BCS502 – COMPUTER NETWORKS

Faculty:

Prof. Ashok Herur

ashok.herur@eastpoint.ac.in

Mobile: 91641 01399

Module 5

Application Layer

Main topics in Module 5

- Client-Server applications and Protocols
- The Web (www) and HTTP
- FTP – File Transfer Protocol
- Electronic mail
- DNS – Domain Name Service
- TELNET
- Secure Shell (SSH)

Application Layer

- A collection of applications that are useful for the users !
- Networks exist to support these applications.
- Popular applications, in the 1970's and 1980's:
  - Text email
  - File transfer
  - Remote access to computers
- In 1990's:
  - World Wide Web (www) – surfing, searching, ecommerce.

Application Layer

- After 2000:
  - VoIP (Voice over IP)
  - Videoconferencing – Skype, Facetime, Google Hangouts, ..
  - YouTube
  - Movies on Demand – Netflix
  - Multiplayer Online games
  - Social networking – Facebook, Instagram, Twitter
  - Messaging Apps - WhatsApp, WeChat
  - Payment Apps – Google Pay, PayTM
  - Transport Apps – Ola, Uber
  - Location-based Apps – Maps, traffic, services (shops, hotels, petrol pumps,..)

Principles of Network applications

- Network application programs should be capable of running on different end systems, and be able to communicate with each other over the network.
- For example, in a Web application, there are two distinct programs that communicate with each other:
  - The Browser program running in the user's host (laptop, smartphone, etc);
  - The Web Server program running in the Web Server host.
- As another example, in a video-on-demand application, like Netflix, there is:
  - A Netflix-provided program running in the user's host (TV, smartphone);
  - The Netflix Server program running in the Netflix Server host.

Network Application Architecture

- Network application architecture is one of the following two types:
  - Client-server architecture
  - Peer-to-Peer (P2P) architecture
- In a Client-server architecture, there is an always-on host, called the Server, which services requests from many other hosts, called Clients.
  - As an example, a Web Server services requests from many browsers running on Client hosts.
  - It responds by sending the requested object.

Client-Server architecture

- Here, the server has a fixed, well-known IP address.
  - Since it is always on, and since it services many requests, the address is fixed for quick access.
- Common applications using this architecture include the Web, FTP, email.
- Often, a single server is incapable of handling the volume of requests.
- There are multiple servers housed in a data centre.
- Further, there could be multiple data centres spread across the globe.
  - Google (search engine) has 19 data centres across the globe.

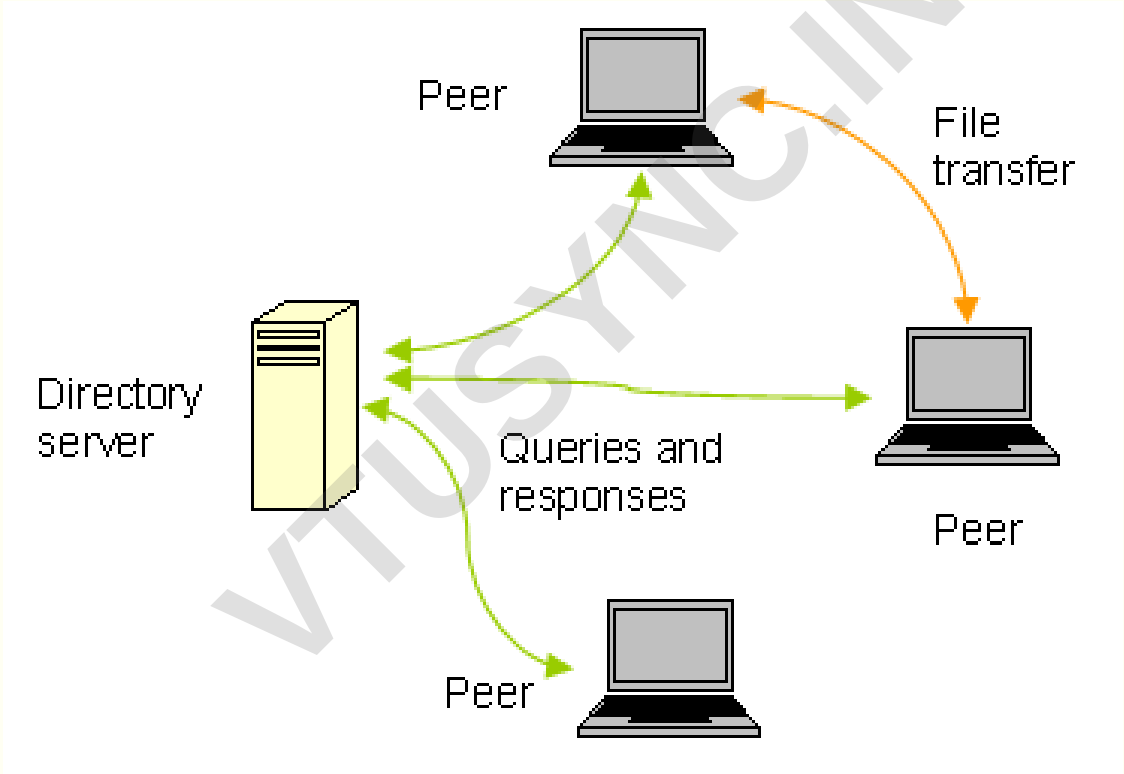
P2P architecture

- Here, there is minimal (or no) reliance on dedicated servers.
- The application exploits direct communication between pairs of intermittently connected hosts, called peers.
- These peers are not owned by the service provider but instead are desktops and laptops owned by the end users (in homes and offices).
- A popular P2P application is the file-sharing application (BitTorrent, InShare, Zapya, etc).
- This architecture is distributed, scalable and cost effective, since it does not need server infrastructure (data centres) and server bandwidth.

P2P architecture

- Since there are no fixed servers, peers must rely on some method to locate fellow peers.
- The most basic approach is a centralized directory where resources are indexed on a central server, and peers query this server for a lookup to find the peer with the desired resource, and then make a connection to the peer.
- BitTorrent uses a centralized directory server, calling it the Tracker.
- Note that while resource lookup is still client-server, the actual resource transmission, which accounts for the bulk of the network capacity usage, is P2P.

