

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.

-----	-----	-----	-----
Sequence	TTL	Payload	Payload
Number		Length	bytes
(4 bytes)	(1 byte)	(2 bytes)	(P bytes)
-----	-----	-----	-----

- Sequence Number: Uniquely identifies individual packet.
- Time-to-live field (TTL): A non-negative even integer with an initial value of T.
- Payload Length: The length of the payload bytes P.
- Payload Bytes: 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. The packet is dropped at the Server if it is found to be inconsistent. Otherwise it decrements the TTL value by one in the datagram and sends the same datagram (with the new TTL) back to the Client.

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

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.

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).

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).

Note: Once the above problem is completed, then attempt the following extension –

This assignment is an extension of above problem. Create a copy of the developed code and make the required changes in this extension. Only `ClientPacketGen.c/cpp` will need some modification.

Technique

On receiving a datagram from Server, the Client program decrements the TTL value and checks if this new value is zero. If the new TTL is not zero, the Client sends the datagram (with the decremented TTL value) back to Server. However, if TTL is zero, the Client calculates the difference between the current time and the time of datagram first sent. Let's call this as the "Cumulative RTT". The Client saves the "Cumulative RTT" value to a file (in a new line).

Running Environment:

As stated earlier, run the Serve on your laptop/terminal computer, and the Client is at Hamsa (10.2.1.40) or Hanau (10.2.1.41).

Every time you run the Client, it generates new datagrams with TTL set to T and payload length P. The value of P and T and the name of the output file for storing the "Cumulative RTT" should be entered as command line arguments when executing the Client. The P should be within the range of 100 to 1000 bytes, and the T between 2 and 20 (and must be even). The Client program initiates the packet transmission process for a total of 50 datagrams (using the sequence numbers 0, 1, 2, ..., 49) and the stops.

At first, run the Client for $T = 2$ and different values of $P = 100, 200, 300, \dots, 1000$. Plot a scatter-plot (using any suitable software, such as Matlab, Gnuplot, etc.) of "Cumulative RTT" for all 50 datagrams vs. P for the different values of P when $T = 2$.

What do you observe? What information does the slope of the graph contain?

Repeat the process when $T = 8$ and 16 and mention your understanding.