APPLICATION LAYER

Standard Application Layer Protocols

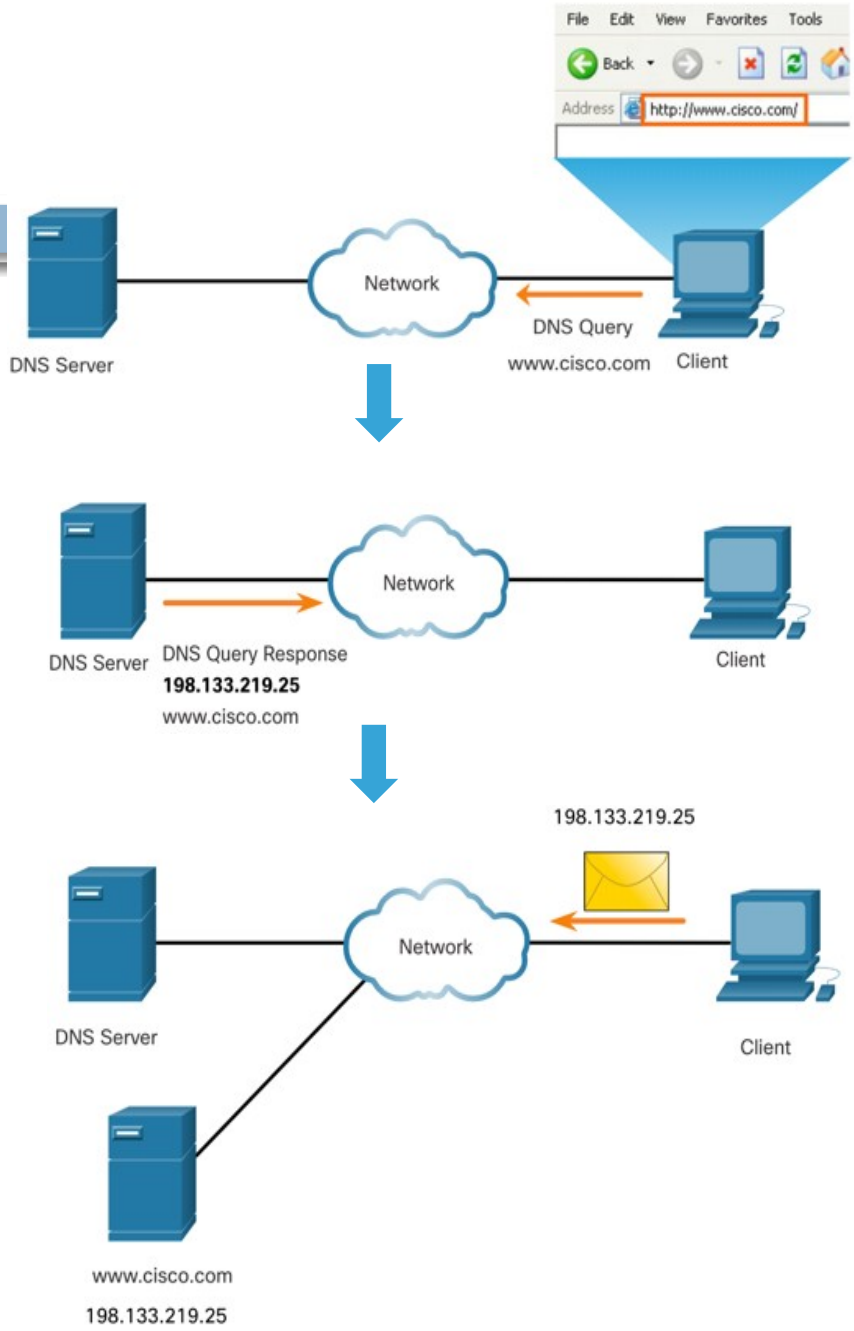
DNS

DNS

- Stands for Domain Name System
- Devices on the network has IP addresses
- But Human always prefers names instead of numeric addresses
- DNS maps the human readable addresses to corresponding IP addressees
- DNS uses UDP

DNS

- Domain names were created to convert the numeric IP addresses into a simple, recognizable name.
- Fully-qualified domain names (FQDNs), such as http://www.cisco.com, are much easier for people to remember than 198.133.219.25.
- The DNS protocol defines an automated service that matches resource names with the required numeric network address. It includes the format for queries, responses, and data.



The nslookup Command

- Nslookup is a computer operating system utility that allows a user to manually query the DNS servers configured on the device to resolve a given host name.
- This utility can also be used to troubleshoot name resolution issues and to verify the current status of the name servers.
- When the **nslookup** command is issued, the default DNS server configured for your host is displayed.
- The name of a host or domain can be entered at the **nslookup** prompt.

```
C:\Users> nslookup
Default Server:  dns-sj.cisco.com
Address:  171.70.168.183
> www.cisco.com
Server:  dns-sj.cisco.com
Address:  171.70.168.183
Name:    origin-www.cisco.com
Addresses:  2001:420:1101:1::a
          173.37.145.84
Aliases:  www.cisco.com
> cisco.netacad.net
Server:  dns-sj.cisco.com
Address:  171.70.168.183
Name:    cisco.netacad.net
Address:  72.163.6.223
>
```

DNS Process Overview

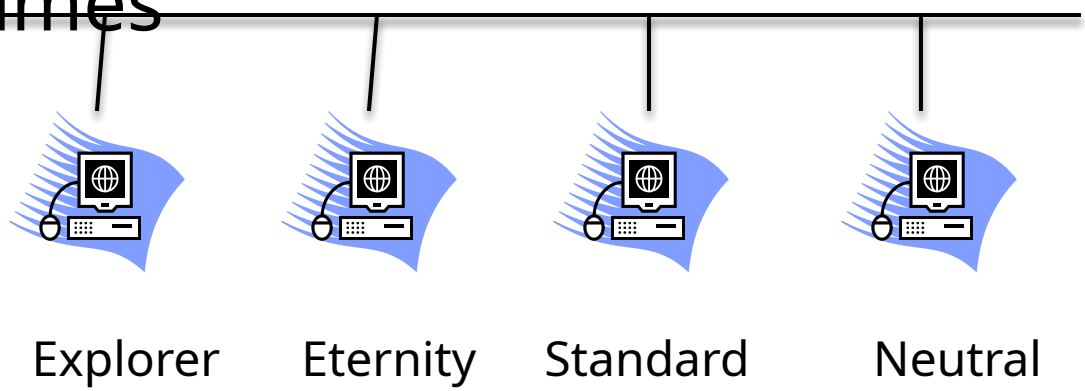
1. User machine runs client side of the DNS application
2. The browser (or any other program that uses URL) extracts the hostname to the client side of the DNS application
3. The DNS client sends a query containing the hostname to a DNS application
4. The DNS client eventually receives a reply, which includes the ip address of the hostname

Name Space

- Maps each address to a unique name
- Organized in two ways
  - Flat Name Space
  - Hierarchical Name Space

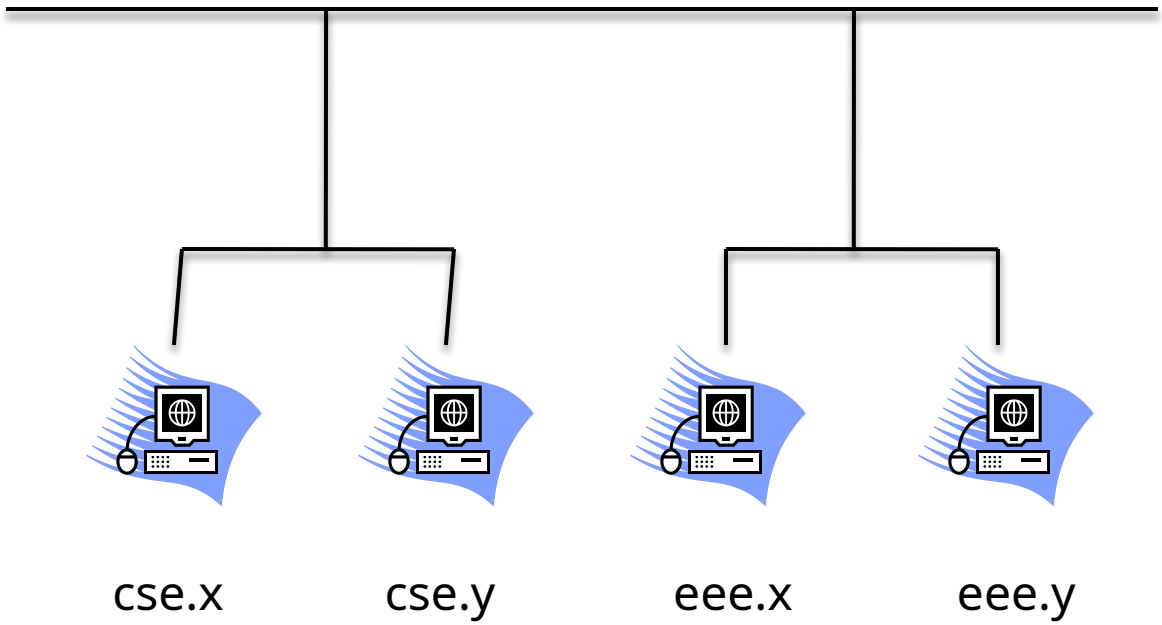
Flat Name Space

- Names are interpreted as a single, whole label
- No internal structure
- No clear relationship between any two names



