

Connection Oriented and Connectionless Services

These are the two services given by the layers to layers above them. These services are:

1. Connection Oriented Service
2. Connectionless Services

Connection Oriented Services

There is a sequence of operation to be followed by the users of connection oriented service. These are:

1. Connection is established.
2. Information is sent.
3. Connection is released.

In connection oriented service we have to establish a connection before starting the communication. When connection is established, we send the message or the information and then we release the connection.

Connection oriented service is more reliable than connectionless service. We can send the message in connection oriented service if there is an error at the receivers end. Example of connection oriented is TCP (Transmission Control Protocol) protocol.

Connection Less Services

It is similar to the postal services, as it carries the full address where the message (letter) is to be carried. Each message is routed independently from source to destination. The order of message sent can be different from the order received.

In connectionless the data is transferred in one direction from source to destination without checking that destination is still there or not or if it prepared to accept the message. Authentication is not needed in this. Example of Connectionless service is UDP (User Datagram Protocol) protocol.

Difference: Connection oriented and Connectionless service

1. In connection oriented service authentication is needed, while connectionless service does not need any authentication.
2. Connection oriented protocol makes a connection and checks whether message is received or not and sends again if an error occurs, while connectionless service protocol does not guarantees a message delivery.
3. Connection oriented service is more reliable than connectionless service.
4. Connection oriented service interface is stream based and connectionless is message based.

What are Service Primitives?

A service is formally specified by a set of primitives (operations) available to a user process to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets. The set of primitives available depends on the nature of the service being provided. The primitives for connection-oriented service are different from those of connection-less service. There are five types of service primitives :

1. **LISTEN** : When a server is ready to accept an incoming connection it executes the LISTEN primitive. It blocks waiting for an incoming connection.
2. **CONNECT** : It connects the server by establishing a connection. Response is awaited.
3. **RECIEVE**: Then the RECIEVE call blocks the server.
4. **SEND** : Then the client executes SEND primitive to transmit its request followed by the execution of RECIEVE to get the reply. Send the message.
5. **DISCONNECT** : This primitive is used for terminating the connection. After this primitive one can't send any message. When the client sends DISCONNECT packet then the

server also sends the DISCONNECT packet to acknowledge the client. When the server package is received by client then the process is terminated.

Connection Oriented Service Primitives

There are 5 types of primitives for Connection Oriented Service :

LISTEN	Block waiting for an incoming connection
CONNECTION	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Sending a message to the peer
DISCONNECT	Terminate a connection

Connectionless Service Primitives

There are 4 types of primitives for Connectionless Oriented Service:

UNIDATA	This primitive sends a packet of data
FACILITY, REPORT	Primitive for enquiring about the performance of the network, like delivery statistics.