Peer discovery



Processes

- When applications communicate with each other, there are processes on both sides that enable the communication (sending and receiving data).
- In applications that use the Client-Server architecture, like the Web, the browser on the client is a client process and the one running on the server is the server process.
- A process sends message into, and receives messages from, the network through a software interface called a socket.
- It is the interface between the Application layer and the Transport layer (in the TCP / IP Reference model).
- It is also referred to as Application Programming Interface (API).

Socket

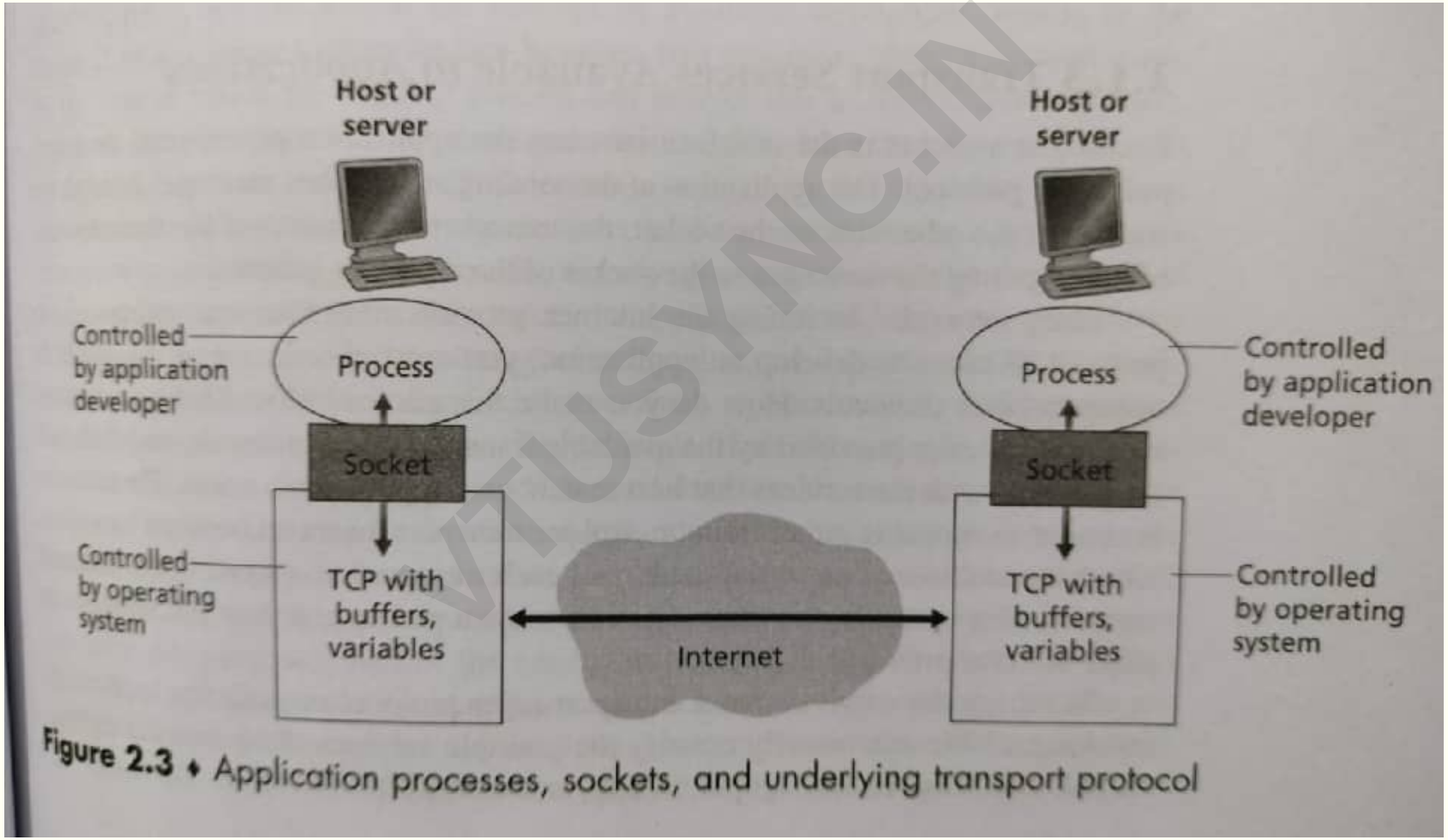


Figure 2.3 ♦ Application processes, sockets, and underlying transport protocol

Socket addressing

- A process on a host has to have an unique identity.
- Two pieces of information are needed here:
  - The address of the host (IP address);
  - An identifier that specifies the process on the host (Port number).
  - The combination of the above two is the socket address.
- Popular applications have been assigned specific Port numbers by IANA:
  - Web server – Port number 80
  - Mail server (using SMTP protocol) – Port number 25.

Transport services available to Applications

Transport services available to Applications

- When a network application is developed, one must choose one of the available Transport layer protocols.
- This is done by studying the services provided by those protocols, and choose the one that best suits the needs of the application.
  - Like how we choose to travel to a distant place – By our own car, or a bus, train, plane, etc.
  - Each one has different pros and cons, and none is the ideal mode of transport for everyone.

Transport services available to Applications

- The services offered by the Transport layer protocols can broadly be classified under the following heads:
  - Reliable data transfer
  - Throughput
  - Propagation time (delay)
  - Security aspects

Transport services – Reliable data transfer

- Reliability means that:
  - Data that is corrupted due to noise should be detected and dealt with;
  - Packets are not lost (due to buffer overruns or due to an error in the IP Header);
  - Packets are not delivered (to the Application) in improper sequence.
- Many applications (Financial transactions, email, remote host access, etc) do not tolerate the data loss mentioned above.
- Some loss-tolerant applications (notably multimedia applications such as conversational audio / video) can tolerate some amount of data loss.

