Unit 6 – Application Layer

Agenda

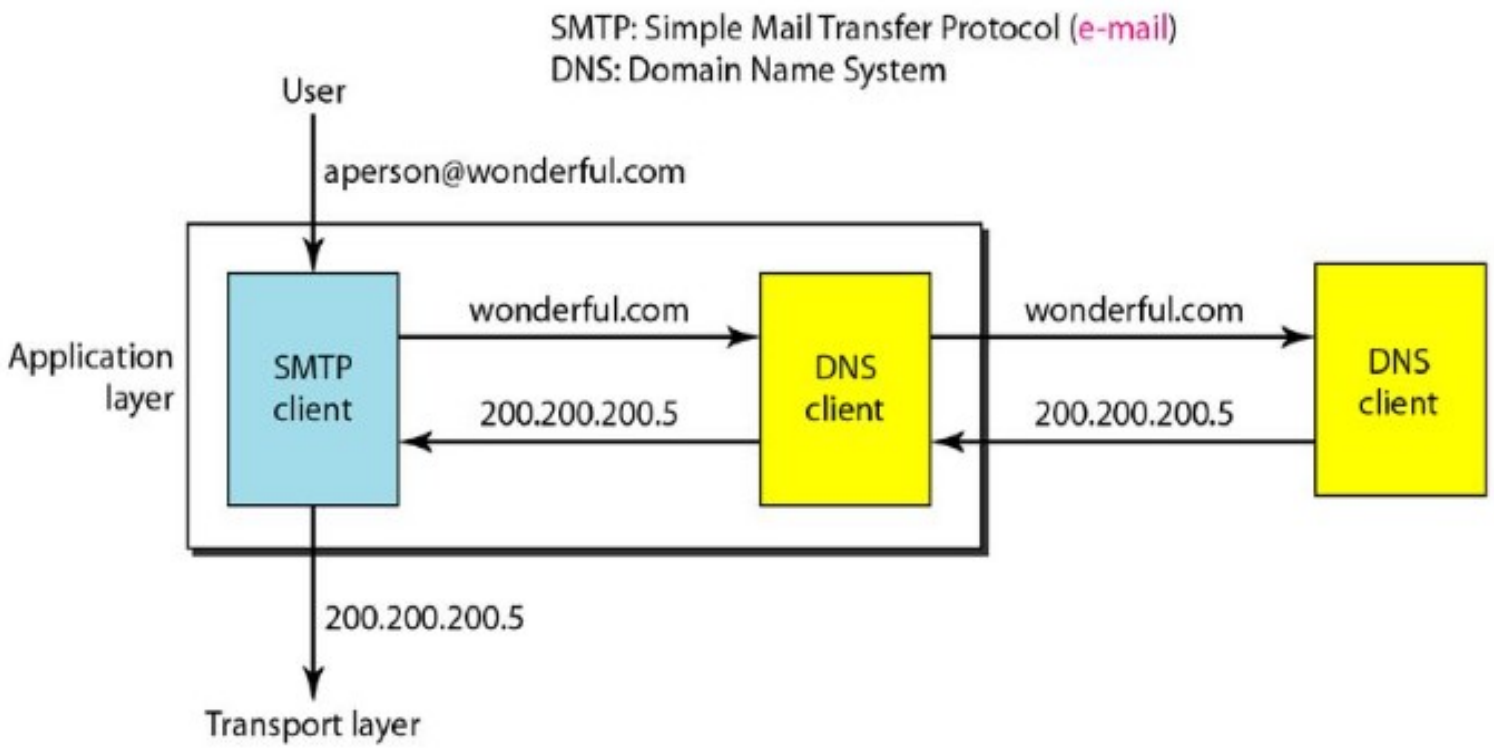
- Name Space: Flat Name Space,
- Hierarchical Name Space
- Domain Name Space,
- Distribution of Name Space: Hierarchy of Name Servers,
- Zone, Root Server, Primary and Secondary Servers,
- DNS in the Internet: Generic Domains, Country Domains,
- Inverse Domain,
- Resolution: Resolver, Mapping Names to Addresses, Mapping Address to Names,
- Recursive Resolution, Iterative Resolution, Caching,
- DNS Messages and Types of Records
- Introduction to Telnet, SMTP, FTP, WWW.

Introduction

- The client/server programs can be divided into two categories: those that can be directly used by the user, such as e-mail, and those that support other application programs.
- The Domain Name System (DNS) is a supporting program that is used by other programs such as e-mail.
- Figure 25.1 shows an example of how a DNS client/server program can support an e-mail program to find the IP address of an e-mail recipient.
- A user of an e-mail program may know the e-mail address of the recipient; however, the IP protocol needs the IP address.
- The DNS client program sends a request to a DNS server to map the e-mail address to the corresponding IP address.

Introduction

Figure 25.1 ***Example of using the DNS service***



What is the need of DNS?

- People prefer to use names than numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name.
- When the Internet was small, mapping was done by using a host file. The host file had only two columns: name and address.
- Every host could store the host file on its disk and update it periodically from a master host file.
- When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.
- Today, however, it is impossible to have one single host file to relate every address with a name and vice versa.
- The host file would be too large to store in every host.

What is the need of DNS?

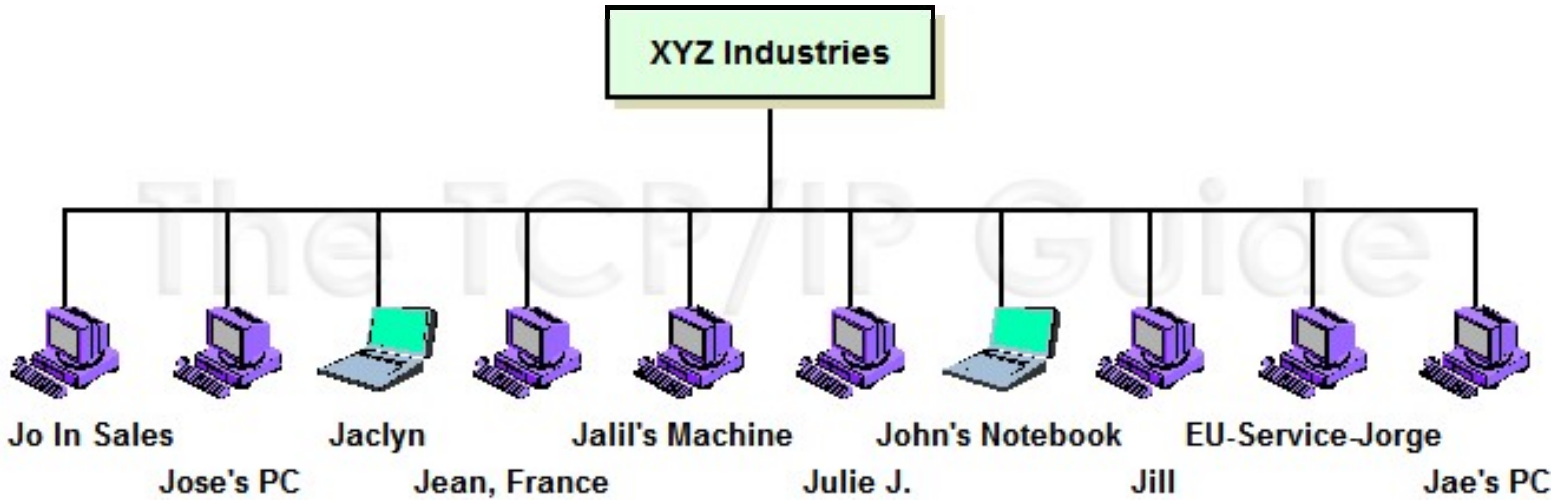
- One solution would be to store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping.
- But we know that this would create a huge amount of traffic on the Internet.
- Another solution is to divide this huge amount of information into smaller parts and store each part on a different computer.
- This method is used by the Domain Name System (DNS).

Namespace

- A namespace is a context within which the names of all objects must be unambiguously resolvable.
- For example, the internet is a single DNS name space, within which all network devices with a DNS name can be resolved to a particular address (for example, www.microsoft.com resolves to 207.46. 131.13).
- A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

Namespace

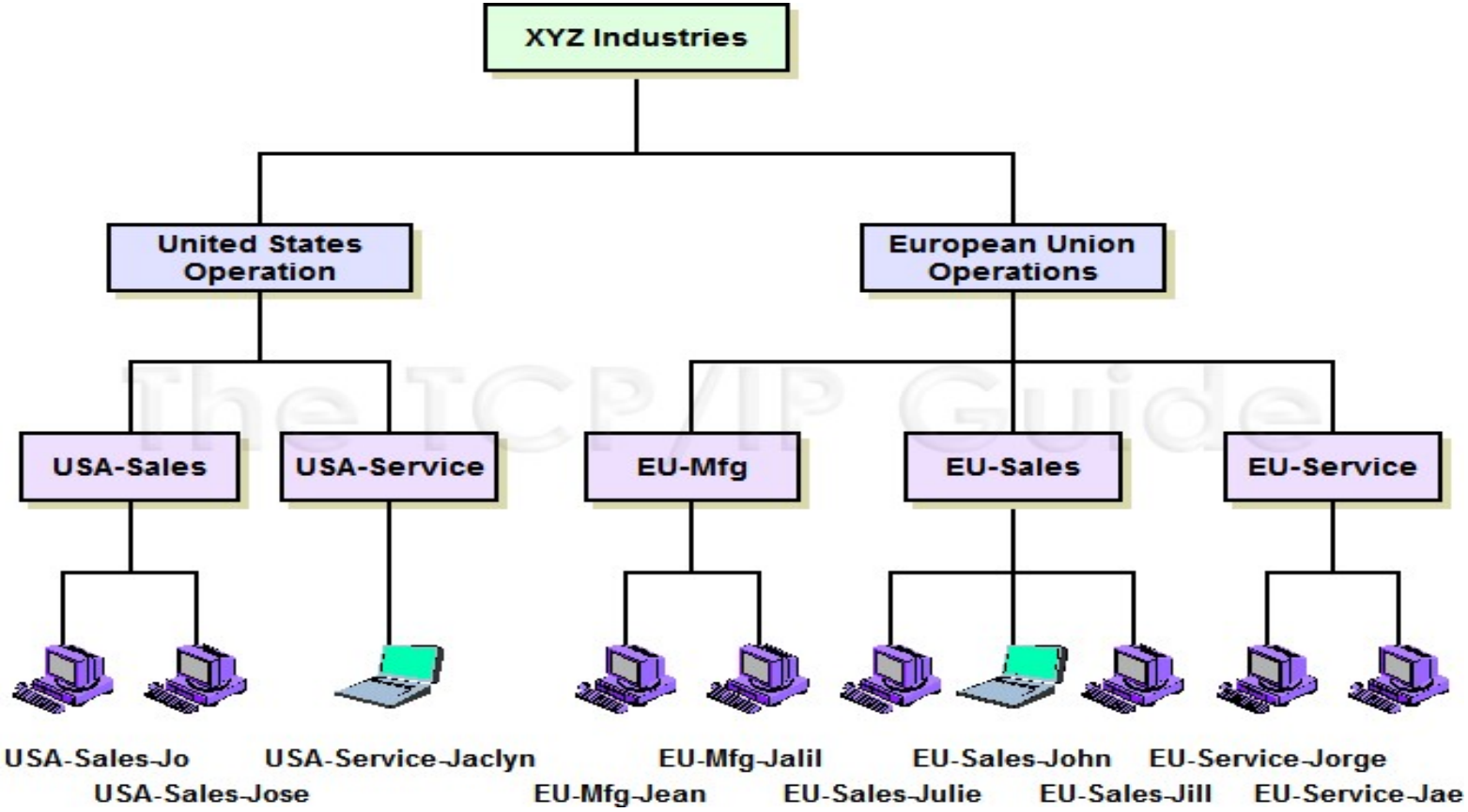
- Flat Name Space
- A name in this space is a sequence of characters, whole label without any internal structure. There is no clear relationship between any name and any other name.
- The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.



Flat Name Architecture (Flat Name Space)

Namespace

- Hierarchical Name Space



Hierarchical Name Architecture (Structured Name Space)

Namespace

- Hierarchical Name Space
- Each name is made of several parts. The first part can define the nature of the organization, the second part - the name of an organization, the third part - departments in the organization, and so on.
- A central authority can assign the part of the name that defines the nature and name of the organization.
- The responsibility of the rest of the name can be given to the organization itself. The organization can add suffixes (or prefixes) to the name to define its host or resources.
- Even if part of an address is the, same, the whole address is different.

