Data Networks
HW 4: Network Layer
Dr. Mohammad reza Pakravan

Introduction

The network layer is concerned with getting packets from the source to the destination. Getting to the destination may require making many hops at the intermediate routers along the way. This function contrasts with that of the data link layer, which has the more modest goal of just moving frames from one end of a wire to the other. To achieve its goal, the network layer must know about the topology of the communication subnet (i.e., the set of all routers) and choose appropriate paths through it. It must also take care to choose routers to avoid overloading some of the communication lines and routers while leaving others idle. It is up to the network layer to deal with them. A router in the network needs to be able to look at a packet's destination address and then determine the output port which is the best choice to get the packet to that destination. The router makes this decision by consulting a forwarding table. Routing algorithms are required to build the routing tables and hence forwarding tables, The basic problem of routing is to find the lowest-cost path between any two nodes, where the cost of a path equals the sum of the costs of all the edges that make up the path. Routing is achieved in most practical networks by running routing protocols among the nodes. The protocols provide a distributed, dynamic way to solve the problem of finding the lowest-cost path in the presence of link and node failures, and changing edge costs. In this exercise, you are going to see some GNS3, NS3, and python features to stimulate network layer's functions.

Question 1: OSPF and BGP (30+15 Points)

Overview

In this problem, We are going to simulate different routing protocols you've learned using GNS3. You have gotten familiar with these powerful network simulators in Mini HW. You can find GNS3's comprehensive documentation here . Moreover, you can find many online tutorials that cover what you need for this problem. As we are going to work with routers after installing the GNS3, follow these instructions for adding a router to your GNS3. You can use c3725 or any other router you want in this implementation. Follow the instructions in each scenario and build your network. Use 192.x.y.z/24 for your VPCs,x is your last two digits of your student number.

part1: OSPF

OSPF is a dynamic routing protocol that routes packets within a single Autonomous System (AS). In OSPF, the network is divided into different areas, each maintaining its Link State Database.Consider Figure 1, in this topology, the network is divided into five different areas. Use this topology and configure routers to use OSPF as their routing protocol. Define areas and their boundaries EXACTLY as specified in this figure. All PCs should be able to ping each other.

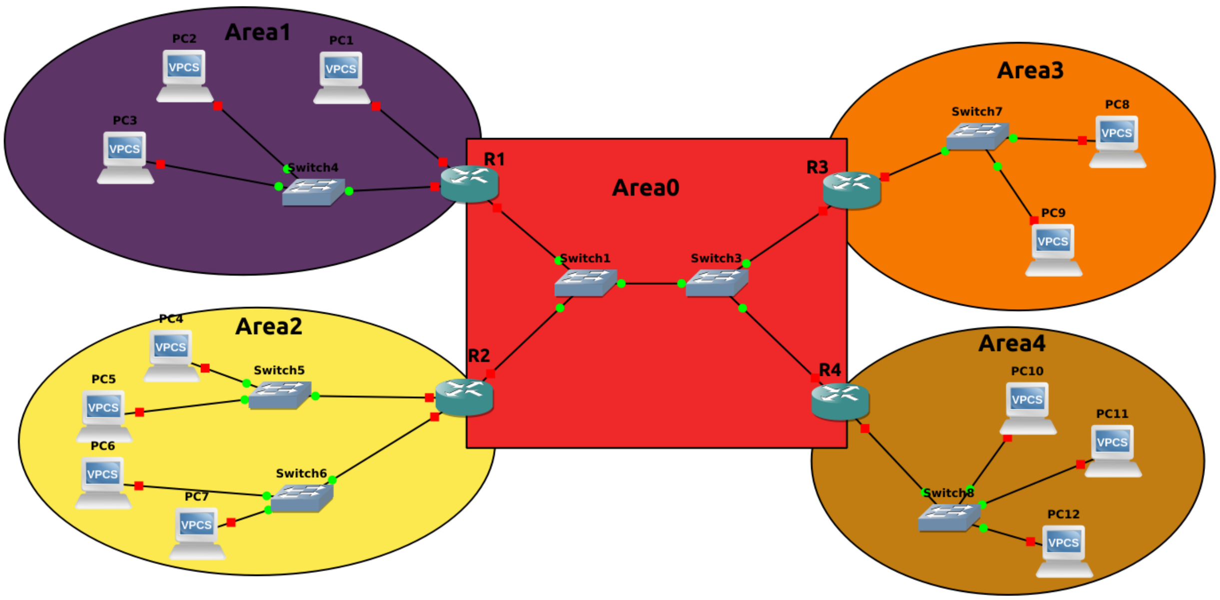

Figure 1: Network Topology Q1

Questions

- Provide screenshots of the following pings: PC1->PC10, PC8->PC12, and PC6->PC3.
- Provide screenshots of captured packets in the intrerface of the VPC pinging(it must contain ARPs and ICMPs)
- Provide screenshots of routing tables of R1, R2, R3, and R4.

part2: BGP(Bonus)

BGP refers to a gateway protocol that enables the internet to exchange routing information between autonomous systems (AS). As networks interact with each other, they need a way to communicate. This is accomplished through peering. BGP makes peering possible. Without it, networks would not be able to send and receive information with each other. Define two more ASes, each of the ASes must have at least two router, two areas, and three switches. Use BGP in order to route Packets between ASes.