Transport services – Throughput

- Throughput is the rate at which the sending process can deliver bits to the receiving process.
- The available throughput will fluctuate with time because:
  - The bandwidth along the network path is shared with other sessions (of differing throughput needs);
  - Sessions come on, get completed and closed at different points in time.
- Can an application request a guaranteed throughput from the Transport layer protocol?
  - And does it need such a guarantee?

Transport services – Throughput

- Bandwidth-sensitive applications are those that are virtually useless in the absence of the required throughput.
  - Many multimedia applications fall in this category, though a few can adjust the coding schemes and settle for a slightly lower throughput.
- Bandwidth-elastic applications are those that can make use of as much, or as little, throughput that is available at the moment.
  - Email, file transfer, Web transfer fall in this category.

Transport services – Propagation time (delay)

- A transport-layer protocol can also provide / should provide some timing guarantees.
- Interactive real-time applications definitely need such a guarantee.
  - Eg.: Internet telephony, videoconferencing, multiplayer games, etc.
- Even for non-real-time applications, a lower delay would always be preferable, though a tight constraint is not placed.

Transport services – Security

- A transport protocol can provide applications with one or more security services, like:
  - End-to-end encryption;
  - End-point authentication;
  - Data integrity (a check to find out if the data has been tampered with).
- Note that the traditional TCP does not provide a facility for the above features.
- If they are required, then the application should use an enhanced feature of TCP, called Transport Layer Security (TLS).
  - Note that TLS is not a third protocol (besides TCP and UDP).
  - The special TLS code should be used in the application development.

Transport services – Security

- TLS has its own socket API that is similar to the traditional socket API.
- When an application uses TLS, the sending process passes cleartext data to the TLS socket.
- TLS in the sending host then encrypts the data and passes it on to the TCP socket.
- The reverse happens on the receiver side.

Transport services – Requirements of network applications

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1 Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Smartphone messaging	No loss	Elastic	Yes and no

Figure 2.4 ♦ Requirements of selected network applications

The Web and HTTP

The World Wide Web (www)

- The World Wide Web (WWW), commonly referred to as the web, is a vast and interconnected network of digital information that is accessible through the internet.
- It consists of a collection of web pages, documents, multimedia content, and resources linked together using hyperlinks.
- The web allows users to access, share, publish, and interact with a diverse range of content, including text, images, videos, audio, and interactive applications.

Components of WWW

- Web pages are individual documents containing information, often presented in HTML format, that can include text, images, multimedia, and links.
- These web pages are grouped together to form websites, which are hosted on web servers and accessible via web browsers.
- Hyperlinks, often called links, are clickable elements within web content that connect to other web pages, websites, or resources.
- Clicking on a hyperlink navigates the user to the linked content, enabling seamless exploration across the web.

Components of WWW

- Uniform Resource Locator (URL) is a web address that specifies the location of a specific resource on the web.
- URL consists of a protocol (such as HTTP or HTTPS), a domain name (e.g., www.wipro.com), and a path to the resource.
- Web browsers are software applications used to access and view web content. They interpret HTML and other web technologies to render web pages in a readable and interactive format for user.
- Web servers are computers that store web content and respond to user requests by sending the requested web pages and resources back to the user's browser.

Components of WWW

- Hyper Text Transfer Protocol (HTTP) and HTTPS: HTTP is the protocol used for transferring data between a web browser and a web server. HTTPS is a secure version of HTTP that encrypts data to enhance security and privacy during data transmission.
- Search engines index web content, making it searchable and discoverable for users. They use algorithms to rank and present search results based on user queries, enabling efficient access to relevant information.
  - Search Engine Optimization (SEO) is the practice of optimizing websites to improve their visibility in search engine results. By following SEO best practices, website owners increase the chances of their sites appearing higher in search rankings.

SEO

- When website owners implement the SEO strategies effectively, they send signals to search engines that their website is valuable and relevant to specific search queries.
  - For example, if a website sells organic skincare products, they would want to optimize their website for keywords related to organic skincare, such as “natural skincare,” “chemical-free skincare,” or “organic beauty products.”
  - By strategically placing these keywords in their website’s content and URLs, the website owner increases the likelihood of their site ranking higher when users search for those terms.
- Additionally, search engines consider other factors when determining website visibility, such as the website’s loading speed, mobile-friendliness, and user experience.

Web cache

- A web cache is a server computer located either on the public Internet or within an enterprise that stores recently accessed web pages to improve response time for users when the same content is requested within a certain time after the original request.
- Most web browsers also implement a browser cache by writing recently obtained data to a local data storage device
- Enterprise firewalls often cache Web resources requested by one user for the benefit of many users.

Key components of a website

- HTML (Hypertext Markup Language): HTML determines how the different elements are organized (defines its structure and content) and presented to the visitors.
- CSS (Cascading Style Sheets): CSS is responsible for the visual presentation and layout of a website. It adds style (font, spacing, etc) colour, and aesthetic appeal to the web pages, making them visually appealing and engaging.
- JavaScript: JavaScript enhances the user experience by adding interactivity and dynamic elements to web pages. It allows websites to respond to user actions, such as clicking on buttons or scrolling through content.

HTTP

- Hypertext Transfer Protocol (HTTP) is an Application layer protocol for distributed, collaborative, information systems like the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access, for example by a mouse click or by tapping the screen.
- HTTP/1 was finalized as version 1.0 in 1996. It evolved (as version 1.1) in 1997.
  - Its secure variant named HTTPS is used by more than 85% of websites.
- HTTP/2, published in 2015, provides a more efficient version of HTTP/1.
  - As of January 2024, it is used by 36% of websites and supported by almost all web browsers (over 98% of users).
- HTTP/3 was published in 2022.
  - It is now used on 28% of websites, and is supported by most web browsers.

HTTP

- HTTP functions as a Request - Response protocol in the Client-Server model.
- The client (process on a browser) submits an HTTP *request* message to the server.
- The server, which provides *resources* such as HTML files and other content returns a *response* message to the client.
- HTTP defines the structure of these messages, and how the messages are exchanged.
- It uses TCP as the transport layer protocol for a reliable, end-to-end transfer.

