Layer 7 Protocols

Layer 7 refers to the Application Layer in the OSI (Open Systems Interconnection) model, which is the last layer and it is responsible for managing communication between applications and end-user services. The communication between applications across a network are defined by a set of protocols that ensure the data is presentable for end users.

Layer 7 protocols support application-level functionalities, providing services for file transfers, web browsing, email exchange and so on. The most common layer 7 protocols include:

1. **HTTP (HyperText Transfer Protocol)**
   Used for web browsing and communication between servers and clients, and it operates on the request-response model.

2. **HTTPS (HTTP Secure)**
   A secure version of HTTP, it uses SSL/TLS for encryption to ensure data integrity and confidentiality. HTTPS is often used for transactions and services that require security, such as banking and online shopping.

3. **FTP (File Transfer Protocol)**
   Uses TCP/IP protocols for file transfers between systems in a network. Includes file uploads, downloads and file management.

4. **SMTP (Simple Mail Transfer Protocol)**
   Utilized for sending emails between mail servers, used along with POP3 and IMAP for email retrieval.

5. **POP3 (Post Office Protocol)**
   Used for email retrieval from mail servers by downloading emails.

6. **IMAP (Internet Message Access Protocol)**
   Used for email retrieval from mail servers, it differs from POP3 for its more sophisticated management of messages on the server.

7. **DNS (Domain Name System)**
   Translates domain names into IP addresses which enables browsers and other applications to access internet resources by names that are readable for humans.

8. **SNMP (Simple Network Management Protocol)**
   Used for device management in a network, such as switches, routers, and printers.

9. **Telnet/SSH (Secure Shell)**
   Used for remote terminal access, SSH provides security for the communication.

HTTP Methods

HTTP methods are used by web browsers and other clients to access resources on a web server. The methods describe the action desired to be performed on a specific resource.

1. **GET:**
   Used to retrieve information from the resource server. It does not alter any data on the server and can be called multiple times. The GET requests are casheable, which makes them suitable for data that does not get changed often.

2. POST:
   Used to send data to the server such as files upload or adding new data.

3. PUT:
   Used for updating or replacing a row of data in the server.

4. PATCH:
   Used for partial updates to data in the server and does not replace the entire row of data.

5. DELETE:
   Used to remove data from the server.

6. HEAD:
   Used to retrieve the headers of a resource and not the resource itself unlike GET. commonly used to check for metadata about a resource such as type or length.

7. OPTIONS:
   Used to check for the HTTP methods provided by a specific resource.

8. TRACE:
   Used to trace the path of a request through the network. Mostly used for debugging and diagnostic purposes.

9. CONNECT:
   Used to create a tunnel between a client and a server for connection security such as SSL encryption.

Method	Description	Safe	Casheable	Use Case
GET	Requests data from a specified resource.	Yes	Yes	Retrieving a webpage or resource.
POST	Submits data to be processed to a specified resource.	No	No	Submitting form data, creating resources.
PUT	Updates a current resource with new data.	No	No	Updating an existing resource (e.g., modifying a record).
DELETE	Removes the specified resource.	No	No	Deleting a resource.
PATCH	Partially updates a resource.	No	No	Modifying part of an existing resource.
HEAD	Similar to GET but only retrieves headers (no body).	Yes	Yes	Checking headers, such as for content type, without downloading the resource itself.
OPTIONS	Describes the communication options for the target resource.	Yes	Yes	Discovering allowed methods, CORS preflight requests.
TRACE	Echoes back the received request, used for debugging.	Yes	No	Debugging request-response cycle.
CONNECT	Establishes a tunnel to the server, often used for SSL (HTTPS) connections.	No	No	Used by HTTP proxies for establishing SSL connections.