Application Layer

Compiled by: Hiranya Prasad Bastakoti

Contents

- Introduction and Functions
- Web & Overview of HTTP
- Non-Persistent and Persistent Connections,
- HTTP Message Format
- DNS and the Query Types Services provided by DNS
- Overview of how DNS works
- DNS records and messages
- File Transfer and Email Protocols FTP, SFTP, SMTP, IMAP, POP3
- Overview of Application Server Concepts Proxy, Web, Mail
- Network Management SNMP and Transport mapping

Introduction

- The application layer is the top-most layer of OSI model.
- It provides services directly to user applications.
- It enables the to access the network.
- It provides user interfaces and support for services such as email, remote file access and transfer, shared database management and other types of distributed information services.
- The functions of the application layer are:
- It facilitates the user to use the services of the network.
- It is used to develop network-based applications.
- It provides user services like user login, naming network devices, formatting messages, and e-mails, transfer of files etc.
- It is also concerned with error handling and recovery of the message as a whole

Functions of Application Layer

- **Network virtual terminal:-** It is software terminal that allows user to logon and interact with the machine
- **File Transfer:** It allows a user to access, retrieve and manage files in a remote computer.
- **Mail services:** It provides the basis for email forwarding and storage facilities.
- **Directory services:** It provides distributes database sources and access for global information about various objects and services.

Application layer protocols

- *Hyper Text Transfer Protocol, HTTP:* It is the underlying protocol for world wide web. It defines how hypermedia messages are formatted and transmitted.
- *File Transfer Protocol, FTP:* It is a client-server based protocol for transfer of files between client and server over the network.
- *Simple Mail Transfer Protocol, SMTP:* It lays down the rules and semantics for sending and receiving electronic mails (e-mails).
- *Domain Name System, DNS:* It is a naming system for devices in networks. It provides services for translating domain names to IP addresses.
- *TELNET:* It provides bi-directional text-oriented services for remote login to the hosts over the network.
- *Simple Network Management Protocol, SNMP:* It is for managing, monitoring the network and for organizing information about the networked devices.

Web & Overview of HTTP

- The Web, or World Wide Web (W3), is basically a system of Internet servers that support specially formatted documents.
- Web page consists of objects
- The World Wide Web (WWW) is a repository of information linked together from points all over the world
- Object can be HTML file, JPEG image, Java applet, audio file,…
- The documents are formatted in a markup language called HTML (*HyperText Markup Language*) that supports links to other documents, as well as graphics, audio, and video files
- Web page consists of base HTML-file which includes several referenced objects
- Each object is addressable by a URL

.

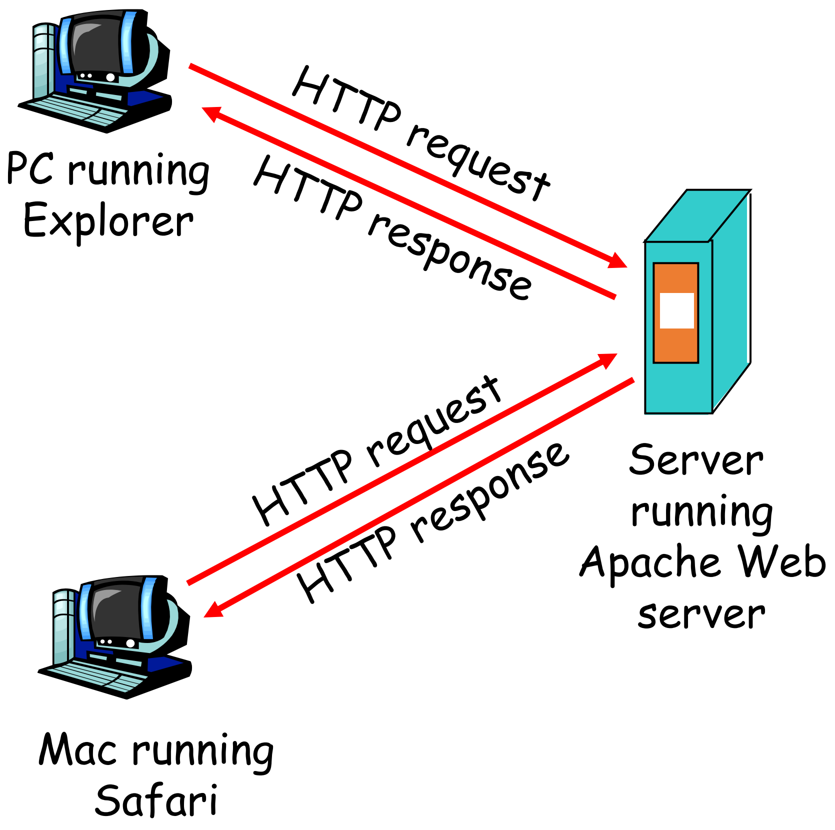
HTTP

- HTTP, the Hypertext Transfer Protocol, is the application-level protocol that is used to transfer data on the Web.
- HTTP comprises the rules by which Web browsers and servers exchange information
- Hyper Text Transfer Protocol is a file transfer protocol specifically designed to facilitate access to the WWW.
- This protocol transfers data in the form of plain text, hypertext, audio, video, and so on
- **HTTP uses the services of TCP on a well-known port 80**

Features of HTTP

- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.
- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.
- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages

How Does HTTP Work?



How Does HTTP Work?

- The above fig illustrates the **HTTP transaction between the client and the server.**
- **The client initializes the transaction by sending a request message.**
- **The server replies by sending a response.**
- **Although HTTP uses the services of TCP, HTTP itself is a stateless protocol(**server maintains no information about past client requests)

It uses Uses TCP:

- client initiates TCP connection (creates socket) to server, port 80 (default)
- server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

How Does HTTP Work?

HTTP Is a request-response protocol. For example, a Web browser initiates a request to a server, typically by opening a TCP/IP connection. The request itself comprises

- a request line,
- a set of request headers, and
- an entity.

The server sends a response that comprises

- a status line,
- a set of response headers, and
- an entity.

The entity in the request or response can be thought of simply as the payload, which may be binary data. The other items are readable ASCII characters. When the response has been completed, either the browser or the server may terminate the TCP/IP connection, or the browser can send another request.

HTTP Connections

Persistent Versus Non-persistent Connection

- **Non-Persistent Connection**: It requires connection setup again and again for each object to send.
- **Persistent connection**: It does not require connection setup again and again. Multiple objects can use connection
- HTTP prior to version 1.1 specified a nonpersistent connection, while a persistent connection is the default in version 1.1.

Nonpersistent HTTP

- At most one object is sent over a TCP connection.
- HTTP/1.0 uses nonpersistent HTTP

Persistent HTTP

- Multiple objects can be sent over single TCP connection between client and server.
- HTTP/1.1 uses persistent connections in default mode

Nonpersistent Connection

In a nonpersistent connection, one TCP connection is made for each request/response.

The following lists the steps in this strategy:

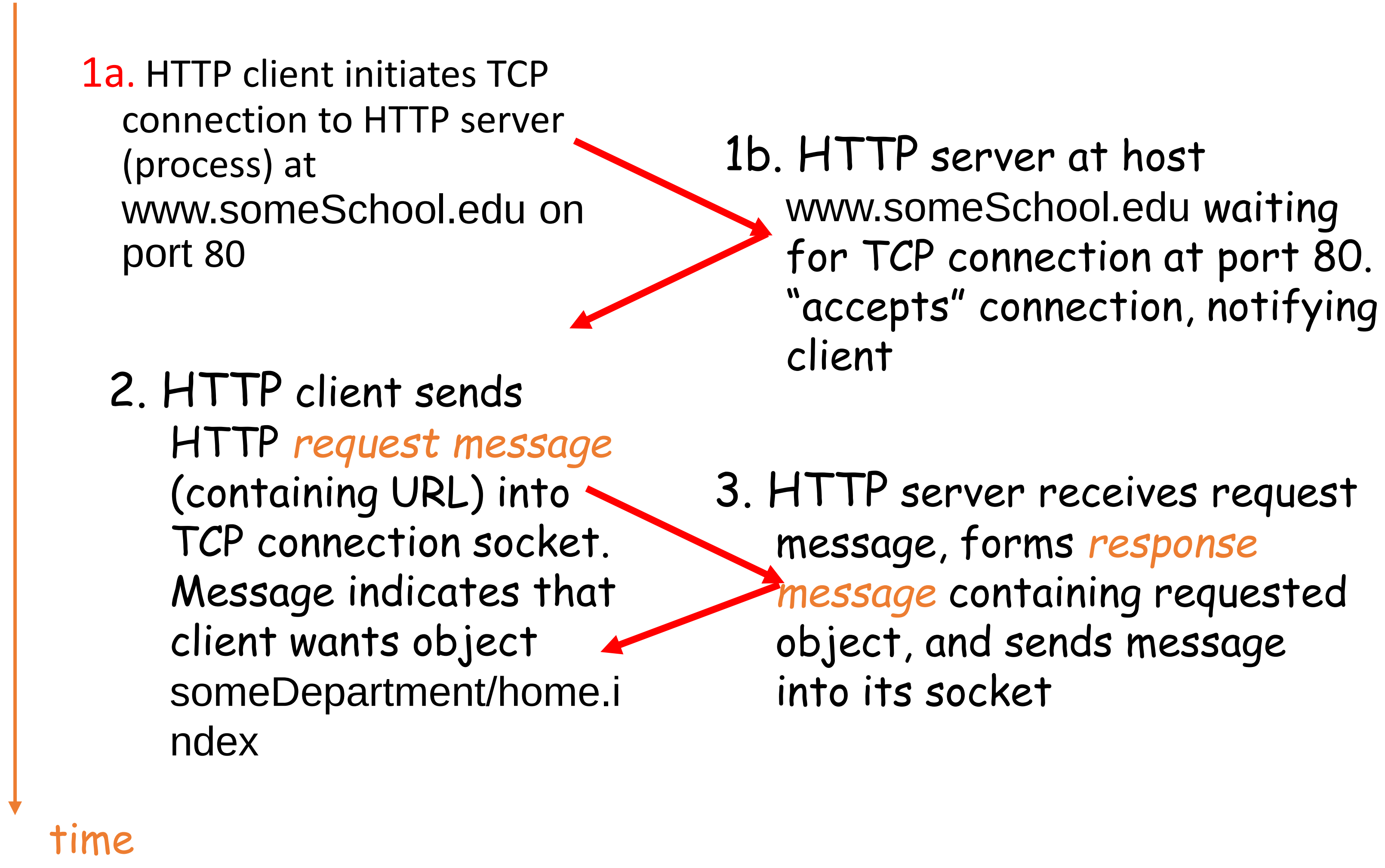
1. The client opens a **TCP connection and sends a request.**
2. The server sends the response and closes the connection.
3. The client reads the data until it encounters an end-of-file marker; it then closes the connection.

- In this strategy, for *N different pictures in different files, the connection must be opened and closed N times. The non persistent strategy imposes high overhead on the server because the server needs N different buffers and requires a slow start procedure each time a connection is opened.*

Nonpersistent HTTP

Suppose user enters URL
`www.someSchool.edu/someDepartment/home.index`

(contains text, references to 10 jpeg images)



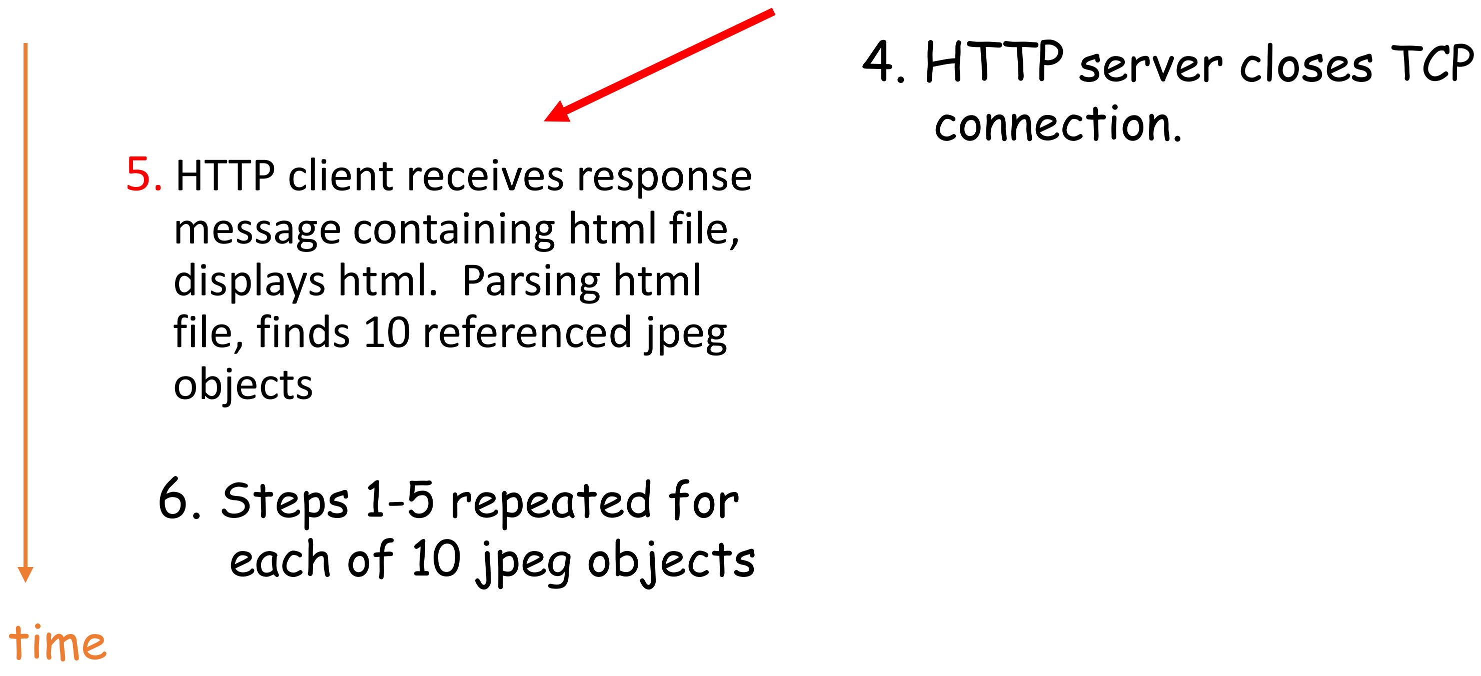
1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. HTTP client sends HTTP *request message* (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

Nonpersistent HTTP (cont.)