HTTP – Persistent and non-persistent connections

- As with the TCP handshake, the Request - Response way of working takes time to initiate and transfer the requested web page.
- Therefore, in many applications, where the client and the server interact for an extended period of time, the TCP connection is kept “open” for a (configurable) amount of time.
  - This is called Persistent connection (default mode of operation)
- On the other hand, in non-Persistent connection mode, a new TCP connection is opened for each request.

HTTP – Persistent and non-persistent connections

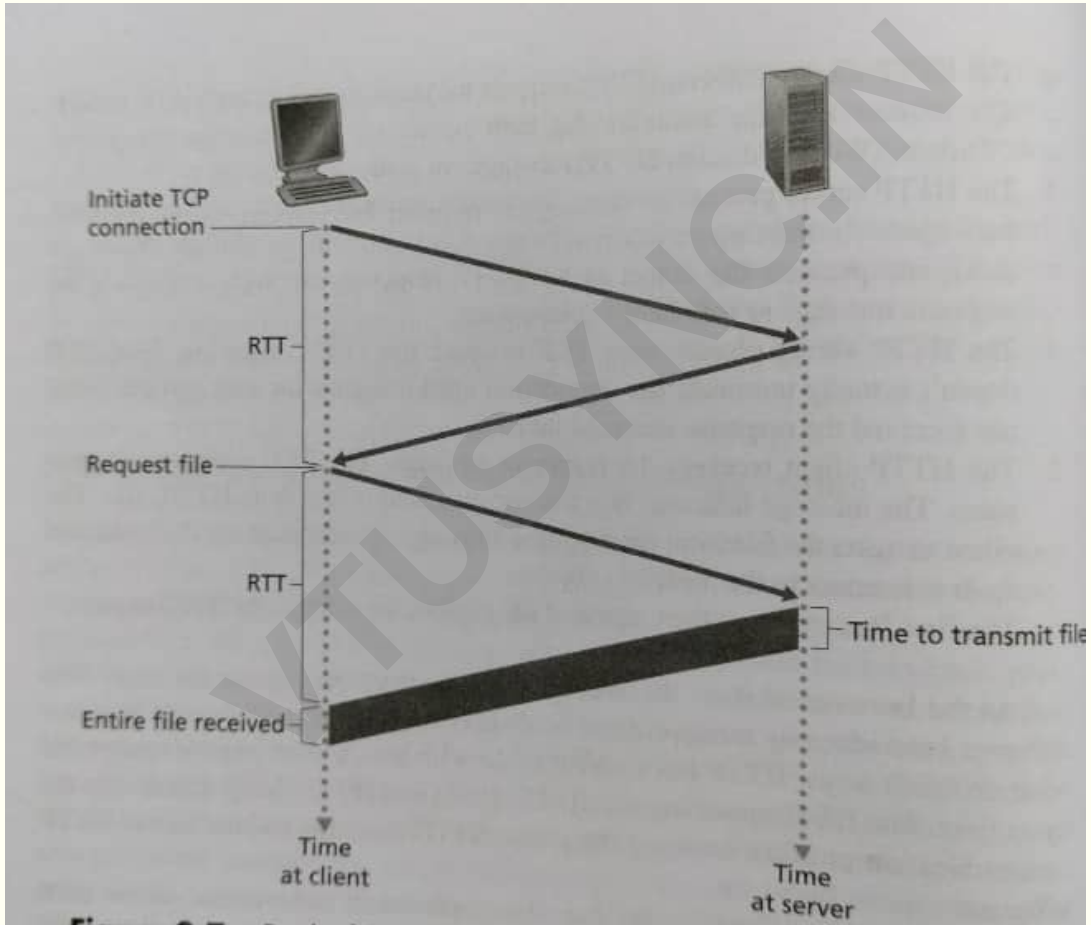


Figure 2.7 ♦ Back-of-the-envelope calculation for the time needed to request and receive an HTML file

HTTP – Message format

- An HTTP request contains a series of lines that each end with a Carriage return (cr) character (takes to the beginning of a line), followed by a Line feed (lf) character (takes to next line)
- The first line is called the Request line, and subsequent ones are called Header lines.
- The Request line has 3 fields: Method field, the URL field and the HTTP version field.
- The Method field can take on different values, but generally is GET, but can also be PUT, DELETE, POST, and HEAD.
  - The GET method is used when the browser requests something that is identified in the URL field.
  - The PUT method allows an user to upload an object to a specific path on the server.