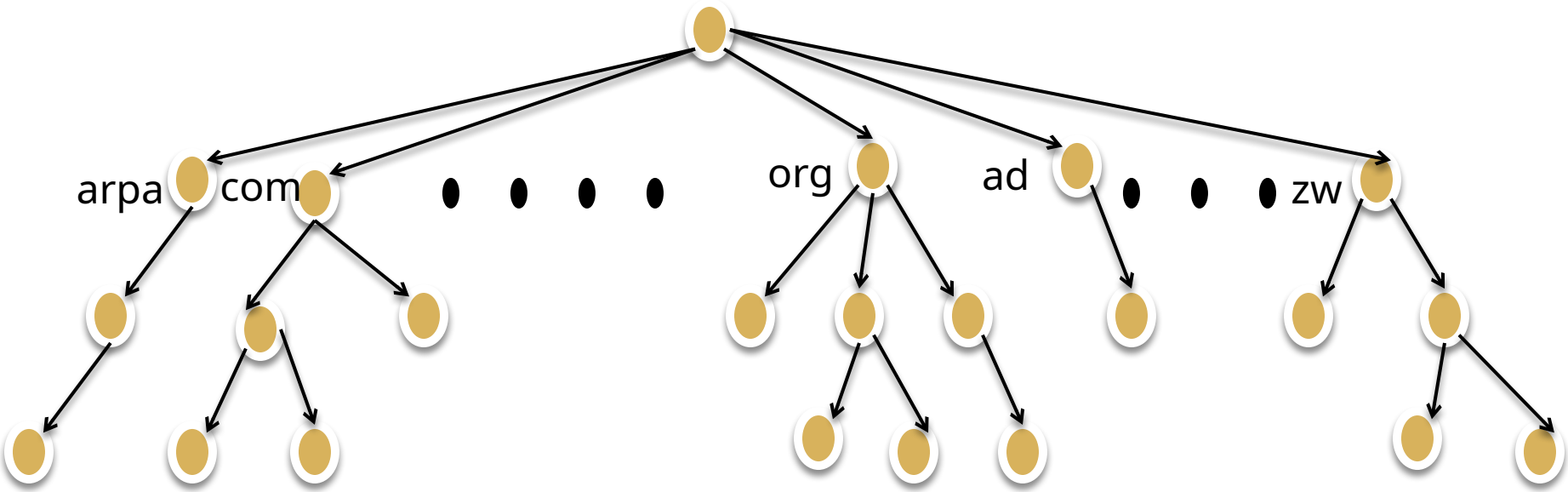
Hierarchical Name Space

- Logical grouping is possible based on the names



Domain Name Space

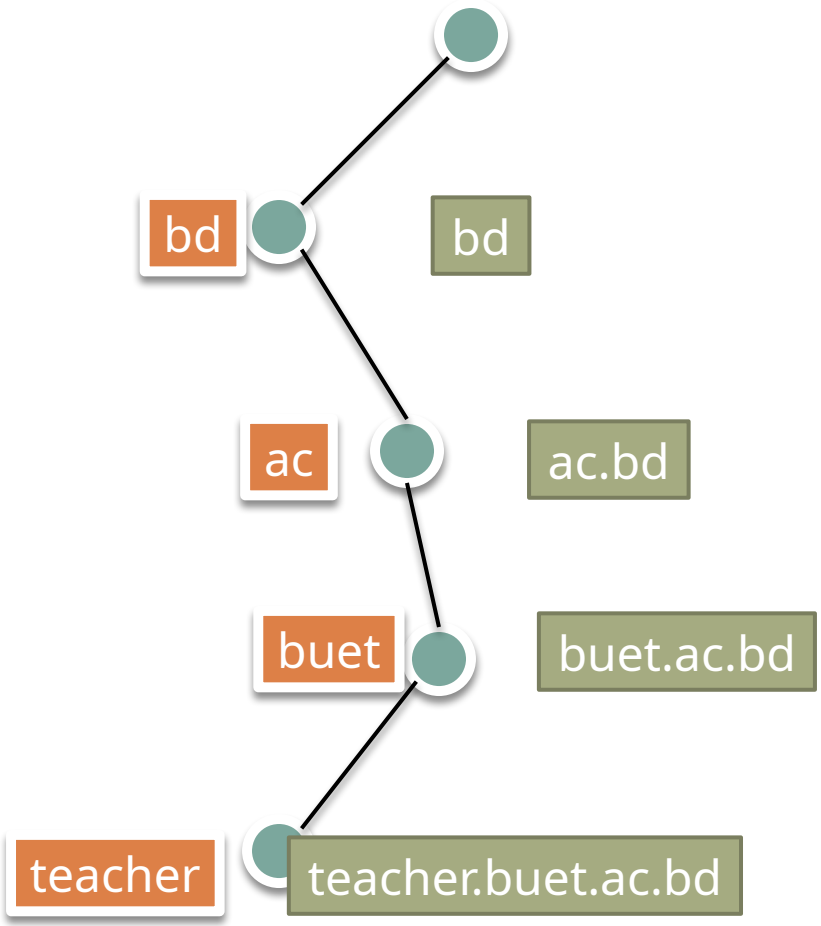
- Names are defined in an inverted-tree structure with the root at the top
- Tree can have at most 128 levels



Labels & Domain Name

- Label
  - Each node has a label of maximum 63 characters
  - Root is labeled with null string
  - Children of a node must have different labels for uniqueness
- Domain Name
  - Each node of a tree has a domain name
  - A full domain name is a sequence of labels separated by dots(.)
  - Read from node up to the root.

Label & Domain Name (Explained)



Fully Qualified Domain Name

- An FQDN is a domain name that contains the full name of a host
- Starts from a node and read up to the root
- It uniquely defines the host machine irrespective of the location of client

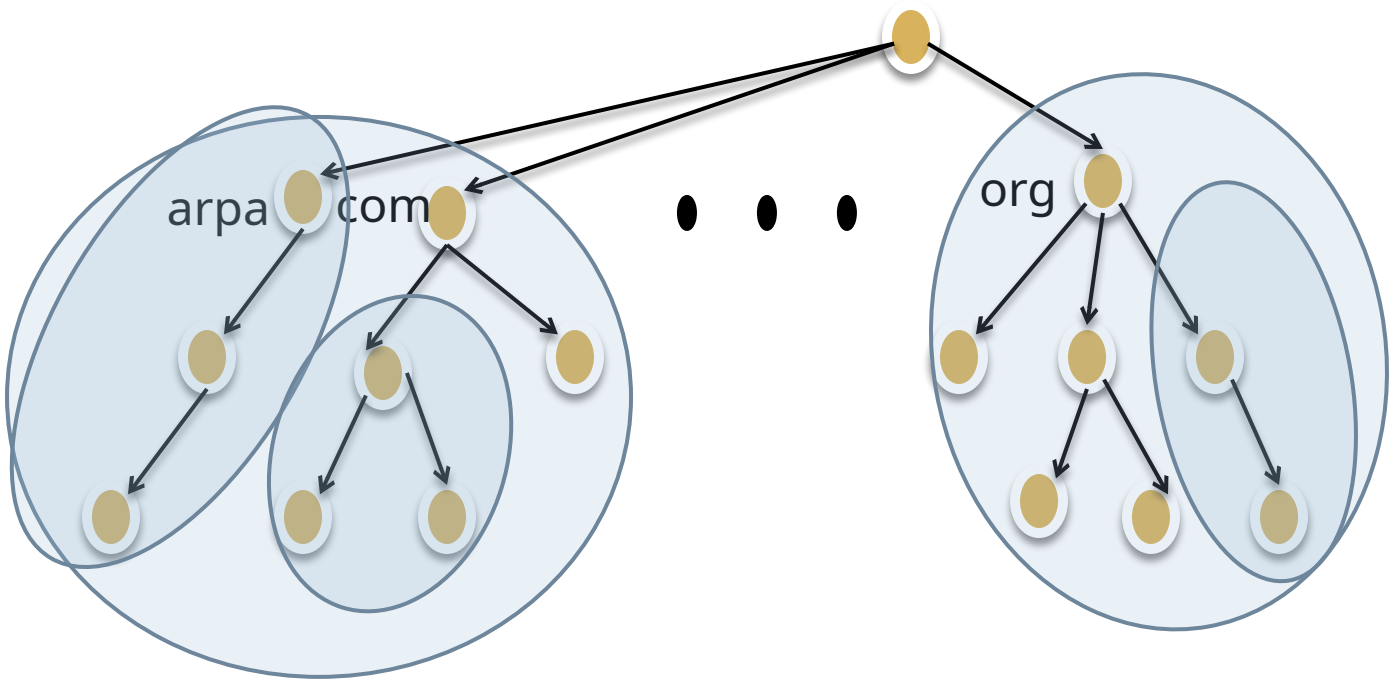
teacher.buet.ac.bd

Partially Qualified Domain Name

- PQDN starts from a node but does not reach the root
- Used when the name to be resolved belongs to the same site as the client
- Resolver can supply the suffix to create an FQDN

Domain

- Subtree of the DNS
- The name of the domain is the Domain Name of the node at the top of the subtree
- Domain is divided in subdomains



DNS Information storing

- In a single server
  - Inefficient
  - Unreliable
- Multiple and Distributed
  - Efficient
  - Reliable
  - But Processing Requires structure and protocols

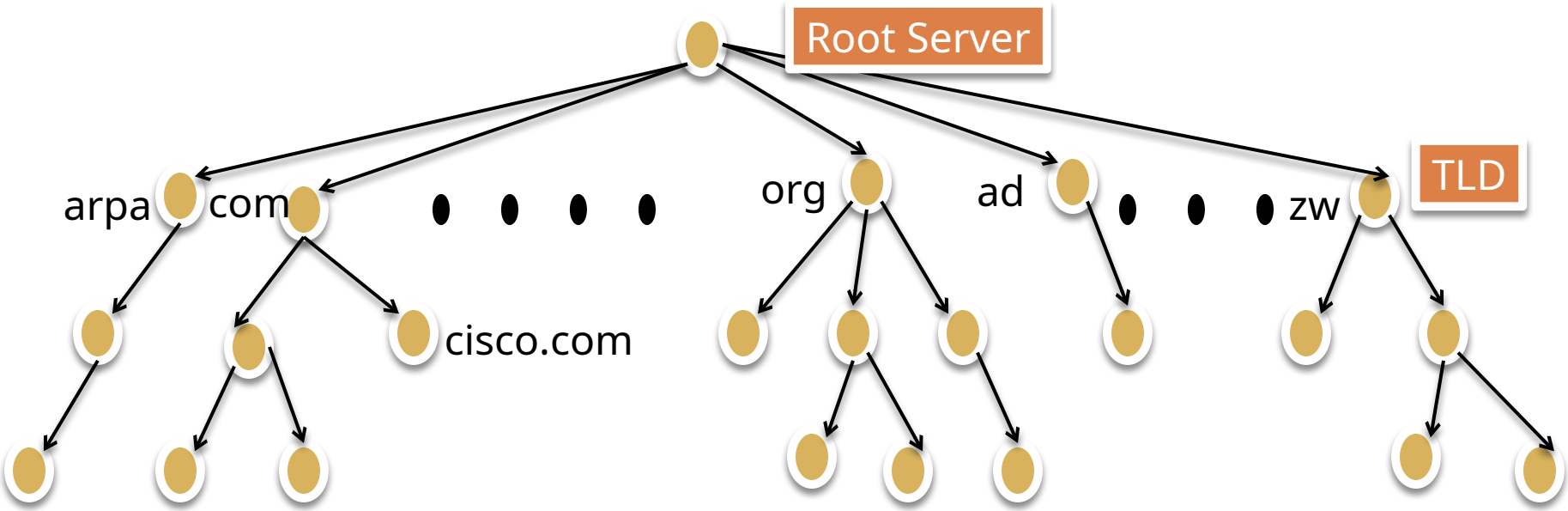
Inefficiency of Single Server

1. Single point of failure
2. Traffic Volume high. All requests have to be handled by a single DNS server
3. A single DNS server cannot be close to all querying clients
4. It would be updated frequently for every new host .

Hierarchy of Name Server

- Each server known as DNS server
- Each server is responsible(authoritative) for a domain
- Three classes of Server
  - Root DNS Server
  - Top-level Domain (TLD) Server
  - Authoritative DNS Server

Types of DNS Server



Root DNS Server

- Root DNS Server
  - Number of Root servers distributed throughout the world
  - Some servers are in fact cluster

TLD Server & Authoritative DNS Server

- TLD Server
  - These servers are responsible for Generic Top-level Domains such as com, org, net, edu and gov and all of the Country Top-level Domains such as uk, fr, ca, bd
- Authoritative DNS Server
  - Every organization with publicly accessible host must provide publicly accessible DNS records that map names of those host to IP Addresses
  - An organization authoritative DNS houses these DNS records

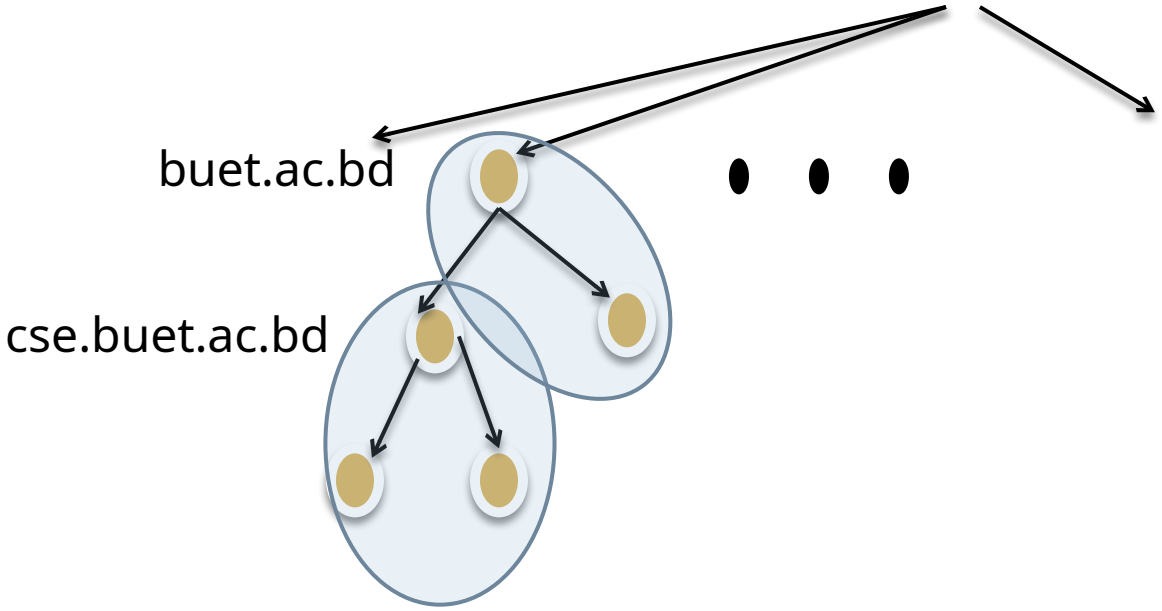
Local DNS Server

- Does not strictly belong to the hierarchy of servers
- Each ISP has a local DNS server
- When a host connects to the ISP, the ISP provides IP addresses one or more of its local DNS servers
- When a host makes a DNS query, the query is sent to the local DNS server which acts as a proxy, forwarding the query into the DNS server hierarchy