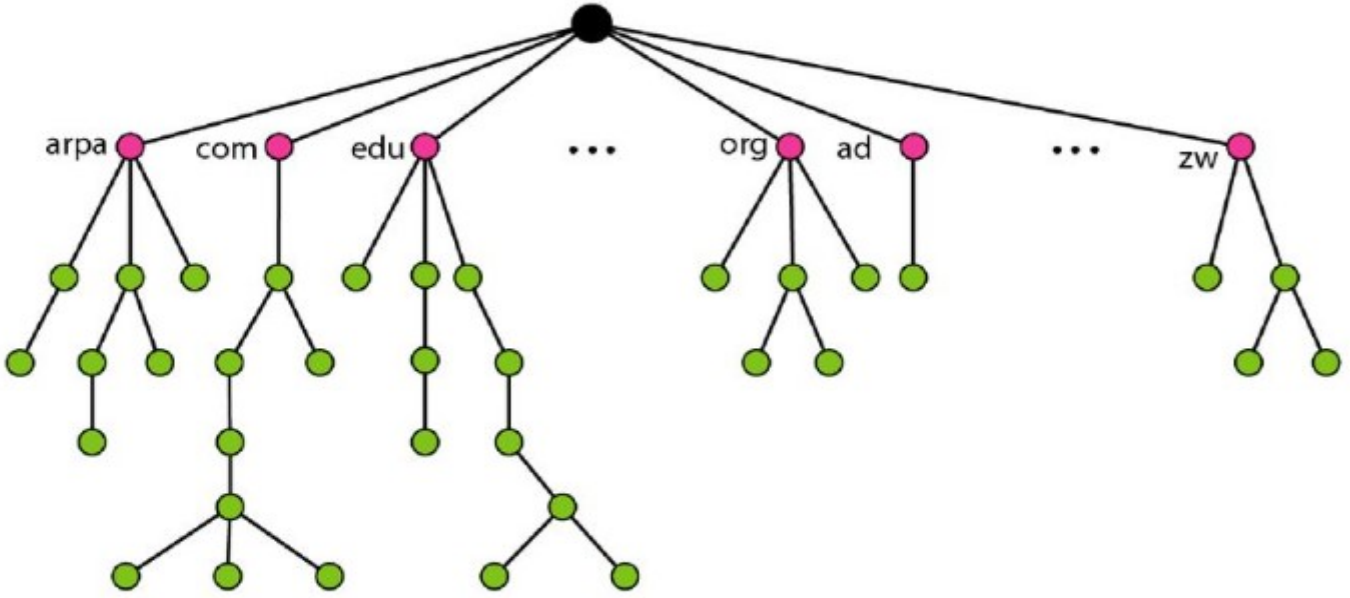
Namespace

- Example
- For example, assume two colleges and a company call one of their computers challenger.
- The first college is given a name by the central authority such as jhda.edu, the second college is given the name berkeley.edu, and the company is given the name smart. com
- When these organizations add the name challenger to the name they have already been given, the end result is three distinguishable names: challenger.jhda.edu, challenger.berkeley.edu, and challenger.smart.com.
- The names are unique without the need for assignment by a central authority.

Domain Name Space

- To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels: level 0 (root) to level 127 (see Figure 25.2)

Figure 25.2 *Domain name space*

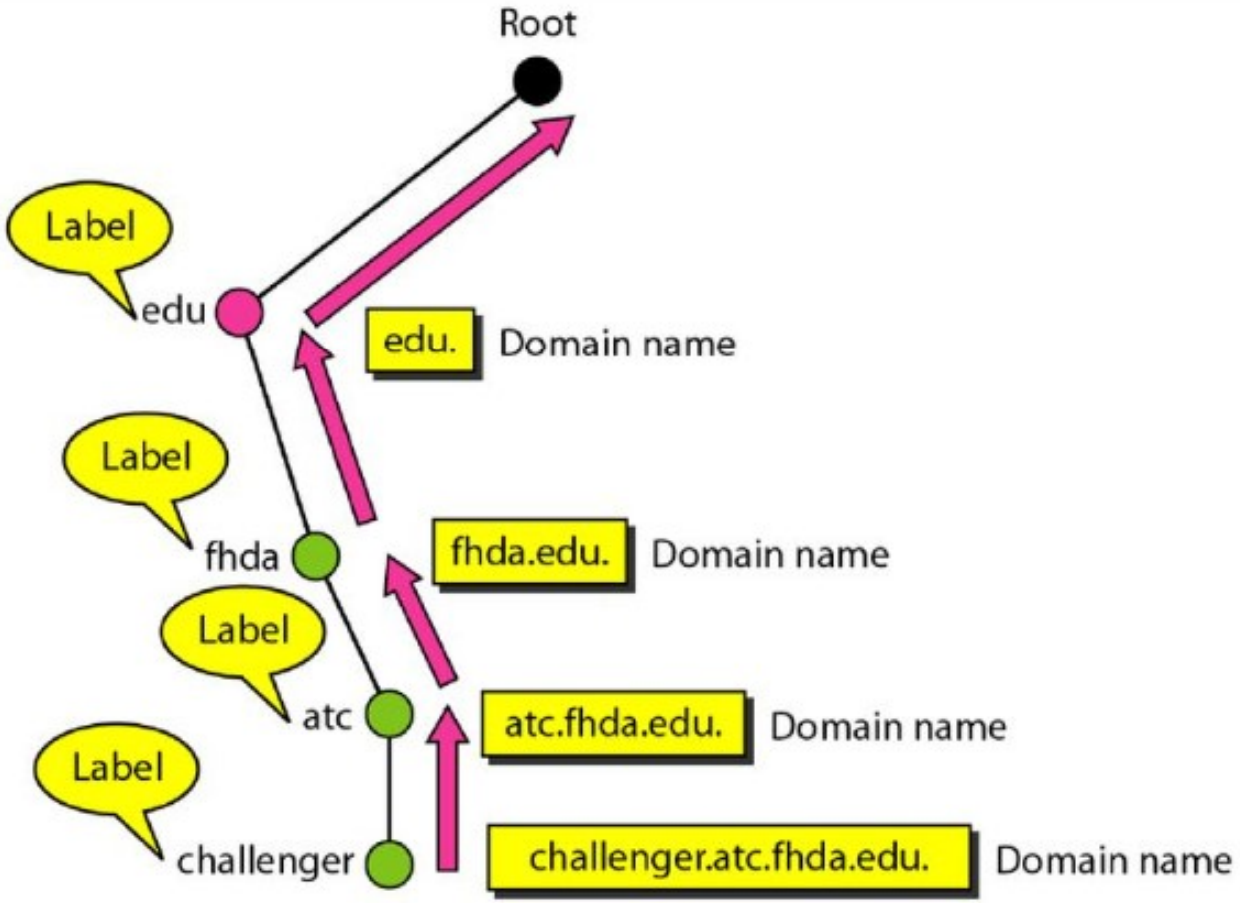


Domain Name Space

- Label
- Each node in the tree has a label, which is a string with a maximum of 63 characters.
- The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.
- Domain Name
- Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.).
- The domain names are always read from the node up to the root. The last label is the label of the root (null).
- This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.
- Figure 25.3 shows some domain names

Domain Name Space

Figure 25.3 *Domain names and labels*



Domain Name Space

- Fully Qualified Domain Name (FQDN)
- If a label is terminated by a null string, it is called a fully qualified domain name (FQDN). It is a domain name that contains the full name of a host.
- It contains all labels, from the most specific to the most general, that uniquely define the name of the host.
- For example, the domain name challenger.ate.tbda.edu. is the FQDN of a computer named challenger installed at the Advanced Technology Center (ATC) at De Anza College.
- A DNS server can only match an FQDN to an address.
- Note that the name must end with a null label, but because null means nothing, the label ends with a dot (.).

Figure 25.4 shows some FQDNs and PQDNs.

Figure 25.4 *FQDN and PQDN*

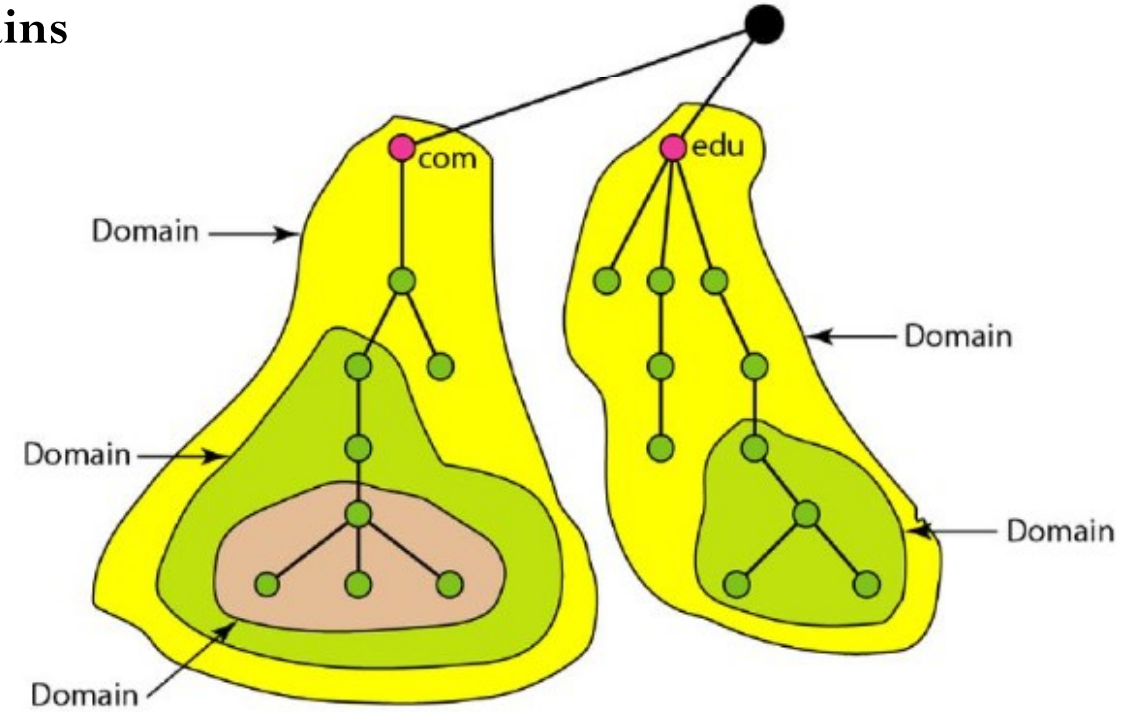
Domain Name Space

- Partially Qualified Domain Name (PQDN)
- If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN).
- A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client.
- Here the resolver can supply the missing part, called the suffix, to create an FQDN.
- For example, if a user at the jhda.edu. site wants to get the IP address of the challenger computer, he or she can define the partial name challenger
- The DNS client adds the suffix atc.jhda.edu. before passing the address to the DNS server. The DNS client normally holds a list of suffixes.
- The following can be the list of suffixes at De Anza College. The null suffix defines nothing. This suffix is added when the user defines an FQDN.

Domain Name Space

- Domain
- A domain is a subtree of the domain name space. The name of the domain is the domain name of the node at the top of the subtree.
- Figure 25.5 shows some domains. Note that a domain may itself be divided into domains (or subdomains as they are sometimes called) called

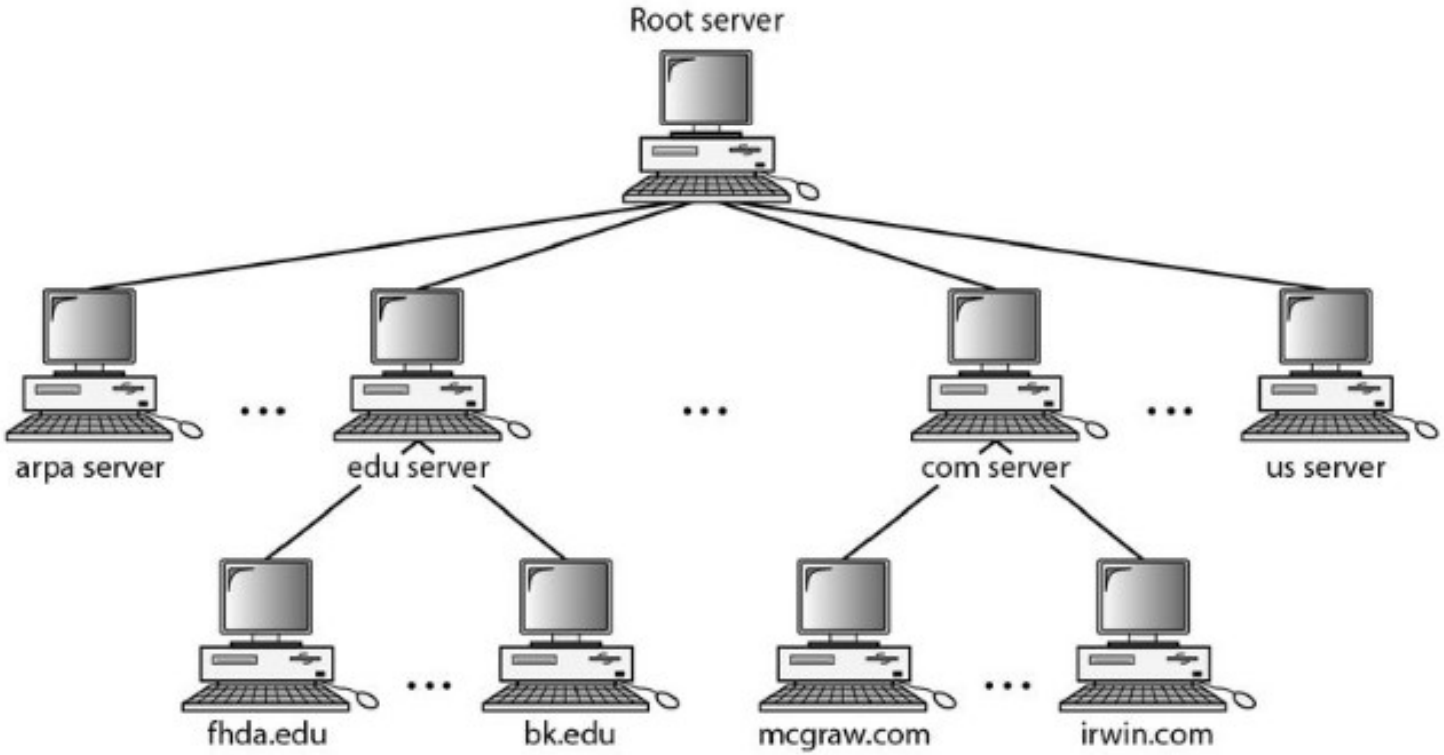
Figure 25.5 Domains



Distribution of Name Space

- The information contained in the domain name space cannot be stored in just one computer because responding to requests from all over the world places a heavy load on the system. It is not unreliable because any failure makes the data inaccessible.

