Data networks

DR.Pakravan

electrical engineering department

Ahmadreza Majlesara 400101861

assignment 2

May 30, 2024

Data networks

assignment 2

Ahmadreza Majlesara 400101861

1: Ethernet simulation

In this question, we are going to simulate Ethernet using a hub and a switch and investigate the performance of the network under different conditions. For this question, you can get help from the examples in the directory "inet4.5/examples/ethernet/lans".

part 1

1. In this part, we are going to simulate a simple star-topology network. As shown in Figure 2, one `EthernetHub` module and 8 `EthernetHost` modules are used. Use `Eth10M` for the channel and set length parameter to 10m. You can find these modules in `inet.node.ethernet` package. Assume one of the hosts as a server(sending no traffic) and others as clients. Run simulation for 50 sec. Set the other parameters as follows:

   - `**.cli.sendInterval = exponential(100ms)`
   - `**.cli.reqLength = intuniform(50,1400)*1B`
   - `**.cli.respLength = intWithUnit(truncnormal(3000B,3000B))`

   In the results folder of your project, you should be able to find a `.sca` file. Double-click on it to create a `.anf` file. Then, go to the 'Browse Data' tab in the `.anf` file, select the 'Scalers' tab, and collect the following statistics to include in your report: `rx channel idle(%)`, `rx channel utilization(%)`, `rx channel collision(%)` for each host. Further, calculate *network throughput(bit/sec) = total number of bits successfully arrived / simulation time*. To do so, you can use `packetReceived:sum(packetBytes)` in the statistics.

solution

thi .ini file and .ned file are shown in figure 1 and 2. for the ned file it is a simple network with parts from inet framework.in ini file i just set the destination of host1 to null so it doesnt send any packets as client and set other hosts destination to host 1 so they send their data to host 1 and it acts as a server. the results are as below:

1. `rx channel idle(%)` shown in figure 3
2. `rx channel utilization(%)` shown in figure 4
3. `rx channel collision(%)` shown in figure 5
4. `packetReceived:sum(packetBytes)` shown in figure 6

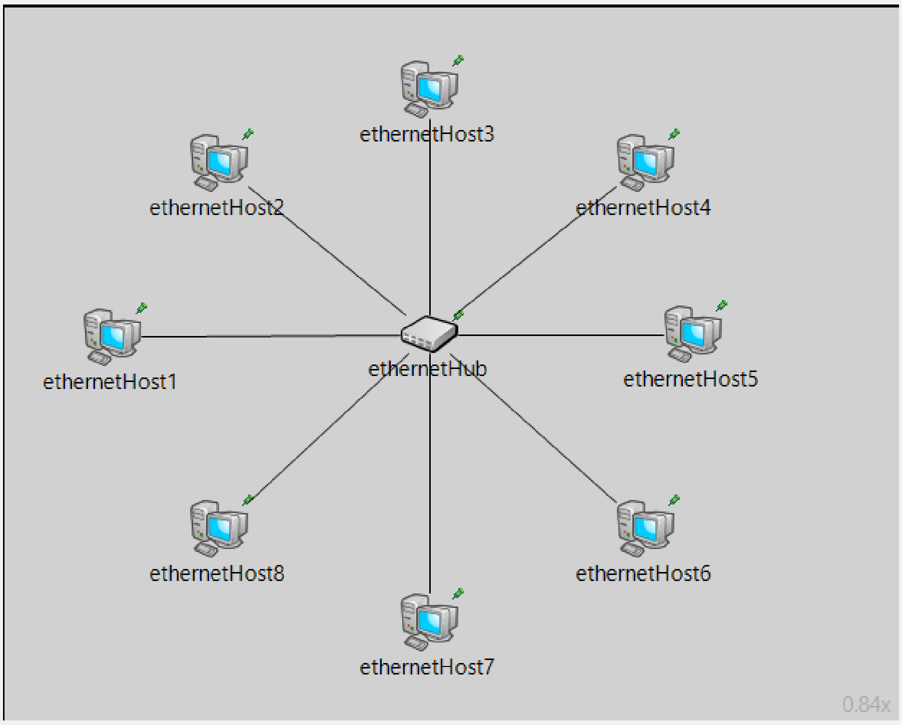


Figure 1. network for Q1 part 1

```
[General]
network = hw3_q1.simulations.Ethernet_network

sim-time-limit = 50s

# Specify that ethernetHost2 to ethernetHost8 send data to ethernetHost1
**.ethernetHost1.cli.destAddress = ""
**.cli.destAddress = "ethernetHost1"
**.cli.sendInterval = exponential(100ms)
**.cli.reqLength = intuniform(50,1400)*1B
**.cli.respLength = intWithUnit(truncnormal(3000B,3000B))
```