Zone

- Refers to a certain portion or administrative space within the global DNS
- All DNS zones form the DNS namespace
- Depending on the administrative rights delegated to a certain entity, DNS zones may consist of only one domain, or of many domains and sub-domains.
- Further authority over a sub-space could be delegated to other servers, if necessary.
- Zone File keeps all information for every node under the domain

Zone Explained



Authoritative DNS Server Types

- Primary DNS Server
  - Stores a file about the zone for which it is an authority
  - Responsible for creating, maintaining and updating Zone file
  - Stores zone file in a local disk
- Secondary DNS Server
  - Transfers the complete information about a zone from another server (primary or secondary) and stores the file in the local disk

Primary and Secondary Servers

- Both are authoritative for the zone that they serve
- The method of downloading information by secondary from primary is known as Zone Transfer
- Utility
  - Create Redundancy to mitigate failure

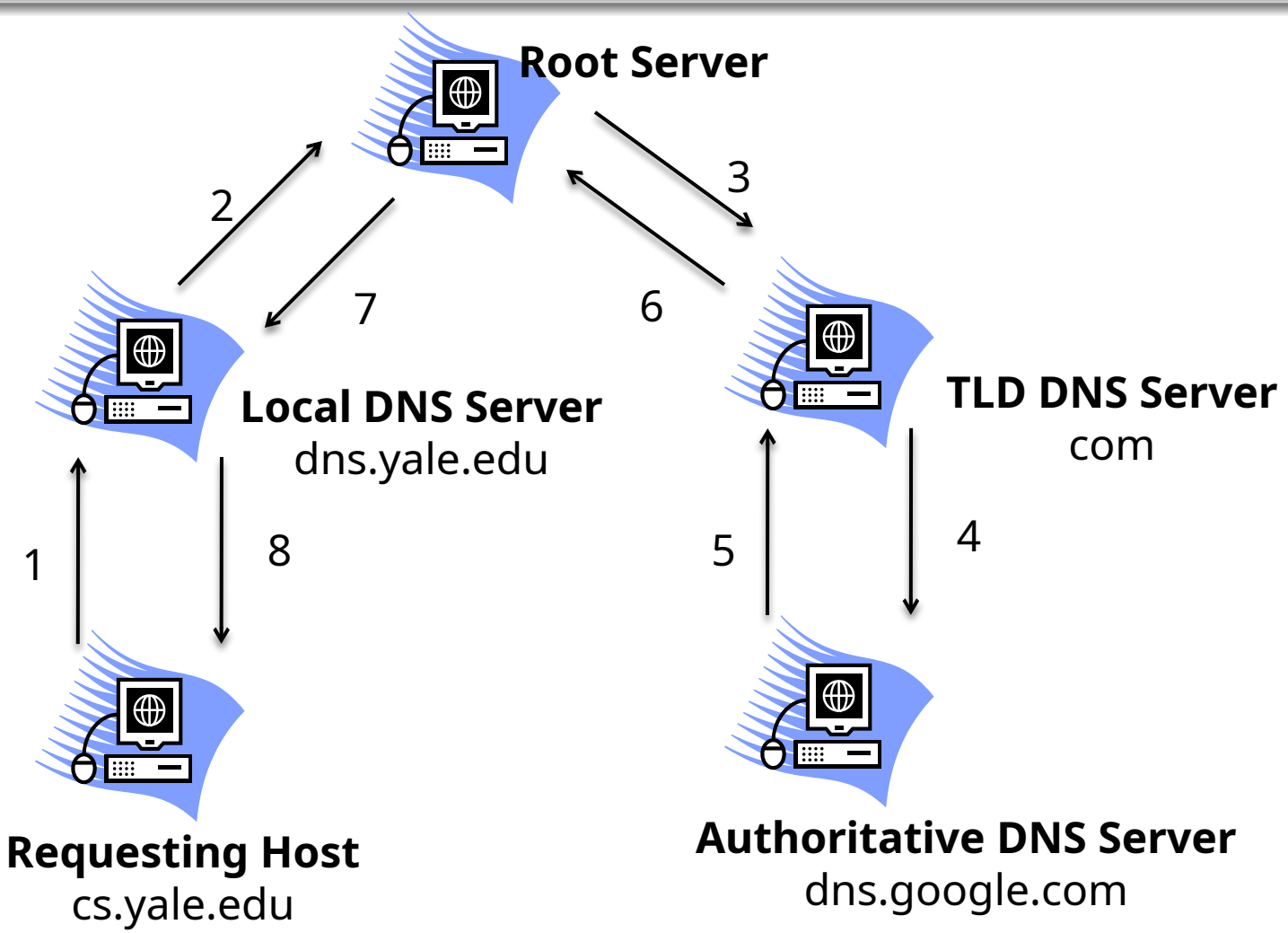
Resolver

- 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 other servers to provide information
- It also checks the occurrence of any error

Recursive Resolution

- Resolver accepts the server to supply the final answer
- If the server is the authority of the DNS, it supplies the answer
- If the server is not authority, it send the request to other server (usually parent) and waits for the response.
- If the parent is authority, it responds, otherwise, it sends request to another one
- When the query is resolved, then the response travel backs to the client

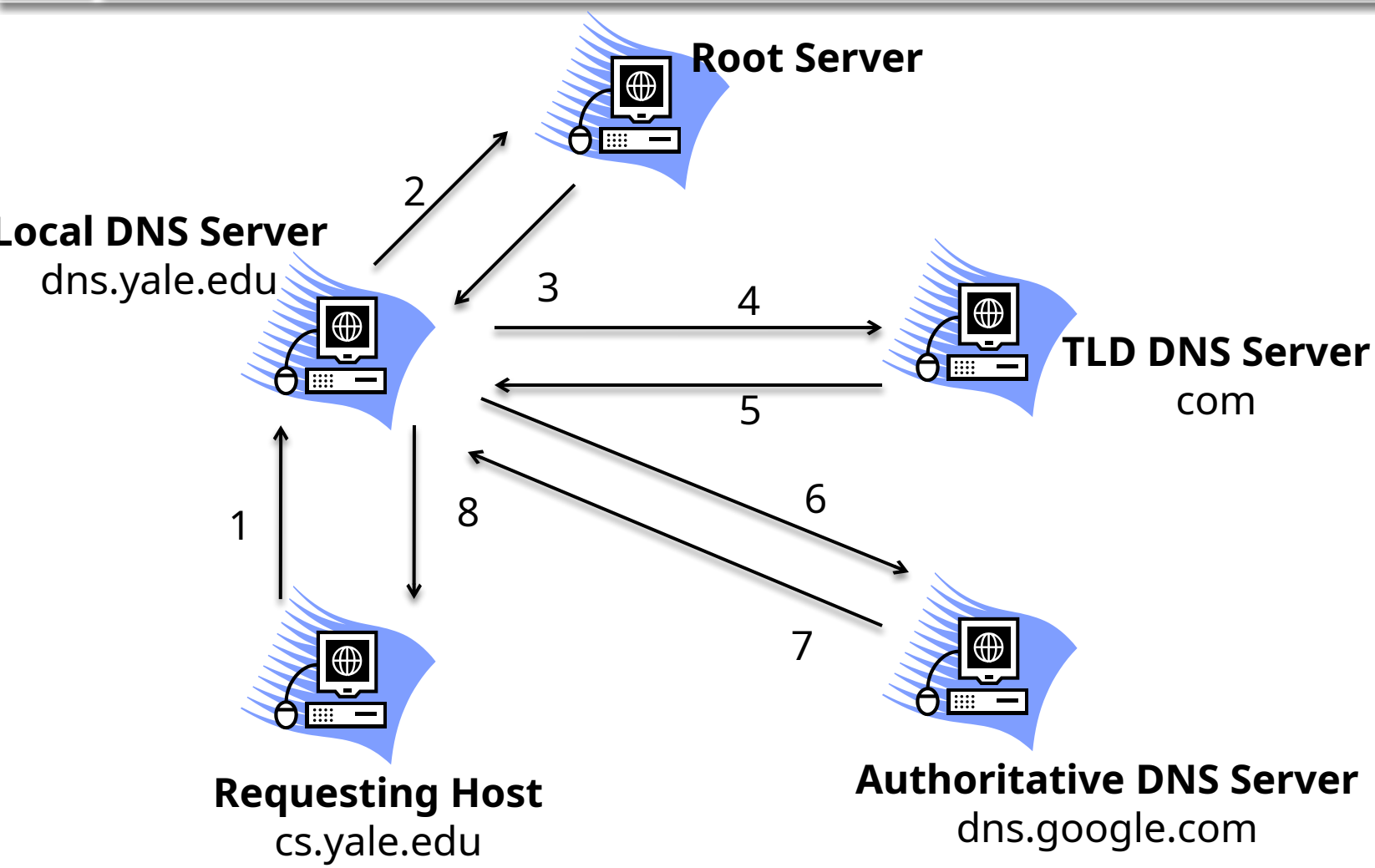
Recursive Resolution Explained



Iterative Resolution

- The client requests the local DNS.
- Local DNS then request another DNS Server
- If the server is an authority for the name, it sends the answer
- If the server is not an authority, it returns the ip addresses of the server which it thinks can resolve the query.
- The local DNS then repeats the query to the second server and so on
- Finally Client receives the resolution from Local DNS server

Iterative Resolution Explained



Caching

- Reduction in DNS lookup can be reduced at a greater scale if the mapping is stored locally
- When a server asks for a mapping from another server and receives response, it stores the information in its cache memory before sending it to client to satisfy other clients afterwards without further query
- The response coming from the cache is marked as Non-authoritative
- TTL is used to keep track of older records. The records are purged periodically.