Figure 25.6 Hierarchy of Name Servers



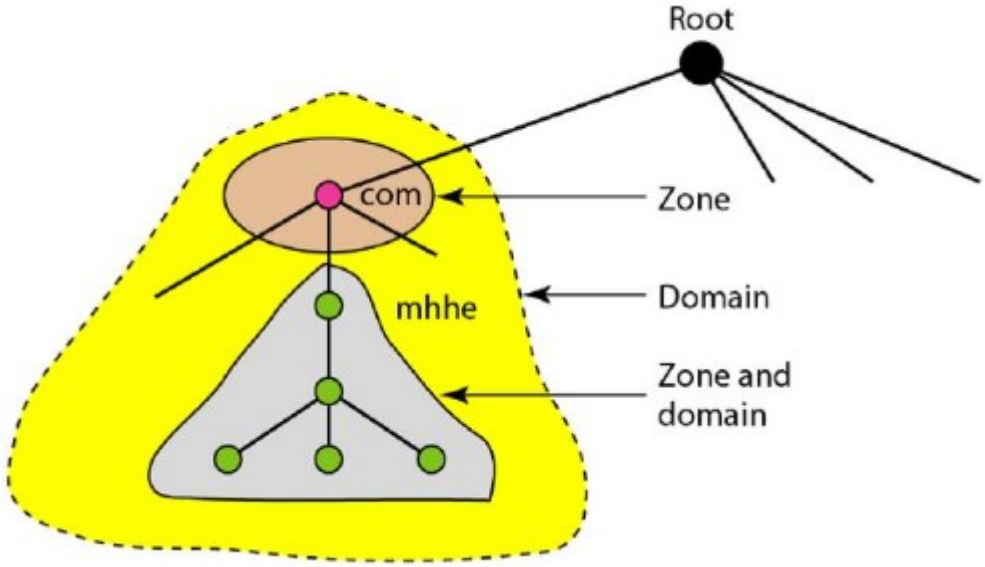
Distribution of Name Space

- Hierarchy of Name Servers
- One way to do this is to divide the whole space into many domains based on the first level.
- In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes.
- Because a domain created could be very large, DNS allows domains to be divided further into smaller domains (subdomains).
- Each server can be responsible (authoritative) for either a large or a small domain. In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names (see Figure 25.6).

Distribution of Name Space

- Zone
- The complete domain name hierarchy cannot be stored in one server so it is divided among many servers.
- What a server is responsible for or has authority over is called a zone.
- The server makes a database called a zone file and keeps all the information for every node under that domain.
- If a server divides its domain into subdomains and delegates part of its authority to other servers, domain and zone refer to different things otherwise they are same.

Figure 25.7 Zones and Domains



Distribution of Name Space

- Root Server
- A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
- There are several root servers, each covering the whole domain name space. The servers are distributed all around the world.

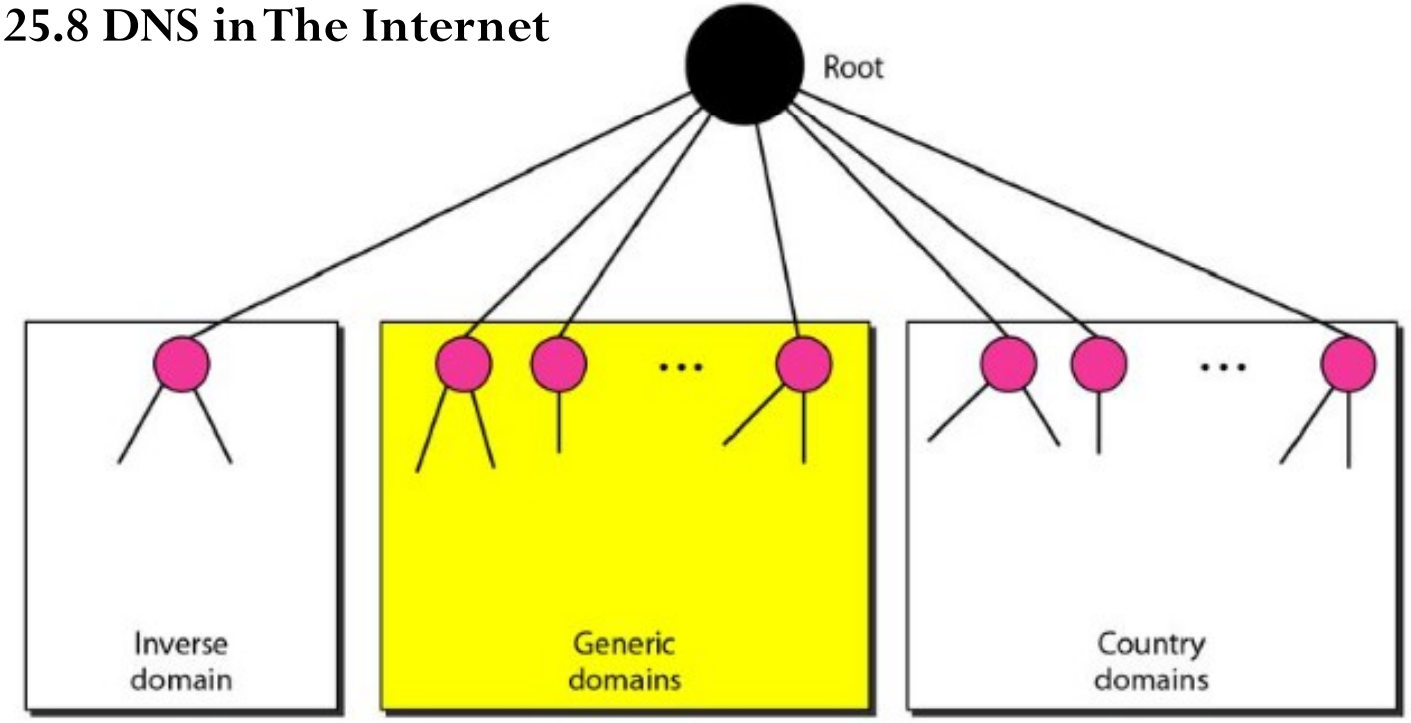
Distribution of Name Space

- Primary and Secondary Servers
- DNS defines two types of servers: primary and secondary.
- A primary server stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file.
- It stores the zone file on a local disk.
- A secondary server transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk.
- The secondary server neither creates nor updates the zone files.
- If updaing is required, it must be done by the primary server, which sends the updated version to the secondary.
- The primary and secondary servers are both authoritative for the zones they serve

DNS IN THE INTERNET

- DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain (see Figure 25.8).

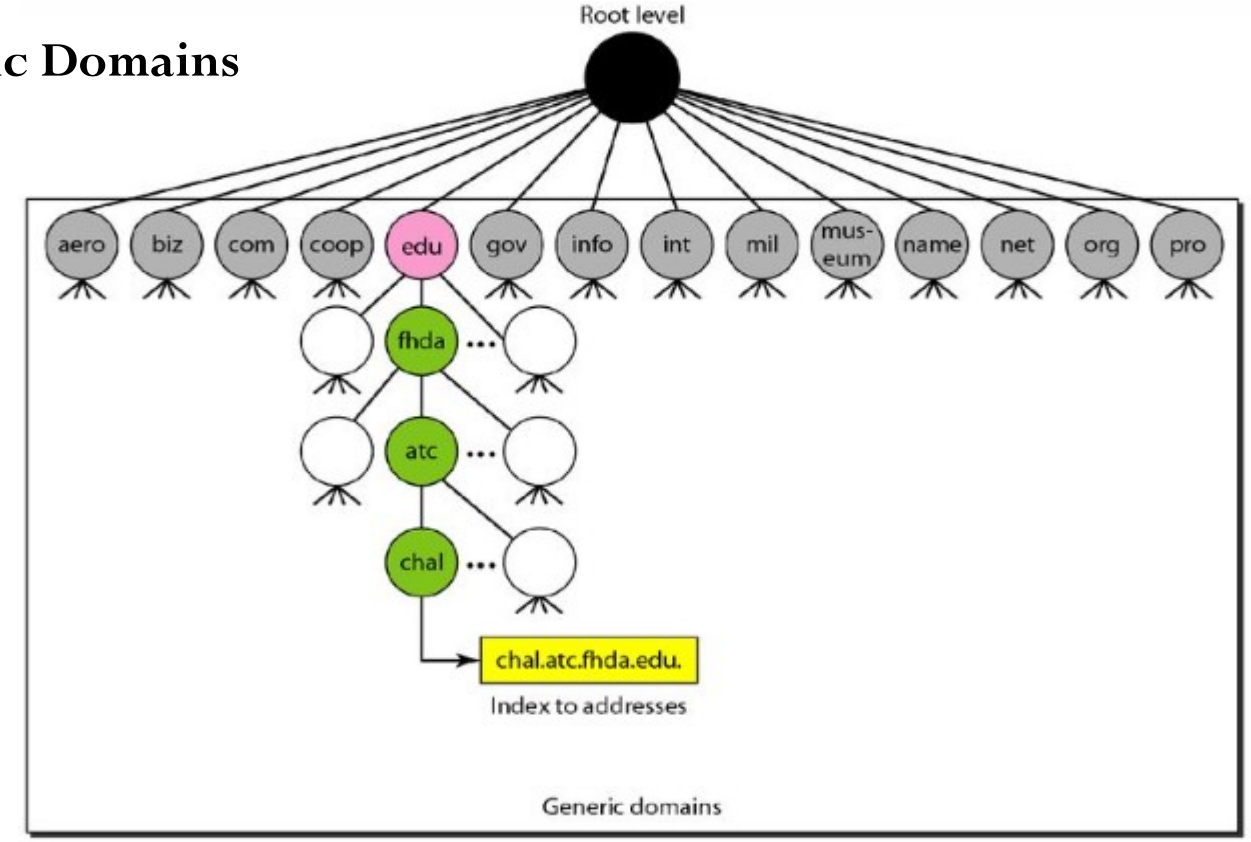
Figure 25.8 DNS in The Internet



DNS IN THE INTERNET

- Generic Domains
- The generic domains define registered hosts according to their generic behavior.
- Each node in the tree defines a domain, which is an index to the domain name space database (see Figure 25.9).

Figure 25.9 Generic Domains



DNS IN THE INTERNET

Table 25.1 *Generic domain labels*

Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

DNS IN THE INTERNET

- Country Domains
- The country domains section uses two-character country abbreviations (e.g., us for United States).
- Second labels can be organizational, or they can be more specific, national designations.
- The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.).
- Figure 25.10 shows that the address anza.cup.ca.us can be translated to De Anza College in Cupertino, California, in the United States.

