response.
- In the case of electronic mail, situation is different. First, e-mail is in general a one-way transaction. When A sends e-mail to B, A may expect a response, although it is not mandatory. B may or may not respond, and if B does respond, it is another one-way transaction. Also, it is neither feasible nor logical to expect for an intended recipient to be on-line all the time.
- This means that for electronic mail the idea of client/server programming should be implemented in another way: using some intermediate servers.

Simple Mail Transfer Protocol

- Simple Mail Transfer Protocol (SMTP) can deliver only simple text messages and it is limited to transfer of messages between sender and receiver.
- Thus, its main function is to transfer messages and the rest of mail handling is beyond scope of SMTP and may differ between systems.
- E-mail message is created by user agent program (UA), and it consists of an envelope and a message:
  - the envelope includes the sender and receiver addresses
  - the message contains the header (it defines recipient's address, the subject and some other info) and the body (it contains actual information to be read by recipient)
- Example of command-driven UA program: mail, pine and elm.
- Examples of GUI based UA program: Outlook and Eudora.
- Then an e-mail message is transferred by SMTP client to SMTP server sender.

SMTP Mail Flow

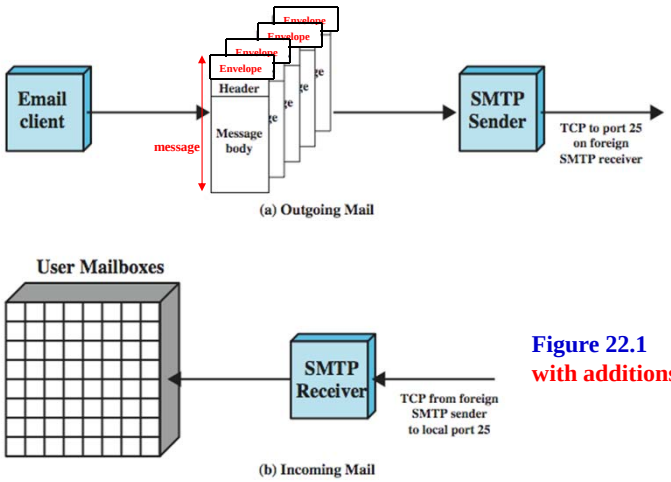


Figure 22.1
with additions in red

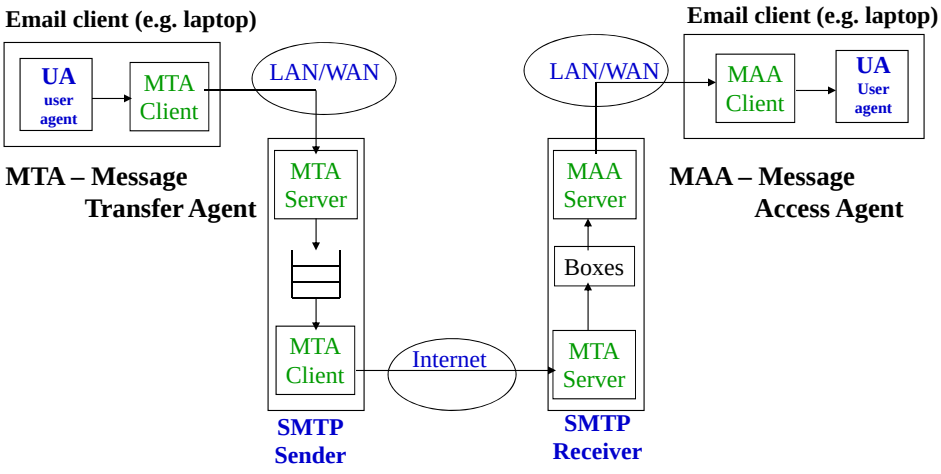
SMTP Sender

- SMTP uses information written on envelope of mail, and does not look at contents of header and message body, except it requires usage of 7-bit VNT ASCII code (and certain log information).
- SMTP sender takes a message from its queue and transmits it to the proper destination server (SMTP receiver) or servers (if a message has multiple receivers)
  - over one or more TCP connections (multiple senders may be active if multiple receivers) to port 25,
  - when delivery complete, sender deletes destination from list for that message and when all destinations processed, message is deleted
  - optimization: if a message destined for multiple users on a given host send it only once and delivery to users handled at the destination server
  - optimization: if multiple messages ready for given server, a single TCP connection can be used and thus saves overhead of setting up and dropping connection

SMTP Receiver

- SMTP server receiver accepts arriving message, places in user mailbox or copies to outgoing queue for forwarding
- SMTP server receiver must:
  - verify local mail destinations
  - deal with errors
- SMTP server sender is responsible for message until SMTP server receiver confirm complete transfer, which only indicates that e-mail message has arrived at the server, not user
- SMTP system mostly does direct transfer from SMTP sender to SMTP receiver
- It may also go through intermediate SMTP servers via forwarding capability
  - target user may have moved
  - sender can specify route

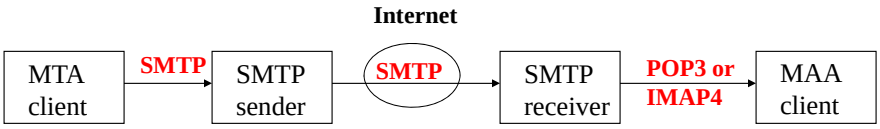
The Common E-Mail Architecture



Some websites (Hotmail, Yahoo and Google) provide web-based mail and that e-mail architecture is somewhat different from one above.

Message Access Agent: POP and IMAP

- The first and second stage of mail delivery uses SMTP, but it is not appropriate for the last stage
- SMPT is a *push* protocol, since it pushes the message from the client to the server
- The third stage needs a *pull* protocol, since the client must pull messages from the server.



- Currently two message access protocols are available:
  - Post Office Protocol, version 3 (POP3) and
  - Internet Mail Access Protocol, version 4 (IMAP4)

Multi-purpose Internet Mail Extension

- SMTP has limitations:
  - can not transmit executables
  - can not transmit text including international characters (e.g. â, å, ä, è, é, ê, ë)
  - servers may reject mail over certain size
  - ASCII to EBCDIC translation not standard
- Multi-purpose Internet Mail Extension (MIME)
  - can deliver other types of data: voice, images, video clips