Figure 2. ini file for Q1 part 1

General		#0	Ethernet_network.ethernetHost1.eth.mac	rx channel idle (%)	95.856264
General		#0	Ethernet_network.ethernetHost2.eth.mac	rx channel idle (%)	76.020758
General		#0	Ethernet_network.ethernetHost3.eth.mac	rx channel idle (%)	76.027674
General		#0	Ethernet_network.ethernetHost4.eth.mac	rx channel idle (%)	76.047525
General		#0	Ethernet_network.ethernetHost5.eth.mac	rx channel idle (%)	76.033814
General		#0	Ethernet_network.ethernetHost6.eth.mac	rx channel idle (%)	76.021784
General		#0	Ethernet_network.ethernetHost7.eth.mac	rx channel idle (%)	76.046931
General		#0	Ethernet_network.ethernetHost8.eth.mac	rx channel idle (%)	75.991647

Figure 3. rx channel idle(%) for Q1 part 1

General		#0	Ethernet_network.ethernetHost1.eth.mac	rx channel utilization (%)	3.873446
General		#0	Ethernet_network.ethernetHost2.eth.mac	rx channel utilization (%)	21.718498
General		#0	Ethernet_network.ethernetHost3.eth.mac	rx channel utilization (%)	21.801536
General		#0	Ethernet_network.ethernetHost4.eth.mac	rx channel utilization (%)	21.795496
General		#0	Ethernet_network.ethernetHost5.eth.mac	rx channel utilization (%)	21.750154
General		#0	Ethernet_network.ethernetHost6.eth.mac	rx channel utilization (%)	21.723837
General		#0	Ethernet_network.ethernetHost7.eth.mac	rx channel utilization (%)	21.636925
General		#0	Ethernet_network.ethernetHost8.eth.mac	rx channel utilization (%)	21.820165

Figure 4. rx channel utilization(%) for Q1 part 1

General		#0	Ethernet_network.ethernetHost1.eth.mac	rx channel collision (%)	0.270290
General		#0	Ethernet_network.ethernetHost2.eth.mac	rx channel collision (%)	2.260744
General		#0	Ethernet_network.ethernetHost3.eth.mac	rx channel collision (%)	2.170790
General		#0	Ethernet_network.ethernetHost4.eth.mac	rx channel collision (%)	2.156979
General		#0	Ethernet_network.ethernetHost5.eth.mac	rx channel collision (%)	2.216032
General		#0	Ethernet_network.ethernetHost6.eth.mac	rx channel collision (%)	2.254379
General		#0	Ethernet_network.ethernetHost7.eth.mac	rx channel collision (%)	2.316144
General		#0	Ethernet_network.ethernetHost8.eth.mac	rx channel collision (%)	2.188189

Figure 5. rx channel collision(%) for Q1 part 1

Experiment	Measurement	Replica	Module	Name	Value
General		#0	Ethernet_network.ethernetHost1.cli	packetReceived:	0
General		#0	Ethernet_network.ethernetHost1.srv	packetReceived:	2,325,900
General		#0	Ethernet_network.ethernetHost2.cli	packetReceived:	1,656,794
General		#0	Ethernet_network.ethernetHost2.srv	packetReceived:	0
General		#0	Ethernet_network.ethernetHost3.cli	packetReceived:	1,619,771
General		#0	Ethernet_network.ethernetHost3.srv	packetReceived:	0
General		#0	Ethernet_network.ethernetHost4.cli	packetReceived:	1,700,507
General		#0	Ethernet_network.ethernetHost4.srv	packetReceived:	0
General		#0	Ethernet_network.ethernetHost5.cli	packetReceived:	1,774,337
General		#0	Ethernet_network.ethernetHost5.srv	packetReceived:	0
General		#0	Ethernet_network.ethernetHost6.cli	packetReceived:	1,637,212
General		#0	Ethernet_network.ethernetHost6.srv	packetReceived:	0
General		#0	Ethernet_network.ethernetHost7.cli	packetReceived:	1,805,313
General		#0	Ethernet_network.ethernetHost7.srv	packetReceived:	0
General		#0	Ethernet_network.ethernetHost8.cli	packetReceived:	1,529,315
General		#0	Ethernet_network.ethernetHost8.srv	packetReceived:	0
General		#0	Ethernet_network.ethernetHub	packetReceived:	15,845,412

Figure 6. packetReceived:sum(packetBytes) for Q1 part 1

so the network throughput is $\frac{11723249\times 8 \ bits}{50 \ s} \simeq 1.872 Mbps$

part 2

Replace the `EthernetHost` module with the `EthernetSwitch` module. Repeat Part 1 for this new network configuration (other configuration setups are the same as part 1). Compare the collision rate and network throughput with those from Part 1 and explain the results. Additionally, include the MAC address table of the switch in your report.