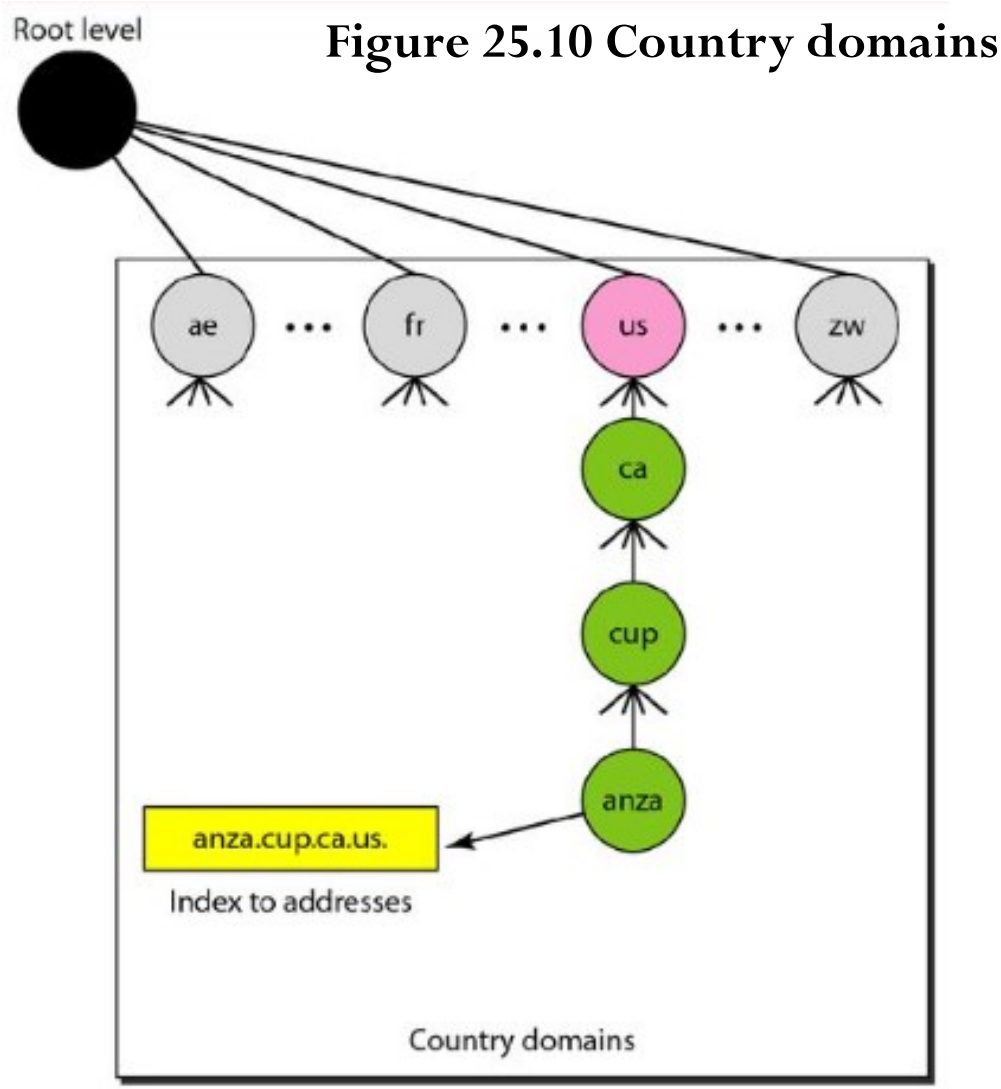


Figure 25.10 Country domains

DNS IN THE INTERNET

- Inverse Domain
- The inverse domain is used to map an address to a name, for example, when a server has received a request from a client to do a task.
- The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list. This type of query is called an inverse or pointer (PTR) query.
- To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called arpa (for historical reasons).
- The second level is also one single node named in-addr (for inverse address). The rest of the domain defines IP addresses.

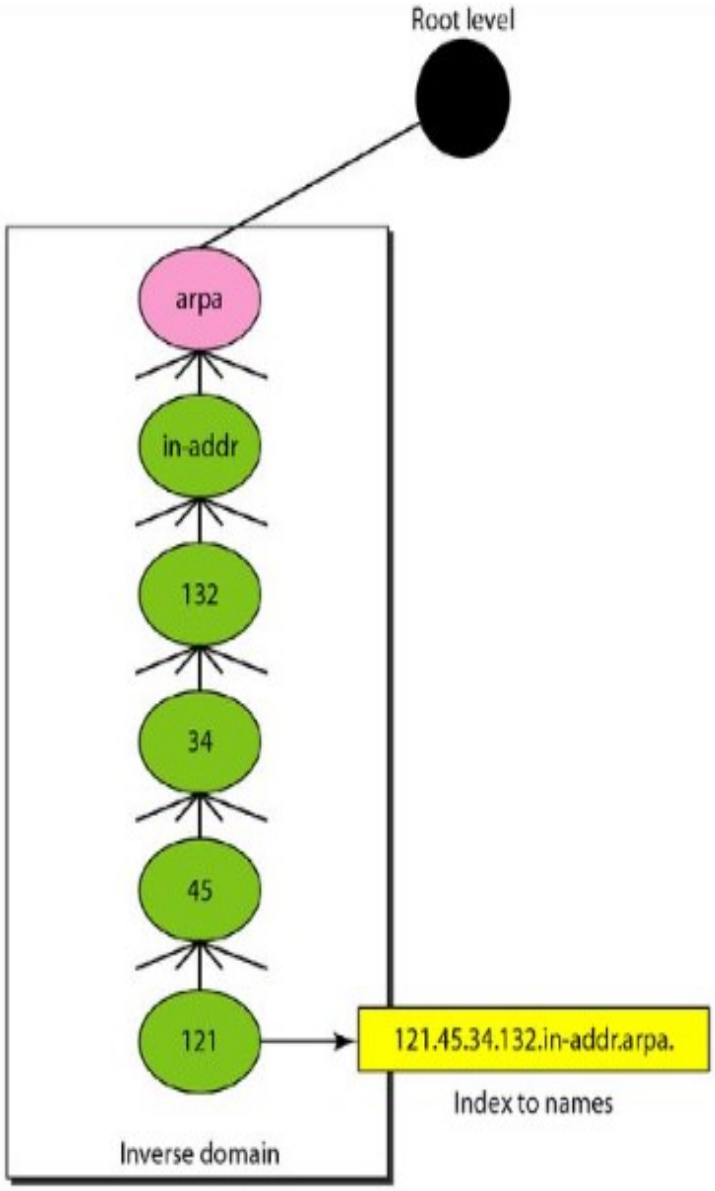


Figure 25.11 Inverse Domain

RESOLUTION

- Resolver DNS is designed as a client/server application.
- A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver.
- The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.

RESOLUTION

- Mapping Names to Addresses
- Most of the time, the resolver gives a domain name to the server and asks for the corresponding address.
- In this case, the server checks the generic domains or the country domains to find the mapping.
- If the domain name is from the generic domains section, the resolver receives a domain name such as "chal.atc.jhda.edu.".
- The query is sent by the resolver to the local DNS server for resolution. If the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly.
- If the domain name is from the country domains section, the resolver receives a domain name such as "ch.jhda.cu.ca.us.".

RESOLUTION

- Mapping Addresses to Names
- A client can send an IP address to a server to be mapped to a domain name. As mentioned before, this is called a PTR query.
- To answer queries of this kind, DNS uses the inverse domain. However, in the request, the IP address is reversed and the two labels in-addr and arpa are appended to create a domain acceptable by the inverse domain section.
- For example, if the resolver receives the IF address 132.34.45.121, the resolver first inverts the address and then adds the two labels before sending.
- The domain name sent is "121.45.34.132.in-addr.arpa." which is received by the local DNS and resolved.

RESOLUTION

- Recursive Resolution
- The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer.
- If the server is the authority for the domain name, it checks its database and responds. If the server is not the author city, it sends the request to another server (the parent usually) and waits for the response.
- If the parent is the authority, it responds; otherwise, it sends the query to yet another server.
- When the query is finally resolved, the response travels back until it finally reaches the requesting client. This is called recursive resolution and is shown in Figure 25.12.

RESOLUTION

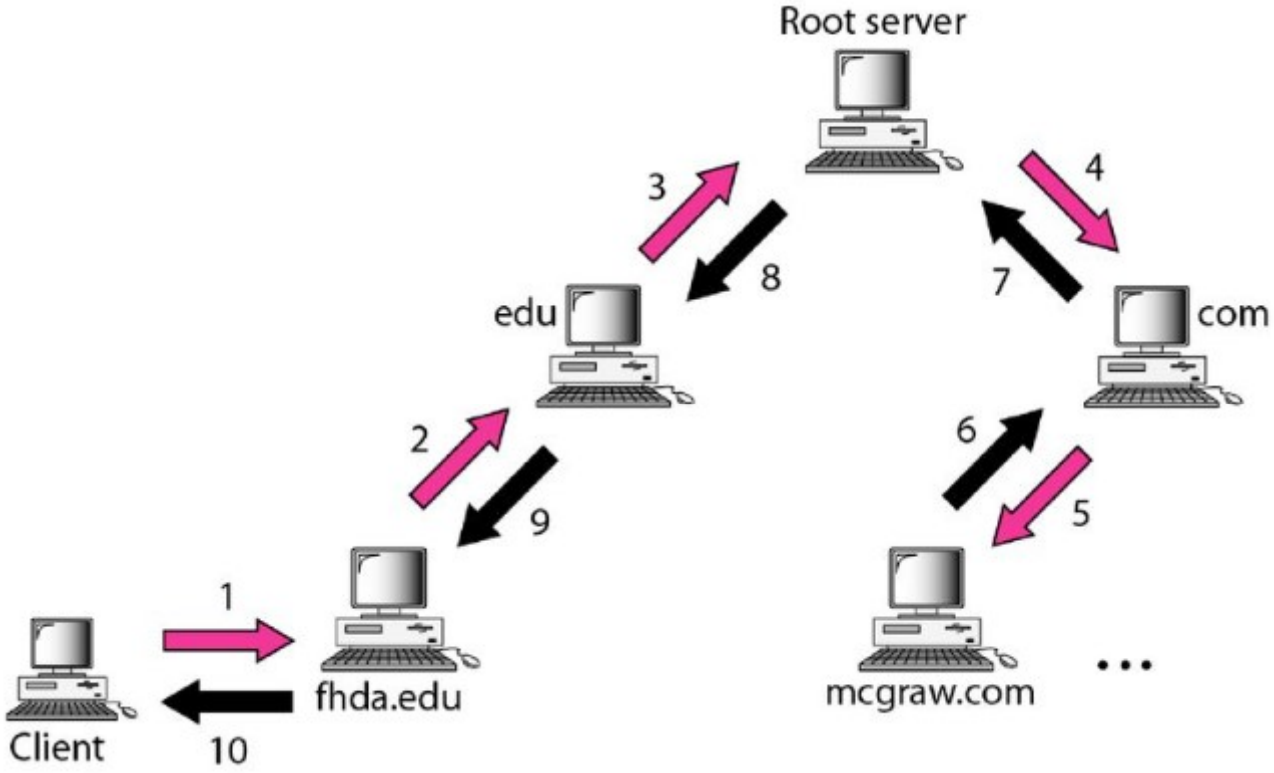


Figure 25.12 Recursive resolution

RESOLUTION

- Iterative 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.
- Now the client must repeat the query to the third server. This process is called iterative resolution because the client repeats the same query to multiple servers.
- In Figure 25.13 the client queries four servers before it gets an answer from the mcgraw.com server.