4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

Persistent Connection

- HTTP version 1.1 specifies a persistent connection by default.
- In a persistent connection, the server leaves the connection open for more requests after sending a response. The server can close the connection at the request of a client or if a time-out has been reached.
- The sender usually sends the length of the data with each response.
- However, there are some occasions when the sender does not know the length of the data.
- This is the case when a document is created dynamically or actively.
- In these cases, the server informs the client that the length is not known and closes the connection after sending the data so the client knows that the end of the data has been reached

HTTP message

- Two types of HTTP messages: *request, response*
- *Both uses almost same format*
- HTTP request message:
  - ASCII (human-readable format)

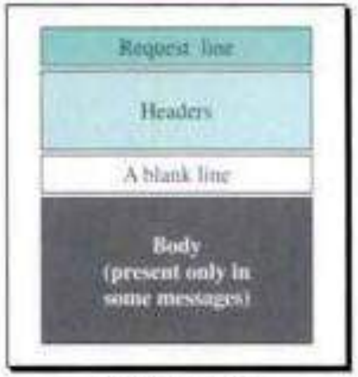
request line (GET, POST, HEAD commands)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language:fr
```

header lines

(extra carriage return, line feed; i.e., a blank line)

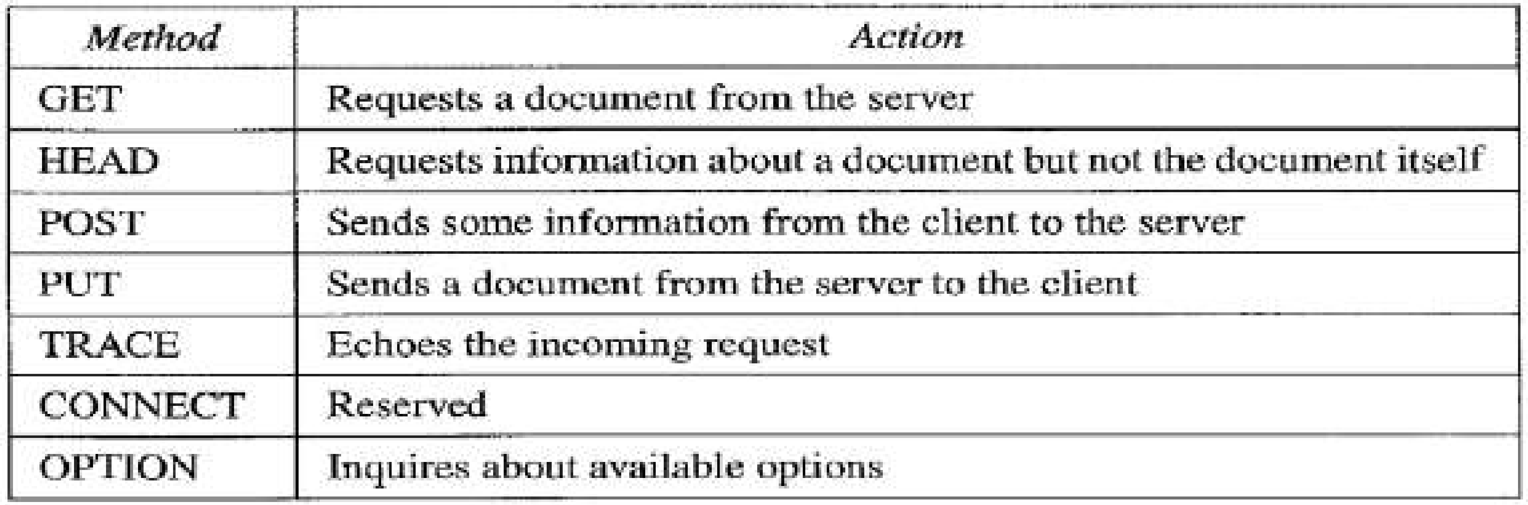
Carriage return, line feed indicates end of message



Request message

HTTP request message

- **Request line :**defines the request type, resource (**URL), and HTTP version**
- **Request Type: In version 1.1 HTTP, several request types are defined. The request type categorizes the request messages into several methods.**
- These methods are defined as several kinds of messages. The request method is the actual command or request that a client issues to the server

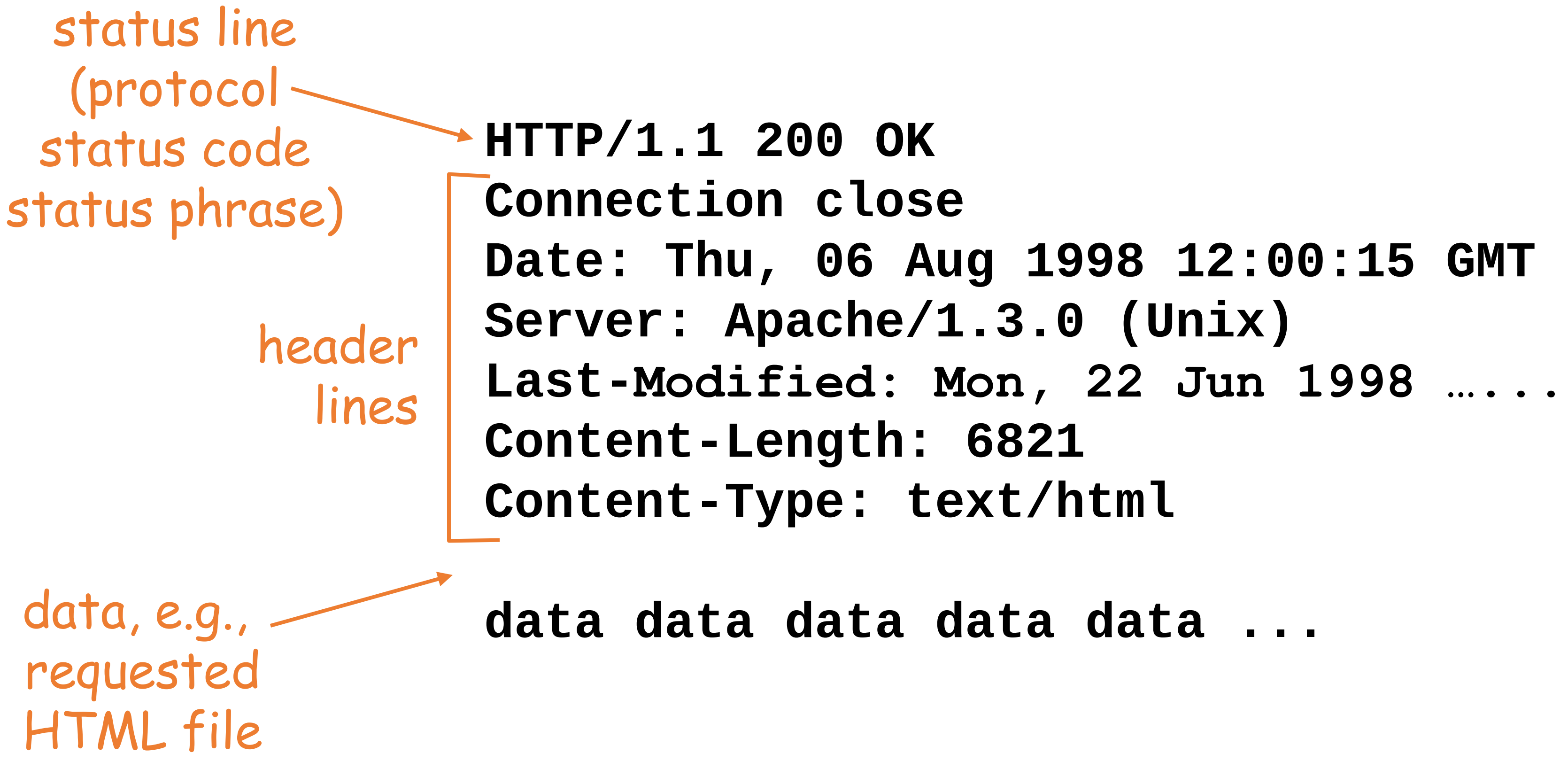


Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

Response Message:

- A response message consists of a status line, a header, and sometimes a body.
- Status Line: The status line defines the status of the response message.
- It consists of the HTTP version, a space, a status code, a space, and a status phrase.
- HTTP version: This field is the same as the corresponding field in the request line.
- Status code: The status code field is similar to those in the FTP and the SMTP protocols.
- It consists of three digits. Whereas the codes in the 100 range are only informational, the codes in the 200 range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site

HTTP response message



status line (protocol status code status phrase)

header lines

data, e.g., requested HTML file

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 …...
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```

Other status codes are

`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`

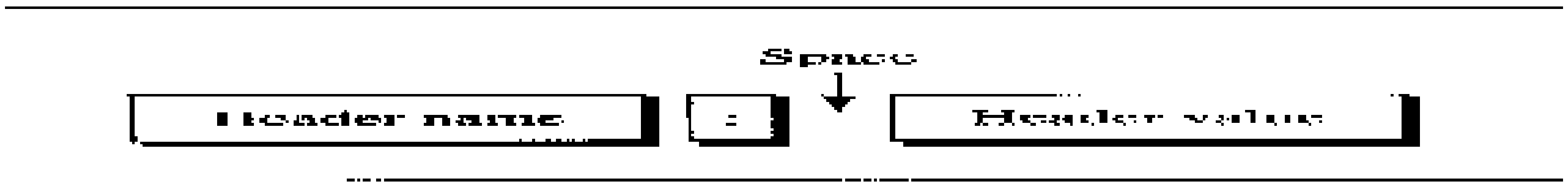
- request message not understood by server

`404 Not Found`

- requested document not found on this server

`505 HTTP Version Not Supported`

Header and Body



- The header exchanges additional information between the client and the server.
- For example, the client can request that the document be sent in special format, or the server can send extra information about the document.
- The header can consist of one or more header lines.
- Each header line has a header name, a colon, a space, and a header value). A header line belongs to one of four categories: general header, request header, response header, and entity header.
- A request message can contain only general, request, and entity header. A response message, on the other hand, can contain only general, response, and entity headers

Body:

The body can be present in a request or response message. Usually, it contains the document to be sent or received

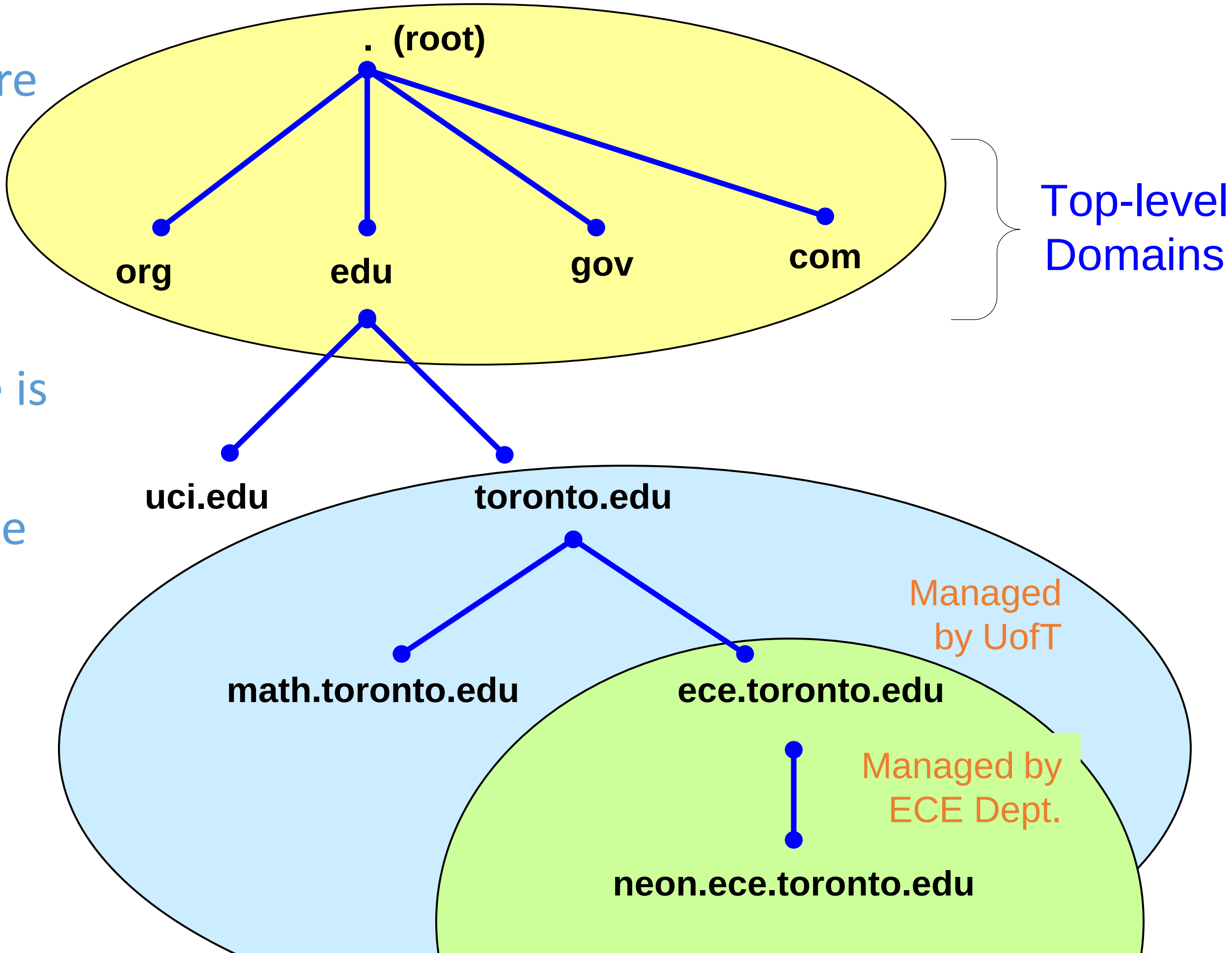
Overview of Domain Name System

Domain Name System is a hierarchical distributed database

- DNS is the foundation of the Internet naming scheme
- People prefer to use easy-to-remember names instead of IP addresses
- Domain names are alphanumeric names for IP addresses e.g., neon.ece.utoronto.ca, www.google.com, ietf.org
- The domain name system (DNS) is an Internet-wide distributed database that translates between domain names and IP addresses
- DNS was created to support the Internet's growing number of hosts