solution

the ini file is as the same as part 1 but the ned file is changed to use switch instead of host the network is shown in figure 7. the results are as below:

1. `rx channel idle(%)` shown in figure 8
2. `rx channel utilization(%)` shown in figure 9
3. `rx channel collision(%)` shown in figure 10
4. `packetReceived:sum(packetBytes)` shown in figure 11
5. `MAC address table` shown in figure 12

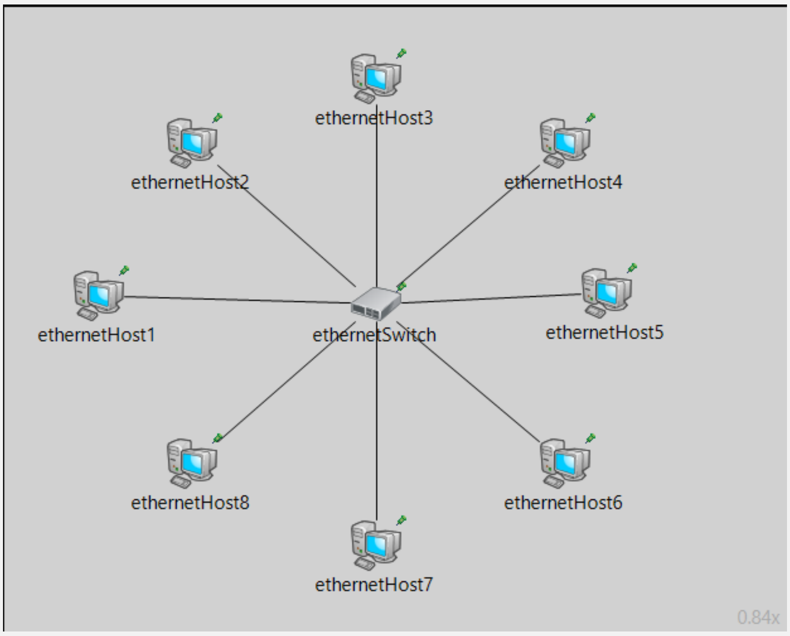
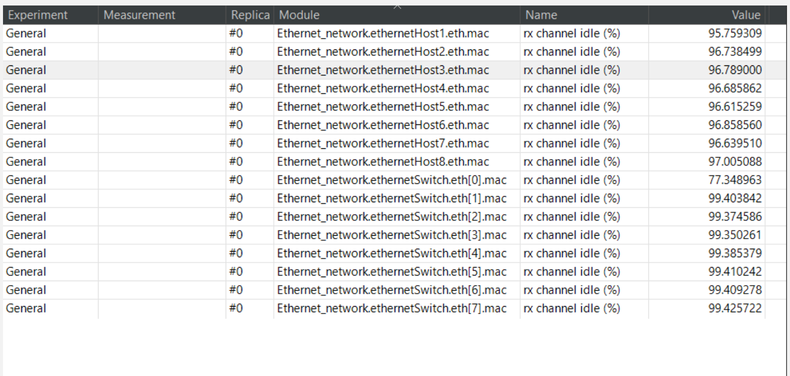


Figure 7. network for Q1 part 2



Experiment	Measurement	Replica	Module	Name	Value
General		#0	Ethernet_network.ethernetHost1.eth.mac	rx channel idle (%)	95.759309
General		#0	Ethernet_network.ethernetHost2.eth.mac	rx channel idle (%)	96.738499
General		#0	Ethernet_network.ethernetHost3.eth.mac	rx channel idle (%)	96.789000
General		#0	Ethernet_network.ethernetHost4.eth.mac	rx channel idle (%)	96.685862
General		#0	Ethernet_network.ethernetHost5.eth.mac	rx channel idle (%)	96.615259
General		#0	Ethernet_network.ethernetHost6.eth.mac	rx channel idle (%)	96.858560
General		#0	Ethernet_network.ethernetHost7.eth.mac	rx channel idle (%)	96.639510
General		#0	Ethernet_network.ethernetHost8.eth.mac	rx channel idle (%)	97.005088
General		#0	Ethernet_network.ethernetSwitch.eth[0].mac	rx channel idle (%)	77.348963
General		#0	Ethernet_network.ethernetSwitch.eth[1].mac	rx channel idle (%)	99.403842
General		#0	Ethernet_network.ethernetSwitch.eth[2].mac	rx channel idle (%)	99.374586
General		#0	Ethernet_network.ethernetSwitch.eth[3].mac	rx channel idle (%)	99.350261
General		#0	Ethernet_network.ethernetSwitch.eth[4].mac	rx channel idle (%)	99.385379
General		#0	Ethernet_network.ethernetSwitch.eth[5].mac	rx channel idle (%)	99.410242
General		#0	Ethernet_network.ethernetSwitch.eth[6].mac	rx channel idle (%)	99.409278
General		#0	Ethernet_network.ethernetSwitch.eth[7].mac	rx channel idle (%)	99.425722

