UNIT - IV:

Transport and Session layer protocols: Transport Layer (TCP, MPTCP, UDP, DCCP, SCTP)-(TLS, DTLS) – Session Layer HTTP, CoAP, XMPP, AMQP, MQTT

Transport Layer (TCP, MPTCP, UDP, DCCP, SCTP) :

The Transport Layer

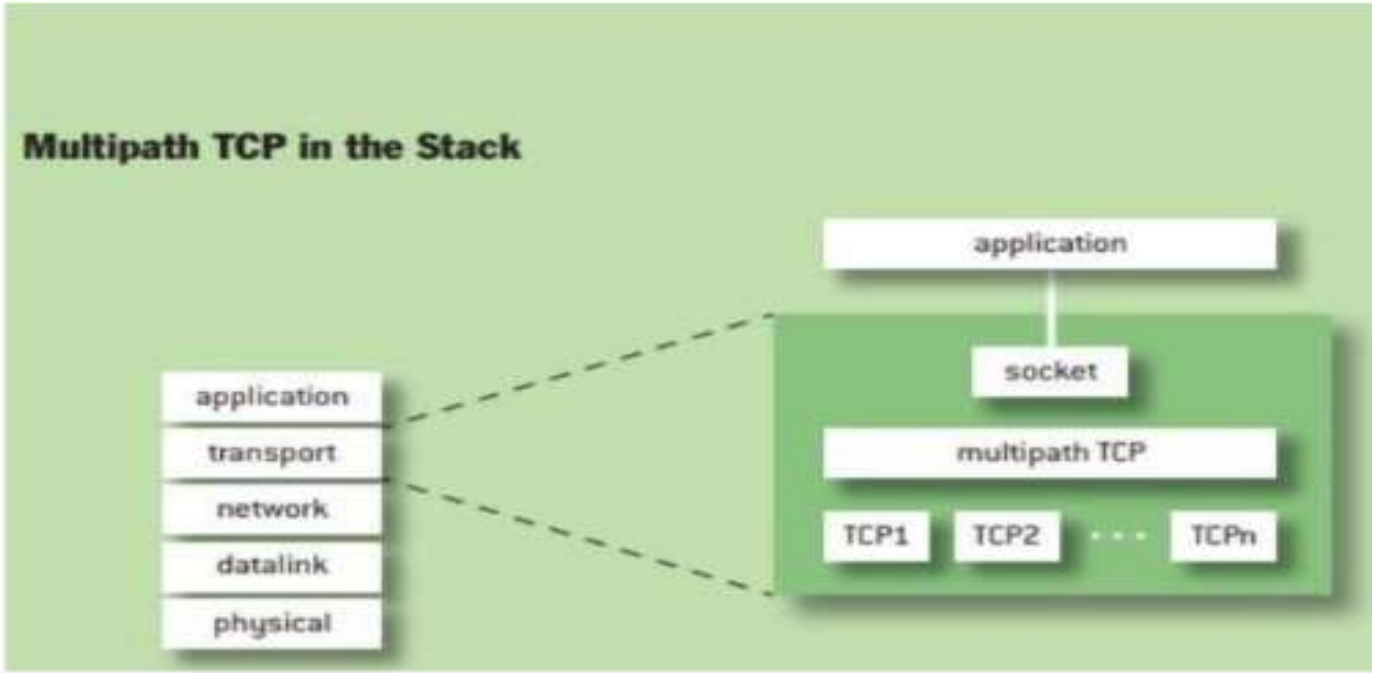
With the TCP/IP protocol, two main protocols are specified for the transport layer:

1) Transmission Control Protocol (TCP): This connection-oriented protocol requires a session to get established between the source and destination before exchanging data. You can view it as an equivalent to a traditional telephone conversation, in which two phones must be connected and the communication link established before the parties can talk.

2) User Datagram Protocol (UDP): With this connectionless protocol, data can be quickly sent between source and destination—but with no guarantee of delivery. This is analogous to the traditional mail delivery system, in which a letter is mailed to a destination. Confirmation of the reception of this letter does not happen until another letter is sent in response.