HTTP – Request Message format

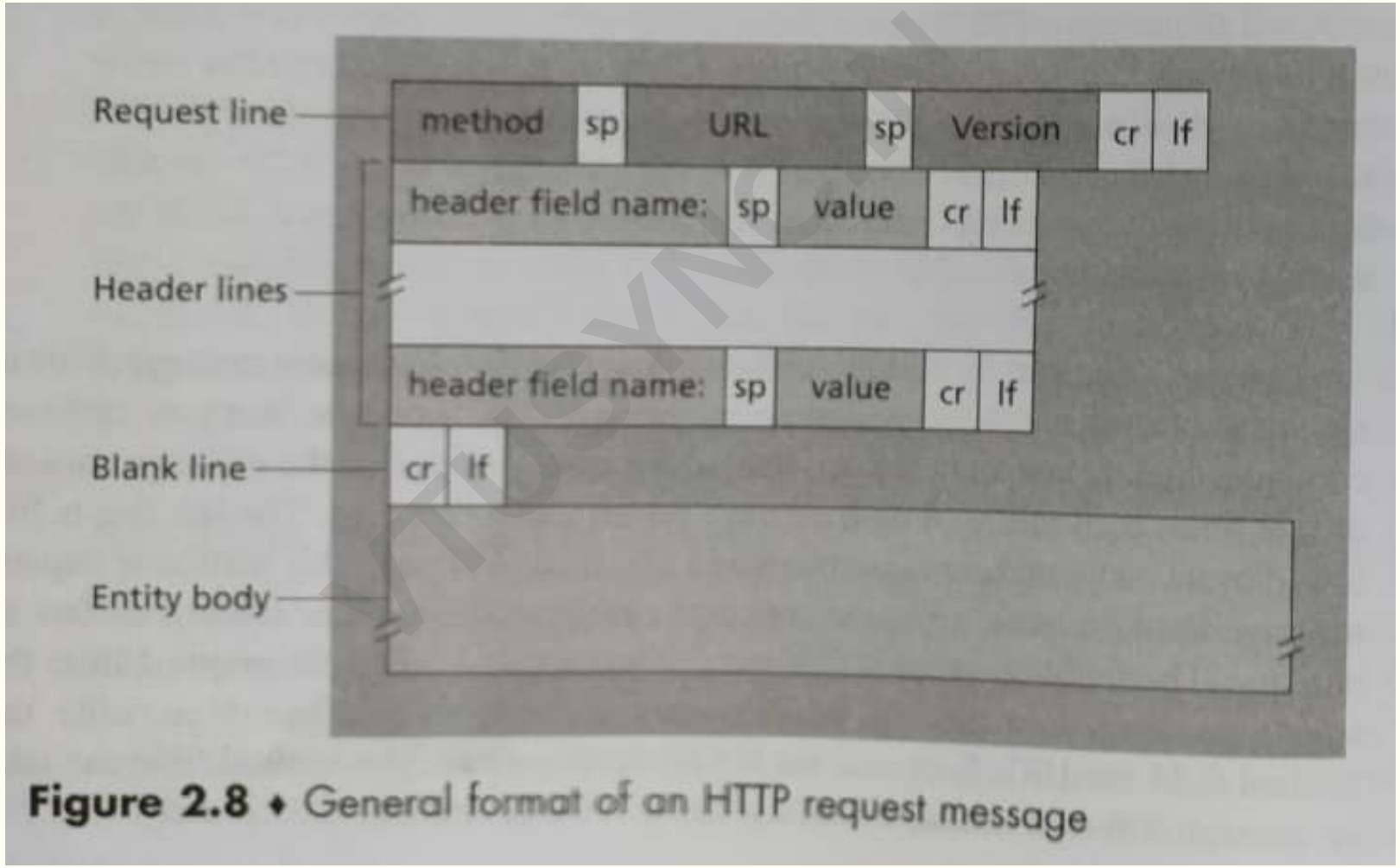


Figure 2.8 ♦ General format of an HTTP request message

HTTP – Request Message format

- The DELETE method is used to delete that one had earlier PUT on the Web.
- The POST method is used when the user fills out a Form with more details for a specific search (the Entity body is now used to describe the Request in detail, by including key search words).
- The HEAD method is used by developers for debugging, wherein the request is responded with an HTTP message but without the requested object.

- The Header lines contain information about various things, including:
  - The host on which the object resides (for use of Web proxy caches);
  - If the connection should be Persistent or Non-persistent;
  - The browser type (Chrome, Firefox, Mozilla,..) that is making the request;
  - Language preference (if the requested object is available in that language).

HTTP – Response Message format

- The first line is called the Status line, and subsequent ones are called Header lines.
- The Status line indicates the HTTP version (/1, /1.1, /2 or /3), followed by the code and phrase representing the status of the Request (found what was requested, not found, not found in the specific language, etc).
- The Header part contains details like Time Stamp, Server type (Apache, etc), length of the content (object) in bytes, the content type (Text /HTML, etc), the Last-modified time, etc.
- The Entity body is a large part of the message and contains the requested object itself.