Figure 8. rx channel idle(%) for Q1 part 2

Experiment	Measurement	Replica	Module	Name	Value
General		#0	Ethernet_network.ethernetHost1.eth.mac	rx channel utilization (%	4.240691
General		#0	Ethernet_network.ethernetHost2.eth.mac	rx channel utilization (%	3.261501
General		#0	Ethernet_network.ethernetHost3.eth.mac	rx channel utilization (%	3.211000
General		#0	Ethernet_network.ethernetHost4.eth.mac	rx channel utilization (%	3.314138
General		#0	Ethernet_network.ethernetHost5.eth.mac	rx channel utilization (%	3.384741
General		#0	Ethernet_network.ethernetHost6.eth.mac	rx channel utilization (%	3.141440
General		#0	Ethernet_network.ethernetHost7.eth.mac	rx channel utilization (%	3.360490
General		#0	Ethernet_network.ethernetHost8.eth.mac	rx channel utilization (%	2.994912
General		#0	Ethernet_network.ethernetSwitch.eth[0].mac	rx channel utilization (%	22.651037
General		#0	Ethernet_network.ethernetSwitch.eth[1].mac	rx channel utilization (%	0.596158
General		#0	Ethernet_network.ethernetSwitch.eth[2].mac	rx channel utilization (%	0.625414
General		#0	Ethernet_network.ethernetSwitch.eth[3].mac	rx channel utilization (%	0.649739
General		#0	Ethernet_network.ethernetSwitch.eth[4].mac	rx channel utilization (%	0.614621
General		#0	Ethernet_network.ethernetSwitch.eth[5].mac	rx channel utilization (%	0.589758
General		#0	Ethernet_network.ethernetSwitch.eth[6].mac	rx channel utilization (%	0.590722
General		#0	Ethernet_network.ethernetSwitch.eth[7].mac	rx channel utilization (%	0.574278

Figure 9. rx channel utilization(%) for Q1 part 2

Experiment	Measurement	Replica	Module	Name	Value
General		#0	Ethernet_network.ethernetHost1.eth.mac	rx channel collision (%)	0
General		#0	Ethernet_network.ethernetHost2.eth.mac	rx channel collision (%)	0
General		#0	Ethernet_network.ethernetHost3.eth.mac	rx channel collision (%)	0
General		#0	Ethernet_network.ethernetHost4.eth.mac	rx channel collision (%)	0
General		#0	Ethernet_network.ethernetHost5.eth.mac	rx channel collision (%)	0
General		#0	Ethernet_network.ethernetHost6.eth.mac	rx channel collision (%)	0
General		#0	Ethernet_network.ethernetHost7.eth.mac	rx channel collision (%)	0
General		#0	Ethernet_network.ethernetHost8.eth.mac	rx channel collision (%)	0

Figure 10. rx channel collision(%) for Q1 part 2

Experiment	Measurement	Replica	Module	Name	Value
General		#0	Ethernet_network.ethernetHost1.cli	packetReceived:sum(pa	0
General		#0	Ethernet_network.ethernetHost1.srv	packetReceived:sum(pa	2,548,033
General		#0	Ethernet_network.ethernetHost2.cli	packetReceived:sum(pa	1,990,030
General		#0	Ethernet_network.ethernetHost2.srv	packetReceived:sum(pa	0
General		#0	Ethernet_network.ethernetHost3.cli	packetReceived:sum(pa	1,959,926
General		#0	Ethernet_network.ethernetHost3.srv	packetReceived:sum(pa	0
General		#0	Ethernet_network.ethernetHost4.cli	packetReceived:sum(pa	2,022,007
General		#0	Ethernet_network.ethernetHost4.srv	packetReceived:sum(pa	0
General		#0	Ethernet_network.ethernetHost5.cli	packetReceived:sum(pa	2,066,104
General		#0	Ethernet_network.ethernetHost5.srv	packetReceived:sum(pa	0
General		#0	Ethernet_network.ethernetHost6.cli	packetReceived:sum(pa	1,917,093
General		#0	Ethernet_network.ethernetHost6.srv	packetReceived:sum(pa	0
General		#0	Ethernet_network.ethernetHost7.cli	packetReceived:sum(pa	2,050,778
General		#0	Ethernet_network.ethernetHost7.srv	packetReceived:sum(pa	0
General		#0	Ethernet_network.ethernetHost8.cli	packetReceived:sum(pa	1,826,899
General		#0	Ethernet_network.ethernetHost8.srv	packetReceived:sum(pa	0