Questions

- Provide screenshots of the following pings showing ASes connected successfully.
- Provide screenshots of captured packets in the intrerface of the VPC pinging(it must contain ARPs and ICMPs)
- Provide screenshots of routing tables of routers using BGP.

Question 2: Routing in Fixed wired Networks (30 Points)

Overview

In this problem, we want to run a simulation to apply the routing algorithm in figure 2. Without loss of generality, we only want to find the lowest-cost route between nodes "A" and "E". Also note that the routing algorithm is based on Dijkstra's shortest path algorithm. Building physical and data link layer is pretty much the same as before, and you will use *NodeContainer* class to create network topology. You are encouraged to use your previous assignment and NS3 examples for building the corresponding layers.

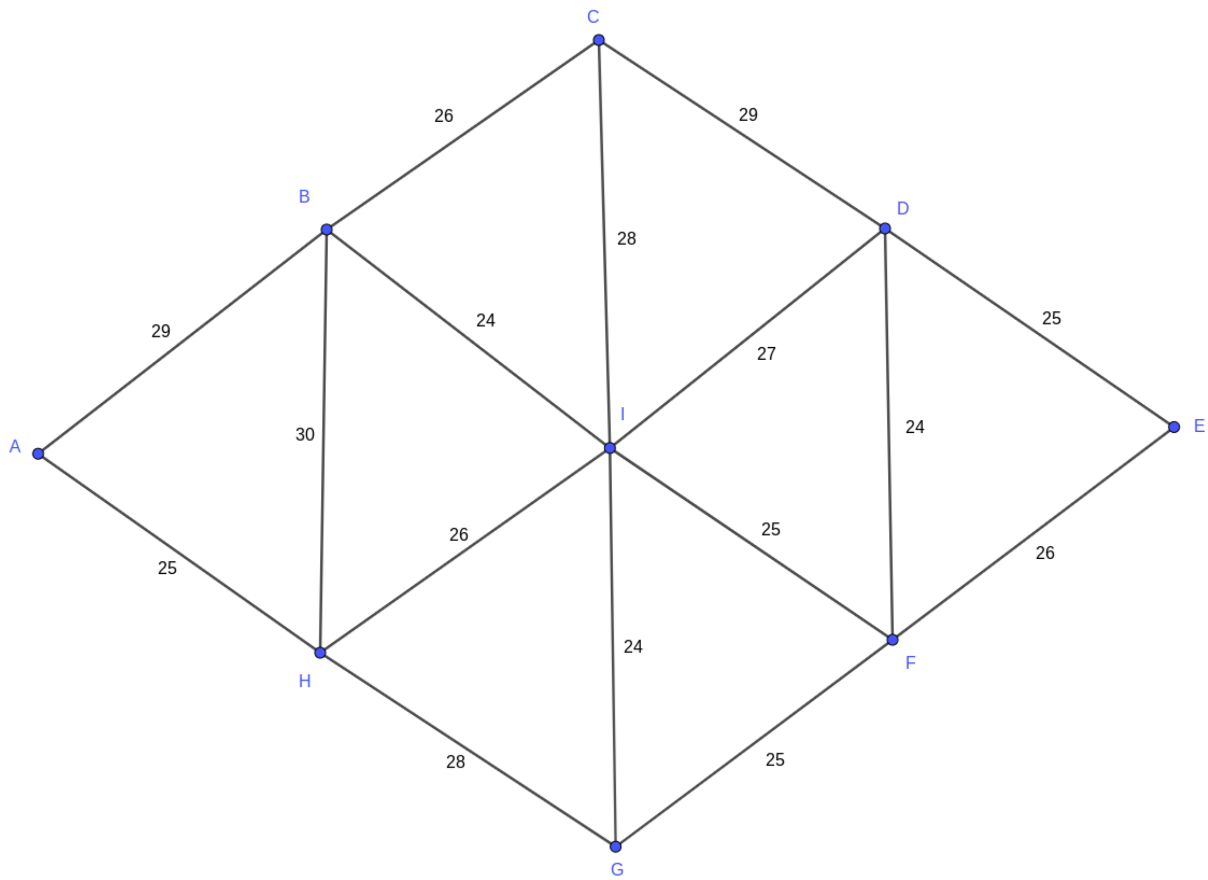

Figure 2: Network Topology Q2

Code

With your little knowledge and experience with NS3, it will be very difficult for you to write the c++ code for this problem from scratch. So we attached the base code file for this problem with the name "Network_Layer_Assignment.cc". Please refer to this code to solve this problem. Clearly the code is not complete and you have to add mentioned parts in the code. Note that the code is well commented and we omitted the explanation from this file. We strongly suggest to carefully read and understand the code and related comments.