DNS Name hierarchy

- DNS hierarchy can be represented by a tree
- Root and top-level domains are administered by an Internet central name registration authority (ICANN)
- Below top-level domain, administration of name space is delegated to organizations
- Each organization can delegate further



Top-level domains

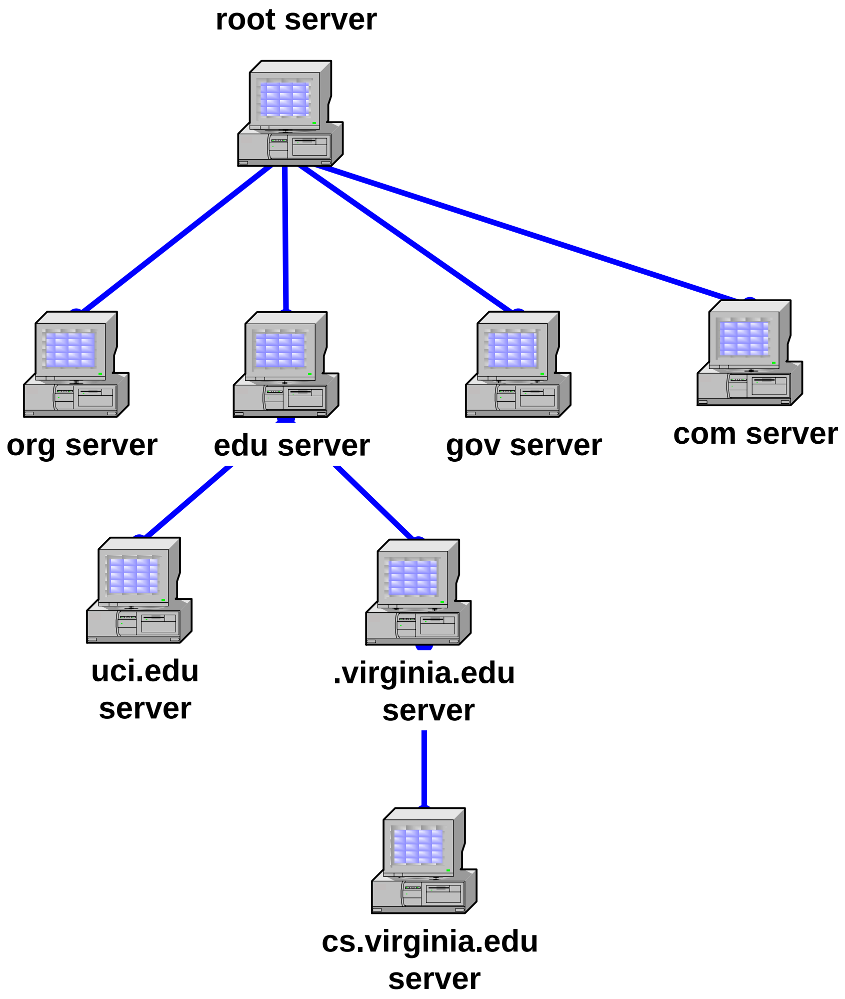
- Three types of top-level domains:
  - Organizational: 3-character code indicates the function of the organization
    - Examples: gov, mil, edu, org, com, net
  - Geographical: 2-character country or region code
    - Examples: us, va, jp, de
  - Reverse domains: A special domain (in-addr.arpa) used for IP address-to-name mapping

Organizational top-level domains

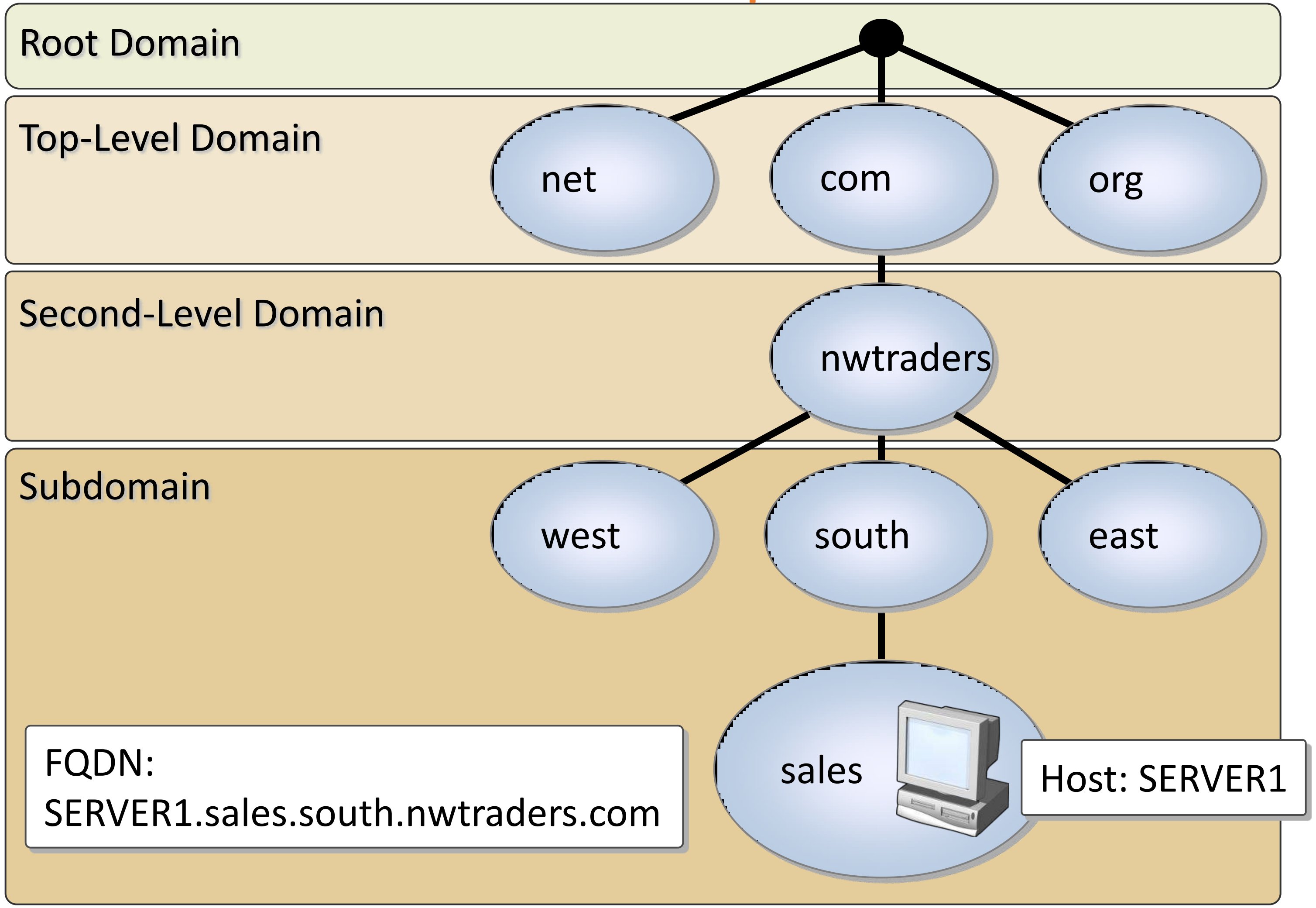
com	Commercial organizations
edu	Educational institutions
gov	Government institutions
int	International organizations
mil	military institutions
net	Networking organizations
org	Non-profit organizations

Hierarchy of name servers

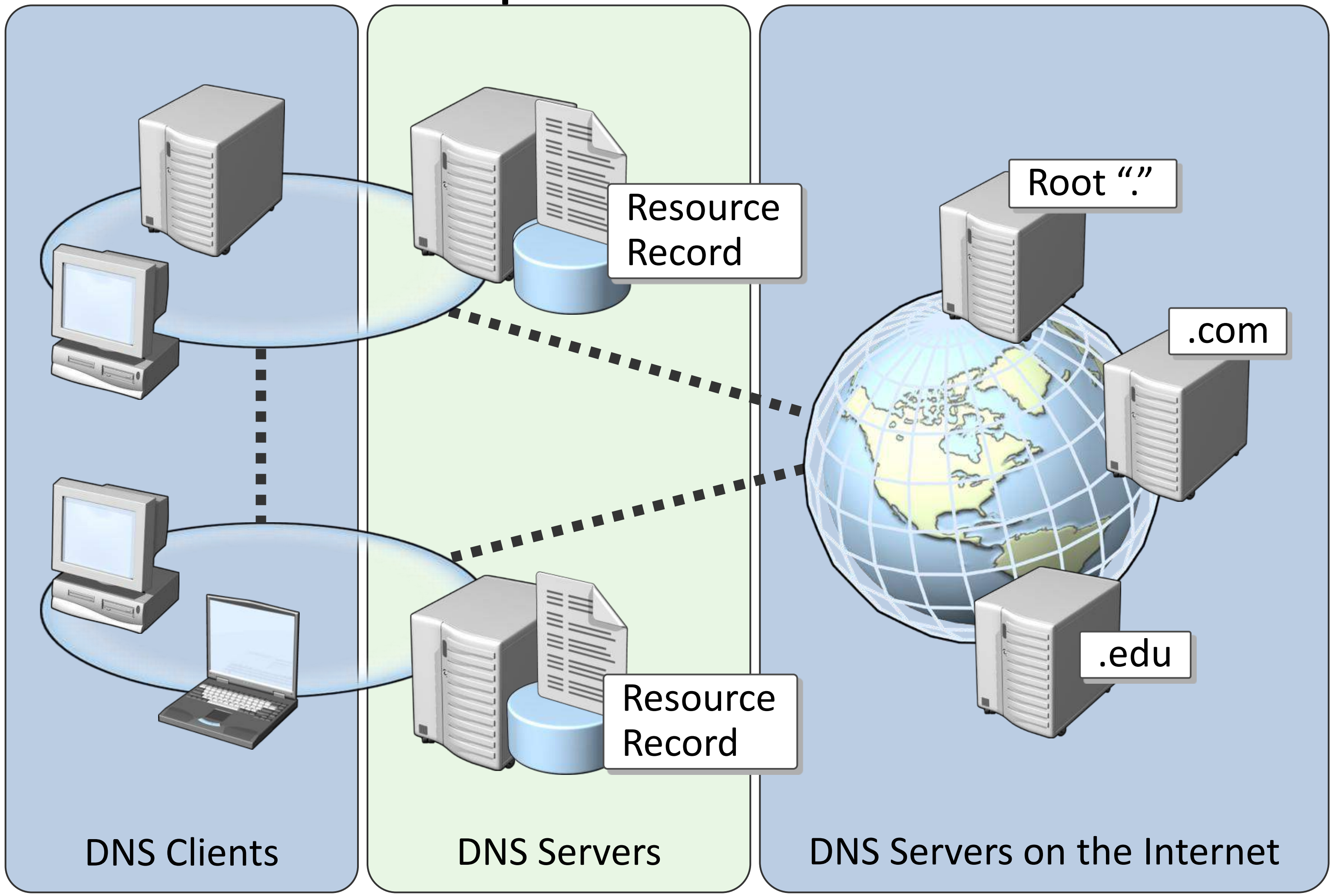
- The resolution of the hierarchical name space is done by a hierarchy of name servers
- Each server is responsible (authoritative) for a contiguous portion of the DNS namespace, called a zone.
- Zone is a part of the subtree
- DNS server answers queries about hosts in its zone



What Is a Domain Namespace?



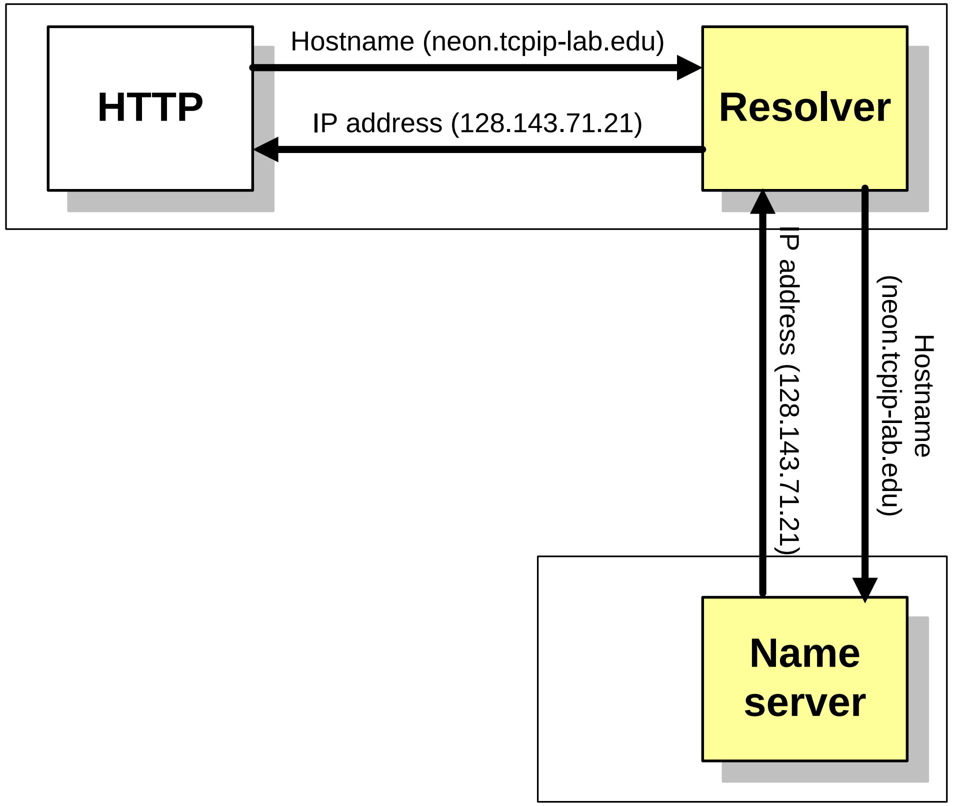
What Are the Components of a DNS Solution?



Resolver and name server

1. An application program on a host accesses the domain system through a DNS client, called the **resolver**
2. Resolver contacts DNS server, called name server
3. DNS server returns IP address to resolver which passes the IP address to application

- Reverse lookups are also possible, i.e., find the hostname given an IP address


HTTP
Hostname (neon.tcpip-lab.edu)
IP address (128.143.71.21)
Resolver
IP address (128.143.71.21)
Hostname (neon.tcpip-lab.edu)
Name server

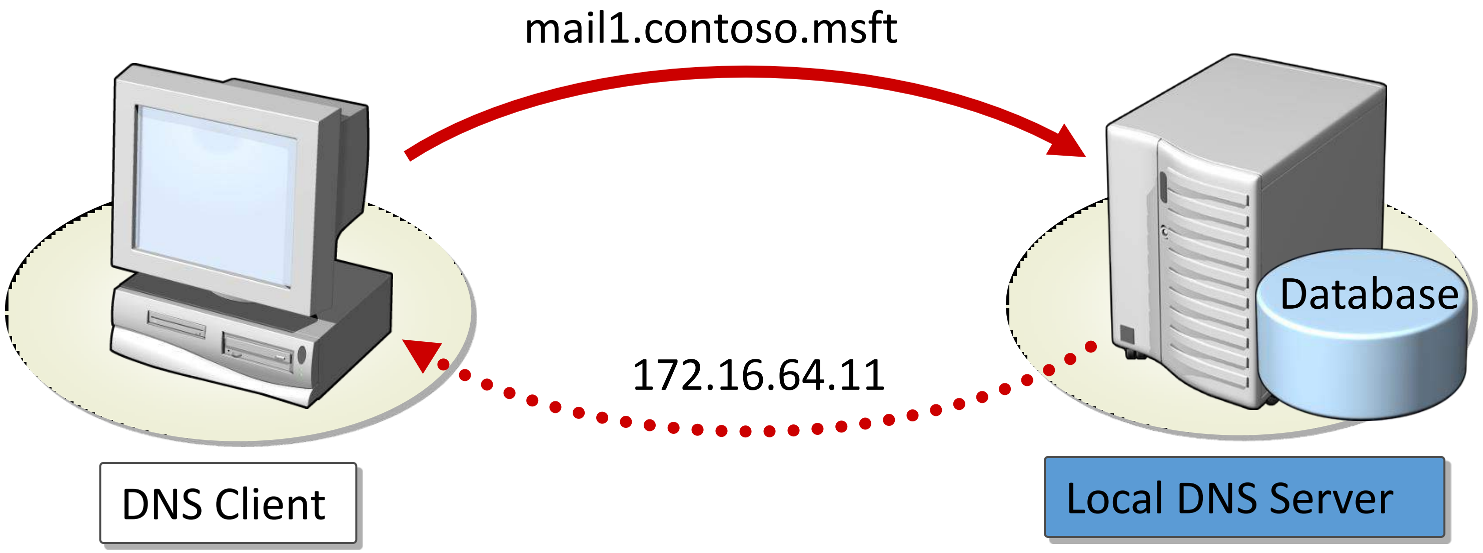
What Is a DNS Query?

A *query* is a request for name resolution and is directed to a DNS server

- Queries are recursive or iterative
- DNS clients and DNS servers both initiate queries
- DNS servers are authoritative or nonauthoritative for a namespace
- An authoritative DNS server for the namespace will either:
  - Return the requested IP address
  - Return an authoritative “No”
- A nonauthoritative DNS server for the namespace will either:
  - Check its cache
  - Use forwarders
  - Use root hints

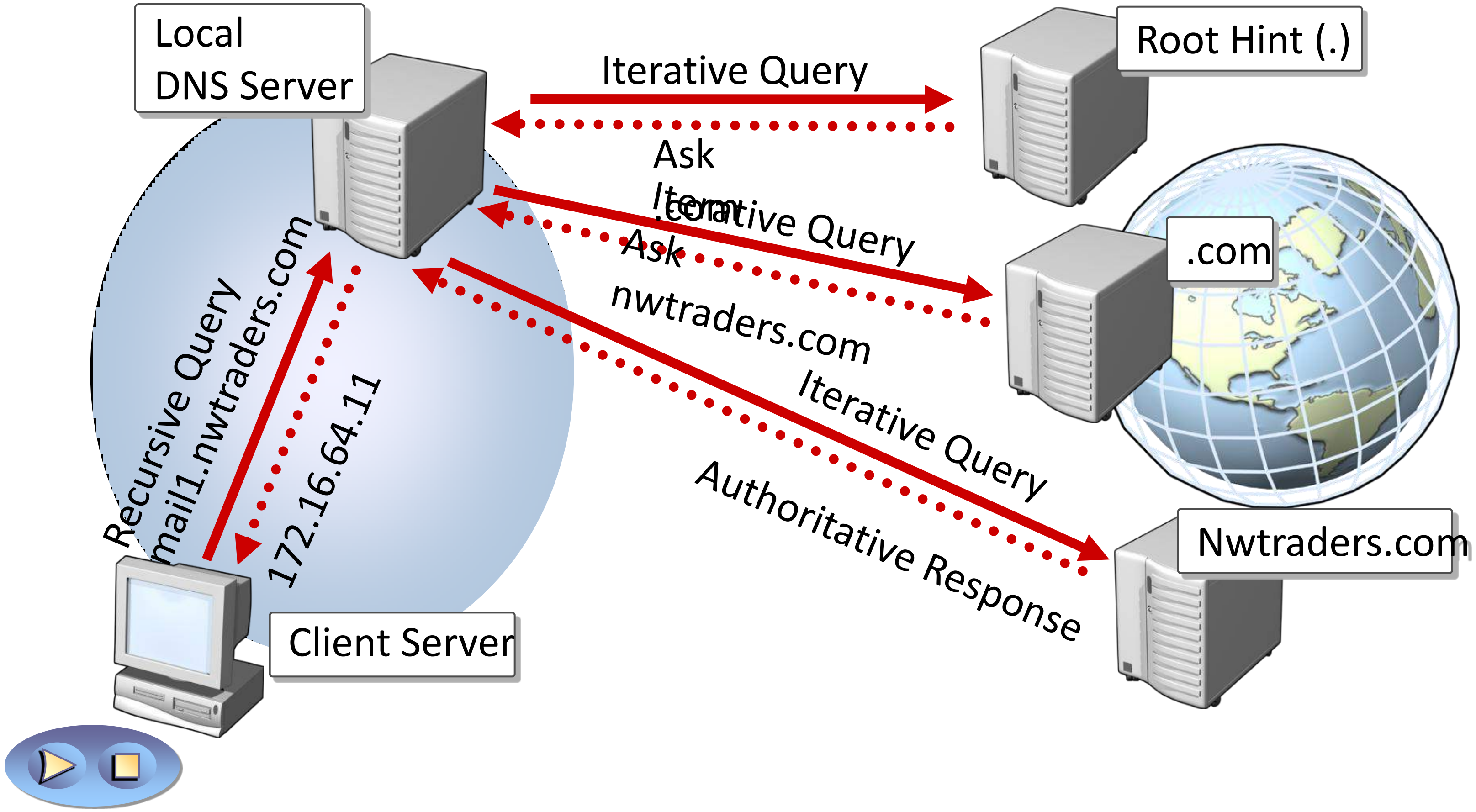
How Recursive Queries Work

A *recursive query* is sent to a DNS server and requires a complete answer



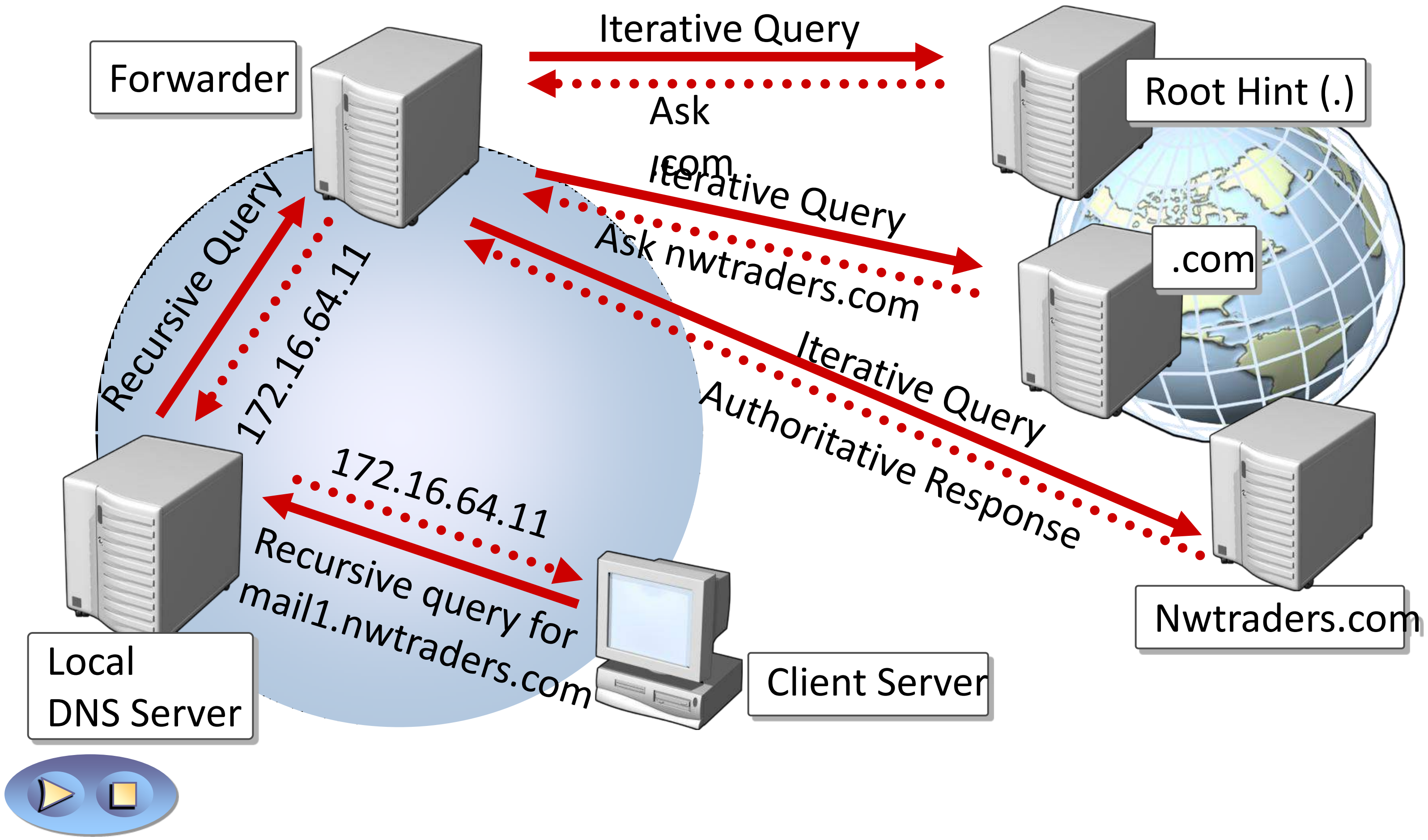
How Iterative Queries Work

An iterative query directed to a DNS server may be answered with a referral to another DNS server

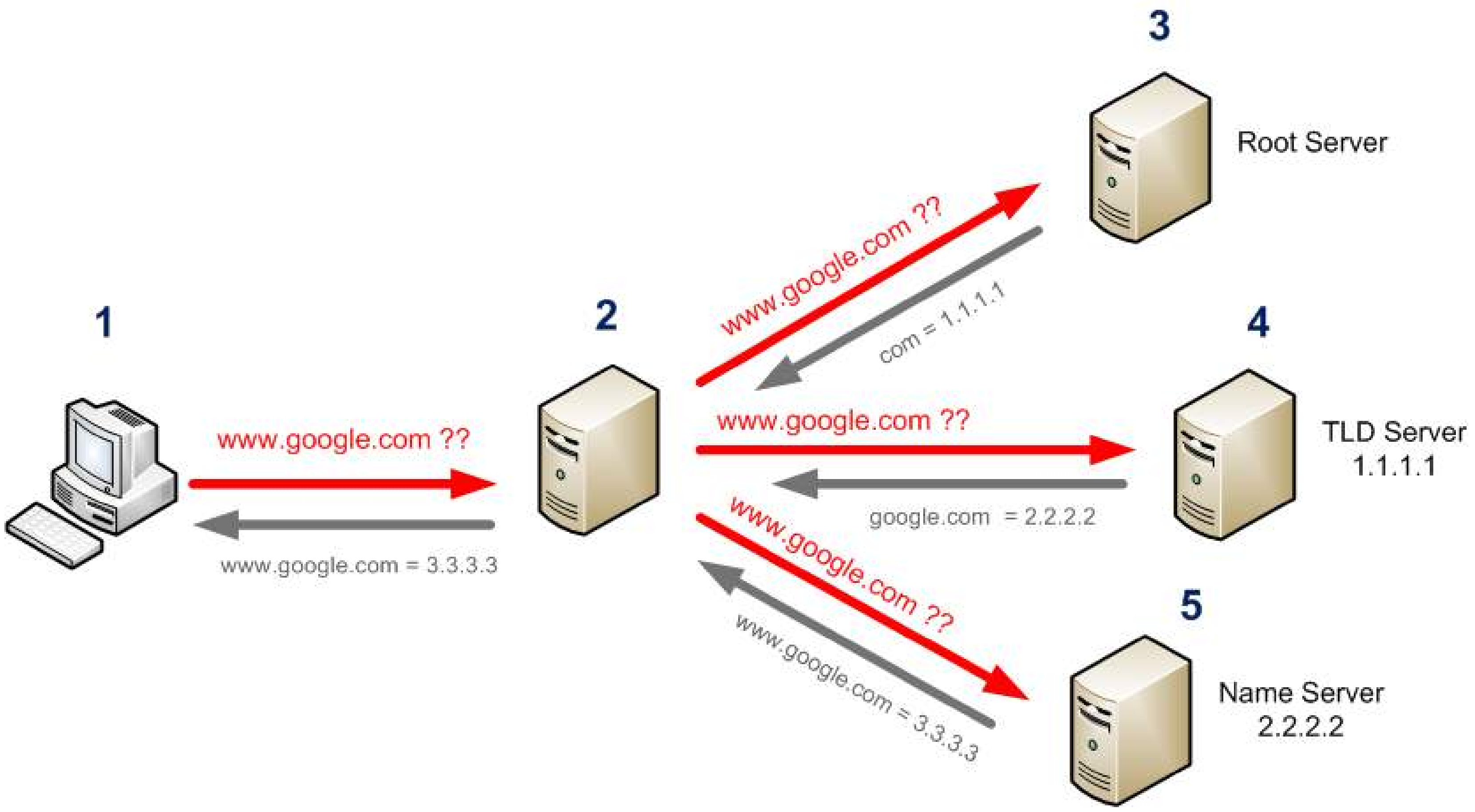


How Forwarders Work

A *forwarder* is a DNS server designated to resolve external or offsite DNS domain names



How DNS works?



How DNS Works?

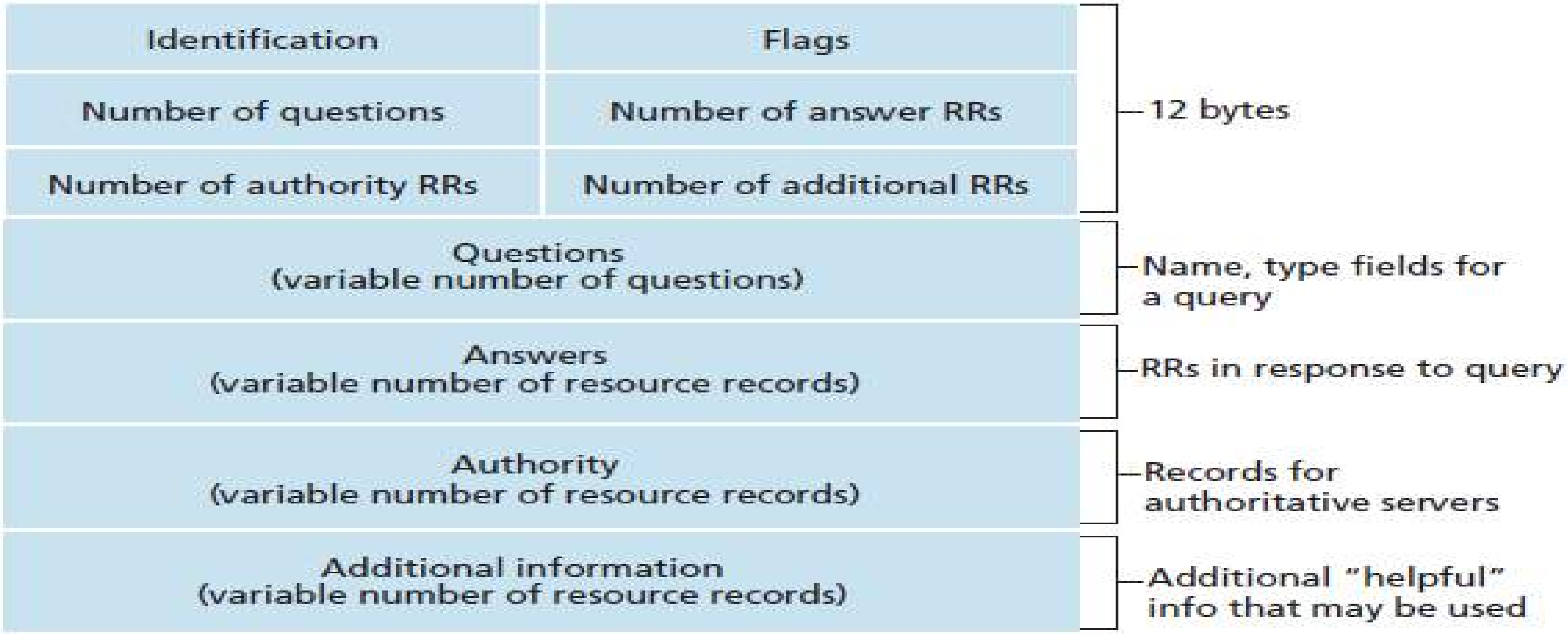
- Step 1: The client proposes a domain name resolution request and sends the request to the local domain name server.
- Step 2: When the local domain name server receives the request, it first queries the local cache. If there is this record, the local domain name server directly returns the result of the query.
- Step 3: If the local cache does not have the record, the local domain name server directly sends the request to the root domain name server, and then the root domain name server returns the primary domain name of the domain (the subdomain of the root) of the local domain name server. The address of the server.
- Step 4: The local server sends a request to the domain name server returned in the previous step, and then the server that accepts the request queries its own cache. If there is no such record, it returns the address of the relevant lower-level domain name server.
- Step 5: Repeat step 4 until you find the correct record.
- Step 6: The local domain name server saves the returned results to the cache for the next use and returns the results to the client

DNS Record and Message

