Ns3 simulation code

Kratik Gupta 2022252
Shashank Mishra 2022603

SIMULATION

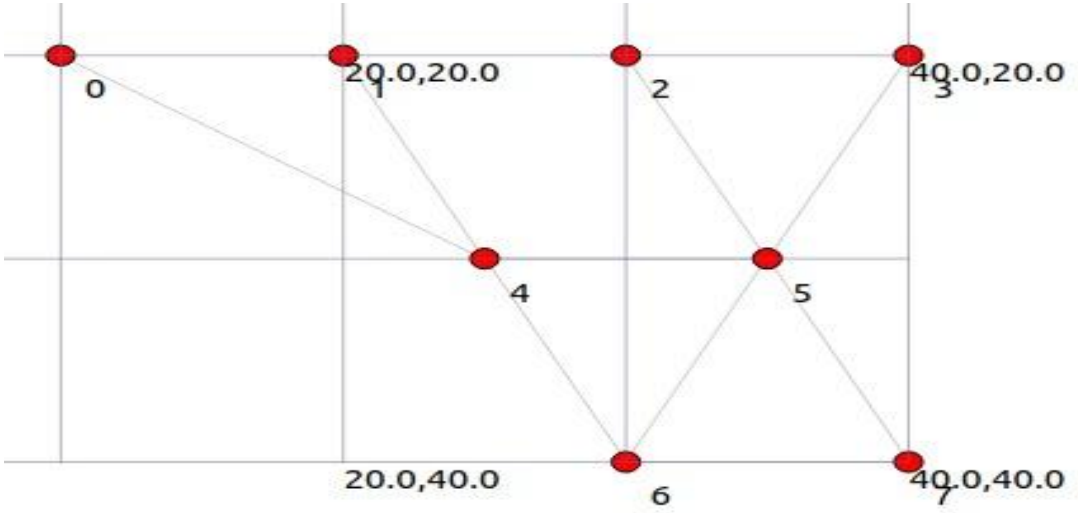
We are asked to analyze the performance of a custom network topology implemented using NS-3. The network consists of 5 servers 0-4 and four routers R1-R4. Point-to-point links are used to connect these components. The objectives of this assignment include the evaluation of end-to-end performance metrics such as average and variance of one-way delays, observation of packet drops, and monitoring of queue lengths at the outgoing links of the routers.

SIMULATION

We will trace the paths of specific packets traveling from source servers 0,1,2,3 to each other through the network, thereby providing a detailed understanding of the routing and transmission process. These observations will be recorded and tabulated in source-destination matrices for better comprehension and presentation of the results. Topology , Routing Table , End-to-end Delay , Queue Lengths and Packet Drops are discussed on subsequent slides.

TOPOLOGY

Nodes 0 , 1 , 2 and 3 are **workstations ,** they are end-user devices and are connected to each other through a router for network traffic. Nodes 4, 5 , 6 and 7 are **routers** , they serve as intermediary devices to route traffic between workstations . They are interconnected , forming a mesh-like structure. The following image illustrates our topology :-



Traffic

Source Node	Node 0	Node 1	Node 2	Node 3
Node 0	0	313	389	121
Node 1	667	0	426	407
Node 2	442	89	0	424
Node 3	458	237	319	0

ROUTING TABLE

The routing table demonstrates the network's hierarchical structure, showing paths between workstation nodes (Node 0–Node 3) and routers (R1–R4). Each entry highlights the next hop for packet delivery, emphasizing efficient routing through direct connections or intermediate routers to optimize communication

	Destination							
Source	Node 0	Node 1	Node 2	Node 3	R1	R2	R3	R4
Node 0	-	R1	R1	R1				
Node 1	R1		R1	R1				
Node 2	R2	R2		R2				
Node 3	R2	R2	R2					
R1(4)						R2	R3	R2
R2(5)					R1		R3	R4
R3(6)					R1	R2		R4
R4(7)					R3	R2	R3	