DNS Records

- DNS servers store information in Resource Records
- A Records provide hostname-to-ip mapping
- Resource Record is a four-tuple
  - (Name, Value, Type, TTL)
- TTL
  - Determines when a resource record should be removed from the cache
- The meaning of Name and Value depend on Type

Type Explained

- Type=A
  - Name- Hostname
  - Value – IP Address
  - Provides standard hostname to IP Address Mapping
  - (sites.google.com, 142.23.23.1, A, 200)
- Type=NS
  - Name – Domain
  - Value – Hostname of an authoritative DNS server that may know the information about the domain
  - Provides routing the DNS query
  - (sites.google.com, dns.google.com, NS, 125)

Type Explained

- Type = CNAME
  - Name – Alias Domain Name
  - Value – Main Domain Name
  - Specifies that the domain name is an alias of another
  - This helps when running multiple services (like an FTP *and* a web server; each running on different ports) from a single IP address. Each service can then have its own entry in DNS.
  - (www.cs.mit.edu, star.cs.mit.edu, CNAME, 120)

Type Explained

- Type=MX
  - Name - Hostname
  - Value – The address of the Mail Server
  - A company can use the same aliased name for all its mail server and for one of its other server
  - (buet.ac.bd, mail.buet.ac.bd, MX, 125)

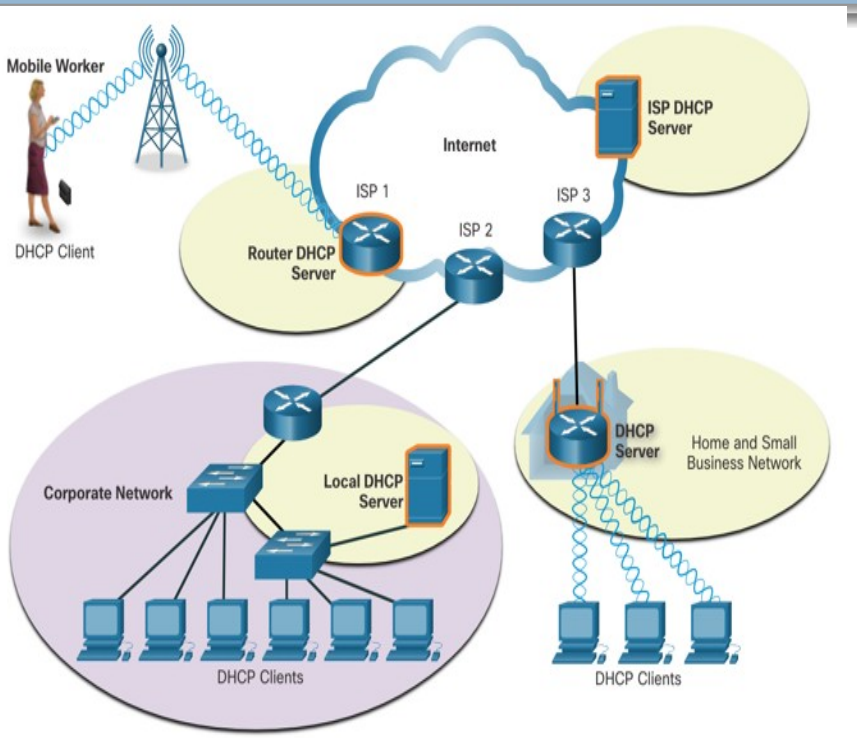
A and NS Record Explained

- If a DNS server is authoritative for a particular domain, it contains a Type A record
- If a server is not authoritative for a hostname, then it contains a type NS record for the domain that includes the hostname; it will also contain a Type A record that provides the ip address of the DNS server in the value field of the NS record
- TLD is not authoritative for cs.mit.edu
  - (mit.edu, dns.mit.edu, NS, 123)
  - (dns.mit.edu, 123.22.32.21, A, 134)

DHCP

Dynamic Host Configuration Protocol

- DHCP service automates the assignment of IPv4 addresses, subnet masks, gateways, and other IPv4 networking parameters.
- DHCP is considered dynamic addressing compared to static addressing. Static addressing is manually entering IP address information.



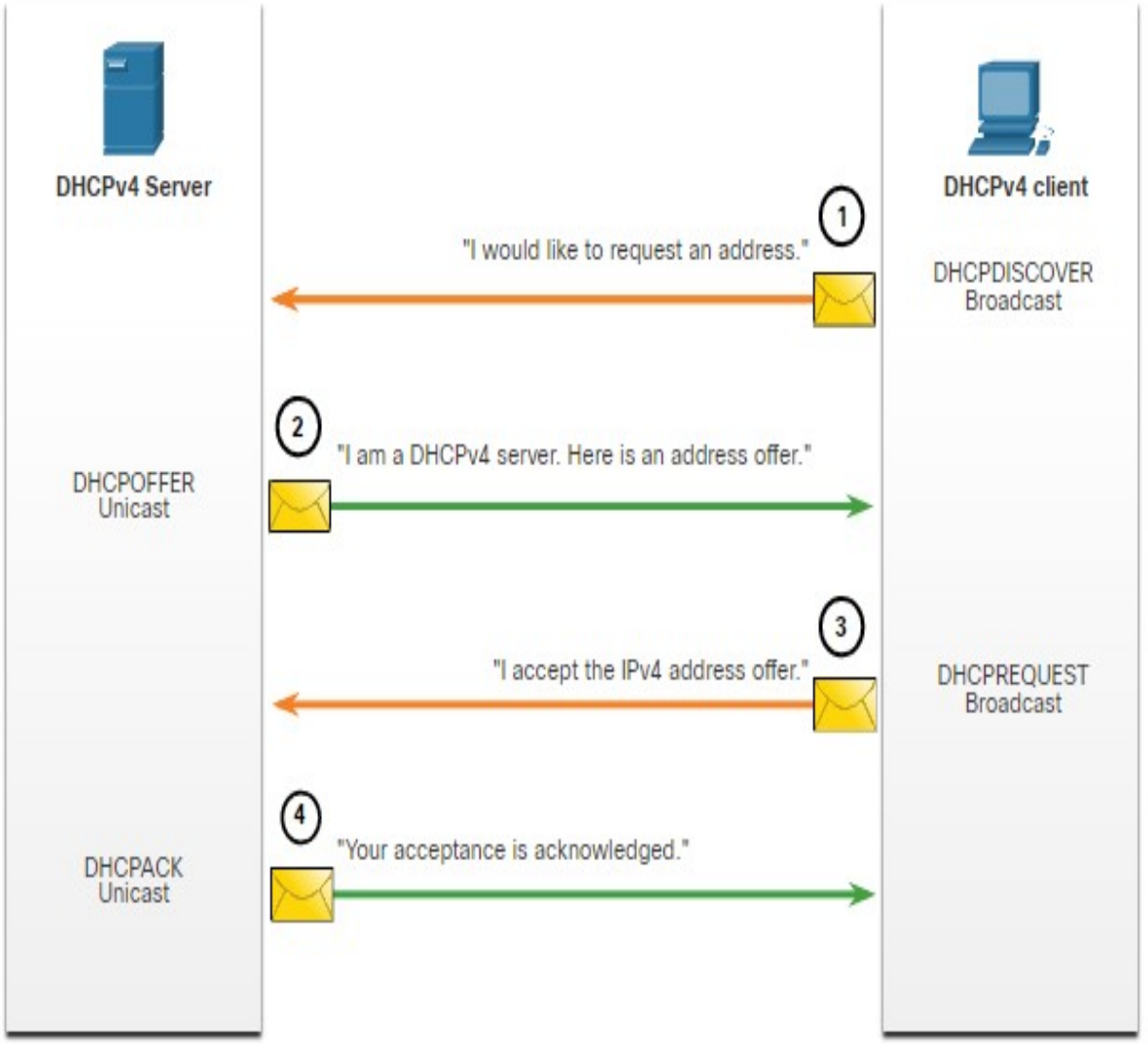
DHCP

- When a host connects to the network, the DHCP server is contacted, and an address is requested. The DHCP server chooses an address from a configured range of addresses called a pool and assigns (leases) it to the host.
- Many networks use both DHCP and static addressing. DHCP is used for general purpose hosts, such as end user devices. Static addressing is used for network devices, such as gateway routers, switches, servers, and printers.

Steps to Obtain a Lease

When the client boots (or otherwise wants to join a network), it begins a four-step process to obtain a lease:

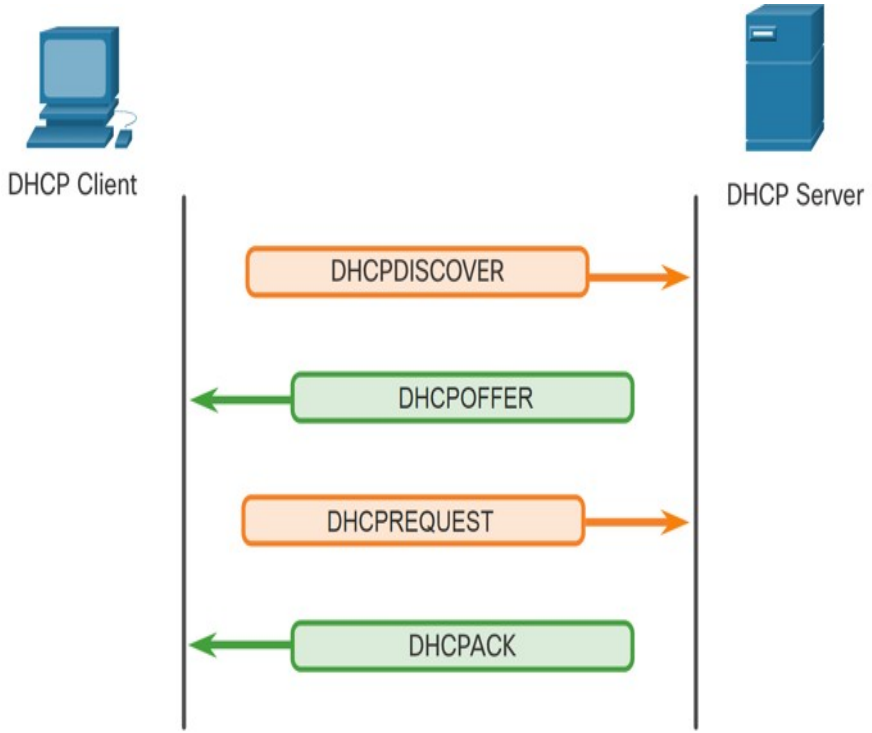
1. DHCP Discover (DHCPDISCOVER)
2. DHCP Offer (DHCPOFFER)
3. DHCP Request (DHCPREQUEST)
4. DHCP Acknowledgment (DHCPACK)



DHCP Operation

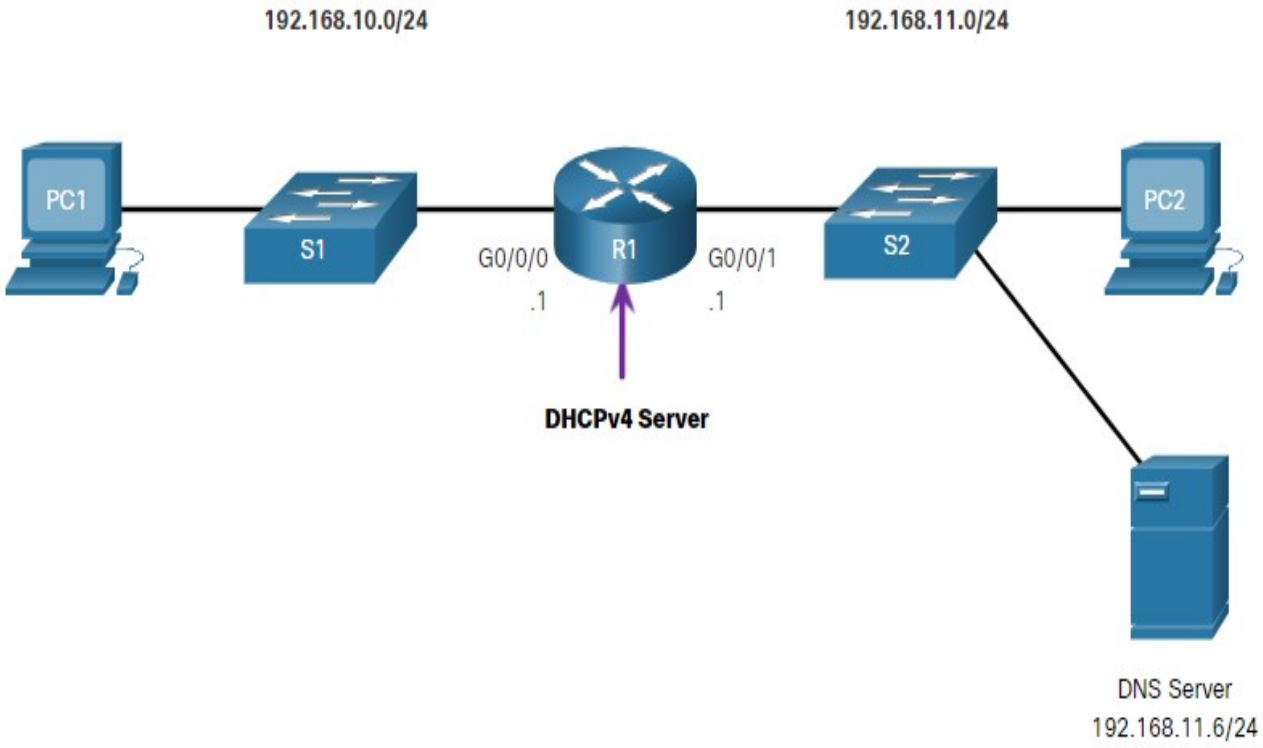
The DHCP Process:

- When an IPv4, DHCP-configured device boots up or connects to the network, the client broadcasts a DHCP discover (DHCPDISCOVER) message to identify any available DHCP servers on the network.
- A DHCP server replies with a DHCP offer (DHCPOFFER) message, which offers a lease to the client. (If a client receives more than one offer due to multiple DHCP servers on the network, it must choose one.)
- The client sends a DHCP request (DHCPREQUEST) message that identifies the explicit server and lease offer that the client is accepting.
- The server then returns a DHCP acknowledgment (DHCPACK) message that acknowledges to the client that the lease has been finalized.
- If the offer is no longer valid, then the selected server responds with a DHCP negative acknowledgment (DHCPNAK) message and the process must begin with a new DHCPDISCOVER message.



