Advanced Computer Networks - Set 2

Study online at https://quizlet.com/_epgoc7

1.	**Server process**	Process that waits to be contacted
2.	**Application Layer**	Service: Service apps Location: Inside the apps Type: Software Messages called: Messages
3.	**Popular Protocols**	HTTP, DNS, SMTP, FTP
4.	**Client-Server Applications**	A client computer requests data or a service from a server
5.	**Peer-to-Peer Applications**	When client hosts provide services directly to other client hosts (ex. two iPads using airdrop). No always on server
6.	**Server**	Always on host, permanent IP address
7.	**Client**	Host which communicates with server
8.	**Process/Interprocess communication**	Process: Program running within a host. Interprocess communication: Two processes within the same host can communicate by exchanged messages
9.	**Client Process**	Process that initiates communication
10.	**Socket**	Address of machine and address of process. IP address & port number pair
11.	**TCP Protocol**	- Connection oriented - Reliable transport - Flow control - Congestion control - Acknowledgements
12.	**UDP Protocol**	- Not reliable - No flow control

		- No acknowledgements - Fast
13.	**SSL**	- Secure Sockets Layer (NOT a layer in TCP/IP model) - SSL is used with HTTPS to encrypt HTTP traffic - Enhances TCP & UDP - SSL uses port 443.
14.	**HTTP**	- Hypertext Transfer Protocol - Web's application layer protocol - Client/server model - Uses TCP on port 80
15.	**Web Page**	- Consists of base HTML file, which can include several referenced objects - Each object addressable by a URL
16.	**Persistent vs. Non-persistent HTTP connections**	Non-persistent: At most one object sent over TCP connection, then connection closed. Multiple objects require multiple connections Persistent: Multiple objects can be sent over single TCP connection
17.	**RTT**	Round trip time: time for a packet to travel from client to server and back 1 RTT to initiate TCP connection + 1 RTT for HTTP request and HTTP response to return Non-persistent HTTP response time: 2RTT + file transmission time per object Persistent HTTP response time: as little as 1RTT for all referenced objects
18.	**HTTP Request Message**	Main components: - Method (GET/POST) - URL

		- Version - Keep-Alive - Connection - body
19.	**HTTP Response Message**	Main components: - Version - Status code - Last-Modified - Keep-Alive - Connection - Content-Length - Content-Type - data
20.	**POST Method**	Data transferred to server via form input (fill out form and press submit)
21.	**GET Method**	Data is transferred to server via URL (www.test.com/testing?id=1&this=test)
22.	**Cookies**	Maintain state at sender/receiver over multiple transactions Used for: - Authorization - Shopping carts - Recommendations - User session state
23.	**Web Caches (Proxy server)**	- Aim to satisfy client request without involving origin server, reducing response time and traffic to origin server - Cache acts as both client and server
24.	**Conditional GET**	Client utilizes "If-Modified-Since" field in HTTP header, server only returns copy of requested object if object has been updated. Else server returns "304 Not Modified"

25. **HTTP 1.1, HTTP/2, HTTP/3**	HTTP 1.1: Introduced multiple pipelined GET's over single TCP connection. Issue is HOL blocking with large objects HTTP/2: Decrease delay in multi-object HTTP requests by dividing objects into frames and scheduling to avoid HOL blocking and client priority HTTP/3: Add security, per object error and congestion control over UDP
26. **Electronic Mail**	Major components: - User agent - Mail reader, e.g. Outlook, Gmail - Mail servers - Mailbox: contains incoming messages - Message queue: contains outgoing messages - SMTP
27. **SMTP**	Simple Mail Transfer Protocol - Uses TCP, port 25 - Direct transfer - Three phases (handshake, transfer, closure)
28. **DNS**	Domain Name System: - Distributed database storing resource records (RR) - Implemented in hierarchy of many name servers - Hosts/name servers communicate to resolve names from addresses - Application layer protocol
29. **TLD Server**	Top-Level Domain server - Responsible for com, org, net, edu, and all top level country domains
30. **Authoritative Server**	Organizations own DNS servers, providing hostname to IP Mappings for organizations names hosts
31. **Local DNS Server**	Each ISP has one, not necessarily part of hierarchy.

	When host makes query, it is sent to its local DNS server
32. **DNS Caching**	- Once any name server learns mapping, it caches the mapping - Cache entry disappears after some TTL