Figure 11. packetReceived:sum(packetBytes) for Q1 part 2

```
Ethernet_network.ethernetSwitch.macTable.forwardingTable (map<pair,AddressEntry>) size=8
  elements[8] (pair<,>)
    [0] {VID=0, addr=0A-AA-00-00-00-01} ==> {interfaceId=102, insertionTime=49.972174061561}
    [1] {VID=0, addr=0A-AA-00-00-00-02} ==> {interfaceId=101, insertionTime=49.962871317223}
    [2] {VID=0, addr=0A-AA-00-00-00-03} ==> {interfaceId=100, insertionTime=49.976868561561}
    [3] {VID=0, addr=0A-AA-00-00-00-04} ==> {interfaceId=107, insertionTime=49.896097552834}
    [4] {VID=0, addr=0A-AA-00-00-00-05} ==> {interfaceId=106, insertionTime=49.774952947222}
    [5] {VID=0, addr=0A-AA-00-00-00-06} ==> {interfaceId=105, insertionTime=49.920229572463}
    [6] {VID=0, addr=0A-AA-00-00-00-07} ==> {interfaceId=104, insertionTime=49.8610833171}
    [7] {VID=0, addr=0A-AA-00-00-00-08} ==> {interfaceId=103, insertionTime=49.543918538498}
```

Figure 12. MAC address table for EthernetSwitch

so we can calculate the network throughput as $\frac{11766733 \times 8 \ bits}{50 \ s} \simeq 1.88 Mbps$.so the network throughput is increased by 1 kbps. the collision rate using switch is 0% because switch store the packets and send them when the channel is free but hub sends the packets to all hosts and if two hosts send packets at the same time the packets collide and the collision rate increases. so the switch is better than hub in terms of collision rate and network throughput.

part 3

In this part we are going to simulate the network shown in Figure 3 (pay attention to distances depicted in the figure). This configuration has both `EthernetSwitch` and `EthernetHub`. We connect the vector of `EthernetHost` modules to the switch. To define the vector you can go to properties of one `EthernetHost` in the Design mode of your `.ned` file and enter the parameter `k` in front of the vector as shown in Figure 4. You should also add parameter `k` to your network definition in Source mode of `.ned` file like below:

```
network MixedLAN
{
    parameters:
        ...
        int k;
        ...
}
```

Then, in Source mode, you can easily connect all `k` hosts to the switch using a single `for` loop. Use `Eth10M` for all connections. We aim to conduct some experiments to determine the impact of packet size and traffic load on network throughput. To accomplish this, you must simulate the network across six scenarios (remember that `k` represents the number of hosts connected to the switch):

- scenario1: `reqLength=500B , k=8`
- scenario2: `reqLength=500B , k=20`
- scenario3: `reqLength=500B , k=100`
- scenario4: `reqLength=500B , k=200`
- scenario5: `reqLength=100B , k=8`
- scenario6: `reqLength=1400B , k=8`

The other settings in the `.ini` file are the same as in Part 1. After calculating the network throughput for each scenario, compare them. What is the impact of traffic load and packet size on network throughput? What is the ideal maximum throughput that can be achieved in this network? Why?

solution

the network is shown in figure 13. to setup k hosts i used the code shown in figure 14 from .ned file. at last we set the reqLength and k in the .ini file.