RESOLUTION

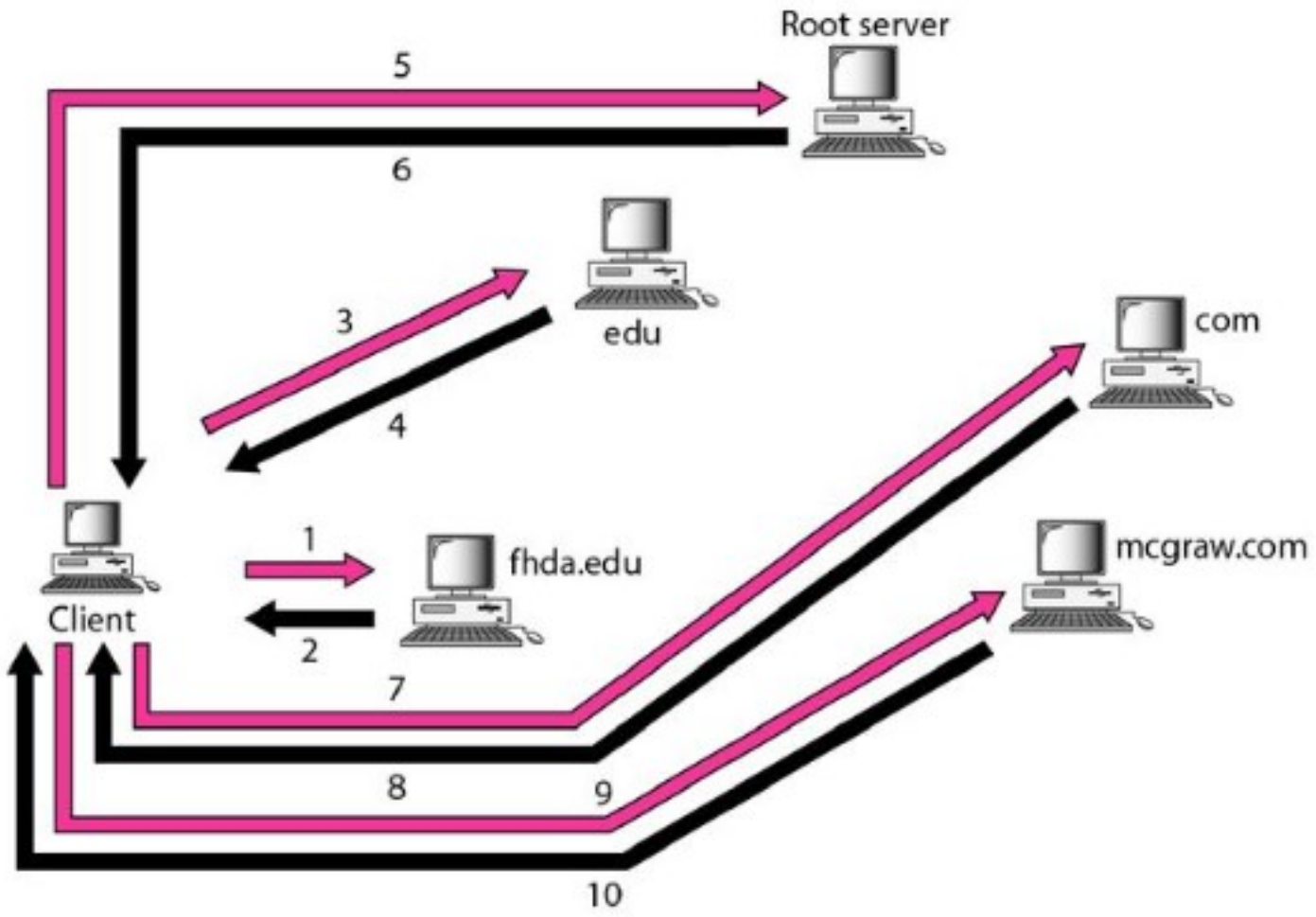


Figure 25.13 Iterative resolution

RESOLUTION

- Caching
- Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
- Caching is used for reduction of this search time to increase efficiency. When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
- If same or another client asks for the same mapping, it can check its cache memory and solve the problem.
- The server marks the response as unauthoritative, when it is sent from cache memory and not from an authoritative source.

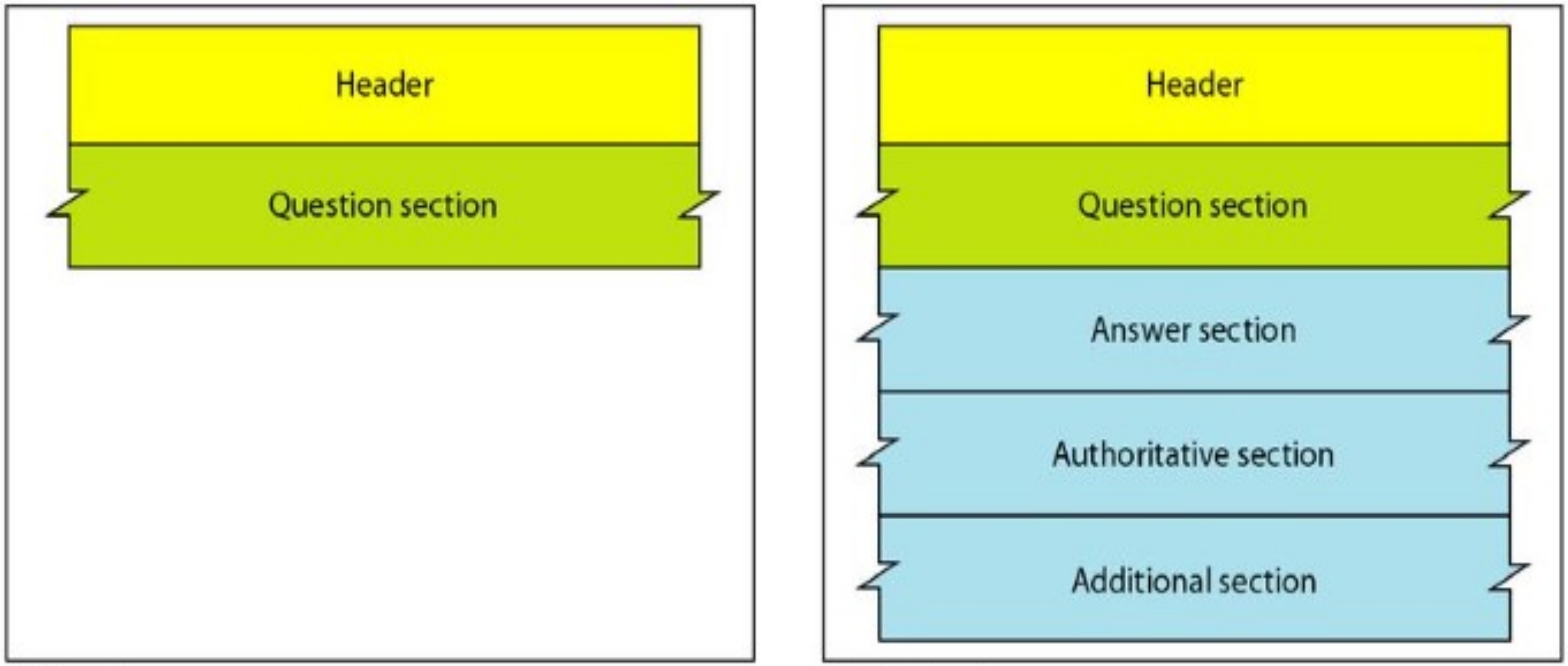
RESOLUTION

- Caching
- Caching speeds up resolution, but it can also be problematic. If a server caches a mapping for a long time, it may send an outdated mapping to the client.
- To counter this, two techniques are used.
- **First**, the authoritative server always adds information to the mapping called time-to-live (TTL).
- It defines the time in seconds that the receiving server can cache the information.
- After that time, the mapping is invalid and any query must be sent again to the authoritative server.
- **Second**, DNS requires that each server keep a TTL counter for each mapping it caches. The cache memory must be searched periodically, and those mappings with an expired TTL must be purged.

DNS Messages

- DNS has two types of messages: query and response. Both types have the same format.
- The query message consists of a header and question records; the response message consists of a header, question records, answer records, authoritative records, and additional records (see Figure 25.14)

Figure 25.14 Query and Response Messages



DNS Messages

- Header
- Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes, and its format is shown in Figure 25.15.

Figure 25.15 Header Format

DNS Messages

- Header
- The identification subfield is used by the client to match the response with the query. The client uses a different identification number each time it sends a query.
- The server duplicates this number in the corresponding response. The flags subfield is a collection of subfields that define the type of the message, the type of answer requested, the type of desired resolution (recursive or iterative), and so on.
- The number of question records subfield contains the number of queries in the question section of the message.

DNS Messages

- Header
- The number of answer records subfield contains the number of answer records in the answer section of the response message. Its value is zero in the query message.
- The number of authoritative records subfield contains the number of authoritative records in the authoritative section of a response message.
- Its value is zero in the query message. Finally, the number of additional records subfield contains the number additional records in the additional section of a response message.
- Its value is zero in the query message.

DNS Messages

- Question Section
- This is a section consisting of one or more question records. It is present on both query and response messages. We will discuss the question records in a following section.
- Answer Section
- This is a section consisting of one or more resource records. It is present only on response messages. This section includes the answer from the server to the client (resolver). We will discuss resource records in a following section.

DNS Messages

- Authoritative Section
- This is a section consisting of one or more resource records. It is present only on response messages.
- This section gives information (domain name) about one or more authoritative servers for the query.
- Additional Information Section
- This is a section consisting of one or more resource records. It is present only on response messages. This section provides additional information that may help the resolver.
- For example, a server may give the domain name of an authoritative server to the resolver in the authoritative section, and include the IP address of the same authoritative server in the additional information section.

Types of Records

- As we saw in Section 25.6, two types of records are used in DNS. The question records are used in the question section of the query and response messages. The resource records are used in the answer, authoritative, and additional information sections of the response message.
- Question Record
- A question record is used by the client to get information from a server. This contains the domain name.
- Resource Record
- Each domain name (each node on the tree) is associated with a record called the resource record. The server database consists of resource records. Resource records are also what is returned by the server to the client.

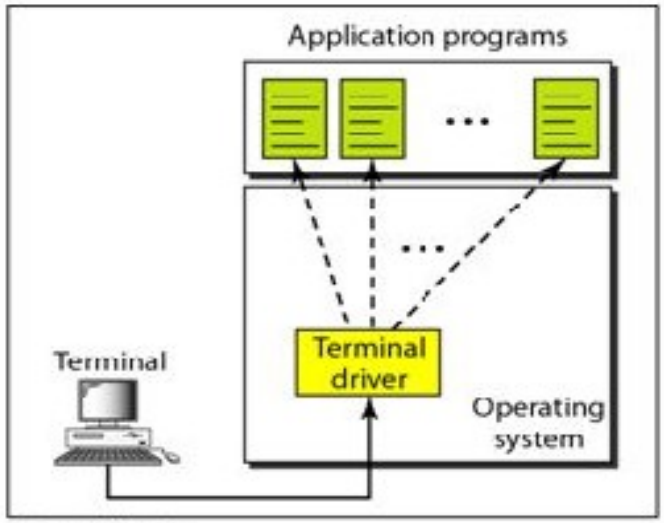
Introduction to Telnet, SMTP, FTP, WWW

- Internet, users may want to run application programs at a remote site and create results that can be transferred to their local site.
- However, it would be impossible to write a specific client/server program for each demand.
- The better solution is a general-purp<Dse client/server program that lets a user access any application program on a remote computer.
- After logging on, a user can use the services available on the remote computer and transfer the results back to the local computer.

TELNET

- TELNET is an abbreviation for TErminaL NETwork.
- It is the standard TCP/IP protocol for virtual terminal service as proposed by the International Organization for Standards (ISO).
- TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.
- *TELNET is a general-purpose client/server application program.*

Figure 26.1 *Local and remote log-in*



TELNET

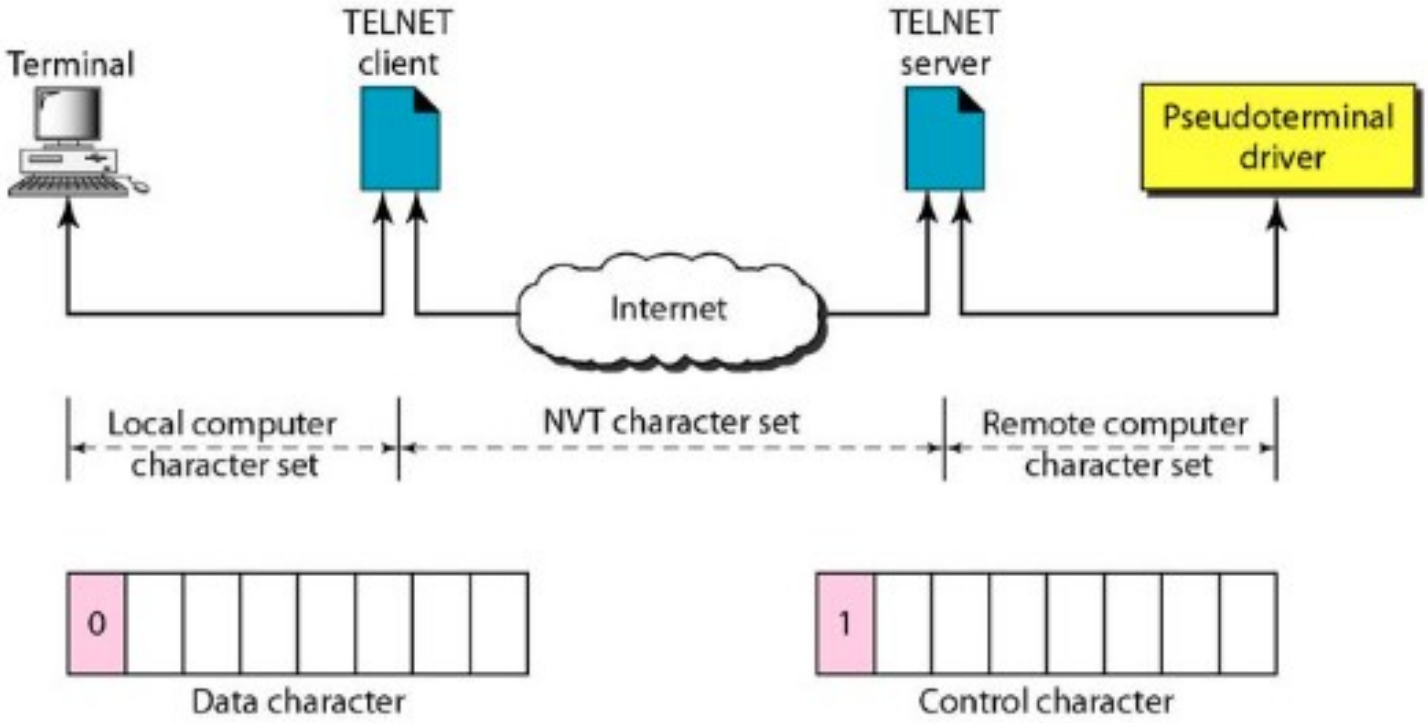
- **Remote Login**
- When the user types something on local computer, then local operating system accepts character.
- Local computer does not interpret the characters, it will send them to TELNET client.
- TELNET client transforms these characters to a universal character set called Network Virtual Terminal (NVT) characters and it will pass them to the local TCP/IP protocol Stack.
- Commands or text which is in the form of NVT, travel through Internet and it will arrive at the TCP/IP stack at remote computer.
- Characters are then delivered to operating system and which later on passed to TELNET server.
- Then TELNET server changes that characters to characters which can be understandable by remote computer.
- Remote operating system receives character from a pseudo-terminal driver, which is a piece of software that pretends that characters are coming from a terminal.
- Operating system then passes character to the appropriate application program.