Configuring a Router as DHCPv4 Server

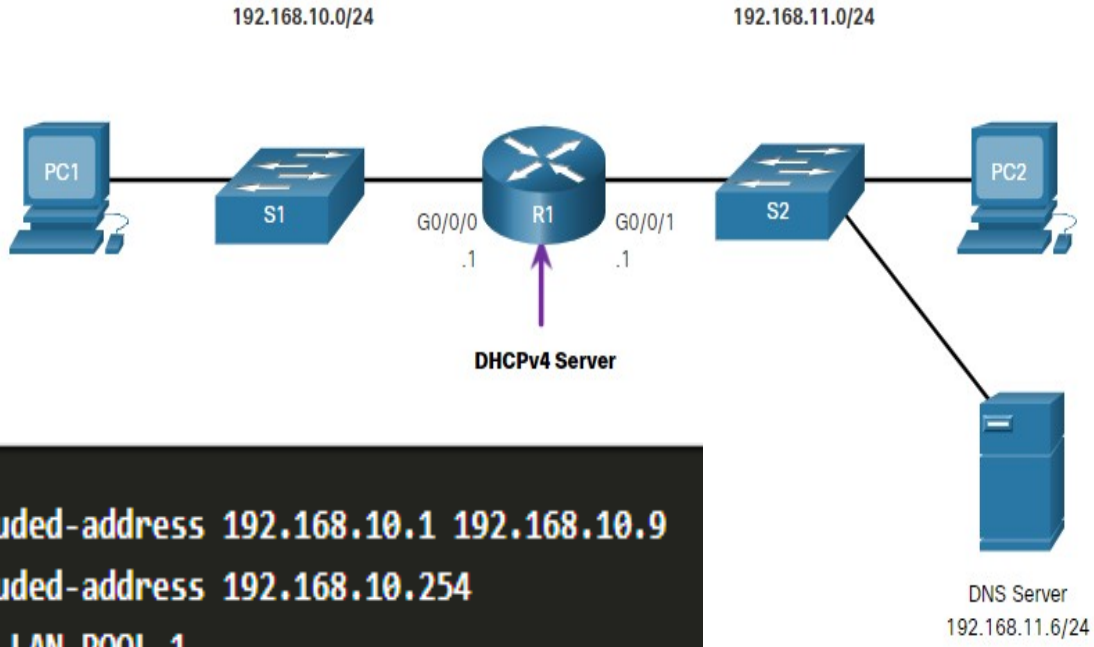
Now you have a basic understanding of how DHCPv4 works and how it can make your job a bit easier. A Cisco router running Cisco IOS software can be configured to act as a DHCPv4 server. The Cisco IOS DHCPv4 server assigns and manages IPv4 addresses from specified address pools within the router to DHCPv4 clients.



Steps to Configure a Cisco IOS DHCPv4 Server (Cont.)

Task	IOS Command
Define the address pool.	**network** *network-number* [*mask* | */ prefix-length*]
Define the default router or gateway.	**default-router** address [*address2....address8*]
Define a DNS server.	**dns-server** *address* [*address2...address8*]
Define the domain name.	**domain-name** *domain*
Define the duration of the DHCP lease.	**lease** {*days* [*hours* [*minutes*]] | **infinite**}
Define the NetBIOS WINS server.	**netbios-name-server** *address* [*address2...address8*]

Configuration Example




```
R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
R1(dhcp-config)# default-router 192.168.10.1
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com
R1(dhcp-config)# end
R1#
```

DHCPv4 Verification

Use the commands in the table to verify that the Cisco IOS DHCPv4 server is operational.

Command	Description
show running-config | section dhcp	Displays the DHCPv4 commands configured on the router.
show ip dhcp binding	Displays a list of all IPv4 address to MAC address bindings provided by the DHCPv4 service.
show ip dhcp server statistics	Displays count information regarding the number of DHCPv4 messages that have been sent and received

Verify DHCPv4 is Operational

Verify the DHCPv4 Configuration: As shown in the example, the **show running-config | section dhcp** command output displays the DHCPv4 commands configured on R1. The **| section** parameter displays only the commands associated with DHCPv4 configuration.

```
R1# show running-config | section dhcp
ip dhcp excluded-address 192.168.10.1 192.168.10.9
ip dhcp excluded-address 192.168.10.254
ip dhcp pool LAN-POOL-1
 network 192.168.10.0 255.255.255.0
 default-router 192.168.10.1
 dns-server 192.168.11.5
 domain-name example.com
```

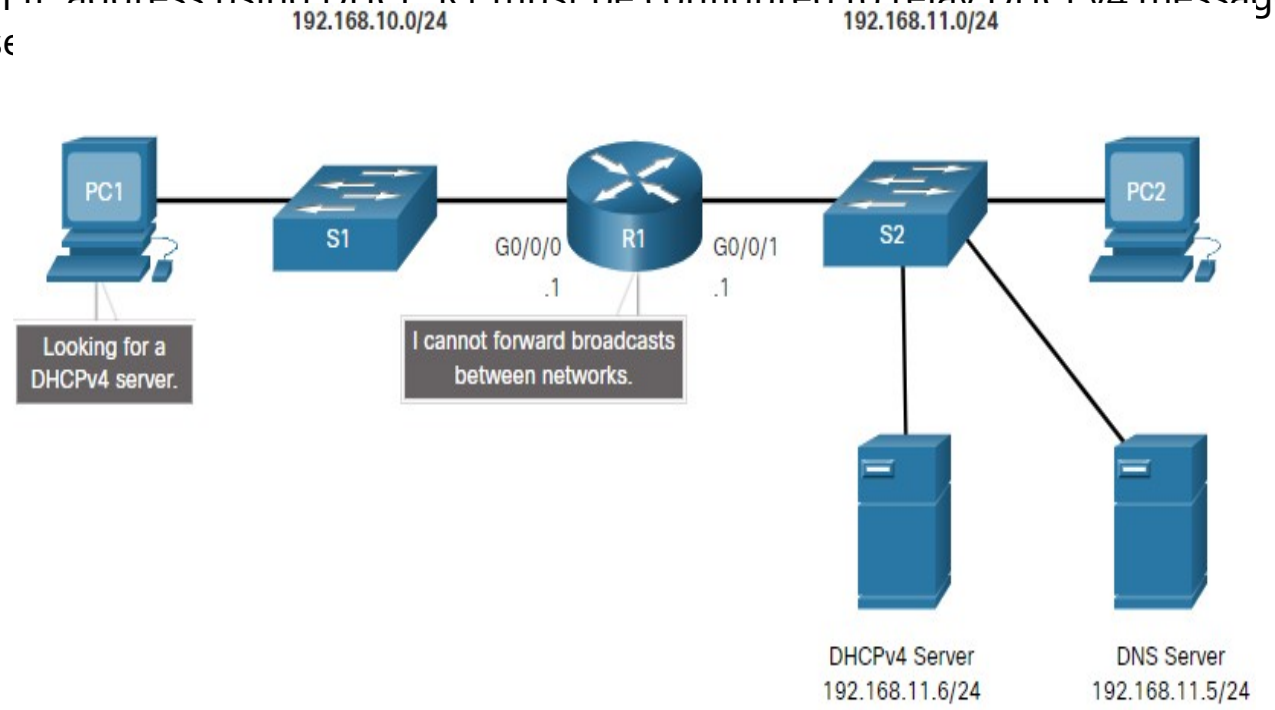
Verify DHCPv4 is Operational (Cont.)

Verify DHCPv4 Bindings: As shown in the example, the operation of DHCPv4 can be verified using the **show ip dhcp binding** command. This command displays a list of all IPv4 address to MAC address bindings that have been provided by the DHCPv4 service.

```
R1# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/              Lease expiration        Type       State      Interface
                Hardware address/
                User name
192.168.10.10   0100.5056.b3ed.d8       Sep 15 2019 8:42 AM     Automatic  Active
GigabitEthernet0/0/0
```

DHCPv4 Relay

- In a complex hierarchical network, enterprise servers are usually located centrally. These servers may provide DHCP, DNS, TFTP, and FTP services for the network. Network clients are not typically on the same subnet as those servers. In order to locate the servers and receive services, clients often use broadcast messages.
- In the figure, PC1 is attempting to acquire an IPv4 address from a DHCPv4 server using a broadcast message. In this scenario, R1 is not configured as a DHCPv4 server and does not forward the broadcast. Because the DHCPv4 server is located on a different network, PC1 cannot receive an IP address using DHCP. R1 must be configured to relay DHCPv4 messages to the DHCPv4 se



DHCPv4 Relay Configuration

- Configure R1 with the **ip helper-address** *address* interface configuration command. This will cause R1 to relay DHCPv4 broadcasts to the DHCPv4 server. As shown in the example, the interface on R1 receiving the broadcast from PC1 is configured to relay DHCPv4 address to the DHCPv4 server at 192.168.11.6.
- When R1 has been configured as a DHCPv4 relay agent, it accepts broadcast requests for the DHCPv4 service and then forwards those requests as a unicast to the IPv4 address 192.168.11.6. The network administrator can use the **show ip interface** command to verify the configuration.

```
R1(config)# interface g0/0/0
R1(config-if)# ip helper-address 192.168.11.6
R1(config-if)# end
R1#
```

```
R1# show ip interface g0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is 192.168.11.6
(output omitted)
```

HTTP

HTTP

- Stands for Hyper Text Transfer Protocol
- Uses the services of TCP on well-known port 80
- HTTP messages are read and interpreted by the HTTP server and HTTP client(browser)
- HTTP is a stateless protocol as HTTP server does not store any information for the clients
- Uses TCP

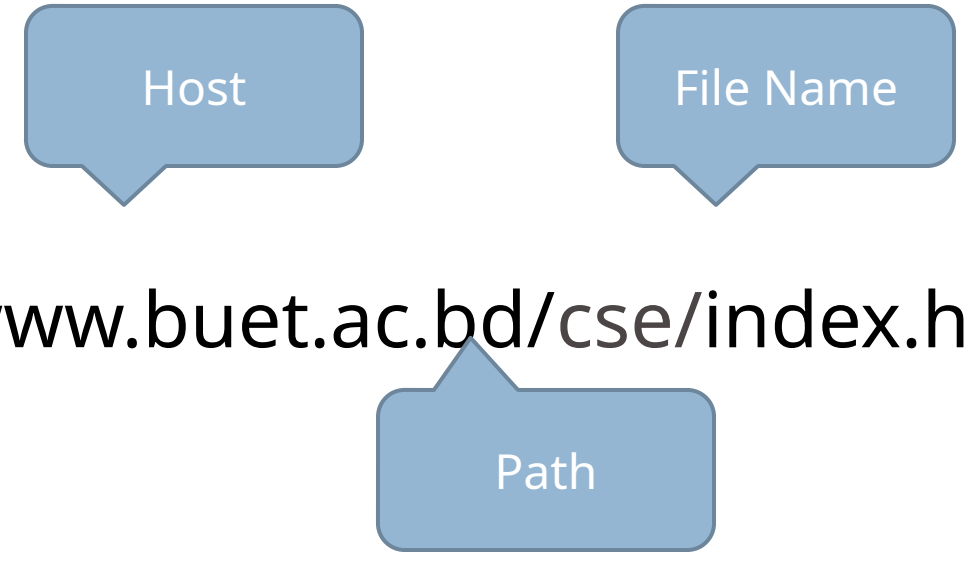
URL

- Abbreviated Form of
  - Uniform Resource Locator
- Components
  - Protocol
  - Host
  - Port (Optional)
  - Path
  - File Name