HTTP – Response Message format

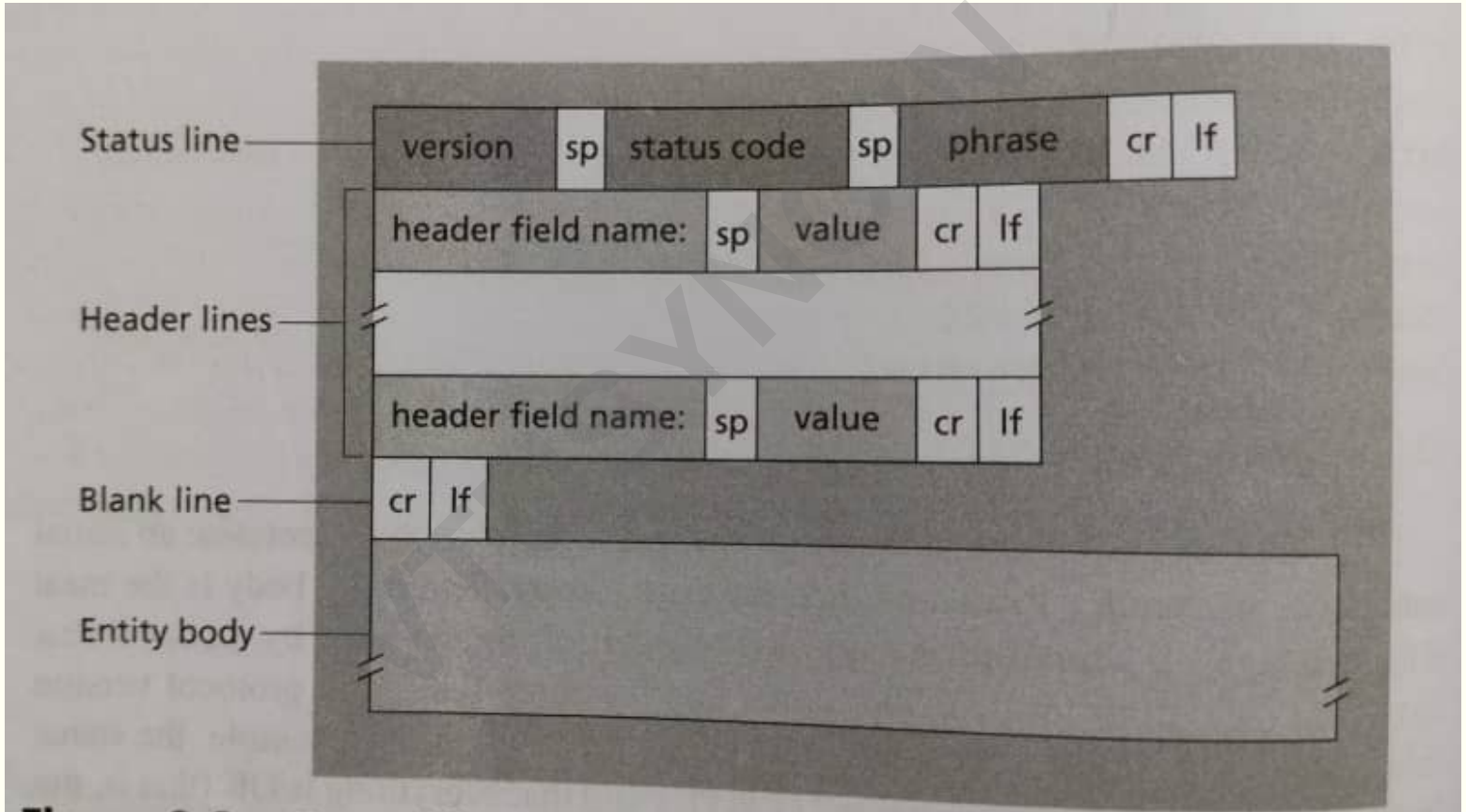
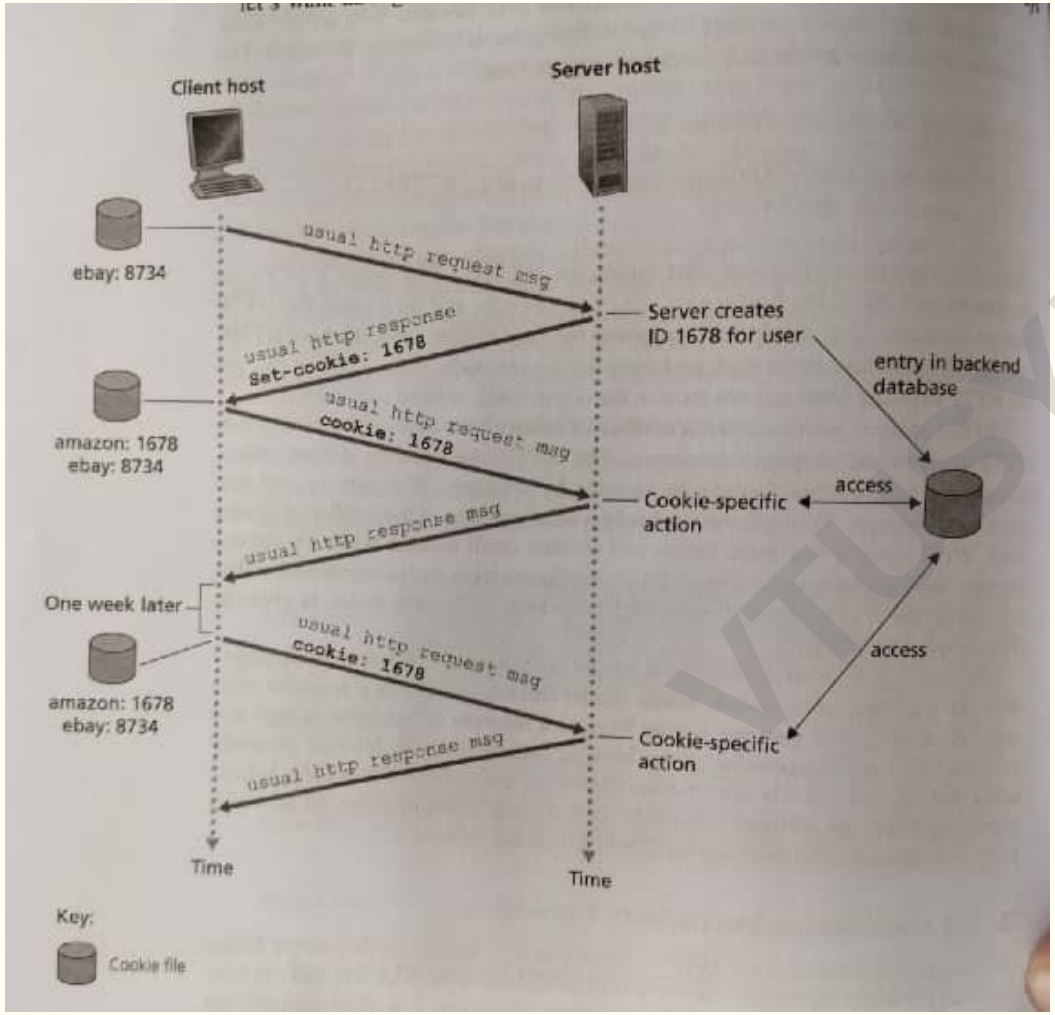


Figure 2.9 ♦ General format of an HTTP response message

Cookies

- Cookies are small files of information that a web server generates and sends to a web browser.
- Web browsers store the cookies they receive for a predetermined period of time, or for the length of a user's session on a website.
- They attach the relevant cookies to any future requests that the user makes to the web server.
- Cookies help inform websites about the user, enabling the websites to personalize the user experience.
  - For example, ecommerce websites use cookies to know what merchandise users have placed in their shopping carts.
  - In addition, some cookies are necessary for security purposes, such as authentication cookies.

Cookies



- The customer has earlier shopped on ebay and has a cookie 8734.
- Now, when he tries to buy on Amazon (with a http request), the server creates a cookie 1678 and tells the client browser to note it down (in a cookie file).
- All subsequent requests (within a specified time) will contain this cookie number in the header line.
- The cookie is stored in the database and also tagged with the transactions.

Electronic mail (email)

Electronic mail

- Most internet systems use Simple Mail Transfer Protocol (SMTP), an application layer protocol, to transfer mail from one user to another.
- At a high-level view, the Internet email system has 3 major components:
  - User agents
  - Mail servers
  - SMTP
- Microsoft Outlook, Apple Mail, Web-based Gmail, Gmail app on a smartphone are some of the popular User Agents.