TELNET

- **Network Virtual Terminal**
- Every computer and its operating system accept a special combination of characters as tokens.
- E.g. the end-of-file token in a computer running the DOS operating system is Ctrl+z, while the UNIX operating system recognizes Ctrl+d.
- We are dealing with heterogeneous systems.
- TELNET solves this problem by defining a universal interface called the network virtual terminal (NVT) character set.
- Via this interface, the client TELNET translates characters (data or commands) that come from the local terminal into NVT form and delivers them to the network.
- The server TELNET, on the other hand, translates data and commands from NVT form into the form acceptable by the remote computer. For an illustration of this concept, see Figure 26.2.

TELNET

Figure 26.2 Concept of NVT



TELNET

- **NVT Character Set**
- With NVT Character set, TELNET client translates characters into NVT form and deliver to network.
- TELNET server translates data and commands from NVT form to the other form that will be understandable by remote computer.
- NVT uses 2 sets of characters, one for data and other for control. Size of both characters is 8-bit bytes.
- For data, NVT is an 8-bit character set in which 7 lowest bits are same as ASCII and highest order bit is 0.
- For control characters, NVT uses an 8-bit character set in which the highest bit is set to 1

Table 26.1 *Some NVT control characters*

Character	*Decimal*	*Binary*	*Meaning*
EOF	236	11101100	End of file
EOR	239	11101111	End of record
SE	240	11110000	Suboption end
NOP	241	11110001	No operation
DM	242	11110010	Data mark
BRK	243	11110011	Break
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase character
EL	248	11111000	Erase line
GA	249	11111001	Go ahead
SB	250	11111010	Suboption begin
WILL	251	11111011	Agreement to enable option
WONT	252	11111100	Refusal to enable option
DO	253	11111101	Approval to option request
DONT	254	11111110	Denial of option request
IAC	255	11111111	Interpret (the next character) as control

SMTP

- To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA.
- The formal protocol that defines the MTA client and server in the Internet is called the Simple Mail Transfer Protocol (SMTP).
- As we said before, two pairs of MTA client/server programs are used in the most common situation (fourth scenario).
- Figure 26. 16 below shows the range of the SMTP protocol in this scenario.

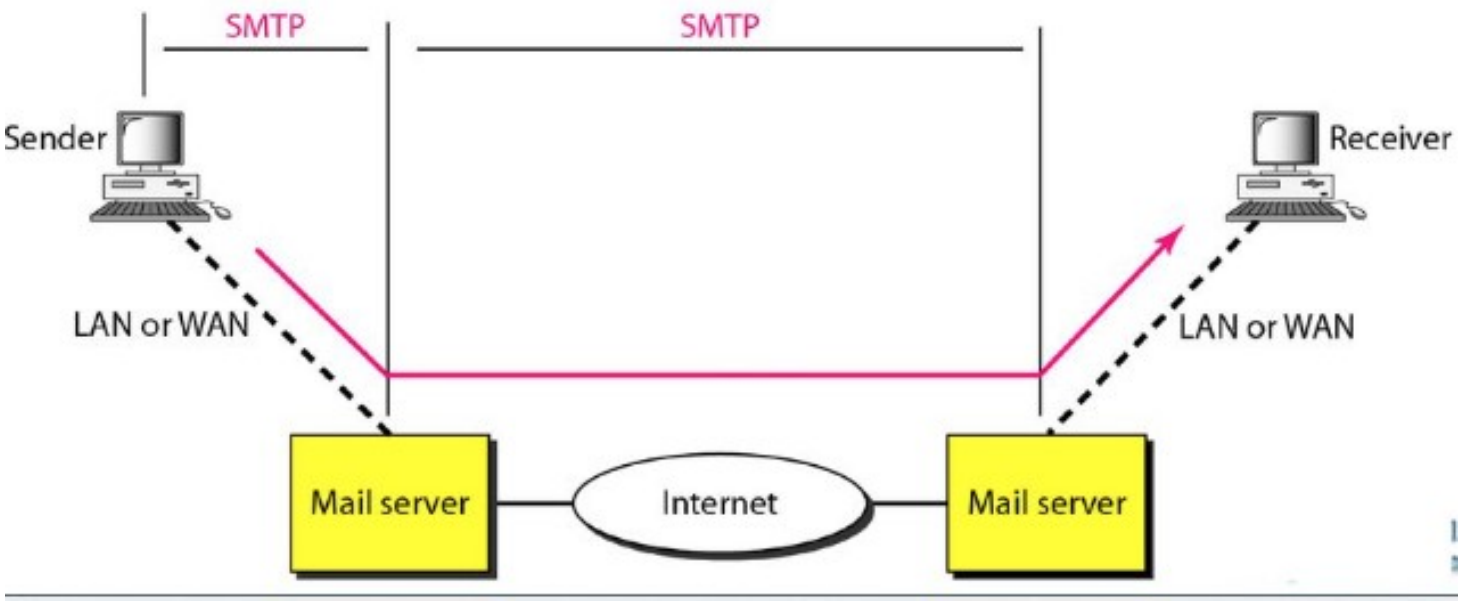


Figure 26.16 SMTP range

SMTP

- SMTP is used two times, between the sender and the sender's mail server and between the two mail servers.
- SMTP simply defines how commands and responses must be sent back and forth. Each network is free to choose Commands and Responses.
- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server (see Figure 26.17).
- Each command or reply is terminated by a two-character (carriage return and line feed) end-of-line token.

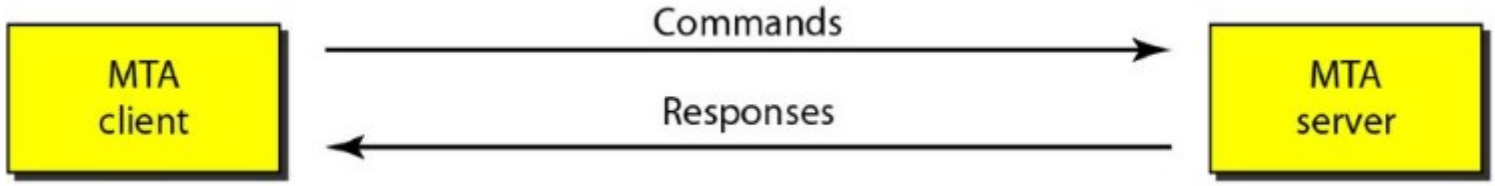


Figure 26.17 Commands and responses

SMTP

- **Commands**
- Commands are sent from the client to the server. The format of a command is shown in Figure 26.18.
- It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands.
- The first five are mandatory; every implementation must support these five commands.
- The next three are often used and highly recommended.
- The last six are seldom used.

Figure 26.18 Command format

Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name
SEND FROM	Intended recipient of the message
SMOL FROM	Intended recipient of the message
SMAL FROM	Intended recipient of the message

Table 26.7 Commands

SMTP

- **Responses**
- Responses are sent from the server to the client.
- A response is a three digit code that may be followed by additional textual information. Table 26.8 lists some of the responses.
- As the table shows, responses are divided into four categories. The leftmost digit of the code (2, 3, 4, and 5) defines the category.

Code	*Description*
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted: insufficient storage

Table 26.8 Responses

SMTP

- **Responses**

Code	*Description*
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Table 26.8 Responses Continue…

SMTP

- **Mail Transfer Phases**
- The process of transferring a mail message occurs in three phases: Connection establishment, Mail transfer, and Connection termination.
- **Message Access Agent: POP and IMAP**
- The first and the second stages of mail delivery use SMTP.
- However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server.
- In other words, the direction of the bulk: data (messages) is from the client to the server.
- On the other hand, the third stage needs a pull protocol; the client must pull messages from the server. The direction of the bulk data is from the server to the client.

SMTP

- The third stage uses a message access agent.
- Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).
- Figure 26.19 shows the position of these two protocols in the most common situation

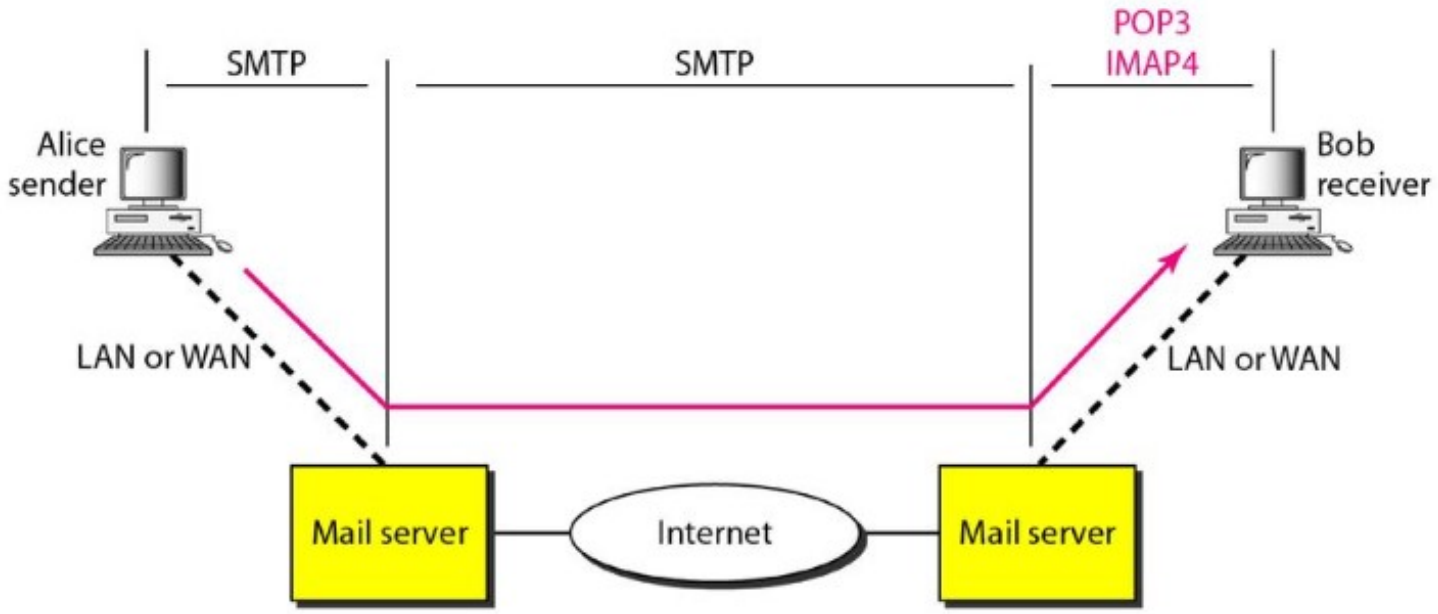


Figure 26.19 POP3 and IMAP4

SMTP

- POP3 Post Office Protocol, version 3 (POP3) is simple and limited in functionality.
- The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server.
- The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one.

SMTP

- Figure 26.20 shows an example of downloading using POP3. POP3 has two modes: the delete mode and the keep mode.
- **In the delete mode**, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval.
- The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.
- **The keep mode** is normally used when the user accesses her mail away from her primary computer (E.g. a laptop).
- The mail is read but kept in the system for later retrieval and organizing.