END-TO-END DELAY

The image attached displays end-to-end delays (average and variance) vary significantly across source-destination pairs. Routes like 10.1.5.1 -> 10.1.4.2 show more delay, which may indicate some performance problems, while others, such as 10.1.1.1 -> 10.1.1.2, maintain lower delay values.

```
End-to-End Delay (Average and Variance)
--------------------------------------------------------------------------------
Source: 10.1.1.1 -> Destination: 10.1.1.2, Value: (1.610005182481752, 0.1604889413417551)
Source: 10.1.1.1 -> Destination: 10.1.2.1, Value: (1.8024899242424242, 0.19438710957423666)
Source: 10.1.1.1 -> Destination: 10.1.2.2, Value: (1.5881757851239668, 0.11128737487892901)
Source: 10.1.1.1 -> Destination: 10.1.3.1, Value: (1.6021019834710744, 0.10497877717457815)
Source: 10.1.1.1 -> Destination: 10.1.3.2, Value: (1.7493546551724137, 0.15370769741798462)
Source: 10.1.1.1 -> Destination: 10.1.4.1, Value: (1.7717436097560977, 0.10770432876355499)
Source: 10.1.1.1 -> Destination: 10.1.4.2, Value: (1.8316618122977348, 0.16225954843749227)
Source: 10.1.1.1 -> Destination: 10.1.5.1, Value: (1.8032876699029126, 0.06466169170719198)
Source: 10.1.1.2 -> Destination: 10.1.1.1, Value: (1.7870546351084815, 0.26585363203315093)
Source: 10.1.2.1 -> Destination: 10.1.1.1, Value: (1.5199255529411766, 0.8189623100519883)
Source: 10.1.2.1 -> Destination: 10.1.2.2, Value: (0.9215191154791152, 1.018641035314697)
Source: 10.1.2.1 -> Destination: 10.1.3.1, Value: (1.9938406756756755, 0.16624472899413806)
Source: 10.1.2.1 -> Destination: 10.1.3.2, Value: (1.5983640438871474, 0.7752970052109193)
Source: 10.1.2.1 -> Destination: 10.1.4.1, Value: (2.0384775105485233, 0.12039394951996299)
Source: 10.1.2.1 -> Destination: 10.1.4.2, Value: (2.0330991304347825, 0.1323842274035917)
Source: 10.1.2.1 -> Destination: 10.1.5.1, Value: (2.0654660344827587, 0.08089671363944709)
Source: 10.1.4.2 -> Destination: 10.1.1.1, Value: (1.4528051528384278, 0.9129096035594744)
Source: 10.1.4.2 -> Destination: 10.1.1.2, Value: (2.023138902953587, 0.1678709228460539)
Source: 10.1.4.2 -> Destination: 10.1.2.2, Value: (2.028463417721519, 0.15348295698958503)
Source: 10.1.4.2 -> Destination: 10.1.3.1, Value: (1.5579179937304075, 0.8197206778486393)
Source: 10.1.4.2 -> Destination: 10.1.3.2, Value: (2.0112960958904105, 0.14586440699366202)
Source: 10.1.4.2 -> Destination: 10.1.4.1, Value: (1.9134057065217391, 0.1067691205114574)
Source: 10.1.4.2 -> Destination: 10.1.5.1, Value: (2.0254478378378376, 0.12324683234262235)
Source: 10.1.2.2 -> Destination: 10.1.2.1, Value: (1.8579949657534247, 0.2737038821534235)
Source: 10.1.5.2 -> Destination: 10.1.1.1, Value: (2.0029107843137255, 0.25702487213402536)
Source: 10.1.5.2 -> Destination: 10.1.1.2, Value: (2.0964402283105024, 0.158925403572094)
Source: 10.1.5.2 -> Destination: 10.1.2.1, Value: (2.0280674342105263, 0.2105140146749956 7)
Source: 10.1.5.2 -> Destination: 10.1.3.1, Value: (2.1199277314814813, 0.10376319837216863)
Source: 10.1.5.2 -> Destination: 10.1.3.2, Value: (2.122277615384616, 0.14378234675662127)
Source: 10.1.5.2 -> Destination: 10.1.4.1, Value: (2.1283753846153846, 0.13140012894331354)
Source: 10.1.5.2 -> Destination: 10.1.4.2, Value: (2.1302858139534884, 0.07875336978635296)
Source: 10.1.5.2 -> Destination: 10.1.5.1, Value: (2.1407273846153845, 0.10882052342239049)
Source: 10.1.3.2 -> Destination: 10.1.1.1, Value: (0.9424888461538461, 1.0333025675165421)
Source: 10.1.3.1 -> Destination: 10.1.4.2, Value: (0.9694268348623855, 1.0692418162564956)
Source: 10.1.3.1 -> Destination: 10.1.3.2, Value: (1.084049030837004, 1.0394664579576516)
Source: 10.1.3.1 -> Destination: 10.1.4.1, Value: (1.9240209900990102, 0.26472408881090087)
Source: 10.1.3.1 -> Destination: 10.1.5.1, Value: (1.9376247916666662, 0.22872548385828986)
Source: 10.1.3.2 -> Destination: 10.1.2.1, Value: (0.9404695754716981, 1.0621076792007633)
Source: 10.1.3.2 -> Destination: 10.1.1.2, Value: (1.9644796629213486, 0.20434886984820103)
```