Electronic mail

- When an user has completed composing the mail, his / her User Agent sends the mail to it's own, always-on server where it is placed in the outgoing message queue.
- When the turn comes up, the message is forwarded to the server of the recipient, using the SMTP protocol (over a TCP connection).
  - If it cannot be delivered for any reason, it retries after every 30 minutes.
  - If it cannot be delivered within a set time (a day or two), the sender is informed about the inability to deliver the message.
  - When delivered, it is stored in the mailbox of the recipient (like a mailbox in the physical Post Office).

Electronic mail

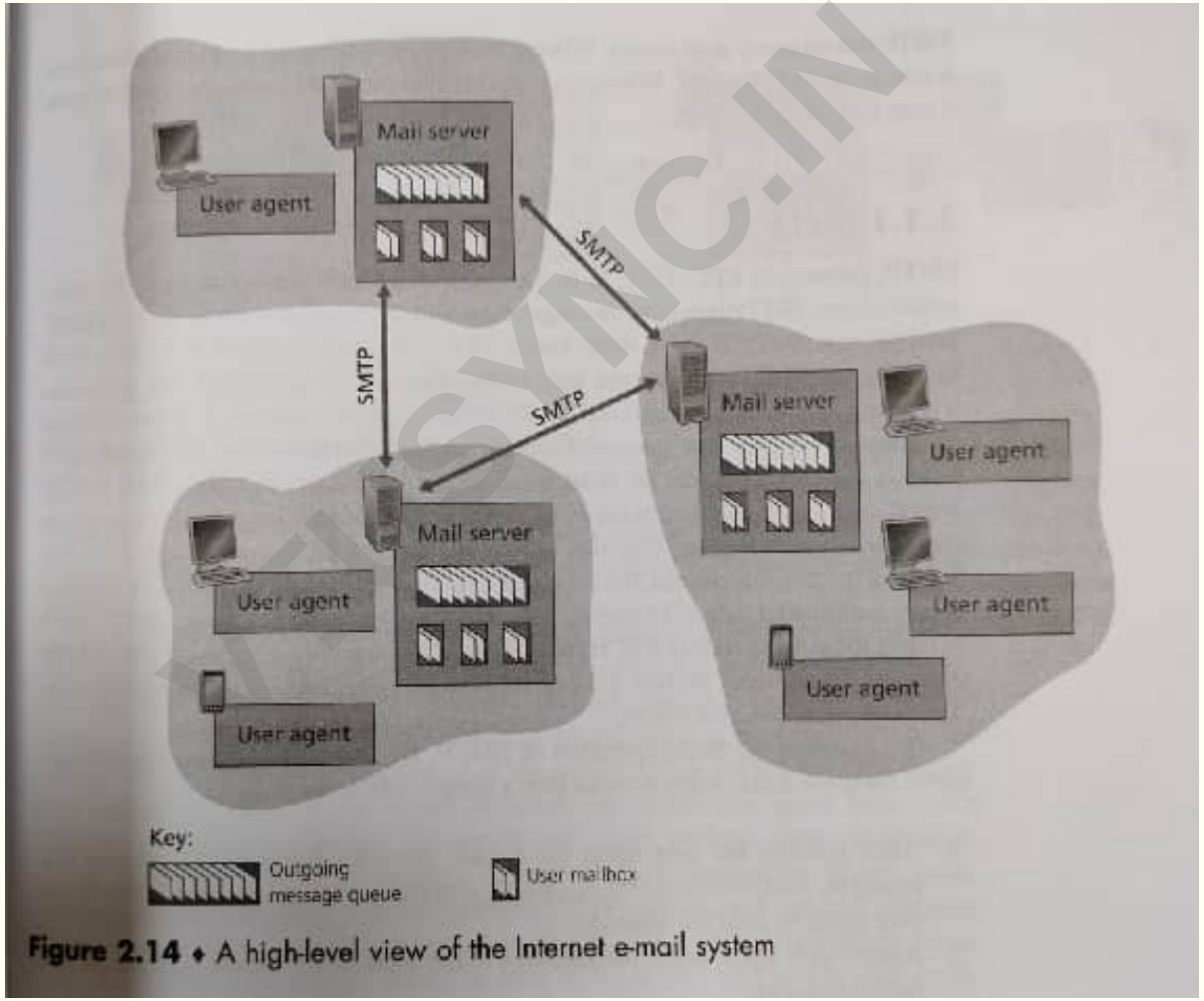


Figure 2.14 ♦ A high-level view of the Internet e-mail system

Electronic mail

- When the recipient comes online, his / her user agent retrieves the message from the mailbox in his / her server, using one of the following protocols:
  - the earlier POP (Post Office Protocol), or
  - the newer IMAP (Internet Message Access Protocol), or
  - HTTP (when the recipient is using a web-based email or a smartphone app)
- When POP is used, the recipient can choose to keep a copy or to delete it from the server after it is downloaded.
  - Further, the mail server can only be accessed from one device.
- However, in the case of IMAP, the copy is always maintained for some management functions, and can be accessed from multiple devices.

Domain Name System (DNS)

Domain Name System (DNS)

- Each one of us is known by our name and our Aadhaar number.
  - Which one is more useful? And where?
- Just like humans, hosts on the Internet are also identified by a human-friendly Hostname (of variable length) like www.google.com, but that will not provide any information about where the host is located within the Internet.
- DNS is the “Internet’s Directory Service” that provides the location (IP address) of the host for the transfer of the request or of the information.
  - Routers effectively forward the packets using the fixed-length, hierarchical IP address.

Domain Name System (DNS)

- The DNS is:
  - A distributed database implemented in a hierarchy of DNS servers;
  - An Application layer protocol that allows hosts to query the database;
  - Also used by other Application layer protocols like HTTP and SMTP to translate user-supplied host names to IP addresses.
- A simple design for DNS would have just one DNS server that contains all the mappings; The problems here would be:
  - Risk of failure of the server – everything crashes.
  - Huge traffic volume.
  - Distance between the querying host and the server.
  - Maintenance – Updating changes to the allocated IP addresses.