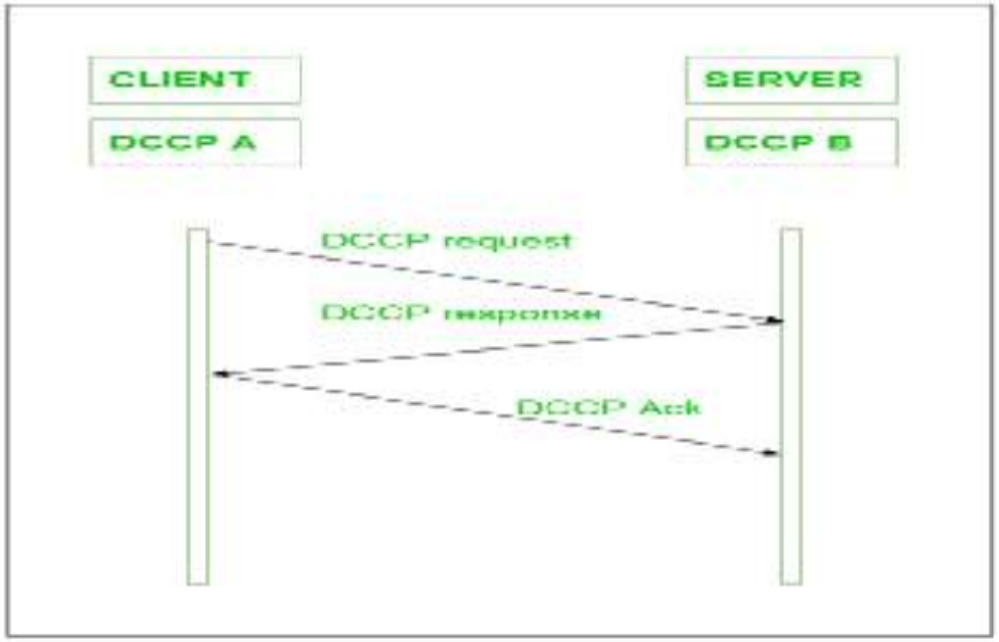
MPTCP

Goals for MPTCP are: • It should be capable of using multiple network paths for a single connection. • It must be able to use the available network paths at least as well as regular TCP, but without starving TCP. • It must be as usable as regular TCP for existing applications. •Enabling MPTCP must not prevent connectivity on a path where regular TCP works.



DCCP: DCCP provides an unreliable transport service, similar to UDP, but with adaptive congestion control, similar to TCP and Stream Control Transmission Protocol (SCTP). DCCP can be viewed equally well as either UDP- plus-congestion-control or TCP-minus-reliability (although, unlike TCP, DCCP offers multiple congestion control algorithms)

DCCP is basically a message-based transport-level protocol. The setting of a secure connection is easily maintained using it, its closure i.e. ECN (Explicit Congestion Notification), congestion control, and negotiation of features. DCCP is a great technique to access congestion control mechanisms, also we don't need to implement them at the application level also.



- **DTLS**

Datagram Transport Layer Security (DTLS) is a protocol used to secure datagram-based communications. It's based on the stream-focused Transport Layer Security (TLS), providing a similar level of security. ... However, DTLS gains the benefits of datagram protocols, too; in particular, the lower overhead and reduced latency .

SCTP Requirements

Number of Inbound and Outbound Streams: An association between the endpoints A and Z provides n streams from A to Z and m streams from Z to A. A pair consisting of two streams with the same stream identifier is considered and used as one bi-directional stream. Thus an SCTP association can be considered as a set of min(n,m) bi- directional streams and (max(n,m) - min(n,m)) unidirectional streams. Fragmentation of User Messages To avoid the knowledge and handling of the MTU inside TLS, SCTP MUST provide fragmentation of user messages, which is an optional feature of [RFC2960].. Thus the supported maximum length of SCTP user messages MUST be at least 2^14+ 2048 + 5 = 18437 bytes, which is the maximum length of a TLS Ciphertext, as defined in [RFC2246]. Note that an SCTP implementation might need to support the partial delivery API to be able to support the transport of user messages of this size. Therefore, SCTP takes care of fragmenting and reassembling the TLS records in order to avoid IP-fragmentation.

Session Layer HTTP :

HTTP stands for HyperText Transfer Protocol.

It is a protocol used to access the data on the World Wide Web (www).

The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.

This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.

HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files. HTTP is used to carry the data in the form of MIME-like format.

HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately. HTTP represents "Hypertext Transfer Protocol." HTTP is the protocol that can transfer information over the network. It is the Internet protocol suite method and defines commands and functions used for sharing web page data.

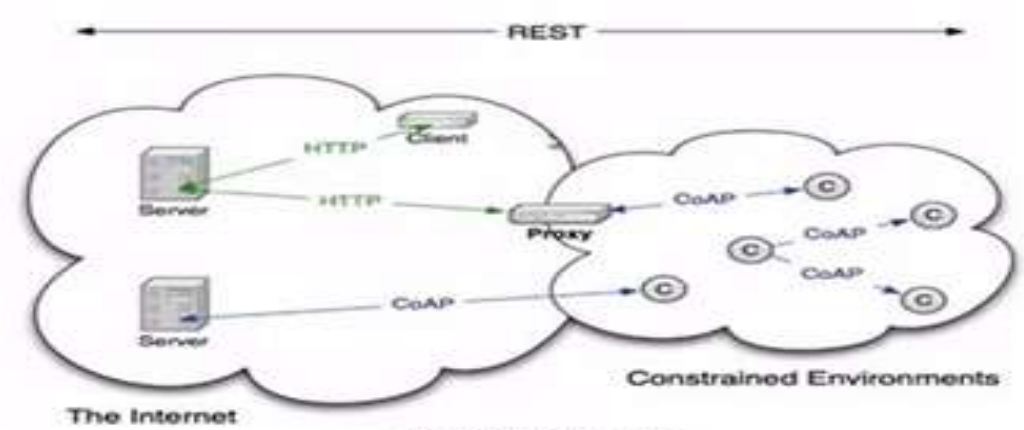
MQTT :