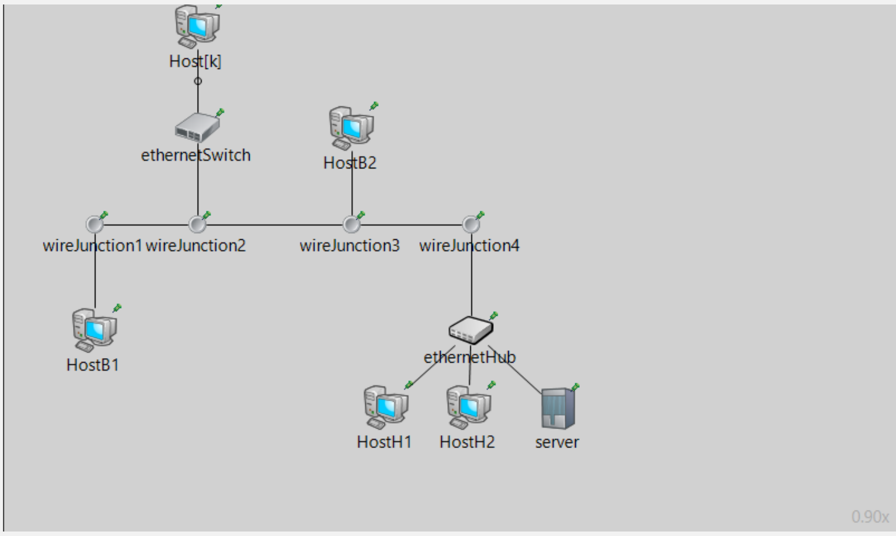


Figure 13. network for Q1 part 3

```
    }
    connections:
        HostB1.ethg <--> Eth10M {  length = 10m; } <--> wireJunction1.port++;
        wireJunction2.port++ <--> Eth10M {  length = 1cm; } <--> ethernetSwitch.ethg[0];
        wireJunction2.port++ <--> Eth10M {  length = 20m; } <--> wireJunction1.port++;
        wireJunction2.port++ <--> Eth10M {  length = 20m; } <--> wireJunction3.port++;
        wireJunction3.port++ <--> Eth10M {  length = 10m; } <--> HostB2.ethg;
        wireJunction3.port++ <--> Eth10M {  length = 20m; } <--> wireJunction4.port++;
        wireJunction4.port++ <--> Eth10M {  length = 1cm; } <--> ethernetHub.ethg++;
        ethernetHub.ethg++ <--> Eth10M {  length = 10m; } <--> HostH1.ethg;
        ethernetHub.ethg++ <--> Eth10M {  length = 10m; } <--> HostH2.ethg;
        ethernetHub.ethg++ <--> Eth10M {  length = 10m; } <--> server.ethg;
        for i=1..k {
            Host[i-1].ethg <--> Eth10M {  length = 10m; } <--> ethernetSwitch.ethg[i];
        }

}
```

Figure 14. setting up k hosts in ned file

```
[General]
network = hw3_q1_3.simulations.MixedLAN

sim-time-limit = 50s

[Config scenario1]
**.server.cli.destAddress = ""
**.cli.destAddress = "server"
**.switch*.bridging.typename = "MacRelayUnit"
**.cli.sendInterval = exponential(100ms)
**.cli.respLength = intWithUnit(truncnormal(3000B,3000B))
**.cli.reqLength = 500B
MixedLAN.k=8
```

Figure 15. ini file for Q1 part 3

by selecting each scenario in simulation menu we can get the outputs as below:

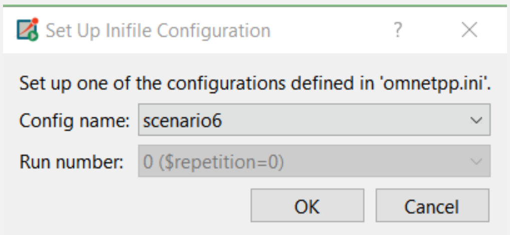


Figure 16. simulation menu

so the throughputs will be calculated as below:

	Throughput
Scenario 1	3.39 Mbps
Scenario 2	6.77 Mbps
Scenario 3	9.54 Mbps
Scenario 4	9.7 Mbps
Scenario 5	3.53 Mbps
Scenario 6	3.02 Mbps

Table 1: Network throughput for different scenarios

the plot of this table is shown in figure 17. as we can see the network throughput increases by increasing the number of hosts but in a fixed number of hosts if the packet size increases the network throughput decreases. the ideal maximum throughput that can be achieved in this network is 10 Mbps because the minimum throuput of channels is 10 Mbps and throughput cant reach a number higher than that.

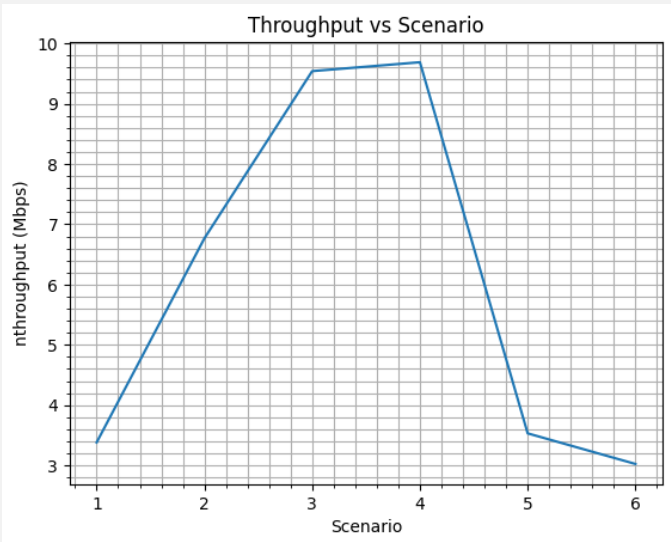


Figure 17. Network throughput for different scenarios

2: VLAN

Virtual Local Area Networks (VLANs) are a way to logically segment a single physical network into multiple virtual networks. VLANs allow network administrators to group devices and users based on their requirements, improving network performance, security, and management. By creating separate broadcast domains, VLANs reduce unnecessary traffic and improve overall efficiency. VLANs are widely used in enterprise networks, data centers, and other environments where network segmentation is required. You have two switches (Switch1 and Switch2) and six hosts (PC1, PC2, PC3, PC4, PC5, and PC6) that need to be configured for three VLANs (VLAN10, VLAN20, and VLAN30). The configuration should be as follows:

Intra-VLAN Ping

I. Configure the switches (Switch1 and Switch2) to support the three VLANs (VLAN10, VLAN20, and VLAN30).

II. Configure the PCs to be part of their respective VLANs:

- VLAN10: PC1, PC4
- VLAN20: PC2, PC5
- VLAN30: PC3, PC6

In your report please explain your code and the configuration steps.

III. Write a script or code to perform the following ping tests

- PC1 pings PC4 (VLAN10)
- PC3 pings PC6 (VLAN30)

solution

the network is shown in figure 18.

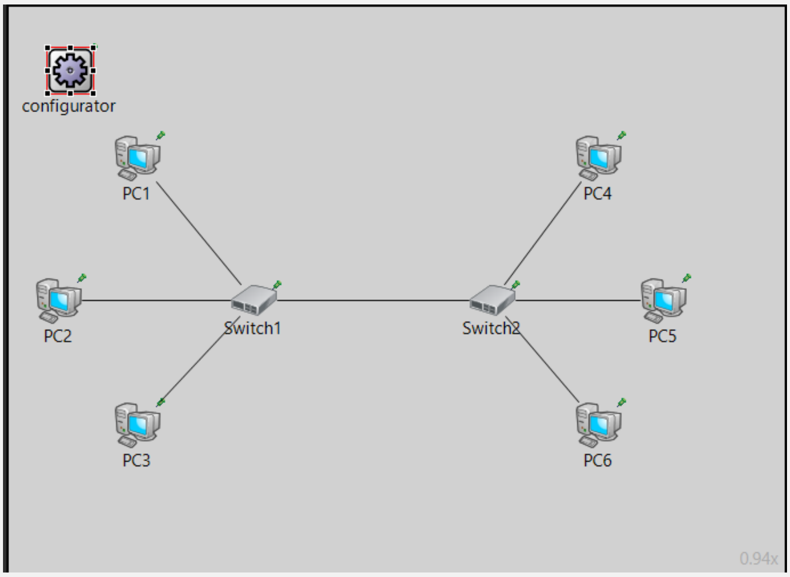


Figure 18. Network diagram

you can see the .NED file in figure 19.

```
network VLANNetwork
{
    @display("bgb=721,526");
    submodules:
        Switch1: EthernetSwitch {
            @display("p=228,273");
            gates:
                ethg[4];
        }
        Switch2: EthernetSwitch {
            @display("p=450,273");
            gates:
                ethg[4];
        }
        PC1: EthernetHost {
            @display("p=120,142");
        }
        PC2: EthernetHost {
            @display("p=47,274");
        }
        PC3: EthernetHost {
            @display("p=120,390");
        }
        PC4: EthernetHost {
            @display("p=549,142");
        }
        PC5: EthernetHost {
            @display("p=609,274");
        }
        PC6: EthernetHost {
            @display("p=549,390");
        }
        configurator: Ipv4NetworkConfigurator {
            @display("p=58,60");
        }
    connections:
        PC4.ethg <--> Eth100M <--> Switch2.ethg[1];
        PC5.ethg <--> Eth100M <--> Switch2.ethg[2];
        PC6.ethg <--> Eth100M <--> Switch2.ethg[3];
        Switch2.ethg[0] <--> Eth100M <--> Switch1.ethg[0];
        Switch1.ethg[1] <--> Eth100M <--> PC1.ethg;
        Switch1.ethg[2] <--> Eth100M <--> PC2.ethg;
```

Figure 19. NED file

The network consists of the following components:

1. Two Ethernet switches (Switch1 and Switch2) to support the VLANs.
2. Six Ethernet hosts (PC1, PC2, PC3, PC4, PC5, and PC6) representing the devices connected to the VLANs.
3. An IPv4 network configurator (configurator) to handle the IP address assignment and other network configuration tasks.

