CENG 435 TERM PROJECT PART-1

Muhammed Yusuf Temizer
2099356

Irfan Can Kaleli
2035996

I. Project Overview

In this assignment, main task is designing a network topology with one source node, one broker node, two routers and one destination node. Briefly, network is designed to send a message from source node to destionation node by using TCP and UDP protocols with packet load distribution. From start to end TCP packet is split into UDP packets and sent through internal the nodes to reach destination.

II. Design and Implementation

A. Design Process

Overall the system has one TCP packet sender in source node, one TCP listener and one UDP sender in broker node, one UDP listener in both routers and one UDP listener in destination node.

1. Source Node : Main task of this node is to get a string input from the user and create a packet including data and sent time of the packet. After packet is created, its sent using TCP protocol to the Broker node.

2. Broker Node : This node is used to achieve getting TCP message from the Source node and then after converting it to UDP message it sends the packets to Routers considering load balancing.

3. Router Nodes : Router nodes are dedicated only for one main task which is getting the UDP packet and sending to given Destination node.

4. Destination Node : This is the arriving node of the packets. Destination nodes task is to get UDP packets from the routers and then sort them in the correct order to compose initial message sent from the source node.

In order to create TCP communication between Source node and Broker node, we used built-in socketserver and threading modules of PYTHON3. The sent time of the packet is obtained from ntplib external module of PYTHON3. After the TCP packets are sent thread receives the packet and triggers the converting and load balancing events.

The connection between the Broker node and Router nodes are achieved by using socket module of PYTHON3. Load balancing in broker node is achieved altering between the routers while sending packets over UDP.

The UDP communication between the Router nodes and Destination nodes are achieved by using socket module of PYTHON3. After Destination node receives all packets the Source node has sent, it sorts the packets and composes the initial message.

All of the TCP and UDP messages are composed with pickle module of PYTHON3.

B. Impelementation Process

All of the node scripts are implemented in PYTHON3. There is only one script running in each node.

a. Source Node Implementation : Source script creates TCP connection with the Broker node and then waits for an input from the user. After user enters an input current time is requested from ”tr.pool.ntp.org” using ntplib then current time and user input is stored in an object of a class named DataWithTime. After that, object is converted to bytes using the pickle module and then sent to the Broker node through the TCP connection that is created.

b. Broker Node Implementation : Broker script has one thread created from the threading module, that listens the TCP connection from the Source node. After TCP request arrives from the Source, thread starts to get byte streams and packetize the stream. Script has one main class named Packet that has the following attributes:

1. Data: Stores the input the user entered in Source and can be at most 128 bytes.

2. Order: While dividing the input in TCP packet into sub strings, orders are assigned to achieve the recomposition in Destionation node.

3. Length: Identifies the number of UDP datagrams to send all of the TCP packet.

4. Group ID: Identifies the group of the UDP datagrams to handle the seperation of which UDP datagram maps to which TCP packet.

5. Source Time : Identifies the sent time of the TCP packet.

c. Router Impelementation : Router script only has the capability of receiving UDP datagrams from the broker and send the exact packet using the UDP.

III. Methodology And Motivation

A. Methodology

Python programming language is used for the implementation process. Used built-in libraries are threading, socket, socketserver and pickle, used external library is ntplib library. Internet connection is required to satisfy the requirements of our topology implementation.

B. Motivation

By implementing this topology we became familiar with the scientific-purpose-platform, GENI. In addition to that UDP and TCP connection knowledge, socket programming notion, and network concepts such as delay and byte streaming information are learned.

IV. Experimental Conclusion

A. Other Used Properties

Netem :

Netem is Network Emulation functionality for testing protocols such as TCP and UDP by emulating the properties of wide area networks.

In our project, we used netem to calculate end-to-end delay loss, packet loss.

NTP :

NTP (Network Time Protocol) is a protocol used to synchronize computer clock times in a network. Each node gets time from common computer or server clock so that each node work at the exact same time.

In our project, we used ntp to get time when first packet is sent from source and when the first packet is arrived to destination. Main concern is to calculate end-to-end-delay.

Experiment :

First we set up our environment in GENI portal and deploy our scripts to the each node. And then we send input from source and get the current time. When input receved by destination, we get the time again and calculate the end-to-end delay.

Before the test we presumed the expected result and we compare them with results. Here are the comparison values:

First Experiment(1ms):

Normal Distribution value: 1 ms

End-to-end delay = 1.15

Second Experiment(20ms):

Normal Distribution value: 20 ms

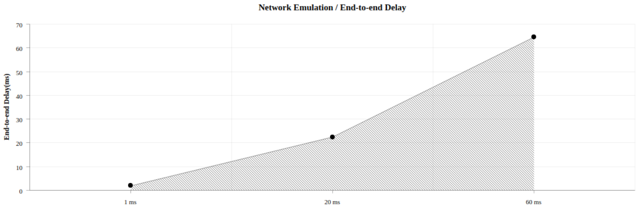
End-to-end delay = 22.4

Third Experiment(60ms):

Normal Distribution value: 60 ms

End-to-end delay = 64.1

Chart at the below is the visualization of the experience results.


Network Emulation / End-to-end Delay
End-to-end Delay(ms)
1 ms
20 ms
60 ms