Application Layer protocols

Overview

The application layer is used by software such as web browsers and email clients. It provides protocols that allow the software to send and receive information and present it in meaningful data to all the users. Examples of application layer protocols are Hypertext Transfer Protocol, File Transfer Protocol, Post Office Protocol, Simple Mail Transfer Protocol, and Domain Naming system. An application layer protocol defines how application processes (clients and servers) running on different end systems transmit messages. In particular, an application layer protocol defines:

- Message types, e.g., ex. B. Request messages and reply to messages.
- The syntax of the different types of messages, i. H. the fields within the message and the kind of delimitation of the areas.
- The semantics of fields, i. H. the importance of the knowledge that the sector must contain;
- The Rules for determining when and how a process sends messages and responds to messages.

HTTP

It is an application protocol for collaborative, distributed hypermedia information systems that enable users to talk about data across the globe. HTTP was invented along with HTML to make the leading interactive text-based web browser - the first World Wide Web. Today, the protocol is one of the first means of using the web. HTTP allows users to interact with web resources, such as HTML files, passing hypertext messages between clients and servers, as a request-response protocol. HTTP clients generally use Transmission Control Protocol (TCP) connections to communicate with servers.

HTTP can be a TCP / IP-based communication protocol that delivers data (HTML files, image files, query results, etc.) over the Planet Wide Web. The standard port is TCP 80, but other ports are often used. It provides computers with a uniform way to communicate with each other. The HTTP specification defines how clients request data and send it to the server and respond to these requests.

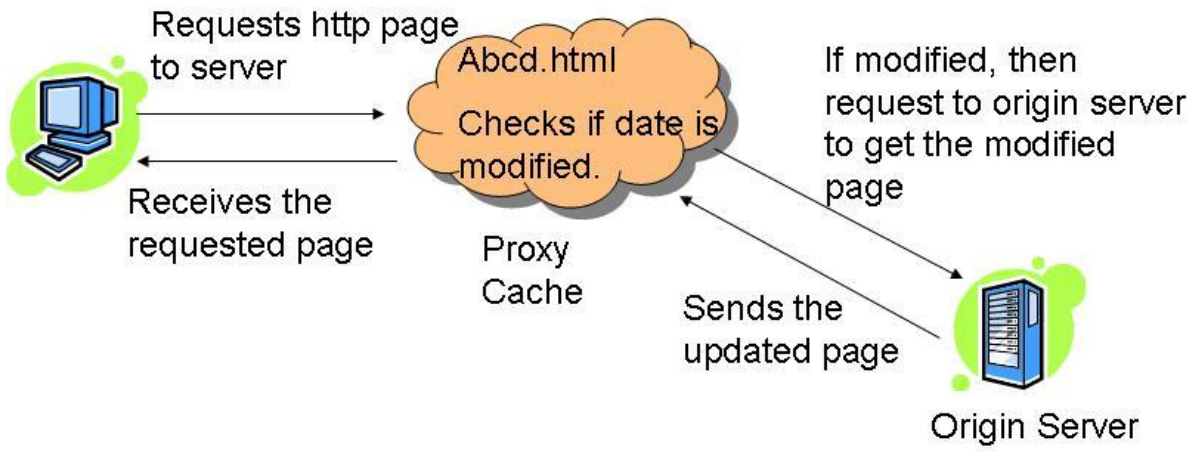

Figure 1: HTTP Request

FTP

File Transfer Protocol (FTP) can be a network protocol used to transfer files between computers using TTCP / IP connections. Within the TCP / IP suite, FTP is considered the application layer protocol. In an FTP transaction, the top user's computer is generally referred to as the localhost. The second computer involved in FTP can be a remote host, which is usually a server. Both computers must be connected to a network and properly configured to transfer files via FTP. Servers must be found to run FTP services, and therefore FTP software must be installed on the client to access these services. Although many file transfers are often done using the Hypertext Transfer Protocol (HTTP), another protocol within the TCP / IP suite, FTP, is still used to transfer files in the background for other applications such as banking services. Sometimes it is also common to download new applications through web browsers.

It is a protocol that usually sends files from computer to computer, and one of them acts as a server as long as the two are connected online. FTP is a network protocol between the client and the server and allows users to download websites, files, and programs from other services. When the user wants to download the knowledge to his computer, he uses FTP. It does not use encryption. It relies on apparent text usernames and passwords for authentication, making data transmissions sent via FTP vulnerable to eavesdropping, spoofing, and other common attacks.

