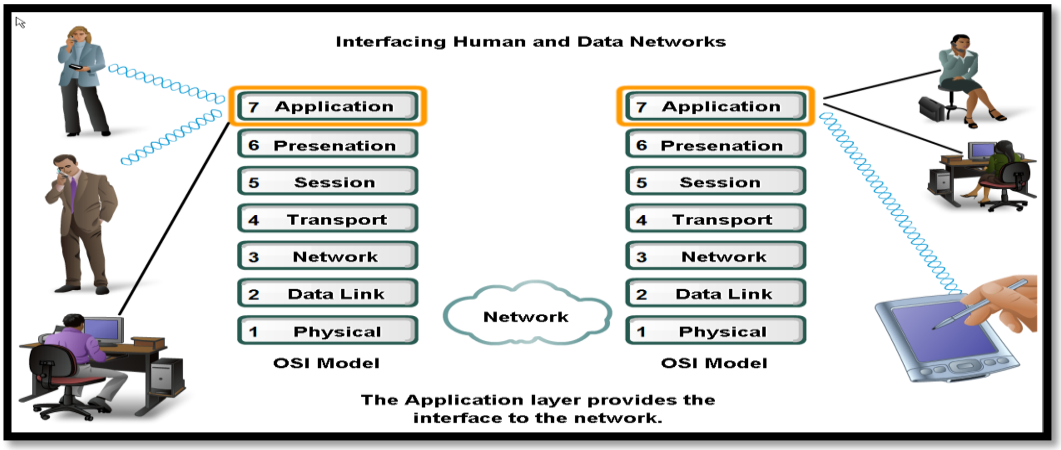
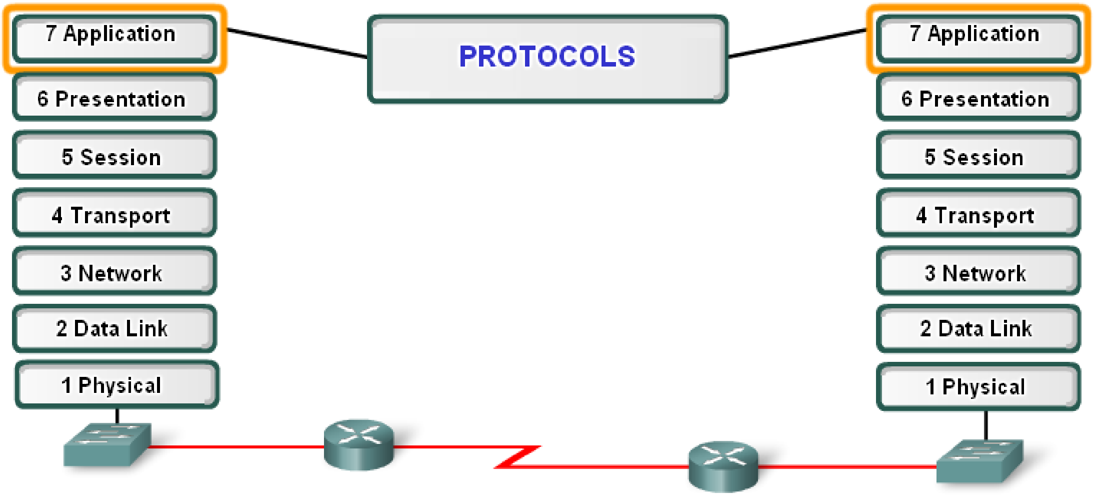
APPLCATION LAYER

The main jobs of application layer are:

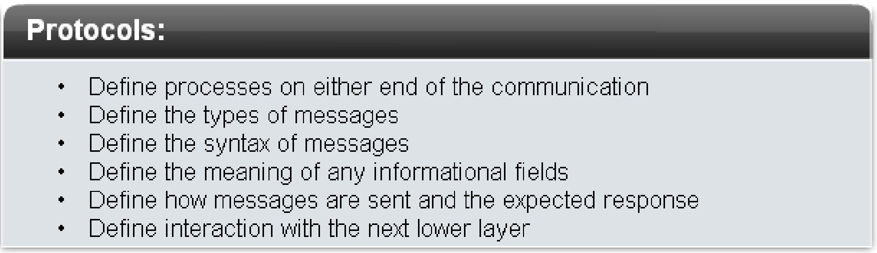
- Allows user to **interface** with the network.
- Provides the **interface** between the applications on either ends of the network.



The Application layer provides the interface to the network.