This network setup allows for the simulation of VLAN configurations, traffic management, inter-VLAN and intra-VLAN communication through the switches.
the functionality of the network is handled using a .XML file and a .ini file. the .xml code snippet which is shown in figure 20 configures the network interfaces of six hosts with specific IP addresses and subnet masks. Each <interface> tag specifies the network configuration for a particular host.this configuration assigns unique IP addresses and a common subnet mask to each of the six PCs, effectively organizing them into three different subnets:

1. Subnet 10.0.10.0/24 for PC1 and PC4.
2. Subnet 10.0.20.0/24 for PC2 and PC5.
3. Subnet 10.0.30.0/24 for PC3 and PC6.

on the other hand the .ini file does all the functionallity as all parts are from inet framework ther is no need for a .cc file.the .ini file which is shown in figure 21 sets up a VLAN network simulation with specific VLAN IDs for each PC's Ethernet interface, configures which PCs will communicate with each other, and specifies an external XML file for detailed network configuration. The simulation is limited to 10 seconds in duration.

```xml
<config>
    <interface hosts="PC1" address="10.0.10.1" netmask="255.255.255.0"/>
    <interface hosts="PC4" address="10.0.10.4" netmask="255.255.255.0"/>
    <interface hosts="PC2" address="10.0.20.2" netmask="255.255.255.0"/>
    <interface hosts="PC5" address="10.0.20.5" netmask="255.255.255.0"/>
    <interface hosts="PC3" address="10.0.30.3" netmask="255.255.255.0"/>
    <interface hosts="PC6" address="10.0.30.6" netmask="255.255.255.0"/>
</config>
```

Figure 20. XML file

```ini
[General]
network = hw3_q2.simulations.VLANNetwork
sim-time-limit = 10s

*.PC1.eth[0].vlanId = 10
*.PC2.eth[0].vlanId = 20
*.PC3.eth[0].vlanId = 30
*.PC4.eth[0].vlanId = 10
*.PC5.eth[0].vlanId = 20
*.PC6.eth[0].vlanId = 30
**.PC2.cli.destAddress = ""
**.PC4.cli.destAddress = ""
**.PC5.cli.destAddress = ""
**.PC6.cli.destAddress = ""
**.PC1.cli.destAddress = "PC4"
**.PC3.cli.destAddress = "PC6"
*.configurator.config = xmldoc("config.xml")
```

Figure 21. INI file

to show that PC1 can ping PC4 and PC3 can ping PC6 i recorded a video which is attached with files that have been sent. the name of the video is ping1.

Inter-VLAN Ping

I. Configure the switches (Switch1 and Switch2) to support inter-VLAN communication (e.g., using a trunk link).

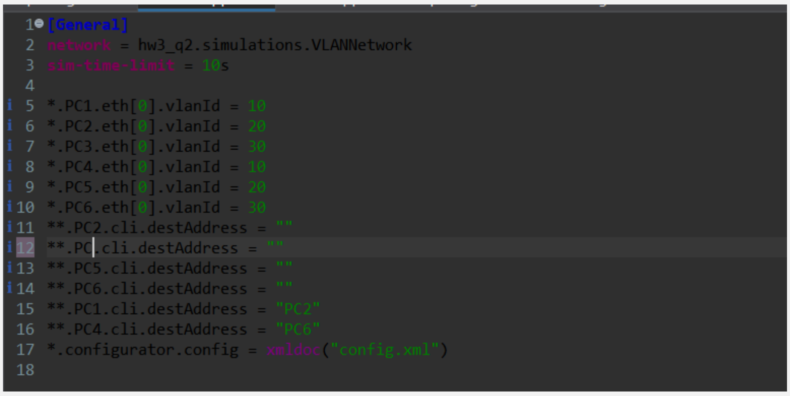
II. Write a script or code to perform the following ping tests:

- PC1 (VLAN10) pings PC2 (VLAN20)
- PC4 (VLAN10) pings PC6 (VLAN30)

Please explain your code and the configuration steps, including how inter-VLAN communication is achieved.

solution

the network topology and .NED and XML file is as the same as before. but the .ini file is changed so the network can support inter-VLAN. the .ini file is shown in figure 22.



```
[General]
network = hw3_q2.simulations.VLANNetwork
sim-time-limit = 10s

*.PC1.eth[0].vlanId = 10
*.PC2.eth[0].vlanId = 20
*.PC3.eth[0].vlanId = 30
*.PC4.eth[0].vlanId = 10
*.PC5.eth[0].vlanId = 20
*.PC6.eth[0].vlanId = 30
**.PC2.cli.destAddress = ""
**.PC.cli.destAddress = ""
**.PC5.cli.destAddress = ""
**.PC6.cli.destAddress = ""
**.PC1.cli.destAddress = "PC2"
**.PC4.cli.destAddress = "PC6"
*.configurator.config = xmldoc("config.xml")
```

Figure 22. INI file

to show that PC1 can ping PC2 and PC4 can ping PC6 i recorded a video which is attached with files that have been sent. the name of the video is ping2.