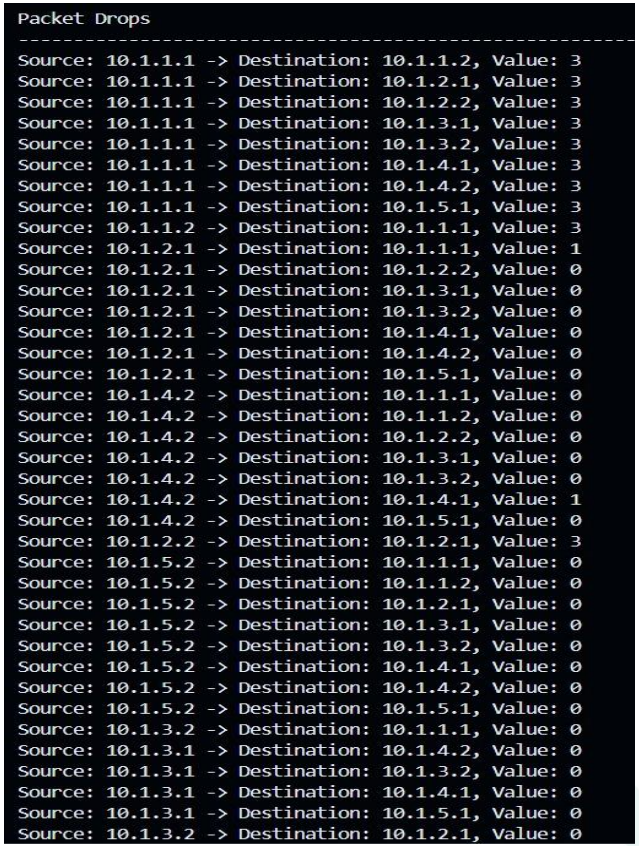
QUEUE LENGTHS

The queue length data suggests that the source-destination pairs have varying levels of congestion. Some links, such as 10.1.1.1 → 10.1.1.2, have very low congestion levels, while others, like 10.1.1.2 → 10.1.1.1, have high queuing levels, which may indicate uneven traffic or capacity.

```
Queue Lengths (Average)
--------------------------------------------------------------------------------
Source: 10.1.1.1 -> Destination: 10.1.1.2, Value: 4.571656050955414
Source: 10.1.1.1 -> Destination: 10.1.2.1, Value: 12.74391805377721
Source: 10.1.1.1 -> Destination: 10.1.2.2, Value: 7.848979591836735
Source: 10.1.1.1 -> Destination: 10.1.3.1, Value: 8.028571428571428
Source: 10.1.1.1 -> Destination: 10.1.3.2, Value: 12.031613976705492
Source: 10.1.1.1 -> Destination: 10.1.4.1, Value: 14.37046004842615
Source: 10.1.1.1 -> Destination: 10.1.4.2, Value: 16.040332147093714
Source: 10.1.1.1 -> Destination: 10.1.5.1, Value: 14.679518072289156
Source: 10.1.1.2 -> Destination: 10.1.1.1, Value: 32.49139865370232
Source: 10.1.2.1 -> Destination: 10.1.1.1, Value: 20.037558685446008
Source: 10.1.2.1 -> Destination: 10.1.2.2, Value: 8.890663390663391
Source: 10.1.2.1 -> Destination: 10.1.3.1, Value: 12.56081081081081
Source: 10.1.2.1 -> Destination: 10.1.3.2, Value: 16.321316614420063
Source: 10.1.2.1 -> Destination: 10.1.4.1, Value: 19.327004219409282
Source: 10.1.2.1 -> Destination: 10.1.4.2, Value: 25.490683229813666
Source: 10.1.2.1 -> Destination: 10.1.5.1, Value: 19.41810344827586
Source: 10.1.4.2 -> Destination: 10.1.1.1, Value: 19.975982532751093
Source: 10.1.4.2 -> Destination: 10.1.1.2, Value: 19.18354430379747
Source: 10.1.4.2 -> Destination: 10.1.2.2, Value: 19.896405919661735