POP3

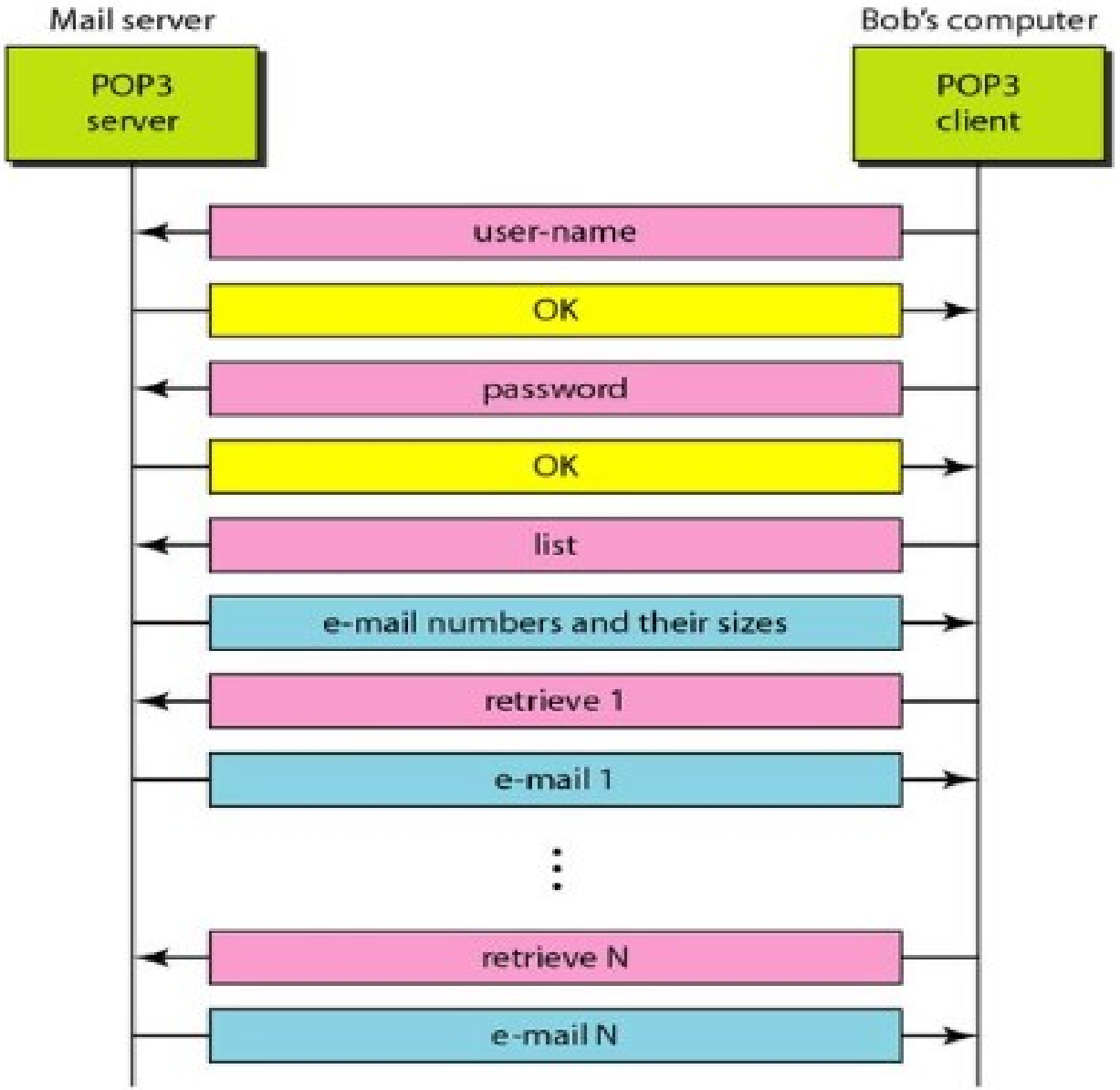


Figure 26.20 The exchange of commands and responses in POP3

SMTP

- IMAP4 Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4).
- IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.
- IMAP4 provides the following extra functions:
  - A user can check the e-mail header prior to downloading.
  - A user can search the contents of the e-mail for a specific string of characters prior to downloading.
  - A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
  - A user can create, delete, or rename mailboxes on the mail server.
  - A user can create a hierarchy of mailboxes in a folder for e-mail storage.

POP3 Vs IMAP

Feature	POP3	IMAP
Where is protocol defined?	RFC 1939	RFC 2060
Which TCP port is used?	110	143
Where is e-mail stored?	User's PC	Server
Where is e-mail read?	Off-line	On-line
Connect time required?	Little	Much
Use of server resources?	Minimal	Extensive
Multiple mailboxes?	No	Yes
Who backs up mailboxes?	User	ISP
Good for mobile users?	No	Yes
User control over downloading?	Little	Great
Partial message downloads?	No	Yes
Are disk quotas a problem?	No	Could be in time
Simple to implement?	Yes	No
Widespread support?	Yes	Growing

FTP

- **File Transfer Protocol (FTP)**
- File Transfer Protocol (FTP) is the standard mechanism provided by *TCP/IP* for copying a file from one host to another.
- Although it looks simple and straightforward, there are certain problems to deal with. E.g. two systems may have different file formats, different ways to represent data, different directory structures.
- FTP differs from other client/server applications in that it establishes two connections between the hosts which makes it more efficient.
- One connection is used for data transfer, the other for control information (commands and responses).

FTP

- In the control connection, We need to transfer only a line of command or a line of response at a time.
- The data connection needs more complex rules due to the variety of data types transferred.
- However, the difference in complexity is at the FTP level, not TCP. For TCP, both connections are treated the same.
- FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection.
- Figure 26.21 shows the basic model of FTP. The client has three components: user interface, client control process, and the client data transfer process.
- The server has two components: the server control process and the server data transfer process.
- The control connection is made between the control processes. The data connection is made between the data transfer processes.

FTP

- 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 commands that involve transferring files are used, and it closes when the file is transferred.
- In other words, when a user starts an FTP session, the control connection opens.
- While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

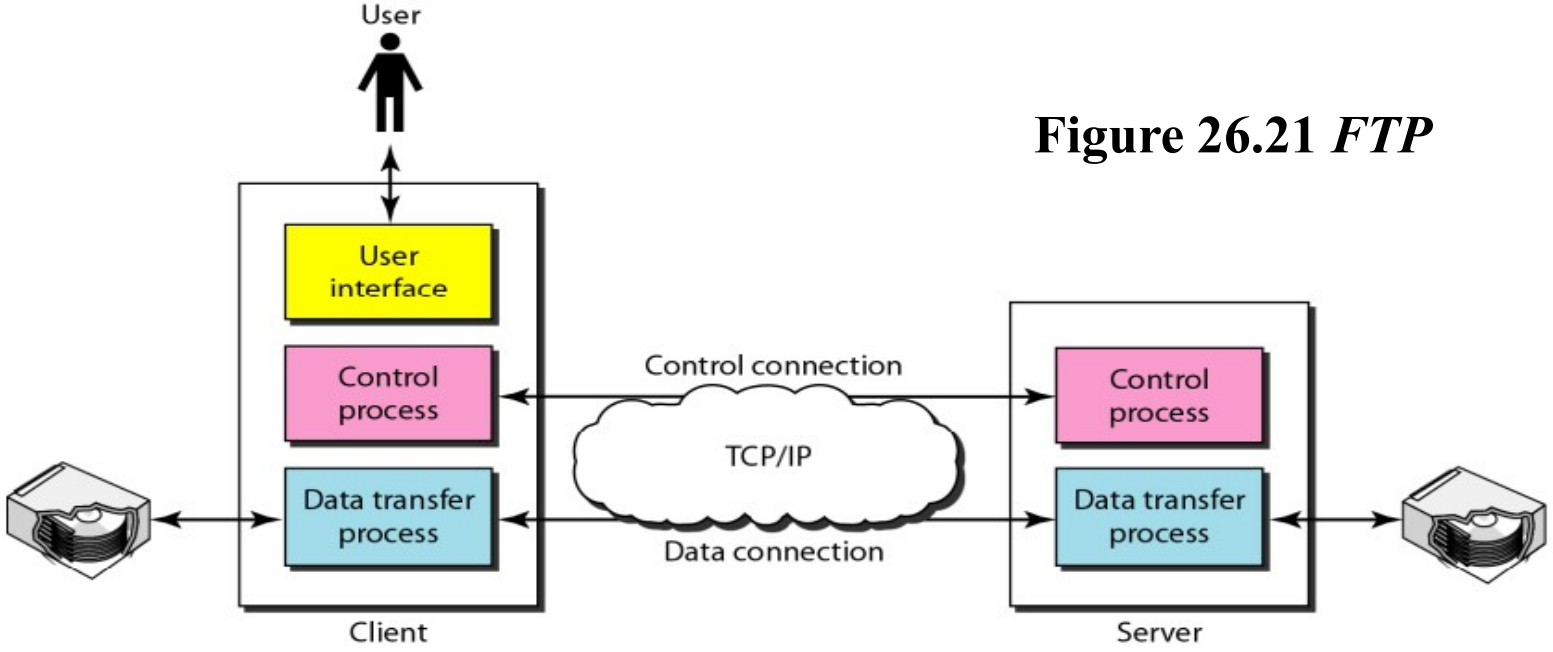


Figure 26.21 ***FTP***

FTP

- **Anonymous FTP**
- To use FTP, a user needs an account (user name) and a password on the remote server. Some sites have a set of files available for public access, to enable anonymous FTP.
- To access these files, a user does not need to have an account or password. Instead, the user can use anonymous as the user name and guest as the password.
- User access to the system is very limited. Some sites allow anonymous users only a subset of commands.
- For example, most sites allow the user to copy some files, but do not allow navigation through the directories multiple times if several files are transferred.

WWW

- The WWW today is a distributed client-server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called sites, as shown in Figure 27.1.

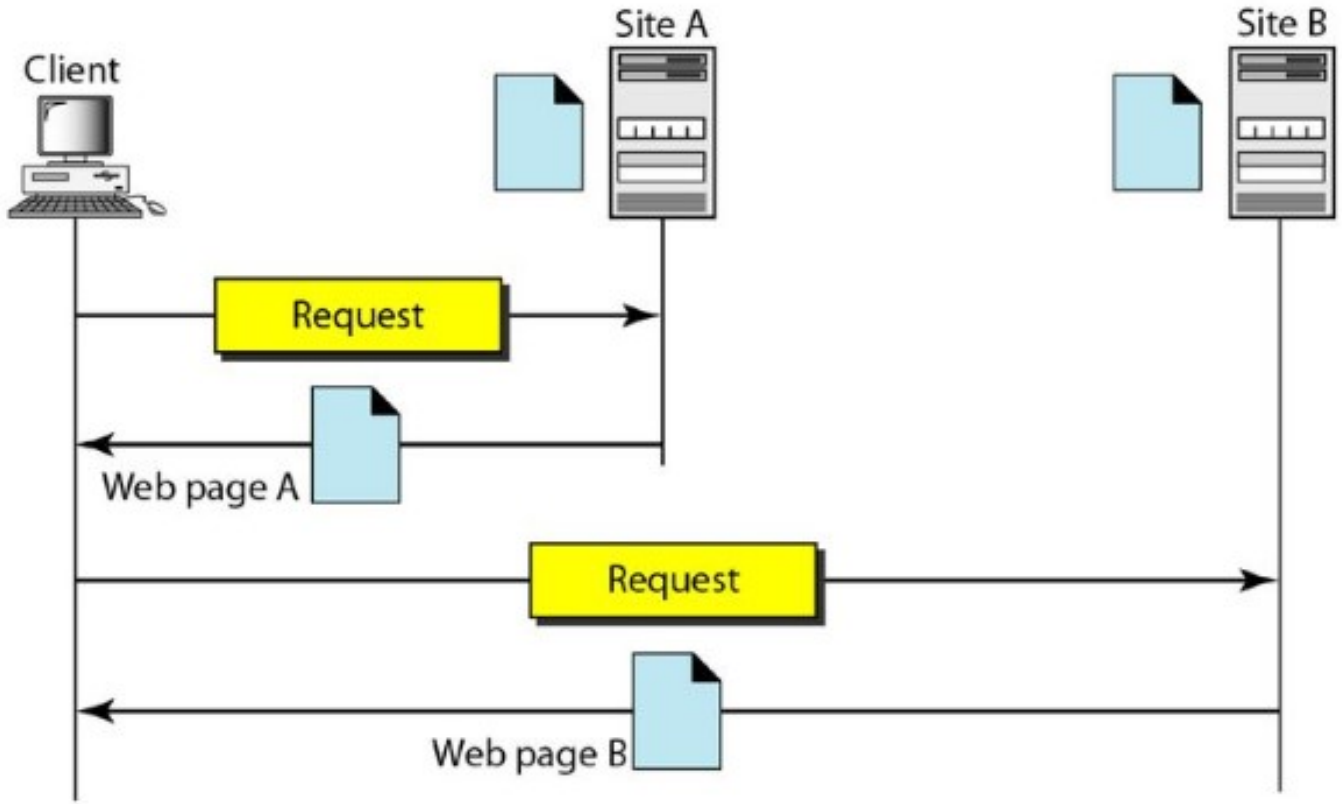


Figure 27.1 Architecture of WWW

WWW

- Each site holds one or more documents, referred to as Web pages.
- Each Web page can contain a link to other pages in the same site or at other sites. The pages can be retrieved and viewed by using browsers.
- Let us go through the scenario shown in Figure 27.1. The client needs to see some information that it knows belongs to site A. It sends a request through its browser, a program that is designed to fetch Web documents.
- The request, among other information, includes the address of the site and the Web page, called the URL.
- The server at site A finds the document and sends it to the client. When the user views the document, she finds some references to other documents, including a Web page at site B.
- The reference has the URL for the new site. The user is also interested in seeing this document. The client sends another request to the new site, and the new page is retrieved.