URL (Explained)

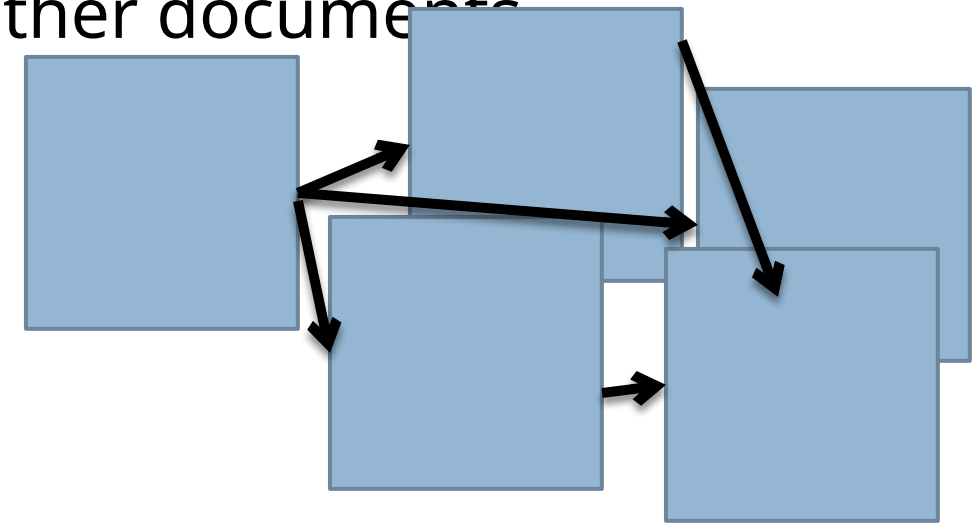
- Protocol
  - The client/server program used to retrieve the document
- Host
  - The computer on which the information is located
- Port
  - Optionally contains the port number of the server
- Path
  - The pathname of the file where the information is located
  - Slashes separates the directories

URL (Explained)

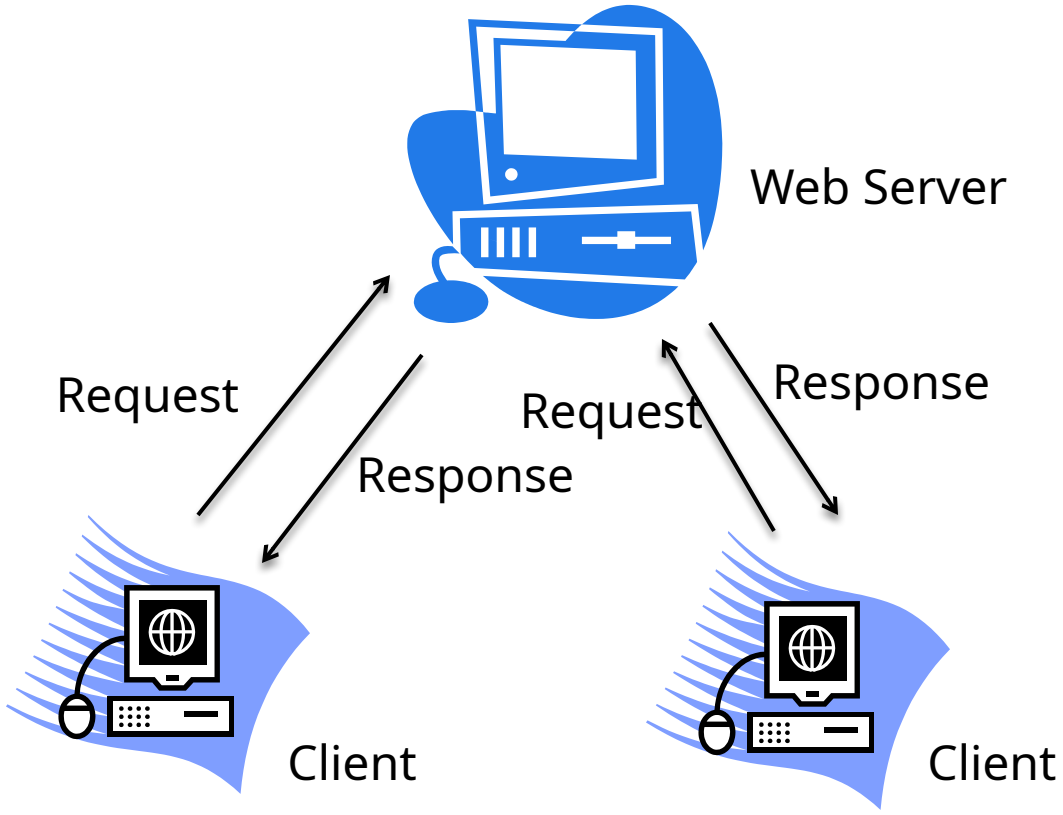


Hypertext

- Obviously, Hypertext file is a Collection of Text, like Document File.
- What is Extra?
  - Certain words within the text are marked as *links* to other areas of the current document or to other documents



HTTP Communication



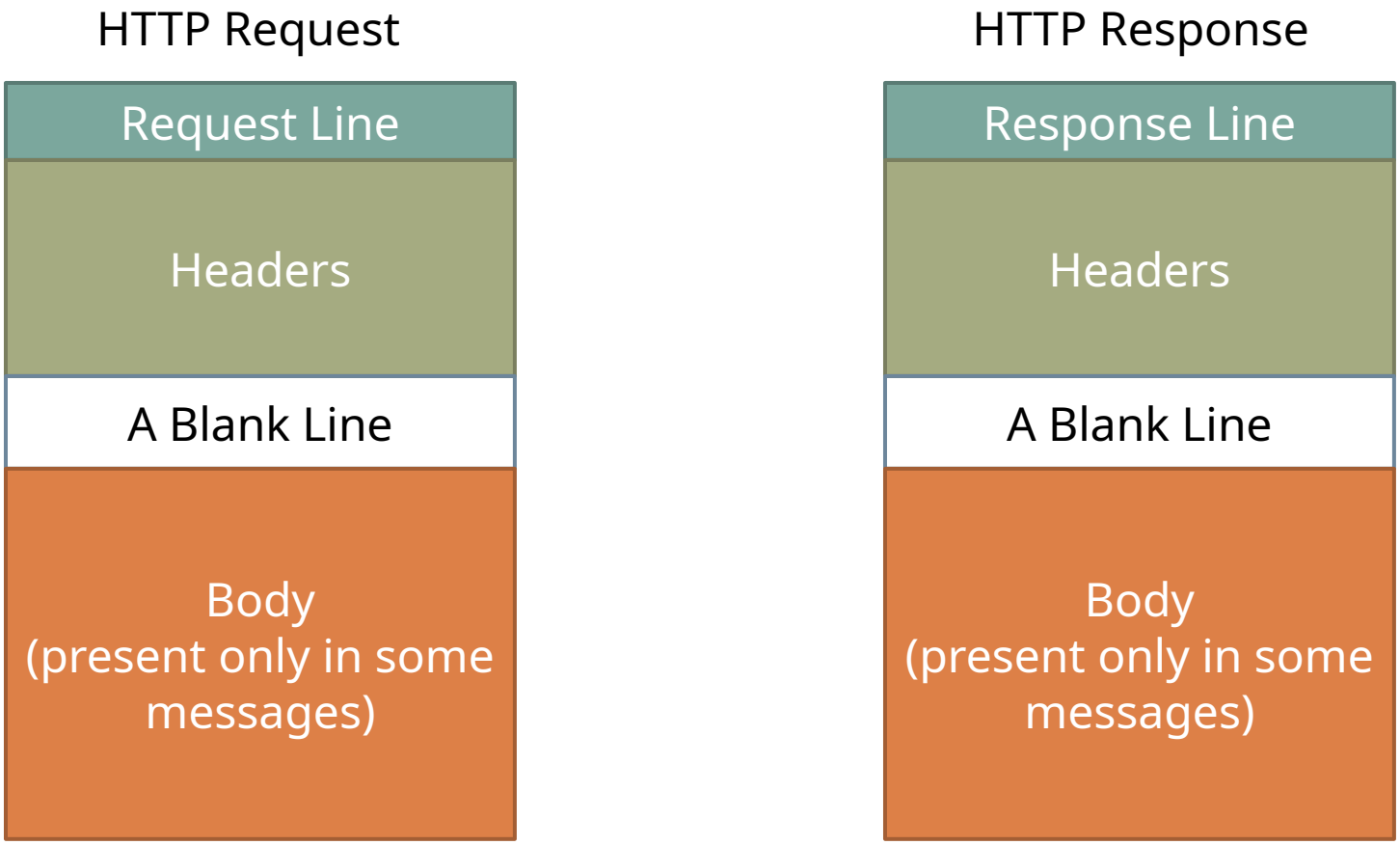
Non-persistent Connection

- One TCP connection is made for each request/response
- Strategy
  - The client opens a TCP connection and sends a request
  - The server sends the response and closes the connection
  - The client reads the data until it encounters and end-of-file marker; it then closes the connection
- HTTP prior to version 1.1 specified non-persistent connection
- Disadvantages
  - High overhead for server to maintain separate connections for all separate requests between a specific host.
  - Slow start

Persistent Connection

- 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
- HTTP version 1.1 specifies persistent connection as default

HTTP Message Format



Request Line

- Request Type- Identifies the request type
- URL – The URL of the requested object

Method	Action
GET	Requests a document from the server
HEAD	Requests information about a document in the server, but not the document
POST	Sends some information from the client to server
PUT	Sends a document from the server to client
TRACE	Echoes the incoming request
CONNECT	Reserved

Status Line

HTTP Version | Status Code | Status Phrase

Code	Phrase	Description
100	Continue	The initial part of the request is received, and the client may continue its request
200	OK	The request is successful
202	Accepted	The request is accepted, but it is not immediately acted upon
204	No content	There is no content in the body
400	Bad Request	There is a syntax error in the request
401	Unauthorized	The request lacks authorization
404	Not found	The document is not found
500	Internal Server Error	There is error, such a crash, at the server side