- The name servers that together implement the DNS distributed database, store Resource Records (RR) for the hostname to IP address mappings.
- Each DNS reply message carries one or more resource record
- A resource record is a four-tuple that contains the following fields:
- (Name, Value, Type, TTL)
- TTL is the time to live of the resource record; it determines the time at which a resource should be removed from a cache
- The meaning of Name and Value depend on Type

- If Type=A, then Name is a hostname and Value is the IP address for the hostname. Thus, a Type A record provides the standard hostname to IP address mapping. As an example, (relay1.bar.foo.com, 145.37.93.126, A) is a Type A record.
- If Type=NS, then Name is a domain (such as foo.com) and Value is the hostname of a server that knows how to obtain the IP addresses for hosts in the domain. This record is used to route DNS queries further along in the query chain. As an example, (foo.com, dns.foo.com, NS) is a Type NS record.
- If Type=CNAME, then Value is a canonical hostname for the alias hostname Name. This record can provide querying hosts the canonical name for a hostname. As an example, (foo.com, relay1.bar.foo.com, CNAME) is a CNAME record.
- If Type=MX, then Value is a hostname of a mail server that has an alias hostname Name. As an example, (foo. com. mail.bar.foo.com, MX) is an MX record. MX records allow the hostnames of mail servers to have simple aliases.

DNS Message Format



Identification	Flags
Number of questions	Number of answer RRs
Number of authority RRs	Number of additional RRs

12 bytes

Questions
(variable number of questions)

Name, type fields for a query

Answers
(variable number of resource records)

RRs in response to query

Authority
(variable number of resource records)

Records for authoritative servers

Additional information
(variable number of resource records)

Additional "helpful" info that may be used

- The first 12 bytes is the header section, which has a number of fields.
- The first field is a 16-bit number that identifies the query. This identifier is copied into the reply message to a query, allowing the client to match received replies with sent queries.
- There are a number of flags in the flag field. A one-bit query/reply flag indicates whether the message is a query (0) or a reply (1).
- A one bit authoritative flag is set in a reply message when a name server is an authoritative server for a queried name.
- A one bit recursion-desired flag is set when a client (host or name server) desires that the name server to perform recursion when it doesn't have the record. A one-bit recursion available field is set in a reply if the name server supports recursion.
- In the header, there are also four "number of" fields. These fields indicate the number of occurrences of the four types of "data" sections that follow the header

- The question section contains information about the query that is being made. This section includes (i) a name field that contains the name that is being queried, and (ii) a type field that indicates the type of question being asked about the name (e.g., a host address associated with a name - type "A", or the mail server for a name - type "MX").
- In a reply from a name server, the answer section contains the resource records for the name that was originally queried. Recall that in each resource record there is the Type (e.g., A, NS, CSNAME and MX), the Value and the TTL. A reply can return multiple RRs in the answer, since a hostname can have multiple IP addresses (e.g., for replicated Web servers, as discussed earlier in this section).
- The authority section contains records of other authoritative servers.
- The additional section contains other "helpful" records. For example, the answer field in a reply to an MX query will contain the hostname of a mail server associated with the alias name Name. The additional section will contain a Type A record providing the IP address for the canonical hostname of the mail server.

FTP

- FTP (File Transfer Protocol) is a protocol for transferring a file from one host to another host.