Domain Name System (DNS)

- In order to deal with the scale of operations, the DNS uses a large number of servers organised in a hierarchical fashion and distributed across the world.
  - No single DNS server has all the mappings for all of the hosts in the internet.
- Broadly, there are 3 classes of DNS servers:
  - Root DNS servers;
  - Top-level domain (TLD) DNS servers;
  - Authoritative DNS servers.

Hierarchy of DNS servers

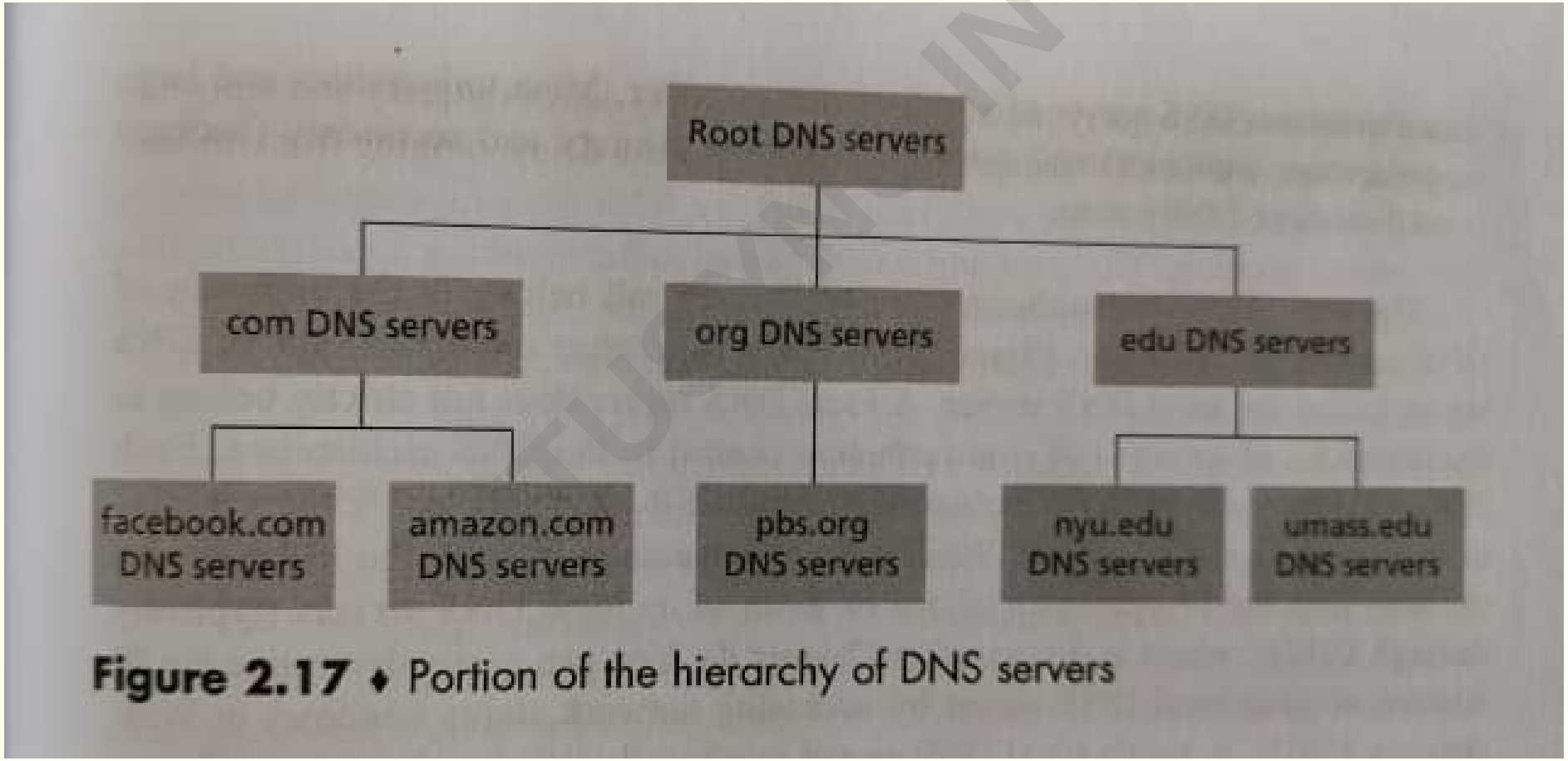


Figure 2.17 ♦ Portion of the hierarchy of DNS servers

Hierarchy of DNS servers - Working

- When a DNS client wants to determine the IP address for a particular hostname (say, www.amazon.com), it contacts one of the nearest Root DNS servers (there are more than 1000 of them scattered across the world, and managed by 12 different organisations, coordinated by IANA.
- The Root DNS server returns the IP addresses of the TLD servers for the top-level domain <.com>, which returns the IP addresses of an authoritative server for amazon.com. This, in turn, returns the IP address for the host name www. amazon.com
- The top-level domains - .com, .edu, .org, .gov, and all the country level domains like .in, .uk, .fr, there is a TLD server (or a cluster of them).

DNS caching

- Whenever a local DNS server receives a DNS reply, containing the mapping of a hostname to an IP address, it stores that in its local memory.
  - This is called DNS caching.
- When another query, for the same hostname, arrives at the DNS server, it can provide the desired IP address immediately.
- Since the mapping is not permanent, the cache is cleared (and updated) regularly.
- The DNS server can also cache the IP addresses of TLD servers, thereby allowing it to bypass the Root DNS servers (in the query chain).

SSH (Secure Shell protocol)

- SSH runs on top of the TCP/IP protocol suite.
- SSH is "secure" because it incorporates encryption and authentication via a process called Public key cryptography.
- Public key cryptography is a way to encrypt data, or sign data, with two different keys.
  - One of the keys, the public key, is available for anyone to use.
  - The other key, the private key, is kept secret by its owner.