HTTP Headers

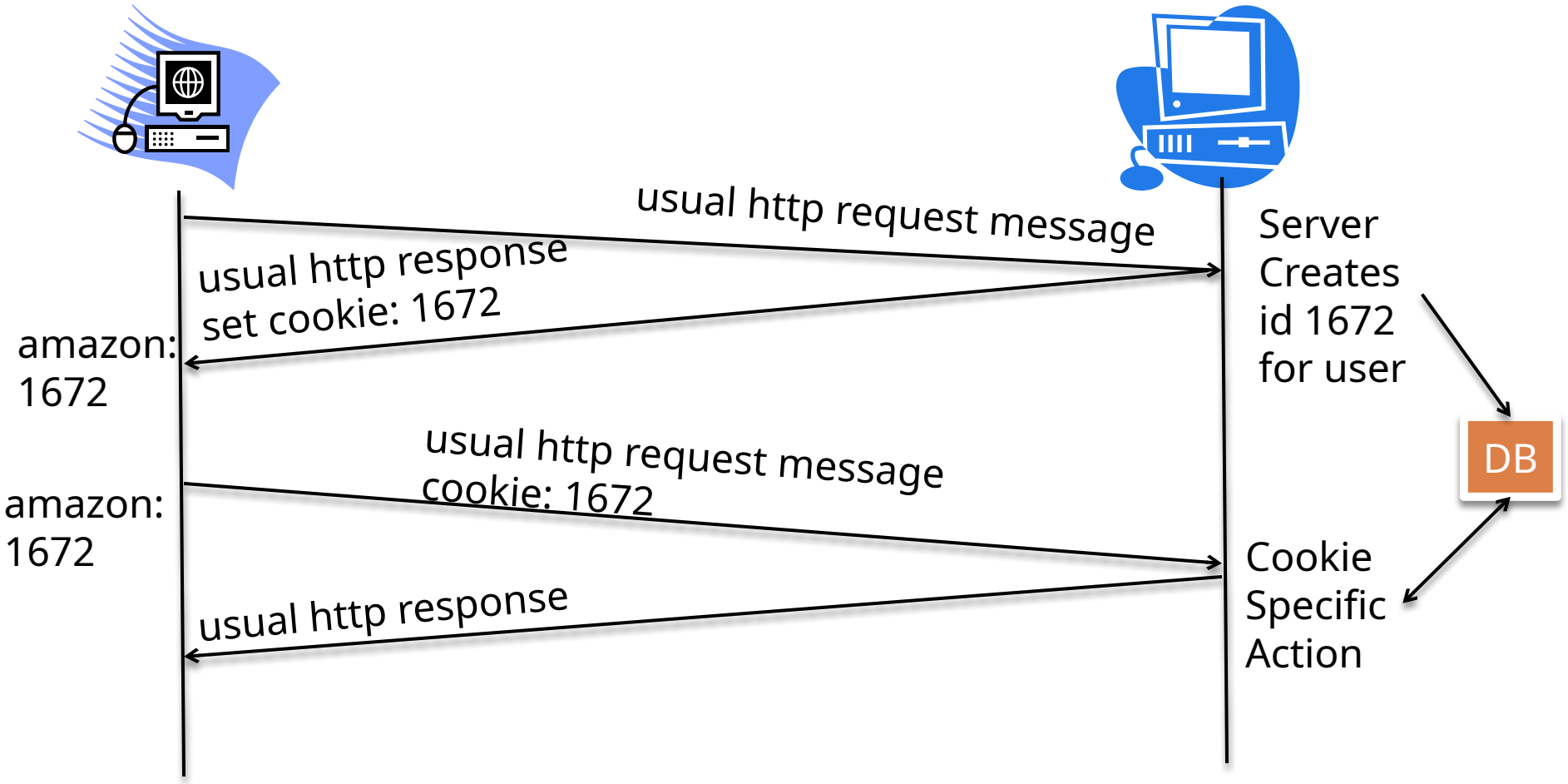
Header Name:	Value

- Three types of Headers
  - General Header
    - Gives general information about the message
    - Can be present in both response and request
  - Request Header
    - Present only in the request message
    - Specifies the client configuration and client's preferred document format
  - Response Header
    - Present only in the response message
    - Provides server configuration and special information about the request

Cookies

- HTTP server is stateless
  - But identification is useful restricting access, personalized content
- Components
  - Cookie header line in HTTP response message
  - Cookie header line in HTTP request message
- Cookie file kept on user's host and managed by user's browser
- Back-end database at Web site

Cookies Explained



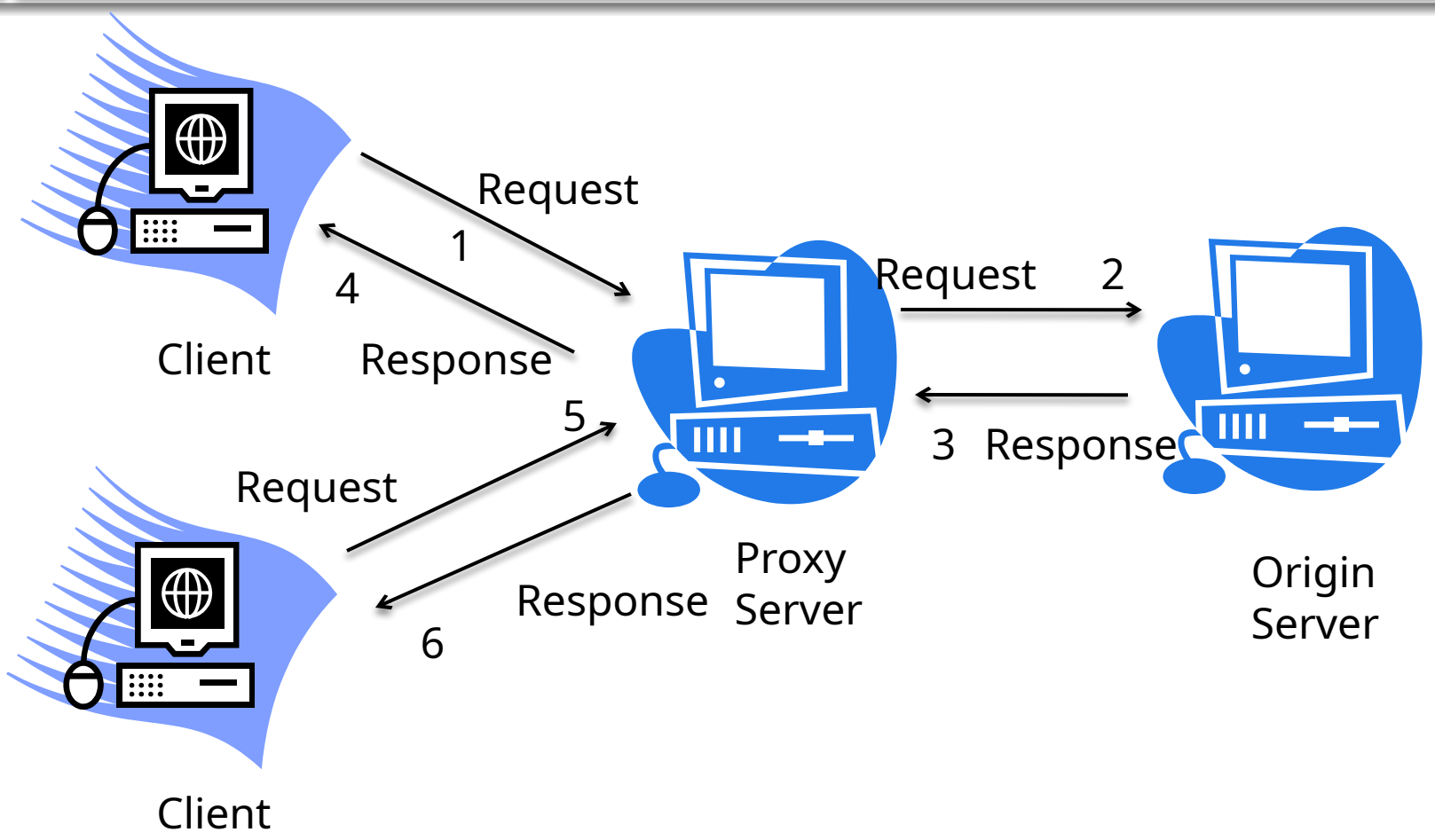
Advantages & Disadvantages

- Advantages
  - Authorization, shopping carts recommendation, user session state
- Disadvantages
  - Websites learns a lot about user

Proxy Server

- User sets browser: all Web accesses via proxy
- Browser sends all HTTP requests to proxy
- If object in proxy, proxy returns the object
- If the object is not there, proxy requests object from origin server, returns object to client, stores
- Reduces
  - Delay
  - Network Traffic