- Transferring files from a client computer to a server computer is called **"uploading"** and transferring from a server to a client is **"downloading".**
- In a typical FTP session, the user is sitting in front of one host (the local host) and wants to transfer files to or from a remote host.
- In order for the user to access the remote account, the user must provide a user identification and a password.
- After providing this authorization information, the user can transfer files from the local file system to the remote file system and vice versa.
- As shown in Figure below the user interacts with FTP through an FTP user agent. The user first provides the hostname of the remote host, which causes the FTP client process in the local host to establish a TCP connection with the FTP server process in the remote host. The user then provides the user identification and password, which get sent over the TCP connection as part of FTP commands.
- Once the server has authorized the user, the user copies one or more files stored in the local file system into the remote file system (or vice versa).

FTP

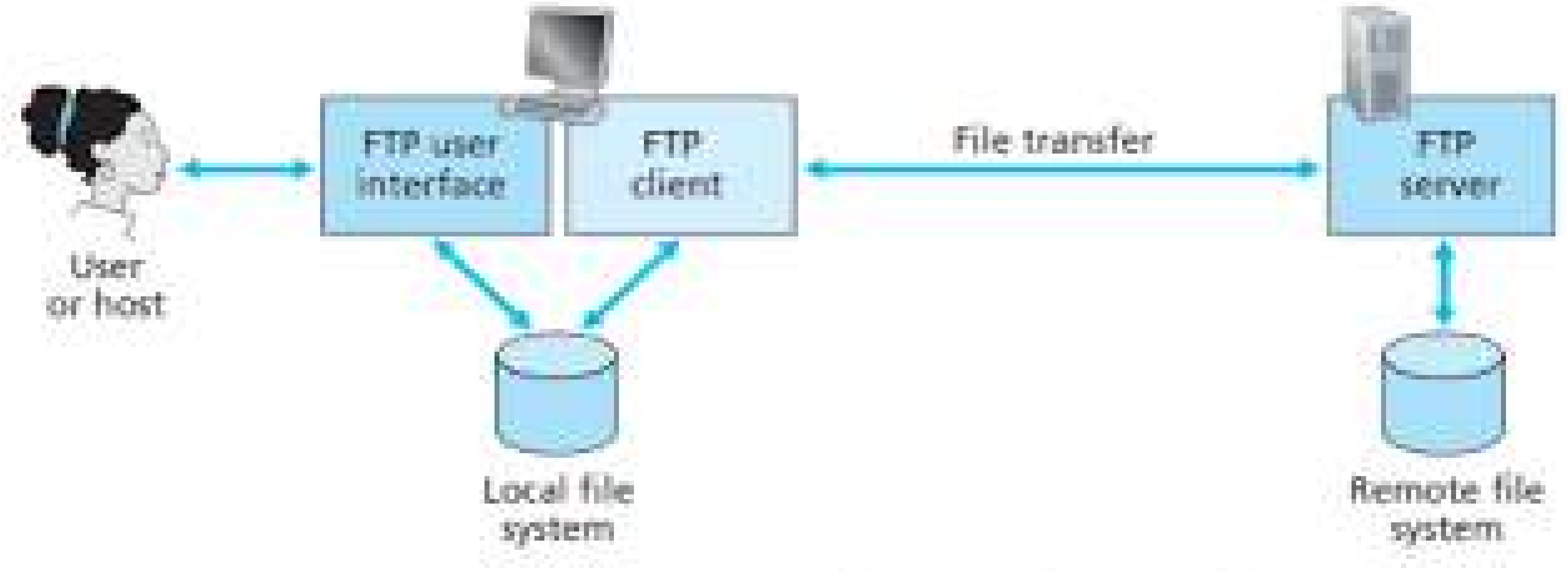
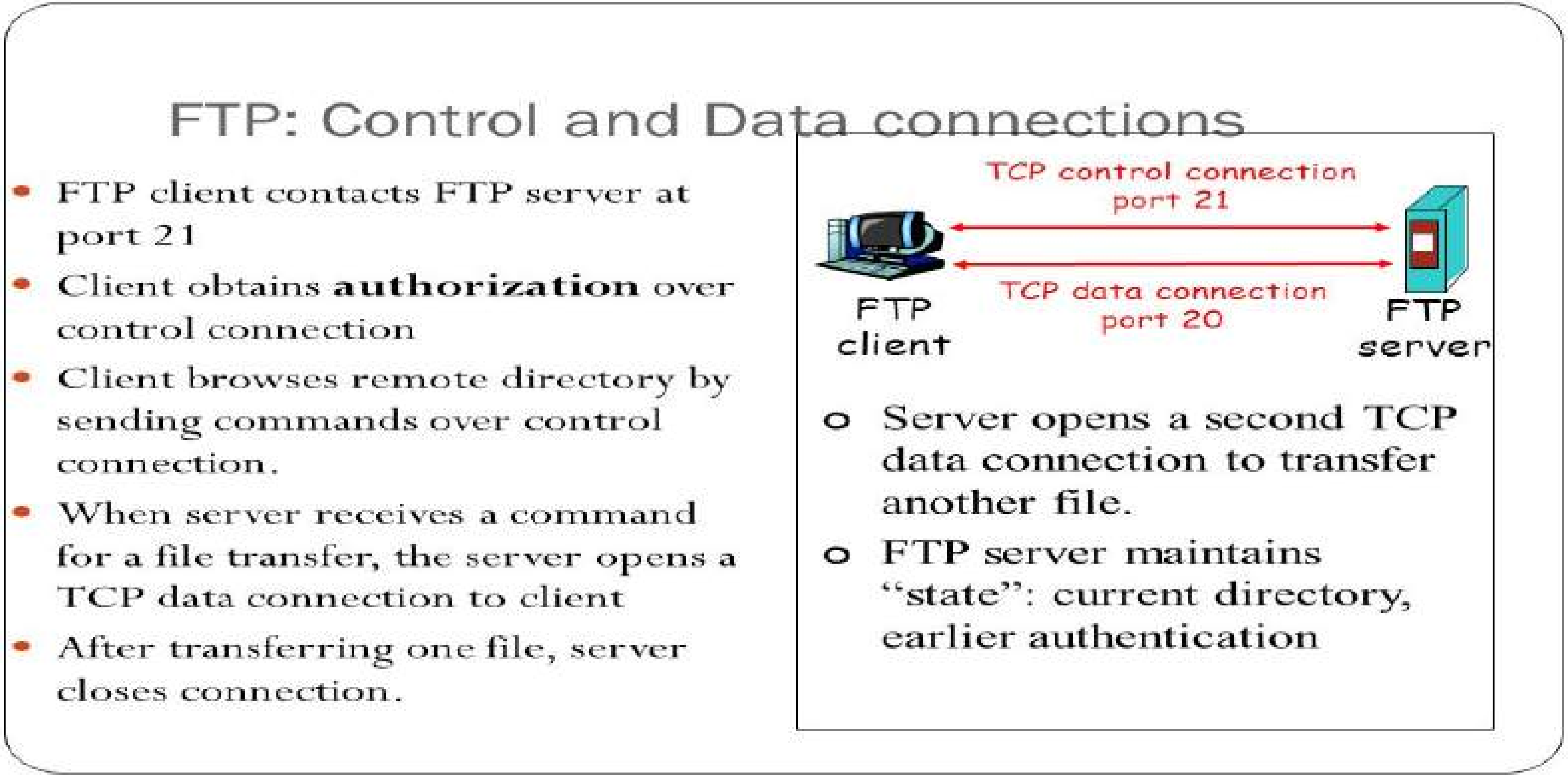


Figure 2.14 • FTP moves files between local and remote file systems

FTP



FTP: Control and Data connections

- FTP client contacts FTP server at port 21
- Client obtains **authorization** over control connection
- Client browses remote directory by sending commands over control connection.
- When server receives a command for a file transfer, the server opens a TCP data connection to client
- After transferring one file, server closes connection.

- o Server opens a second TCP data connection to transfer another file.
- o FTP server maintains "state": current directory, earlier authentication

FTP commands, responses

Sample commands:

- sent as ASCII text over control channel
- **USER** ***username***
- **PASS** ***password***
- **LIST** return list of file in current directory
- **RETR filename** retrieves (gets) file
- **STOR filename** stores (puts) file onto remote host

Sample return codes

- status code and phrase (as in HTTP)
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

SFTP

- SFTP stands for **Secure File Transfer Protocol**. It is a protocol which provides the secure channel, to transfer or copies the file from one host to another host or systems.
- SFTP establishes the control connection under SSH(secure shell) protocol and It is used in port no-22.
- It supports the full security and authentication functionality of SSH.
- SFTP has pretty much replaced legacy FTP as a file transfer protocol, and is quickly replacing FTP/S.
- SFTP also protects against password sniffing and man-in-the-middle attacks. It protects the integrity of the data using encryption and cryptographic hash functions, and authenticates both the server and the user.

FTP and SFTP

.NO	FTP	SFTP
1.	FTP stands for File Transfer Protocol.	SFTP stands for Secure File Transfer Protocol.
2.	In FTP, secure channel is not provided to transfer the files between the hosts.	In SFTP, secure channel is provided to transfer the files between the hosts.
3.	FTP (File transfer protocol) is a part of TCP/IP protocol.	Secure File Transfer Protocol is a SSH protocol.
4.	FTP (File transfer protocol) usually runs on port no-21.	SFTP (Secure File Transfer Protocol) runs on port no-22.
5.	FTP establishes the connection under TCP protocol.	SFTP establishes the control connection under SSH protocol.
6.	FTP do not encrypt the data before sending.	SFTP, data is encrypted before sending.

Email and Email Protocols

- It is a store and forward method of composing, sending, storing, and receiving messages over electronic communication systems.
- One of the most popular network services is electronic mail (email).
- Simple Mail Transfer Protocol (SMTP) is the standard mechanism for electronic mail in the internet; the first e-mail systems simply consisted by file transfer, protocols.

Basic Functions of email:

1. Composition 2. Transfer 3. Reporting 4. Displaying and 5. Disposition

- E-mail Protocols are set of rules that help the client to properly transmit the information to or from the mail server. Email protocols are **SMTP, POP,** and **IMAP**

