NATIONAL INSTITUTE OF TECHNOLOGY SRINAGAR, JAMMU & KASHMIR

Computer Networks Semester Project - Submission

Date of submission : **20 June 2024**

INSTRUCTOR : Dr. Iqra Altaf Gillani

GROUP MEMBERS : 1) Manupati Jaideep - 2021BITE083
2) Kodi Vidya Sagar - 2021BITE091
3) Uppu Vamsi - 2021BITE068

CONTENTS

1 Language and Libraries used

1.1 Language

Python : This project is created and executed on jupyter notebook with all the required extensions downloaded to compile and run a py file.

1.2 Libraries

All the libraries are included in the code with the help of the header files. The list of header files used by the program are listed below.

1. Time
2. Random
3. urllib3

import time : The `import time` statement in Python includes the `time` module, which provides functions for handling time-related tasks such as getting the current time, pausing execution, and measuring performance. It is essential for tasks involving delays, timestamps, and time conversions.
import random : The `import random` statement in Python includes the `random` module, which provides functions for generating random numbers, selecting random elements, and shuffling sequences. It is useful for tasks involving randomness and probability in applications like simulations, games, and randomized algorithms.
import urllib3 : The `import urllib3` statement in Python includes the `urllib3` module, which provides functions for handling HTTP requests, including sending GET and POST requests, handling responses, and managing connections. It is useful for tasks involving web scraping, API interactions, and internet-based data retrieval.

2 Important Functions, Files and their use

2.1 DEVICES

2.1.1 End Device

Different functionalities and properties of End devices are Implemented in header file

PROPERTIES INCLUDED :

1. Device Id
2. Display Name
3. IP address
4. MAC address
5. Connection

2.1.2 Hub

Different functionalities and properties of Hubs are implemented in header file -

PROPERTIES INCLUDED :

1. Device Id
2. Display Name
3. List of Connections

2.1.3 Switch

Different functionalities and properties of switches are implemented in header file -

PROPERTIES INCLUDED :

1. Device Id
2. Display Name
3. List of connections
4. Address learning

2.1.4 Bridge

Different functionalities and properties of Bridges are implemented in header file

PROPERTIES INCLUDED :

1. Device Id
2. Display Name
3. List of connections

2.2 PROTOCOLS

2.2.1 Access Control

Token passing is the only access control protocol which is used and its properties are implemented in header file - AccessControl.h. Whenever a switch is detected at first place during message transfer it passes the token to the source device.

2.2.2 Flow Control

There are three flow control protocols which are implemented in the project. Whenever there is a first time a switch is detected during the message transfer it asks to choose a flow control protocol and provide the needed information.

1. The Stop and Wait protocol is implemented
2. Go Back N is implemented
3. Selective repeat protocol is implemented

2.2.3 Implementing network layer functionalities

1. We successfully created and configured a router using Python.
2. We successfully assigned well-formatted classless IPv4 addresses to the devices using Python. Each device was configured with a unique address within the designated subnets.
3. We successfully used ARP (Address Resolution Protocol) to find the MAC address of a host within a network using Python.
4. We successfully configured static routing using Python.This ensured precise and efficient routing paths for network traffic.
5. We successfully implemented RIP (Routing Information Protocol) for dynamic routing using Python.

2.2.4 Implement Transport and Application layer functionalities.

1. We successfully assigned port numbers to various processes, including well-known and ephemeral ports, using Python.This enabled process-process communication over the network, allowing for data exchange and interaction between different applications running on the network.
2. We successfully implemented the Go Back N or Selective Repeat sliding window flow control protocol at the transport layer using Python. By adapting the protocol from the data link layer (Layer 2) to the transport layer (TCP segment or UDP datagram), we ensured reliable and efficient data transmission over the network. This allowed for controlled data flow between sender and receiver, managing acknowledgments and retransmissions as necessary to maintain data integrity and reliability.
3. We successfully assigned port numbers to various processes, including well-known and ephemeral ports, using Python.

2.3 netsimulator.ipynb File

To execute the program this file is to be compiled and run.This file includes all the custom made functions and gives users different choices to create (end devices, hubs, switches, bridges) establish connection between devices, display all the devices and data transfer between end devices with the help of command line input.

2.4 Other Functions

There is one more function - **import random** which is used as a way or an intermediate to generate the mac and ip address for end devices randomly.

3 Assumptions

All the assumptions made while creating and implementing the code
Are based on the instructions provided.

Following assumptions were made :

1. If selection is to be made then input data type and data type of the value given by user must be the same for example if input data type is integer then data provided by user must be integer otherwise program might enter a infinite loop.
2. In case of collision either the packet was lost(packet not received) or ack was lost(time out). Any one of the cases is chosen randomly.
3. The device id of the end device lies in range 0 and 1000 and is generated by the system.
4. The device id of the hub lies in range 1000 and 2000 and is generated by the system.
5. The device id of switch lies in range 2000 and 3000 and is generated by the system.
6. The device id of the bridge lies in range 3000 and 4000 and is generated by the system.

4 Executing the code

To execute the program, compile and execute the netsimulator.ipynb file. Keeping all the custom made functions in the same executable file.

BASIC REQUIREMENTS :

1. In order to run the code the basic software requirements are the python compiler and the environment path is set for python.
2. The functions included are already present in the basic python compiler. No extra library is needed to be downloaded.
3. In order to open the python file any text editor will do the job.

5 References

- “The C++ Programming Language (4th Edition) Addison-Wesley ISBN. May 2013”- {Text book}
- “C++ Standard Library Documentation.”-en.cppreference.com
- “Behrouz A. Forouzan, Data communications and Networking”-{Text book}
- Github.
- Google.