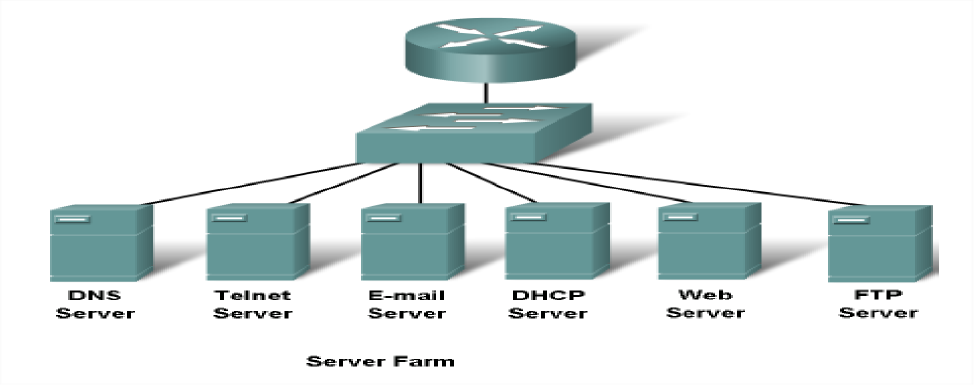


Application layer protocols provide the rules for communication between applications.



Protocols:

- Define processes on either end of the communication
- Define the types of messages
- Define the syntax of messages
- Define the meaning of any informational fields
- Define how messages are sent and the expected response
- Define interaction with the next lower layer



Protocols	Description
DNS	Matches domain names with IP addresses
HTTP	Used to transfer data between clients/servers using a web browser
SMTP & POP3	used to send email messages from clients to servers over the internet
FTP	allows the download/upload of files between a client/server
Telnet	allows users to login to a host from a remote location and take control as if they were sitting at the machine (virtual connection)
DHCP	assigns IP addresses, subnet masks, default gateways, DNS servers, etc. To users as they login the network

Application Layer Software

There are two type of software used in application layer, these are:

1. **Applications**: Provide the human (user) interface. **Relies on lower layers** to complete the communication process.

2. **Services**: Establish an interface to the network where protocols provide the rules and formats that govern how data is treated.

How Data Requests Occur & are filled in application layer?

1. Client/server model
2. Peer-to-peer networking and applications
3. Application layer services and protocols

Client /Server Model

- **Client** – device requesting information (initiates the data exchange) can also UPLOAD data to the servers
- **Server** – device responding to the request

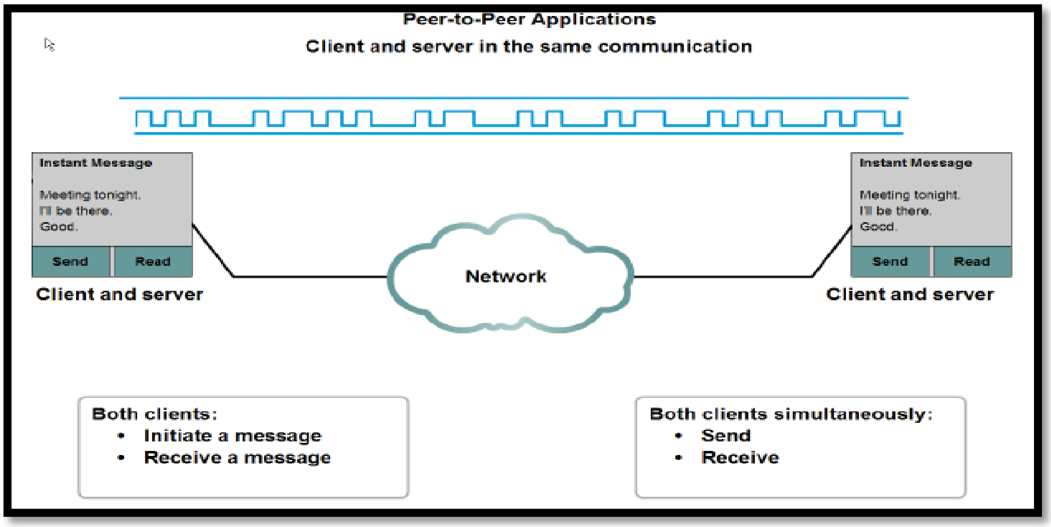
Peer-to-Peer (P2P) Network Model

- Two or more computers are connected and are able to **share resources without having a dedicated server.**
- Every end device can function as a client or server on a 'per request' basis
- Difficult to enforce security and policies
- User accounts and access rights have to be set individually on each peer device.