WWW

- **Client (Browser)**
- A variety of vendors offer commercial browsers that interpret and display a Web document, and all use nearly the same architecture.
- Each browser usually consists of three parts: a controller, client protocol, and interpreters.
- The controller receives input from the keyboard or the mouse and uses the client programs to access the document.
- After the document has been accessed, the controller uses one of the interpreters to display the document on the screen.
- The client protocol can be one of the protocols described previously such as FTP or HTIP (described later in the chapter).
- The interpreter can be HTML, Java, or JavaScript, depending on the type of document.

WWW

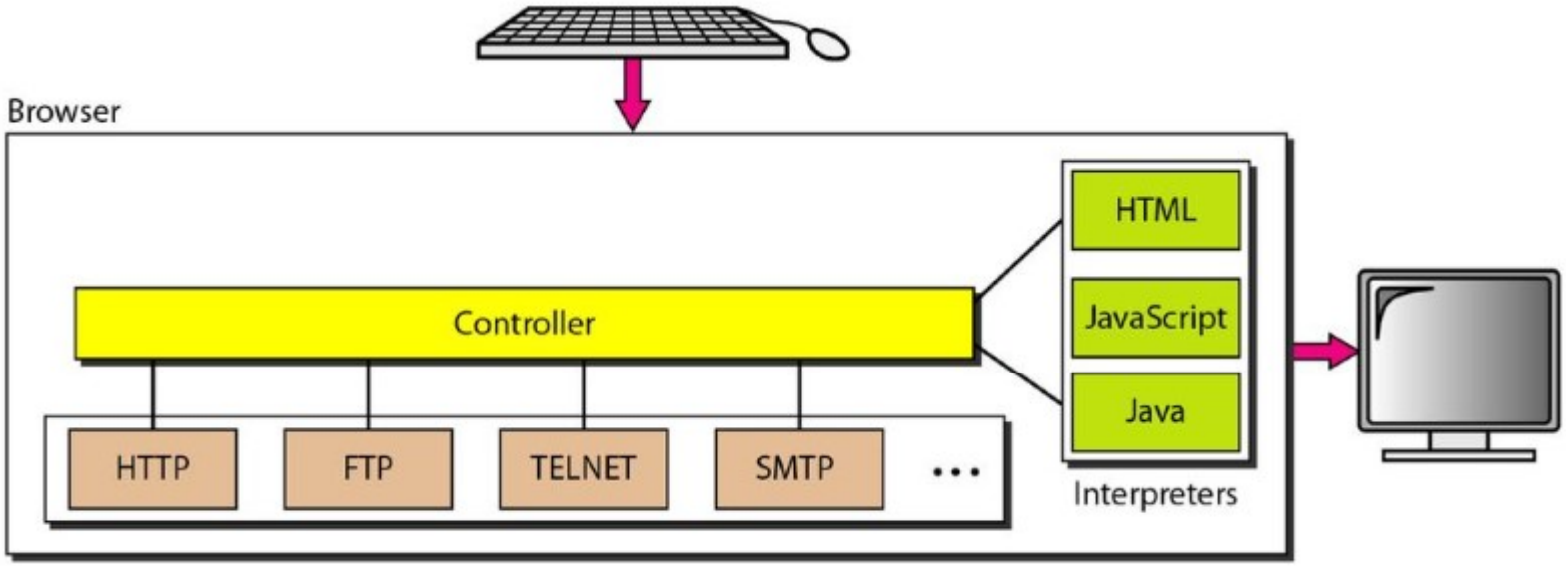
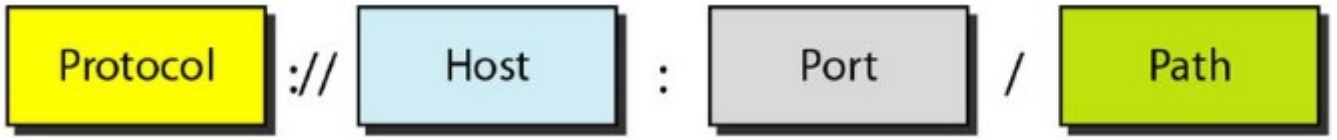


Figure 27.2 Browser

WWW

- **Server**
- The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client.
- To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk.
- A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time
- **Uniform** Resource Locator
- A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators.
- The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet.
- The URL defines four things: protocol, host computer, port, and path (see Figure 27.3).

Figure 27.3 URL



WWW

- The protocol is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP.
- The host is the computer on which the information is located, although the name of the computer can be an alias.
- Web pages are usually stored in computers, and computers are given alias names that usually begin with the characters "www".
- This is not mandatory, however, as the host can be any name given to the computer that hosts the Web page.
- The URL can optionally contain the port number of the server. If the port is included, it is inserted between the host and the path, and it is separated from the host by a colon.
- Path is the pathname of the file where the information is located.
- Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.

WWW

- **Cookies**
- The World Wide Web was originally designed as a stateless entity. A client sends a request; a server responds.
- Their relationship is over. The original design of WWW, retrieving publicly available documents, exactly fits this purpose.
- Today the Web has other functions; some are listed here.
- 1. Some websites need to allow access to registered clients only.
- 2. Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
- 3. Some websites are used as portals: the user selects the Web pages he wants to see.
- 4. Some websites are just advertising. For these purposes, the cookie mechanism was devised.

WWW

- **Creation and Storage of Cookies**
- The creation and storage of cookies depend on the implementation; however, the principle is the same.
- 1. When a server receives a request from a client, it stores information about the client in a file or a string.
- The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.
- 2. The server includes the cookie in the response that it sends to the client.
- 3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the domain server name.

WWW

- **Using Cookies**
- When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server.
- If found, the cookie is included in the request. When the server receives the request, it knows that this is an old client, not a new one.
- Note that the contents of the cookie are never read by the browser or disclosed to the user.
- It is a cookie made by the server and eaten by the server. Now let us see how a cookie is used for the four previously mentioned purposes:

Web Documents

- Documents in WWW can be grouped into three broad categories:
    - Static
    - Dynamic,
    - Active
- The category is based on the time at which the contents of the document are determined.
- **Static documents**
- Static documents are fixed-content documents that are created and stored in server Client can get only copy of document
- HTML (Hypertext Markup Language) is language used to create web pages

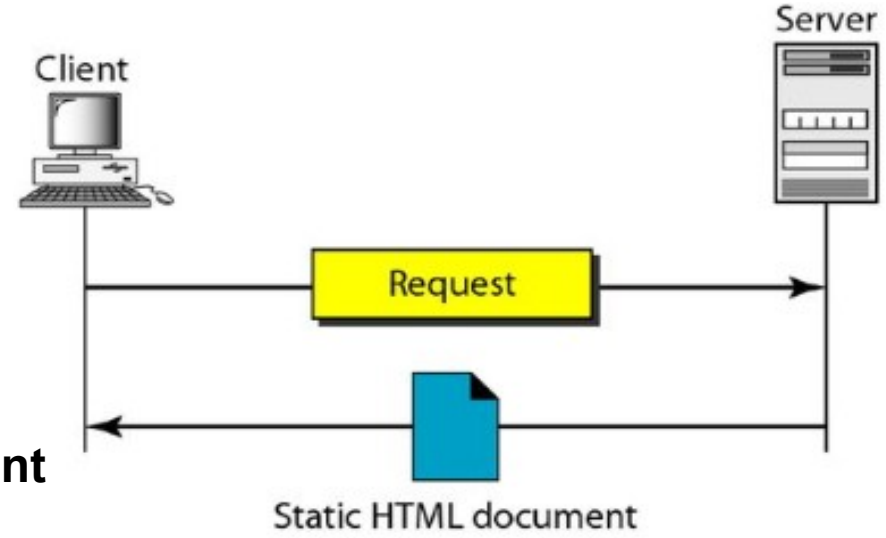


Figure 27.4 Static document

Web Documents

- **Dynamic document**
- Dynamic document is created by Web server whenever browser requests document.
- When request arrives, web server runs an application program or script that creates dynamic document.
- Server returns output of program or script as response to browser that requested the document.
- Contents of dynamic document can vary from one request to another Example: date and time of server.

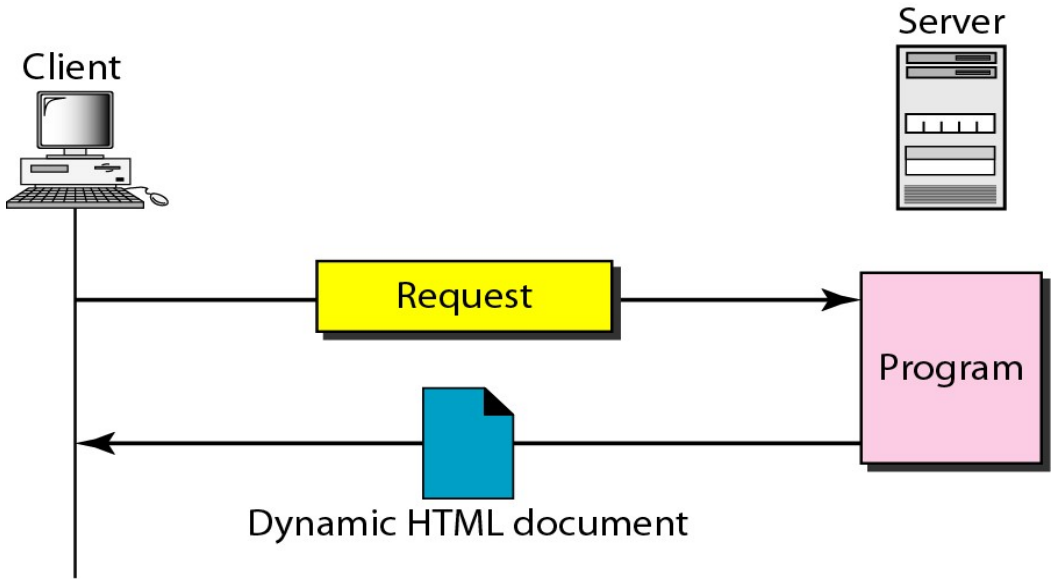


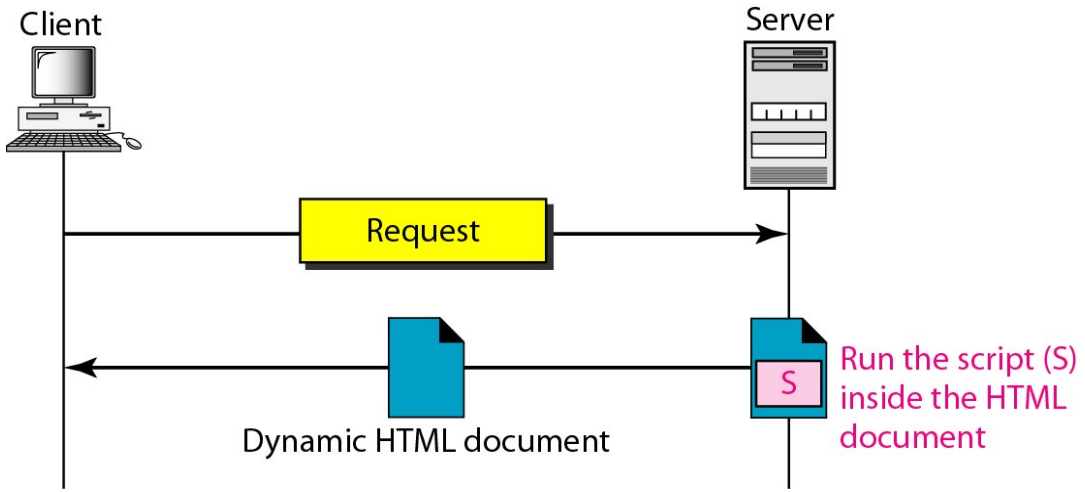
Figure 27.8 Dynamic document using CGI

Web Documents

- **Dynamic document**
- Common Gateway Interface (CGI) is technology that creates and handles dynamic documents.
- CGI is set of standards that defines how dynamic document is written, how data are input to program, and how output result is used.
- Problem with CGI is that program must create an entire document each time request is made.
- Generally document has some part like header as fixed and some part is dynamic.
- Solution is to create file containing fixed part using HTML and embed script, a source code, that can be run by server to provide varying values.

Web Documents

- Technologies for creating dynamic documents using scripts:
  - Hypertext Preprocessor (PHP) using Perl language
  - Java Server Pages (JSP) using Java language
  - Active Server Pages (ASP) by Microsoft
  - ColdFusion for embedding SQL queries in HTML document



Active Documents

- For many applications, we need program or script to be run at client site, These are called active documents.
- For example, suppose we want to run a program that creates animated graphics on the screen or a program that interacts with the user.
- The program definitely needs to be run at the client site where the animation or interaction takes place .
- When browser requests an active document, server sends copy of document or script.
- Document is then run at client (browser) site. Example: Java applet, JavaScript, VBScript.