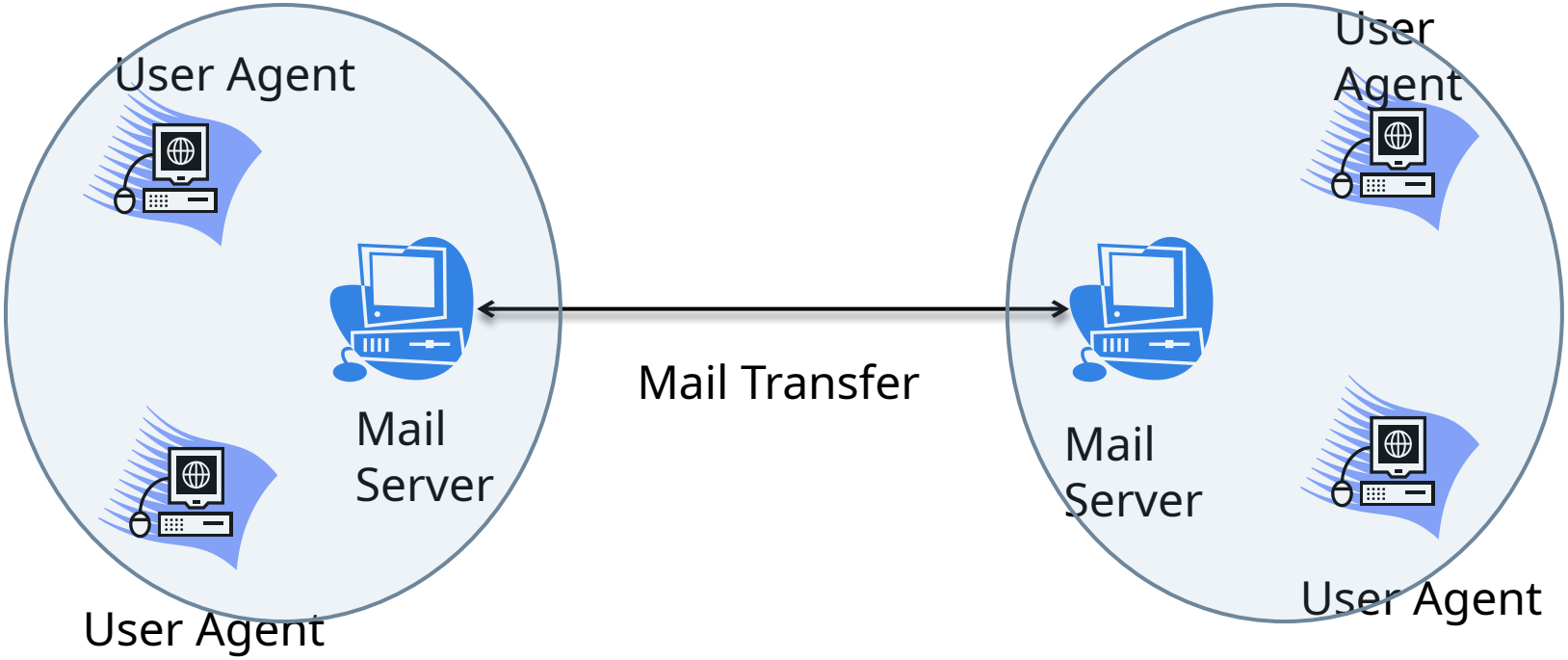
Proxy Server Explained



Electronic Mail

Architecture

- System comprises of
  - User Agents
  - Mail Servers



User Agent

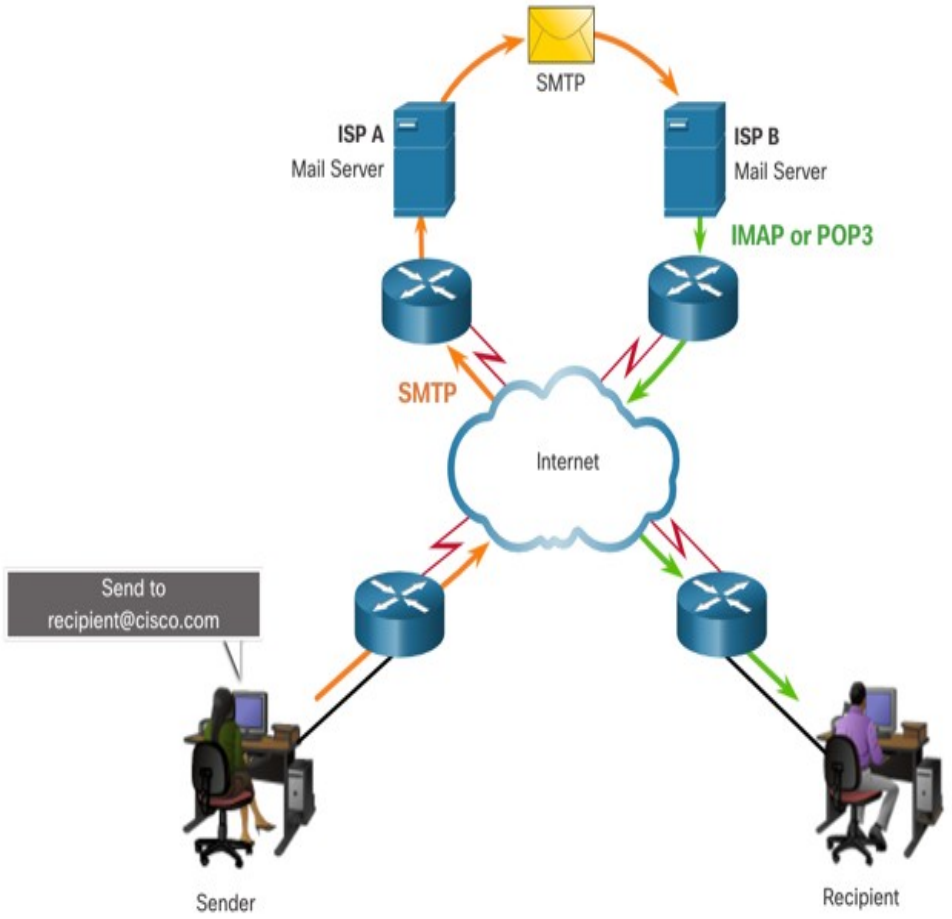
- User Agent
  - Mail Client
  - Purpose - Composing, editing, reading mail messages
  - Eudora, Outlook
- Mail Server
  - Mailbox contains incoming messages for user
  - Message queue of outgoing mail messages
  - SMTP protocol between mail servers to send email messages

Email Protocols

Email is a store-and-forward method of sending, storing, and retrieving electronic messages across a network. Email messages are stored in databases on mail servers. Email clients communicate with mail servers to send and receive email.

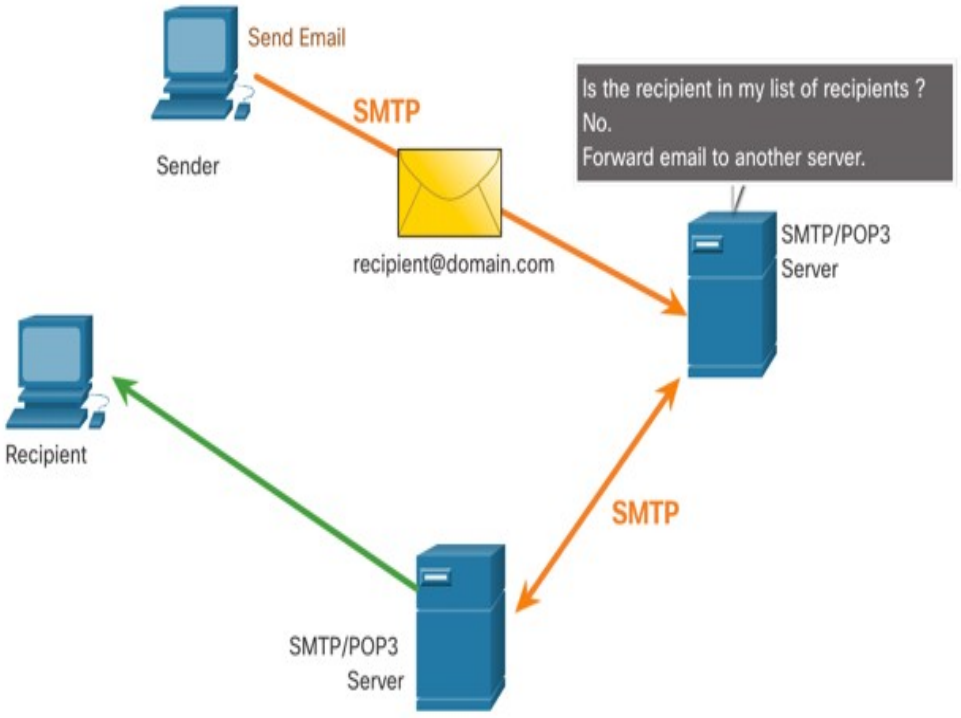
The email protocols used for operation are:

- Simple Mail Transfer Protocol (SMTP) – used to send mail.
- Post Office Protocol (POP) & IMAP – used for clients to receive mail.



SMTP

- When a client sends email, the client SMTP process connects with a server SMTP process on well-known port 25.
- After the connection is made, the client attempts to send the email to the server across the connection.
- When the server receives the message, it either places the message in a local account, if the recipient is local, or forwards the message to another mail server for delivery.
- The destination email server may not be online or may be busy. If so, SMTP spools messages to be sent at a later time.



Note: SMTP message formats require a message header (recipient email address & sender email address) and a message body.

Mail Routing

- Sender uses UA to compose message and send
- Sender's UA sends message to her mail server; message placed in message queue
- Client side of SMTP opens TCP connection with receiver's mail server
- SMTP client sends sender's message over the TCP connection
- Receiver's mail server places the message in receiver's mailbox
- Receiver invokes his user agent to read message

SMTP

- Simple Mail Transfer Protocol
- A simple ASCHII protocol
- After establishing a TCP connection to port 25, the sending machine(client) waits for the receiving machine(server)
- The server starts by sending a line of text giving its identity and telling whether it is prepared to receive mail
- If the server is not ready, the client releases the connection and tries again

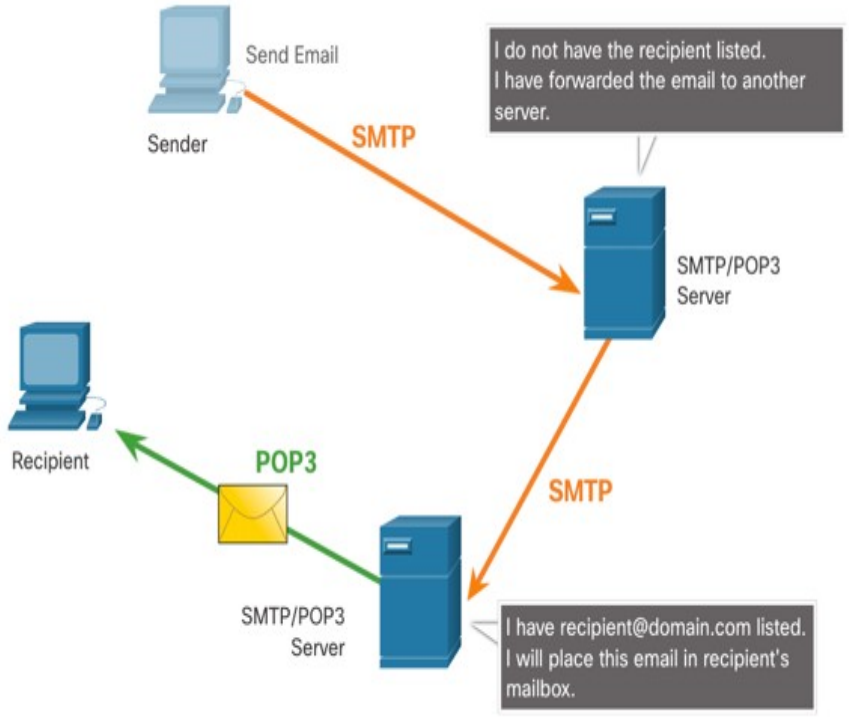
SMTP

- If the server is willing to accept e-mail, the client announces whom the e-mail is coming from and whom it is going to.
- If such recipients exists at the destination, the server gives the client to go-ahead to send the message
- Then the client sends the message and server acknowledges it.

POP

POP is used by an application to retrieve mail from a mail server. When mail is downloaded from the server to the client using POP the messages are then deleted on the server.

- The server starts the POP service by passively listening on TCP port 110 for client connection requests.
- When a client wants to make use of the service, it sends a request to establish a TCP connection with the server.
- When the connection is established, the POP server sends a greeting.
- The client and POP server then exchange commands and responses until the connection is closed or aborted.



Note: Since POP does not store messages, it is not recommended for small businesses that need a centralized backup solution.

POP3

- Post Office 3 Protocol
- Begins when the UA start their mail reader
- The mail reader calls up the mail server and establishes a TCP connection with the mail transfer agent at port 110
- Once the connection is established, POP3 goes through 3 states
  - Authorization
    Authorize the user
  - Transactions
    - Collect mail and mark them for deletion
  - Update
    - Causes mails to be deleted

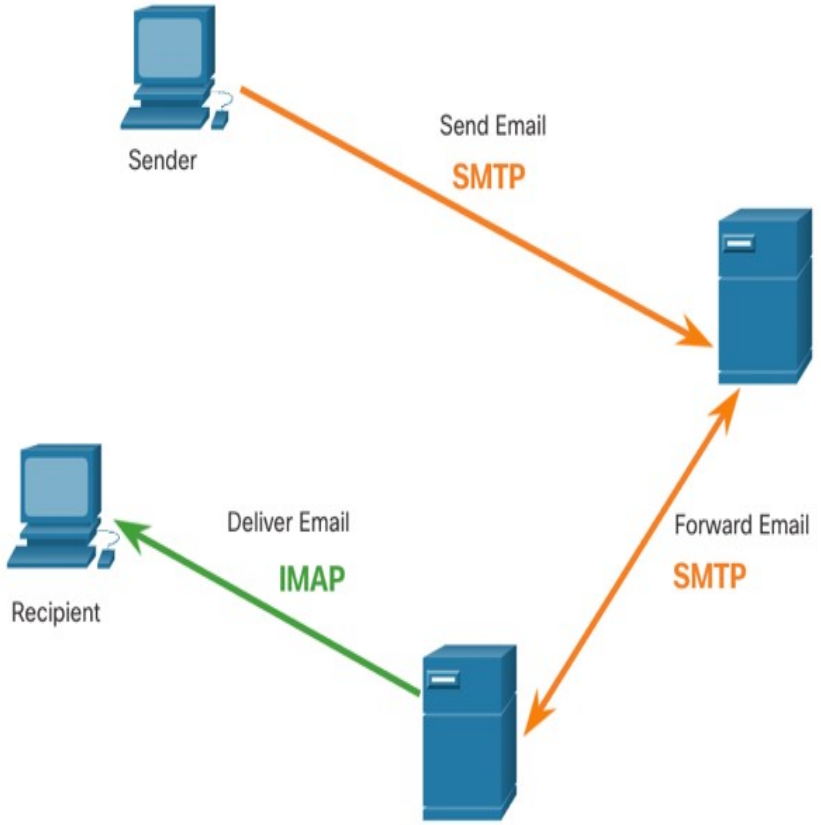
IMAP

- Internet Message Access Protocol
- Assumes all e-mail will remain on the server indefinitely In multipart mailboxes
- Provides mechanism for reading messages or even part of messages
- Provides mechanism of creating, destroying and manipulating multiple mailboxes on the server

IMAP

IMAP is another protocol that describes a method to retrieve email messages.

- Unlike POP, when a user connects to an IMAP server, copies of the messages are downloaded to the client application. The original messages are kept on the server until manually deleted.
- When a user decides to delete a message, the server synchronizes that action and deletes the message from the server.



MIME

- Multipurpose Internet Mail Extensions
- Supplementary protocol to allow non-ASCII data to be sent through e-mail
- Different types of contents are supported by MIME

Type	Subtype	Description
Text	Plain	Unformatted
	HTML	HTML format
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but not ordered
Image	JPEG	Image in JPEG
	GIF	Image in GIF
Video	MPEG	Video in MPEG
Audio	Basic	Single-channel encoding of voice at 8kHz
Application	Postscript	Adobe Postscript
	Octet-	General binary data (8-bit bytes)