P2P Applications

- Unlike P2P networks, a device can act as both the client and server within the same communication
- Each device must provide a user interface and run a background service.
- Can be used on P2P networks, client/server networks and across the internet.

P2P Applications Example



Peer-to-Peer Applications

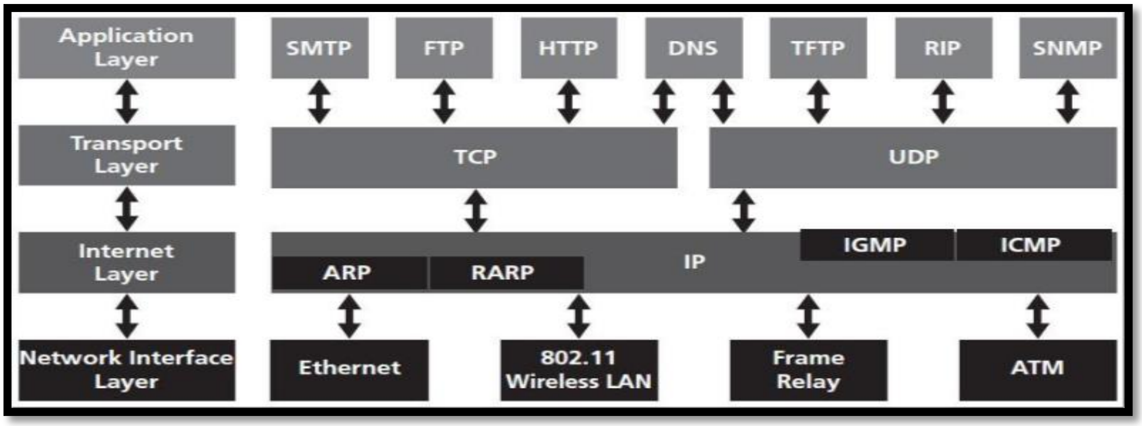
Client and server in the same communication

Both clients:
- Initiate a message
- Receive a message

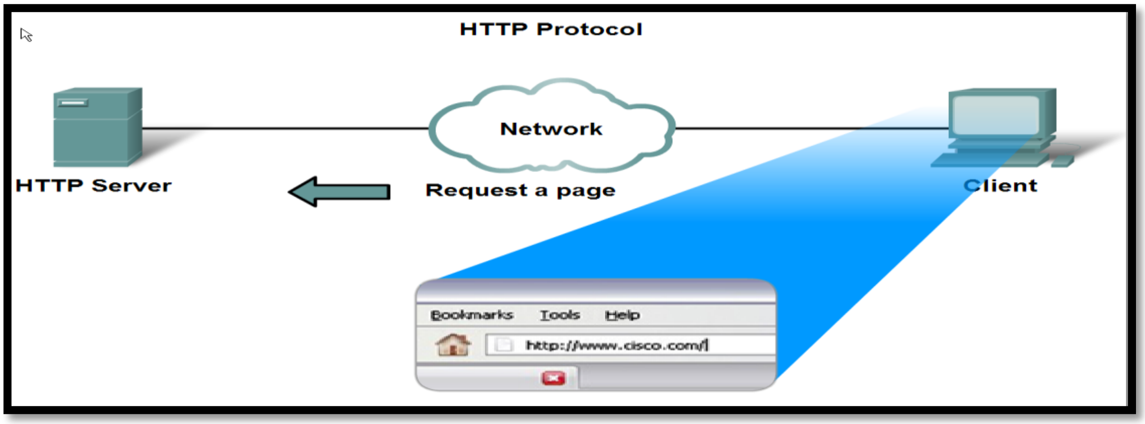
Both clients simultaneously:
- Send
- Receive

<u>Common Port Numbers</u>

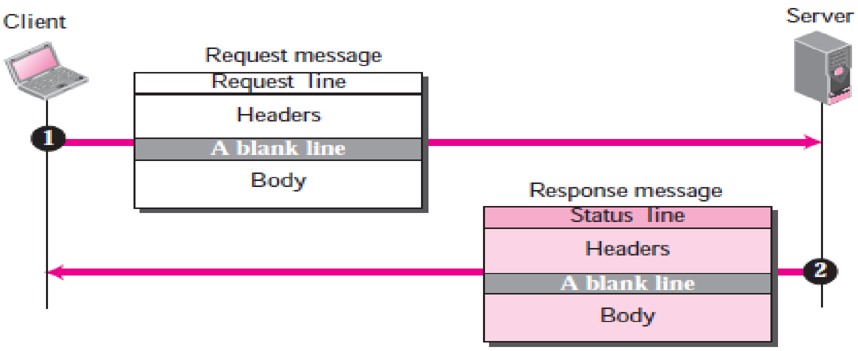
TCP	UDP
• **FTP** – 20-21 • **Telnet** – 23 • **SMTP** – 25 • **DNS** – 53 (Both TCP & UDP) • **HTTP** – 80	• **DHCP** – 67 & 68 • **POP** – 110