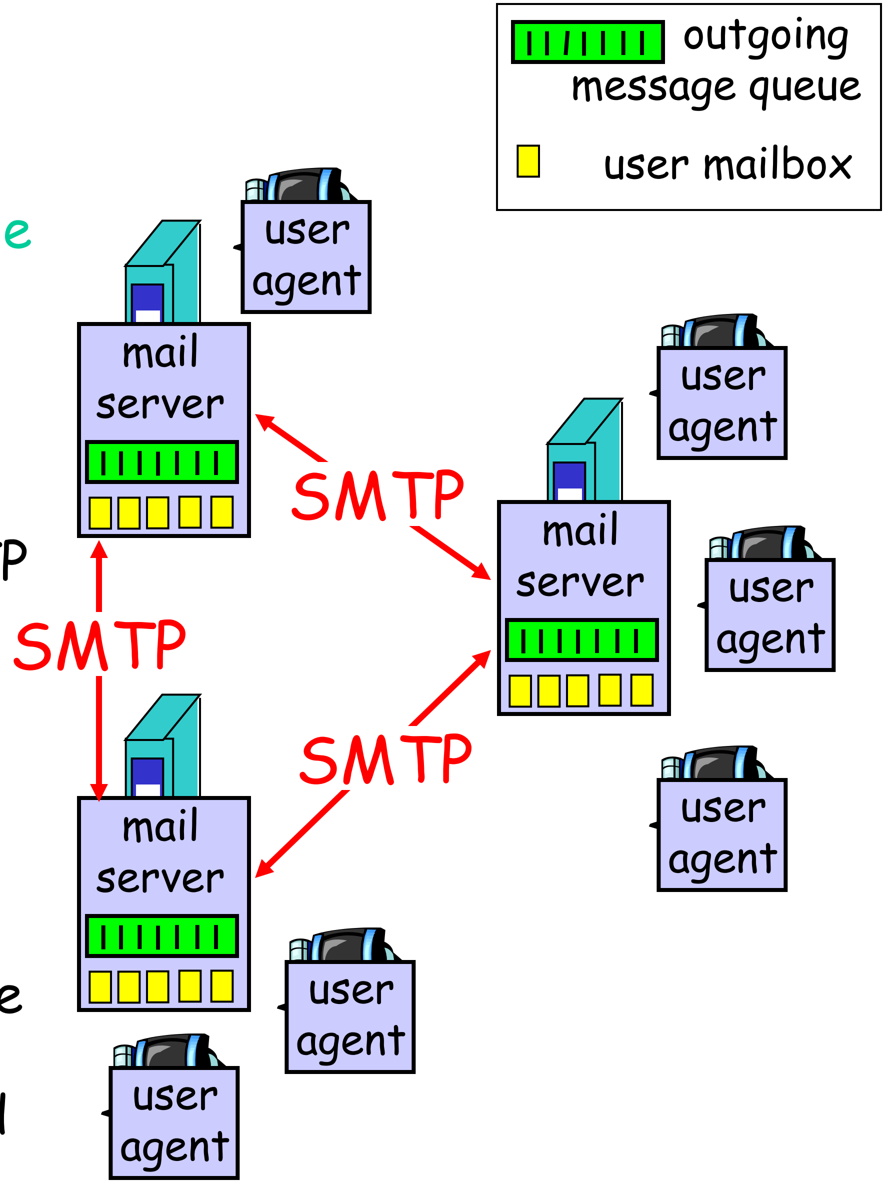
Electronic Mail

Three major components:

- user agents : : They allow the people to read and send e-mail
- mail servers :They move the messages from the source to the destination.
- simple mail transfer protocol: SMTP

- User Agent

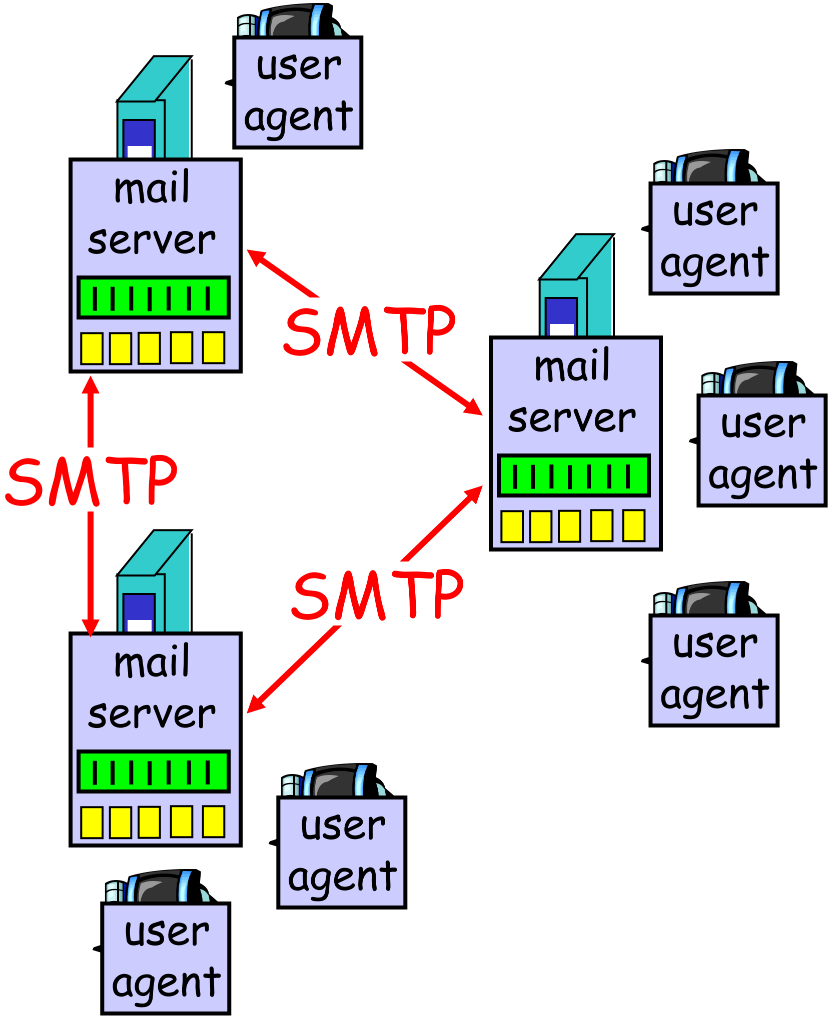
- a.k.a. "mail reader"
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Netscape Messenger
- outgoing, incoming messages stored on server



Electronic Mail: mail servers

Mail Servers

- mailbox contains incoming messages for user
- message queue of outgoing (to be sent) mail messages
- SMTP protocol between mail servers to send email messages
  - client: sending mail server
  - "server": receiving mail server



Scenario: Alice sends message to Bob

1) Alice uses UA to compose message and "to" `bob@someschool.edu`

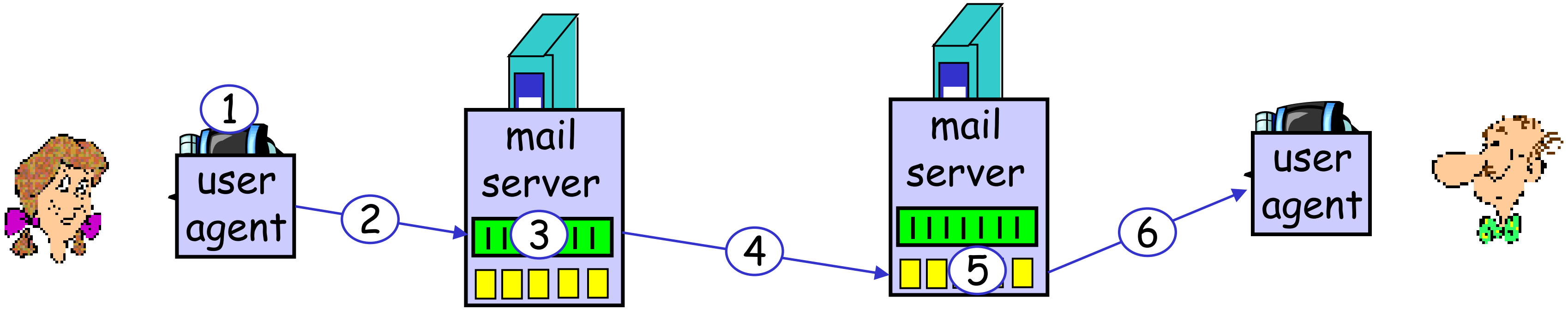
2) Alice's UA sends message to her mail server; message placed in message queue

3) Client side of SMTP opens TCP connection with Bob's mail server

4) SMTP client sends Alice's message over the TCP connection

5) Bob's mail server places the message in Bob's mailbox

6) Bob invokes his user agent to read message



SMTP

- **SMTP** stands for **Simple Mail Transfer Protocol**.
- *Simple Mail Transfer Protocol (SMTP) is the standard protocol for **sending emails** across the Internet*.
- It is a standard protocol used for sending e-mail efficiently and reliably over the internet.
- SMTP is application level protocol.
- SMTP is connection oriented protocol.
- SMTP is text based protocol.
- It handles exchange of messages between e-mail servers over TCP/IP network.
- Apart from transferring e-mail, SMPT also provides notification regarding incoming mail.
- When you send e-mail, your e-mail client sends it to your e-mail server which further contacts the recipient mail server using SMTP client.
- These SMTP commands specify the sender's and receiver's e-mail address, along with the message to be send.
- The exchange of commands between servers is carried out without intervention of any user.
- In case, message cannot be delivered, an error report is sent to the sender which makes SMTP a reliable protocol.
- By default, the SMTP protocol works on three ports:
- **Port 25** - this is the default SMTP non-encrypted port
- **Port 2525** - this port is opened on all SiteGround servers in case port 25 is filtered (by your ISP for example) and you want to send non-encrypted emails with SMTP
- **Port 465** - this is the port used if you want to send messages using SMTP securely

IMAP

- *The Internet Message Access Protocol (IMAP) is a mail protocol used for accessing email on a remote web server from a local client*.
- IMAP allows the client program to manipulate the e-mail message on the server without downloading them on the local computer.
- The e-mail is hold and maintained by the remote server.
- It enables us to take any action such as downloading, delete the mail without reading the mail.
- It enables us to create, manipulate and delete remote message folders called mail boxes.
- IMAP enables the users to search the e-mails.
- It allows concurrent access to multiple mailboxes on multiple mail servers.
- By default, the IMAP protocol works on two ports:
- **Port 143** - this is the default IMAP non-encrypted port
- **Port 993** - this is the port you need to use if you want to connect using IMAP securely

POP

- POP stands for Post Office Protocol.
- *Post Office Protocol version 3 (POP3) is a standard mail protocol used to **receive emails** from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline*
- It is generally used to support a single client.
- There are several versions of POP but the POP 3 is the current standard.
- POP is an application layer internet standard protocol.
- Since POP supports offline access to the messages, thus requires less internet usage time.
- POP does not allow search facility.
- In order to access the messaged, it is necessary to download them.
- It allows only one mailbox to be created on server.
- It is not suitable for accessing non mail data.
- By default, the POP3 protocol works on two ports:
- **Port 110** - this is the default POP3 non-encrypted port
- **Port 995** - this is the port you need to use if you want to connect using POP3 securely

POP and IMAP Comparison

S.N.	POP	IMAP
1	Generally used to support single client.	Designed to handle multiple clients.
2	Messages are accessed offline.	Messages are accessed online although it also supports offline mode.
3	POP does not allow search facility.	It offers ability to search emails.
4	All the messages have to be downloaded.	It allows selective transfer of messages to the client.
5	Only one mailbox can be created on the server.	Multiple mailboxes can be created on the server.
6	Not suitable for accessing non-mail data.	Suitable for accessing non-mail data i.e. attachment.

7	POP commands are generally abbreviated into codes of three or four letters. Eg. STAT.	IMAP commands are not abbreviated, they are full. Eg. STATUS.
8	It requires minimum use of server resources.	Clients are totally dependent on server.
9	Mails once downloaded cannot be accessed from some other location.	Allows mails to be accessed from multiple locations.
10	The e-mails are not downloaded automatically.	Users can view the headings and sender of e-mails and then decide to download.
10	POP requires less internet usage time.	IMAP requires more internet usage time.

Proxy Application Server

- Proxy server is the server that acts as a intermediate between requests from clients seeking resources from other servers.
- A client connects to the proxy server to request for a service.
- The proxy server evaluates the request and simplify its complexity.
- **Proxy server** is an intermediary server between client and the internet.

Proxy servers offers the following basic functionalities:

- Firewall and network data filtering.
- Network connection sharing
- Data caching
- Proxy servers allow to hide, secrete and make your network id anonymous by hiding your IP address.

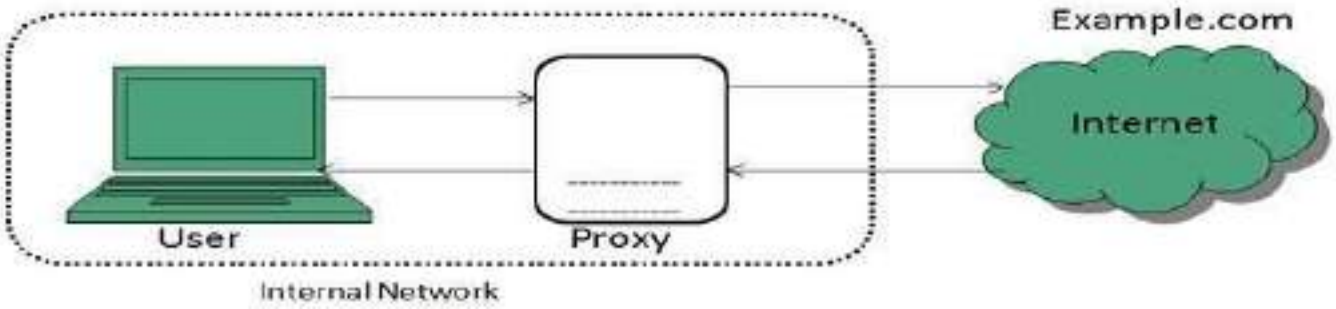
Following are the reasons to use proxy servers:

- Monitoring and Filtering
- Improving performance
- Translation
- Accessing services anonymously
- Security
- Monitoring and Filtering

Proxy servers allow us to do several kind of filtering such as:

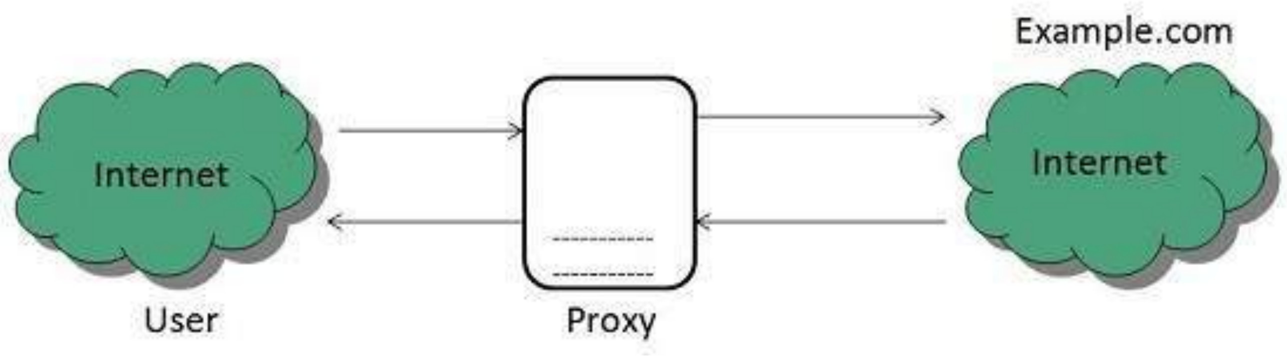
- Content Filtering
- Filtering encrypted data
- Bypass filters
- Logging and eavesdropping

Forward Proxy



- In this the client requests its internal network server to forward to the internet.
- A Forward proxy provides internal clients access through a firewall to resources on the Internet. This service is often provided as part of a larger intranet security strategy.
- Forward proxy allows clients to access resources outside of the firewall without compromising the integrity of the private network.
- A forward proxy can be configured to keep copies of content within their local cache. Subsequent requests for that content can then be serviced from the local cache rather than obtaining the content from the origin server. Caching increases performance by decreasing the time involved in traversing the network.
- A forward proxy is best used to filter content, increase performance, and log user accesses

Open Proxy

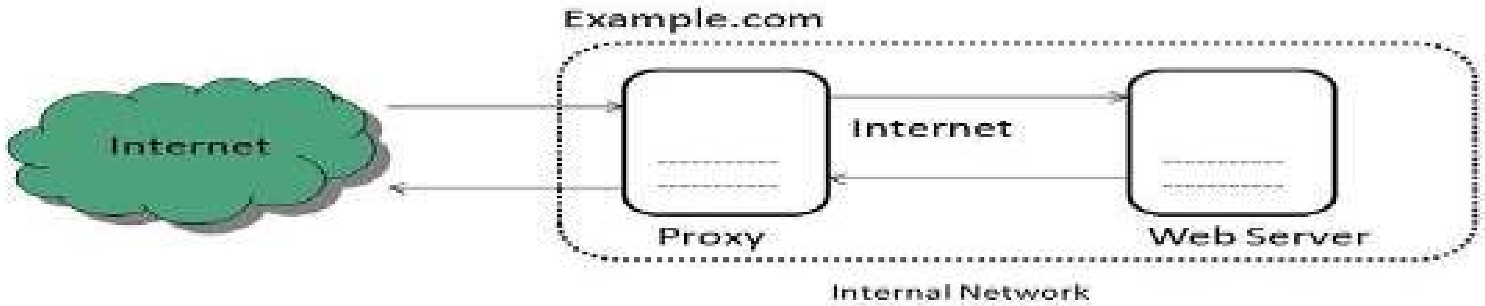


An open proxy forwarding requests from and to anywhere on the Internet..

Anonymous Proxy – This server reveals its identity as a server but does not disclose the initial IP address. Though this server can be discovered easily it can be beneficial for some users as it hides the Internet Protocol address.

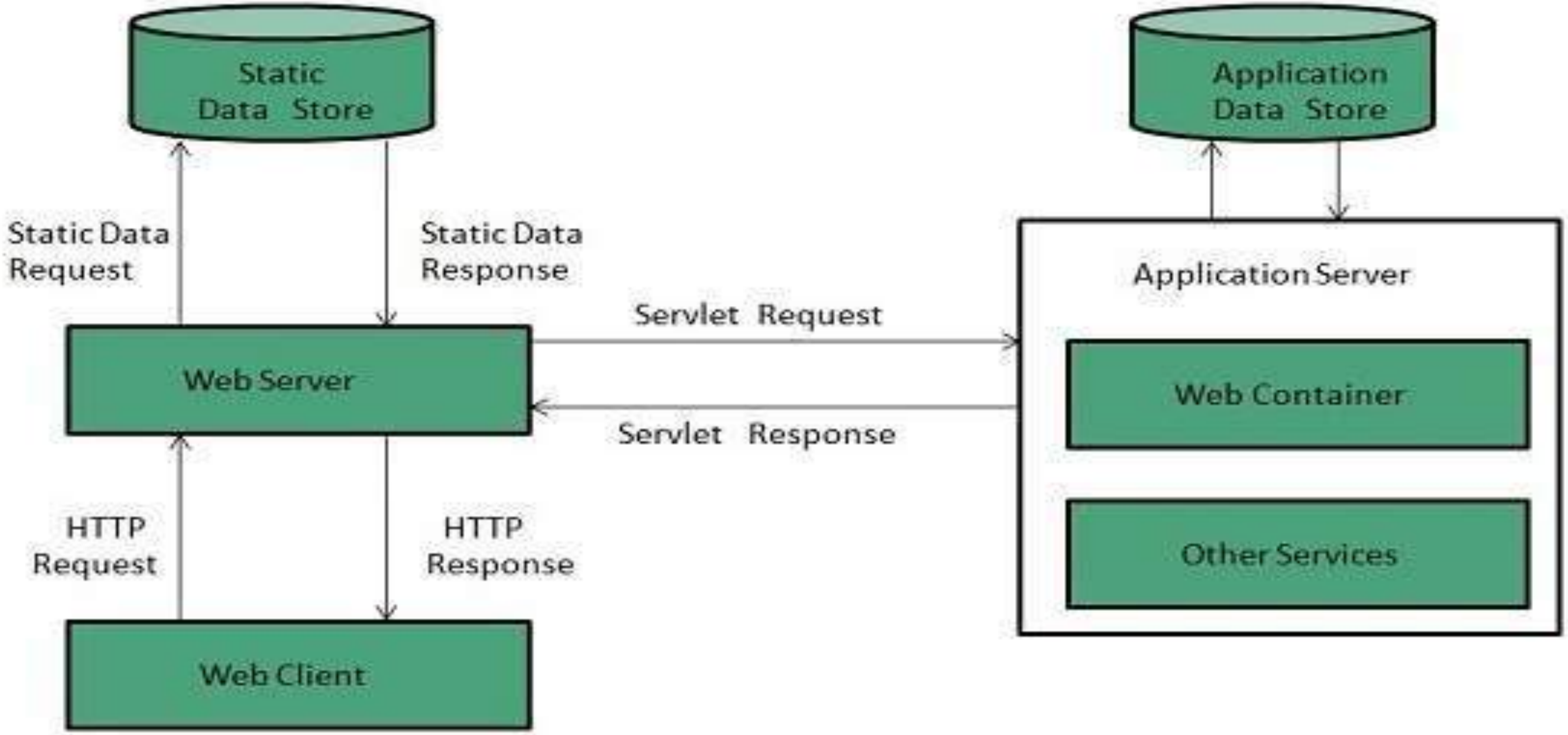
Transparent Proxy – This proxy server again identifies itself, and with the support of HTTP headers, the first IP address can be viewed. The main benefit of using this sort of server is its ability to cache the websites.

Reverse Proxy



- In this the requests are forwarded to one or more proxy servers and the response from the proxy server is retrieved as if it came directly from the original Server.
- A proxy server can also provide external clients with access to internal resources the reside behind the corporate firewall. When a proxy server is used to handle connections into a private network, the process is called reverse proxying.
