| | |
|---|---|
| Name: | Prerna Sunil Jadhav |
| Sap Id: | 60004220127 |
| Class: | S. Y. B.Tech (Computer Engineering) |
| Course: | Computer Networks (DJ12CEL405) |
| Date of Performance: | |
| Date of Submission: | |
| Experiment No.: | 06 |
| Aim: | Socket TCP and UDP |

**AIM:** **TO IMPLEMENT TCP AND UDP SOCKET COMMUNICATION IN JAVA.**

**THEORY:**

- The transport layer is the fourth layer of the OSI model which is responsible for the process-to-process delivery of data.
- The main aim of the transport layer is to maintain the order so that the data must be received in the same sequence as it was sent by the sender.
- The transport layer provides two types of services namely - connection-oriented and connection-less.
- The various protocols used in this layer are:
  - TCP (Transmission Control Protocol),
  - UDP (User Datagram Protocol), etc.

## TCP - Transmission Control Protocol

- ✓ It is a **connection-oriented** protocol that defined the standard of establishing and maintaining the conversation (or connection) that will be used by the applications to exchange the data.
- ✓ In the connection-oriented protocol, we **first need to connect to the receiver before sending** our data.
- ✓ One of the prime reasons for using the transmission control protocol over the other protocol(s) like UDP is that the TCP ensures the **reliable transmission** and delivery of our data packets.
- ✓ The transmission control protocol **can deal with the various issues** that can occur in the data transmission such as packet duplication, packet corruption, packet disordering, packet loss, etc.
- ✓ **Working of TCP**
  - The entire process of establishing a connection, sending data packets, and then removing the connection comes under the working of the transmission control protocol.
    - Step 1: Establish a three-way handshake connection. The three-way handshake connection is used to create a connection between the host or client and the server. As the name suggests, it is a three-step process in which –
      - ➢ First the client (wants to establish a connection) sends an SYN segment (Synchronize Sequence Number segment) which tells the server that the client wants to start the communication.

- Then server responds with an SYN-ACK signal (SYN Acknowledgement). The SYN-ACK signifies the server has received the client's request of establishing the connection.
- Lastly the client again sends the ACK signal to the server and they both establish a reliable connection that will be used to transfer the data packets.
- The three-way handshake is also known as **SYN-SYN-ACK**. Refer to the diagram below for more clarity.

- Step 2: Sending of data packets along with the sequence number from the client. The second computer (server) responds to these sent packets by sending an acknowledgment or ACK. This acknowledgment bit keeps on increasing with the number of packets sent. This ACK bit helps to keep track of three things:
  - The successfully received packets,
  - The lost packets, and
  - The packets which were accidentally sent twice.
- Step 3: ither server or the client can close the connection. The first computer system (either server or the client) initiates the closing of the connection by sending a packet with a FIN bit or finish bit attached to it. The other computer sends back or responds with an ACK bit. Finally, the first computer sends an ACK bit back to the second computer and the connection gets closed.

✓ **Features of TCP**

- **Connection-oriented**
- **Reliable data transfer**.
- **Flow Control** limits the rate of data transfer from the sender's end. The receiving end (receiver) continuously hints the rate at which it can receive the data and the TCP uses the hints to adjust the data transfer rate of the sender.
- **Error Control** - error-checking and error recovery.
- **Sequencing of packets**
- **Congestion Control** - Congestion means the amount of data transferred in the network. So, the transmission control protocol also accounts for the level of congestion and sends the packets paccordingly.
- **Full duplex**

## UDP – User Datagram Protocol

- ✓ UDP protocol is the **connectionless and unreliable** protocol.
- ✓ Since UDP is a connectionless protocol so there is **no need to establish a connection** before transmitting data.
- ✓ UDP packets are called **User Datagram**.
- ✓ User Datagram has **8 bytes fixed-size header**.
- ✓ UDP protocol will work just like an alternative of TCP (Transmission Control Protocol).
- ✓ **Working of UDP protocol**
  - UDP works by collecting data in a UDP packet, and in the packet, it adds its own header information.
  - UDP packet, called user datagram, consists of:
    - **Source port number** is 2 bytes field that defines the port number of the sender.
    - **Destination port numbe**r is 2 bytes field that defines the port number of the destination.
    - **Packet length** is also 2 bytes field for defining the total length of the user datagram (header length + data length)
    - **Checksum** is a 2 bytes optional field for carrying checksum
  - UDP packets are sent to their destination after encapsulating it in an IP packet
  - In UDP, there is no acknowledgment generated for the packet received, so the sender does not wait for acknowledgment of the sent packet.
- ✓ **Features of UDP protocol**
  - Connectionless
  - Since it is connectionless, so packets are sent from different paths between sender and receiver.
  - Ordered delivery of data is not guaranteed.
  - The UDP protocol utilizes different port numbers for transmitting data to the correct destination.
  - The port numbers are defined between 0 - 1023.
  - Faster transmission
  - UDP provides us a faster service of data transmission as there is no prior connection establishment before transmitting the data.
  - UDP does not require any virtual path for data transmission.
  - No acknowledgment mechanism so there is no handshaking.
  - Segments are handled independently.
  - Every segment in UDP takes a different path to reach the destination. So, every UDP packet is handled independent of other UDP packets.
  - Stateless - sender does not wait for an acknowledgment after sending the packet.
- ✓ Applications of User Datagram Protocol (UDP)
  - UDP protocol can be utilized for simple request-response communication when there is a smaller size data since there is very less concern about the error and flow control.
  - UDP carries packet switching, so UDP is considered suitable protocol for multicasting.
  - UDP is also used in s

## DIFFERENCE BETWEEN TCP AND UDP

| CRITERIA | TCP | UDP |
|---|---|---|
| Connection Type | connection-oriented. | connectionless. |
| Reliability | TCP is reliable as there is an absolute certainty that the data sent will arrive intact and in the same order as it was sent. | In UDP, data delivery to the destination cannot be assured. |
| Speed | Speed of TCP is slower than UDP | Error recovery isn't attempted with UDP, therefore it's quicker. |
| Header Size | Header length is variable: 20-60 bytes | Header length is fixed: 8 bytes |
| Acknowledgement | Acknowledgement segments | No Acknowledgement |
| Data Transmission | TCP transmits data in a certain sequence, which assures that packets arrive at the recipient in the proper order. | On the other side, there is no data sequencing in UDP to provide a sequence. |
| Retransmission | In TCP, data packets can be retransmitted if they are lost or need to be resent. | Retransmission of packets, on the contrary, is not feasible with UDP. |
| Error Checking | TCP performs error checking and recovery. Packets that are incorrectly sent from the source are resent to the destination. | Error checking is performed by UDP; however, erroneous packets are simply discarded. There is no attempt at error recovery. |
| Applications | TCP is best suited for applications requiring great reliability, and whose transmission time is less crucial. | UDP is well suited to applications that require quick and efficient transmission, such as video games. The unidirectional characteristic of UDP is especially advantageous for servers that respond to short queries from a large number of clients. |
| Handshake | SYN, SYN-ACK, ACK | No handshake: connectionless protocol |

**CONCLUSION:**

✓ UDP is a much quicker, simpler, and more efficient protocol, nonetheless, only TCP allows for the retransmission of lost data packets.

✓ When comparing the TCP and UDP protocols, TCP is heavier while UDP is lighter.