

Assignment 4: Application development using UDP Socket

Time: 2 weeks

Objective

This assignment aims to find the Round Trip Time (RTT) in a Client to Server communication using a datagram (UDP) socket. The Server and Client should situate on two different physical computers in Ethernet LAN. The aim is also to understand how the Round Trip Time changes in various loading conditions in the network.

Technique

The process starts with the Client generating and sending a packet to the Server. The packet (datagram) format to be used is as follows.

-----	-----	-----	-----	-----
MT	SN	TTL	PL	PB
(1 byte)	(2 bytes)	(1 byte)	(4 bytes)	(P bytes)
-----	-----	-----	-----	-----

- Message Type (MT): Represents type of the message. The value of this field will be set 1, to represent it as a payload packet.
- Sequence Number (SN): Uniquely identifies individual packet.
- Time-to-live field (TTL): A non-negative even integer with an initial value of T.
- Payload Length (PL): The length of the payload bytes P.
- Payload Bytes (PB): Arbitrary payload bytes of size P bytes.

When the Server receives a datagram from the Client, it checks the Payload Length field and compares the number of received bytes to determine the sanity of the packet. If the packet is found to be consistent, then it decrements the TTL value by one in the datagram and sends the same datagram (with the new TTL) back to the Client.

However, the received packet is dropped at the Server if it is found to be inconsistent / malformed and then it responds with an error packet. The format of the error packet is as follows

-----	-----	-----
MT	SN	EC
(1 byte)	(2 bytes)	(1 byte)
-----	-----	-----

- Message Type (MT): Represents type of the message
 - The value of this field is 2, to represent it as an error packet
- Sequence Number (SN): Sequence number of the erroneous / malformed packet.
- Error Code (EC): Error codes can be of these types -
 - 1: TOO SMALL PACKET RECEIVED
 - 2: PAYLOAD LENGTH AND PAYLOAD INCONSISTENT
 - 3: TOO LARGE PAYLOAD LENGTH
 - 4: TTL VALUE IS NOT EVEN

On reception of the packets at the Client back from the Server, calculate the Round Trip Time (RTT) delay for each packet and calculate the average RTT delay across the packets.

However, in case the packet is reported as error packet, it prints the equivalent string of the error code in the console.

Development and Running Environment:

Write two socket programs in `ClientPacketGen.c/cpp` and `ServerPacketFwd.c/cpp`, which communicate using Datagram Sockets (UDP).

The server application should be executed using the following command:

```
$>./serverPacketfwd <ServerPort>
```

The client application should be executed using the following command:

```
$>./clientPacketGen <ServerIP> <ServerPort> <P> <TTL> <NumPackets>
```

Here `NumPackets` could be anything between 1 to 50, and for each packet, a different sequence number should be generated starting from 0. The P value should be within 100 to 1000 bytes and T between 2 and 20 (and must be even).

Note that the Server should minimize the processing delay by not incorporating any printing or queuing of the received packet.

Make sure server and client applications are running on two different physical computers. Run the Server on your host computer (laptop/terminal), and the Client is at Hamsa (10.2.1.40) or Hanau (10.2.1.41).