Questions

- Do the simulation and compare the result with what calculated in the class. Now change the metrics of links AB and BI to 24 and 25, respectively. Again do the simulation and investigate whether the path for sending packets from "A" to "E" has been changed or not. Change the metric of link AH to this number: $\left\lfloor \frac{\textit{Last two digits of your student number}}{10} \right\rfloor + 26$. Again do the simulation and find the path by means of which "A" sends its packets to "E".

- Now, we want to evaluate the algorithm when a node sets down. You have to shut down all the links connected to the node to set it down. Refer to the attached code to find out how you can shut the links down.

- Now check the tables to find the new sink tree after the alternation. You can use NetAnim to see the routing process easily.

- In all above parts, you should take some snapshots from NetAnim to compare the results obtained by simulation with what you calculate manually by the method mentioned in the class.

Question 3: Traffic Shaping Protocols (40+5 Points)

In this question, you will experiment with two different traffic shaping protocols. You need to simulate these protocols using Python. If you need more information on Leaky and Token Bucket, feel free to refer to section 5.4.2 of your main textbook[1].

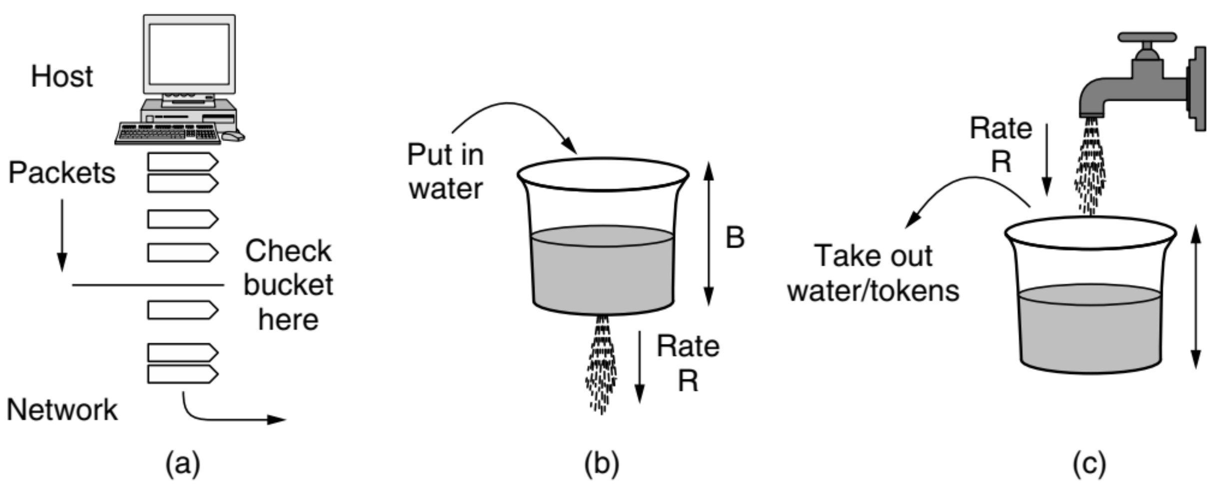

Figure 3: (a)Shaping packets. (b) A leaky bucket. (c) A token bucket.

Leaky Bucket

Let us consider a client that only generates 5kb packets, and we aim to limit its data usage by 1 Mbps. One way to achieve this goal is to ensure the client has at least 5 ms between its packet transmissions. This is

[1]Computer networks / Andrew S. Tanenbaum, David J. Wetherall. – 5th ed.

what we call Leaky Bucket. For example, let's say the client has generated its packets at the *arrival* times. a Leaky Bucket would allow these packets to be transmitted at the *departure* times given in milliseconds:

$$arrival = [6, 16, 17, 18, 20, 40] \rightsquigarrow departure = [6, 16, 21, 26, 31, 40]$$

Also, if we consider a capacity of 100kb, after receiving 100kb packets, the next packet will be discarded. For example, let's say the client has generated its packets at the following times:

$$arrival = [1, 2, 3, 4, 5, 6, ..., 26, 27]ms$$

the packets with the $arrival\ time = [26, 27]ms$ will be droped.