- The term “reverse” refers to the fact that traffic flows in the opposite direction from normal proxy traffic flow.
- We can use a reverse proxy to load balance across multiple servers, provide failover capabilities, and provide access to corporate resources in a safe and secure manner.

Web Application Server



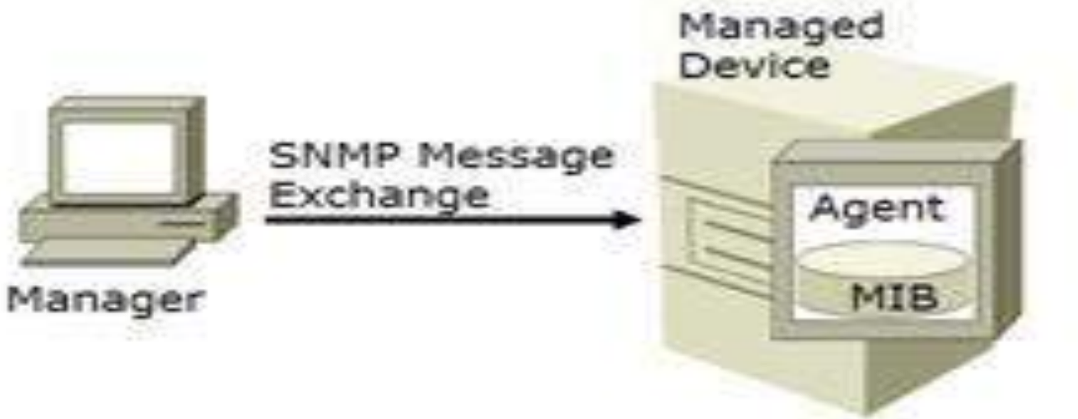
- A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients.
- Dedicated computers and appliances may be referred to as Web servers as well.
- The process is an example of the client/server model. All computers that host Web sites must have Web server programs. Leading Web servers include Apache (the most widely-installed Web server), Microsoft's Internet Information Server (IIS) and nginx (pronounced *engine X*) from NGNIX. Other Web servers include Novell's NetWare server, Google Web Server (GWS) and IBM's family of Domino servers.
- Web servers often come as part of a larger package of Internet- and intranet-related programs for serving email, downloading requests for File Transfer Protocol(FTP) files, and building and publishing Web pages.
- Considerations in choosing a Web server include how well it works with the operating system and other servers, its ability to handle server-side programming, security characteristics, and the particular publishing, search engine and site building tools that come with it.

SNMP

- SNMP stands for simple network management protocol.
- It is a way that servers can share information about their current state, and also a channel through which an administer can modify pre-defined values.
- While the protocol itself is very simple, the structure of programs that implement SNMP can be very complex.
- SNMP is a set of protocols that describes management data and the protocols for exchanging that data between heterogeneous systems.
- The protocols include both the description of the management data, defined in the Management Information Base (MIB), and the operations for exchanging or changing that information.
- By implementing common protocols, management data can be exchanged between different platforms with relative ease.
- SNMP defines an architecture that consists of:
- Network management applications
- Network management agents and subagents
- Network elements, such as hosts and gateways

SNMP Overview

- Manager:
  - Polls agents on the network
  - Correlates and displays information
- SNMP:
  - Supports message exchange
  - Runs on IP
- Agent:
  - Collects and stores information
  - Responds to manager requests for information
  - Generates traps
- MIB:
  - Database of objects (information variables)
  - Read and write community strings for controlling access

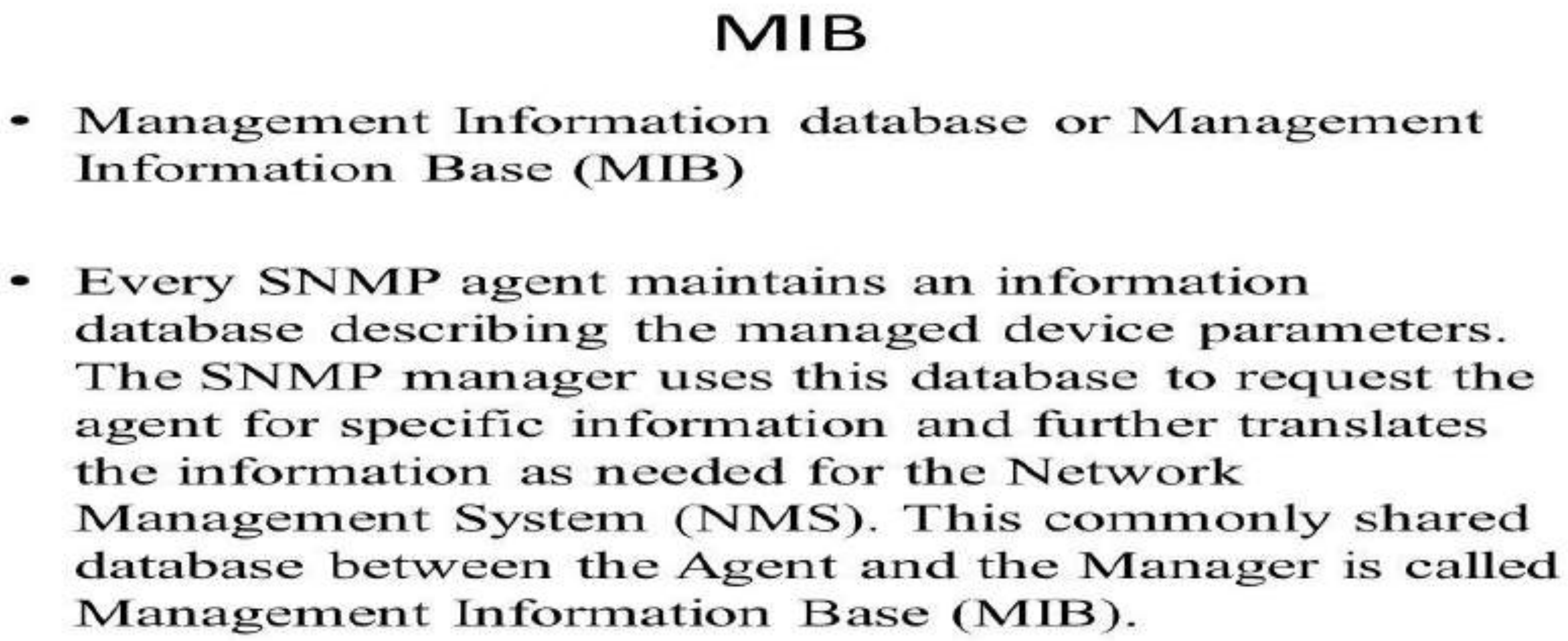


CertificationKits

MIB

- A management information base (MIB) is a formal description of a set of network Objects that can be managed using the Simple Network Management Protocol (SNMP).
- The format of the MIB is defined as part of the SNMP. (All other MIBs are extensions of this basic management information base.) MIB-I refers to the initial MIB definition; MIB-II refers to the current definition. SNMPv2 includes MIB-II and adds some new objects.
- There are MIBs (or more accurately, MIB extensions) for each set of related network entities that can be managed.
- For example, there are MIB definitions specified in the form of Requests for Comments (RFCs) for AppleTalk, domain name system (DNS), Fiber Distributed-Data Interface, and RS-232C network objects. Product developers can create and register new MIB extensions.
- Companies that have created MIB extensions for their sets of products include Cisco, Fore, IBM, Novell, QMS, and Onramp.

MIB



MIB

- Management Information database or Management Information Base (MIB)

- Every SNMP agent maintains an information database describing the managed device parameters. The SNMP manager uses this database to request the agent for specific information and further translates the information as needed for the Network Management System (NMS). This commonly shared database between the Agent and the Manager is called Management Information Base (MIB).