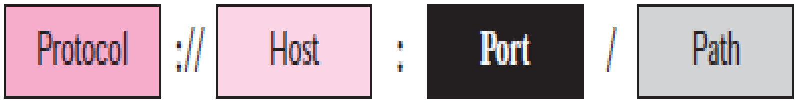
WWW Service and HTTP(Hypertext Transfer Protocol)

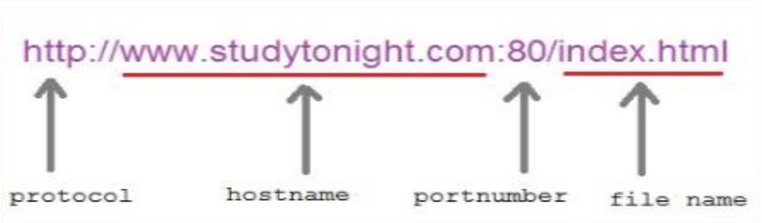


- The (HTTP) is a protocol used mainly to **access** **data on the World Wide Web.**
- HTTP use **TCP connection and port 80**.
- HTTP messages are not destined to be read by humans; (**read and interpreted by the HTTP server and HTTP client (browser).**
- HTTP is called a ***stateless*** protocol because each **command is executed independently**, without any knowledge of the commands that came before it and **the server does not keep information about the client**. The client initializes the transaction by sending a **request**. The server replies by sending a **response**.



- A client that wants to access a Web page needs the **file name and the address.**
- The **Uniform Resource Locator** (URL) is a standard locator for specifying any kind of information on the Internet, The URL defines four things:





Operation Steps:

1. URL is typed in the address bar.
2. Browser checks with DNS server to convert it to an IP address
3. Connects to the server requested
4. Using HTTP or HTTPS protocol requirements, the browser sends a GET request to the server to ask for the desired html document (usually index.html)
5. The server sends the HTML code for the web page to the browser.
6. The browser interprets the HTML code and formats the page to fit the browser window.

HTTPS(HTTP + SSL)

- **HTTPS =** HTTPS stands for Hypertext Transfer Protocol over **Secure Socket Layer** (SSL), or HTTP over SSL.
- HTTPS by default uses port 443.
- URL's beginning with HTTPS indicate that the connection is encrypted using SSL.

Disadvantage of HTTPS:

- HTTPS is slightly slower than HTTP because of the encryption of data.

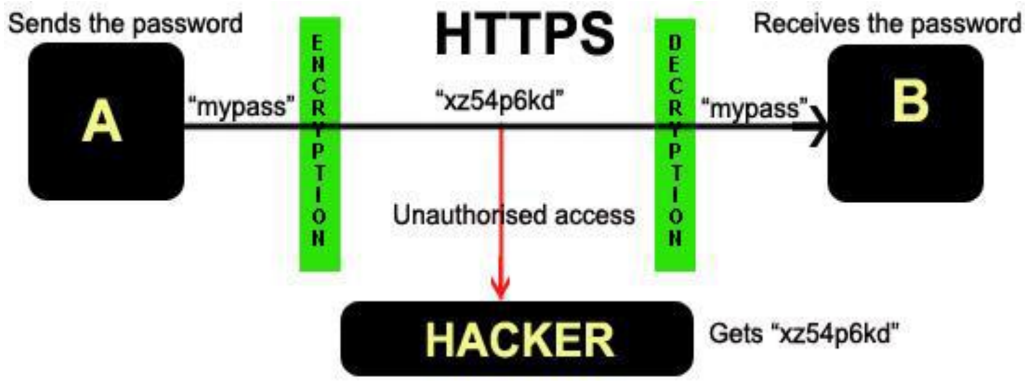


Figure: Working of HTTPS

Compare between HTTP & HTTPS?

HTTP	HTTPS
1. URL begins with "http://"	**1.** URL begins with "https://"
2. unsecured	**2.** secured
3. uses port 80 for communication	**3.** Uses port 443 for communication.
4. operates at Application Layer	**4.** Operates at Transport and application Layer.
5. No encryption	**5.** uses encryption
6. No certificates required	**6.** certificates required