We are interested in two parameters: 1- average delay 2- throughput. These will serve as quantities with which we can measure the performance of our protocol. Before you start with the question itself, let us define one more variable which will come in handy in our analysis, and that is server utilization rate:

$$\rho = \frac{\lambda}{\mu}$$

Where λ is rate of packet arrival and μ is rate of service.

Additional note for interested students: If you are familiar with queuing theory, you might have noticed that we can think of Leaky Bucket as a G/D/1 queuing system where we are only interested in the time that packets spend on the Que itself. G in G/D/1 refers to "General Distribution" and describes the packet arrival process, D denotes "Deterministic Distribution" since each packet has a fixed time of service (in our example 5ms) and describes the service process. Finally, 1 means we have only one server. As mentioned before, we only care about the time packets spend in the Que. If we consider the whole time packets spend on the system, we introduce an artificial delay that does not exist in the Leaky Bucket.

Leaky Bucket simulation

Write a function that takes two parameters as input - first, a vector containing all the packet arrival times, and second, service time. Then outputs are a vector containing the times that these packets exit the system, and a vector denoting the discarded packets. The arrival time vector should be constructed according to the following parts:

- **Deterministic:** Packets are generated with a fixed inter-arrival time = T.
- **Deterministic with Batch Arrival:** Packets are generated with a fixed inter-arrival time = 5T. However, when a packet is generated, so are four others. In other words, five packets are generated at the same time. This way, the average rate of packet generation remains the same. This can be seen as a simple emulation of bursty traffic. Note that in reality, even bursty packets are not generated at the exact same time; however, we made this assumption to simplify the simulation code.
- **Poisson Process**: The number of packets in any time interval is given by Poisson distribution. This is equivalent to saying that the inter-arrival time is given by an exponential distribution with mean T. Remember that exponential distribution is of interest in queuing theory mainly because it is memory-less.
- **Poisson Process with Batch Arrival**: This is a combination of the last two parts meaning that inter-arrival time is given by exponential distribution with mean 5T and 5 packets are generated at the same time.

Questions

- What is the relation between server utilization rate, average inter-arrival time and service time (Which is 5ms)?

- Sweep the value T in the range of 20ms to 3ms. Generate arrival time vector according to instructions. With the help of the function you wrote in 1.1.1, calculate the departure time vector and finally calculate the average delay imposed by the protocol as well as throughput. Note that your result depends on the time vector generated when calculating these values for non-deterministic arrival times. To ensure your values are not far from the average, choose your vector to be large enough (10000 is a good starting point).

- Finally, you must submit two plots. One where you compare the average delay as a function of ρ for all four distributions mentioned above and one where your compare throughput under the same conditions. Don't forget to interpret the results you have obtained.

Token Bucket

This is a slightly different method to achieve the goal of traffic shaping. In Token Bucket, tokens are deposited in the bucket with a fixed rate R until it reaches its maximum capacity of B tokens. Tokens can be viewed as "the right to transmit a packet." This means, when the client requests a packet to be transmitted, its request is granted if at least one token is available in the bucket; Otherwise, it has to wait. Let us consider the example in part one:

$$arrival = [6, 16, 17, 18, 20, 40]$$

Since tokens are created each 5 ms we will have:

$$departure = [6, 16, 17, 20, 25, 40]$$

For now, let us assume that B is infinite, or in other words, there is no limit to the number of tokens our bucket can hold. Now that you have understood how it works, it is time to implement a function that simulates this behavior. After that, you should repeat the same steps discussed in Leaky Bucket and answer the second and third questions. Additionally, you will answer the following questions as well.

Questions

- For each of the four distributions discussed in part 1, submit two plots: One comparing the average delay of Leaky Bucket with that of Token Bucket. And one comparing throughput of Leaky Bucket with that of Token Bucket. In total, eight plots should be submitted for this question. Again, make sure to explain the trends in your report

- What is the effect of changing B to a finite value? In other words, discuss the pros and cons of increasing or decreasing B. Hint: start by answering what happens when we set B=1.

- Which one of the distributions proposed in this assignment do you think represents the data pattern we encounter in the real world more accurately?

- (Bonus) We managed to describe Leaky Bucket as a G/D/1 queuing system. Can you do the same for Token Bucket?

What Should I Do?

You must upload anything that you've been asked to upload and you should also answer the questions in your report. Your report should be complete and you should fully explain your code and the API that you have used in your code. Make a folder for each section and put code and output files of each section in its corresponding folder. Compress all files and rename the compressed file to "STUDENT_ID_HW4.zip".
If you have any questions regarding the problem statement or understanding the concept, feel free to ask in Telegram comments platform.