- MQTT stands for Message Queuing Telemetry Transport.
- MQTT is a machine to machine internet of things connectivity protocol. It is an extremely lightweight and publish-subscribe messaging transport protocol.
- This protocol is useful for the connection with the remote location where the bandwidth is a premium.
- These characteristics make it useful in various situations, including constant environment such as for communication machine to machine and internet of things contexts.
- It is a publish and subscribe system where we can publish and receive the messages as a client. It makes it easy for communication between multiple devices.
- It is a simple messaging protocol designed for the constrained devices and with low bandwidth, so it's a perfect solution for the internet of things applications.

CoAP

- The Constrained Application Protocol (CoAP) is another session layer protocol designed by IETF Constrained RESTful Environment (Core) working group to provide lightweight RESTful (HTTP) interface. Representational State Transfer (REST) is the standard interface between HTTP client and servers. However, for lightweight applications such as IoT, REST could result in significant overhead and power consumption. CoAP is designed to enable low-power sensors to use RESTful services while meeting their power constrains. It is built over UDP, instead of TCP commonly used in HTTP and has a light mechanism to provide reliability. CoAP architecture is divided into two main sublayers: messaging and request/response. The messaging sublayer is responsible for reliability and duplication of messages while the request/response sublayer is responsible for communication. As shown in Figure 7 , CoAP has four messaging modes: confirmable, non- confirmable, piggyback and separate. Confirmable and nonconfirmable modes represent the reliable and unreliable transmissions, respectively while the other modes are used for request/response.

CoAP:



CoAP Architecture

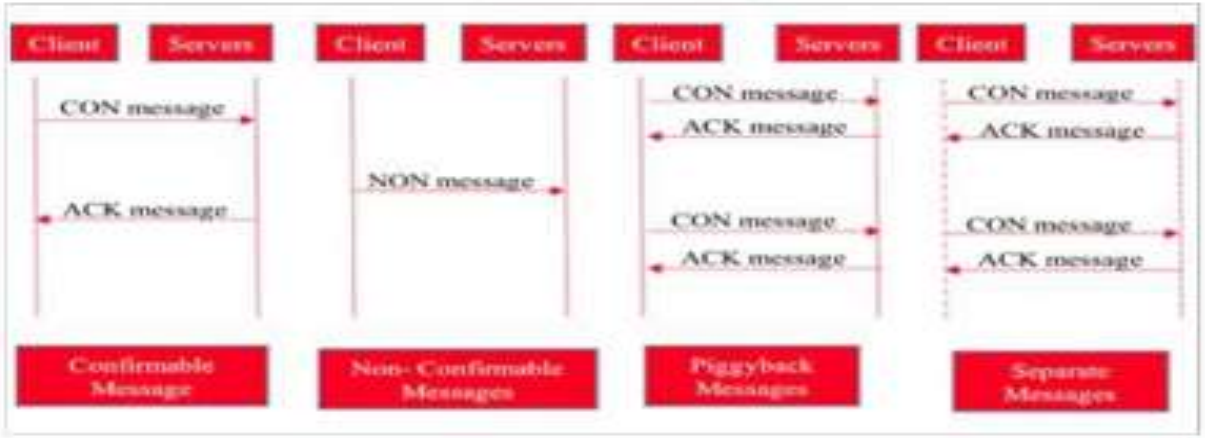


Fig: COAP messages

XMPP:

Extensible Messaging and Presence Protocol (XMPP) is a messaging protocol that was designed originally for chatting and message exchange applications. It was standardized by IETF more than a decade ago. Hence, it is well known and has proven to be highly efficient over the internet. Recently, it has been reused for IoT applications as well as a protocol for SDN. This reusing of the same standard is due to its use of XML which makes it easily extensible. XMPP supports both publish/ subscribe and request/ response architecture and it is up to the application developer to choose which architecture to use. It is designed for near real-time applications and, thus, efficiently supports low-latency small messages. It does not provide any quality of service guarantees and, hence, is not practical for M2M communications. Moreover, XML messages create additional overhead due to lots of headers and tag formats which increase the power consumption that is critical for IoT application. Hence, XMPP is rarely used in IoT but has gained some interest for enhancing its architecture in order to support IoT applications