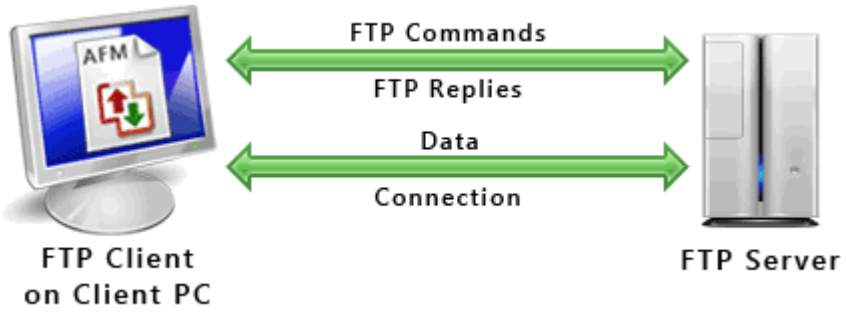

Figure 2: FTP Request

SMTP

It is used to send and receive emails. It is sometimes paired with IMAP or POP3 (for example, via a user-level application) that performs message retrieval, while SMTP primarily sends messages to be forwarded to a server. SMTP can send and receive email, but it is terrible at queuing incoming messages, hence the usual delegation to other protocols. Proprietary systems like Gmail have their email transfer protocols when using their servers, but they still use old SMTP to send emails on top of that. SMTP is an asymmetric protocol, which means that many clients interact with a server, using a base model popular in the 1980s that essentially no longer exists outside of email protocols today. SMTP runs on TCP / IP and listens on port 25. The actual transmission of mail is done through Message Transfer Agents (MTAs). Therefore, the system must have the MTA client send the mail, and the system must have an MTA server to receive the mail.
In principle, the respective mail transmission is done through message transfer agents (MTAs). To send the mail, the system must have the MTA client, and to receive the mail; the system must have an MTA server. To define the MTA client and server on the web, there is a convenient way called Simple Mail Transfer Protocol (SMTP).

- SMTP also uses TCP / IP to send and receive emails.
- SMTP is based on the client/server model.
- The original standard port for SMTP is port 25.
- With this protocol, the client that wants to send the email first opens a TCP connection to the SMTP server and then sends the email through the TCP connection. It is important to note that the SMTP server is usually in listening mode. As soon as it listens for a client's TCP connection, the connection starts on port 25, and after a successful relationship, the client immediately sends the email/message.

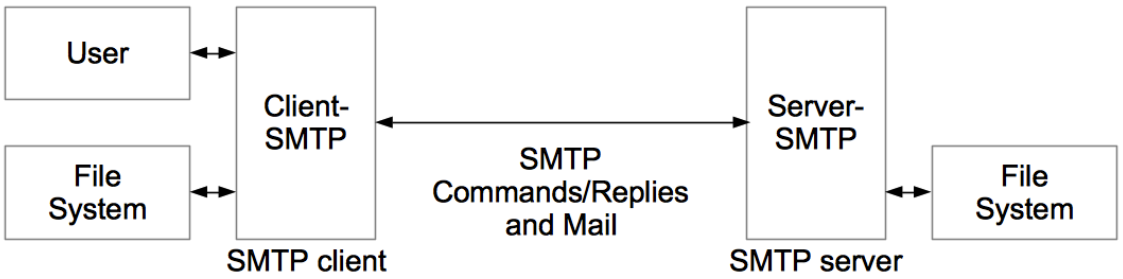

Figure 3: SMTP Request

DNS

The DNS is that the phone book of the web. People access information online using domain names such as nytimes.com or espn.com. Web browsers interact using Internet Protocol (IP) addresses. DNS translates domain names into IP addresses so that browsers can load Internet resources. Each device connected to the Internet has a unique IP address that other devices use to find the device. DNS servers make it unnecessary to remember IP addresses like 192.168.1.1 (on IPv4) or newer, more complex alphanumeric IP addresses like 2400: cb00: 2048: 1:: c629: d7a2 (on IPv6). The DNS resolution method involves converting a hostname (such as www.example.com) to an IP address supported by the computer (192.168.1.1). Every device on the web is assigned an IP address, which is important in finding an acceptable internet device. An address is used to find a specific house. When a user wants to load a web page, a translation must be between what the user typed in their browser (example.com) and the machine-friendly address needed to find the instance's .com web page.

To know the method behind DNS resolution, it is essential to know which hardware components a DNS request must pass. For the online browser, the DNS lookup is done "behind the scenes" and does not require interaction from the user's computer other than the initial query. The DNS process works as follows:

1. A browser, application, or device called a DNS client issues a DNS query or DNS address lookup and returns a hostname such as "example.com."
2. The request is received by a DNS resolver, which is responsible for finding the correct IP address for this hostname. The DNS resolver looks for a DNS name server containing the hostname's IP address in the DNS request.

3. The resolver starts with the Internet root DNS server. It works its way down the hierarchy to the top-level domain (TLD) DNS servers (here, ".com") to the name server that is responsible for the exact environment "Example. com ".
4. When the resolver reaches the authoritative DNS name server for "example.com," it receives the IP address and other relevant details and sends them back to the DNS client. The DNS request is now resolved.
5. The DNS client device can connect directly to the server with the correct IP address.

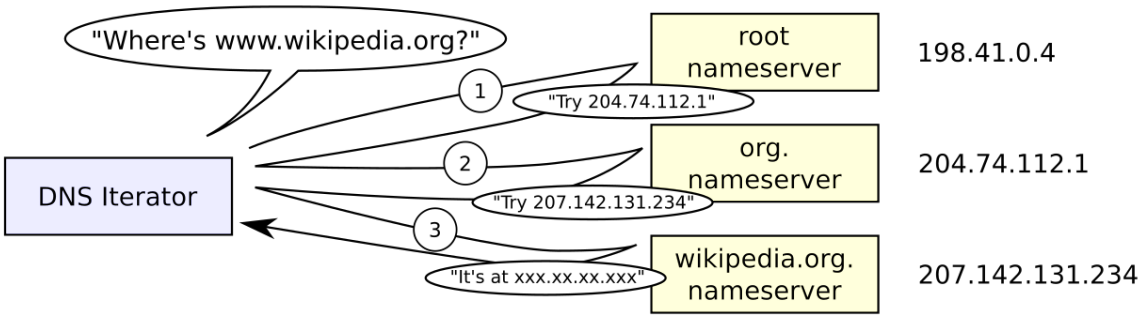

Figure 4: DNS Request