Source: 10.1.4.2 -> Destination: 10.1.3.1, Value: 15.725705329153605
Source: 10.1.4.2 -> Destination: 10.1.3.2, Value: 12.636986301369863
Source: 10.1.4.2 -> Destination: 10.1.4.1, Value: 10.305772230889236
Source: 10.1.4.2 -> Destination: 10.1.5.1, Value: 12.783783783783784
Source: 10.1.2.2 -> Destination: 10.1.2.1, Value: 40.24911785462244
Source: 10.1.5.2 -> Destination: 10.1.1.1, Value: 24.320261437908496
Source: 10.1.5.2 -> Destination: 10.1.1.2, Value: 17.924657534246574
Source: 10.1.5.2 -> Destination: 10.1.2.1, Value: 23.74013157894737
Source: 10.1.5.2 -> Destination: 10.1.3.1, Value: 17.962962962962963
Source: 10.1.5.2 -> Destination: 10.1.3.2, Value: 11.338461538461539
Source: 10.1.5.2 -> Destination: 10.1.4.1, Value: 11.307692307692308
Source: 10.1.5.2 -> Destination: 10.1.4.2, Value: 21.74031007751938
Source: 10.1.5.2 -> Destination: 10.1.5.1, Value: 11.353846153846154
Source: 10.1.3.2 -> Destination: 10.1.1.1, Value: 8.203619909502262
Source: 10.1.3.1 -> Destination: 10.1.4.2, Value: 8.490825688073395
Source: 10.1.3.1 -> Destination: 10.1.3.2, Value: 6.433920704845815
Source: 10.1.3.1 -> Destination: 10.1.4.1, Value: 9.054455445544555
Source: 10.1.3.1 -> Destination: 10.1.5.1, Value: 8.46875
Source: 10.1.3.2 -> Destination: 10.1.2.1, Value: 7.962264150943396
```

PACKET DROPS

The data in the image depicts packet drops between different IP addresses. Most of the packet drops are between 10.1.1.1 to 10.1.1.2, 10.1.3.2, and 10.1.4.2, with Value = 3. Other connections have 1 or 0 packet drops.



scratch-simulator.cc network-simulation.cc

```
    // Create Point-to-Point Links with capacities and delays
    PointToPointHelper p2p;
        NetDeviceContainer devices[11];

        // A -> R1
        p2p.SetDeviceAttribute("DataRate", StringValue("500Kbps"));
        p2p.SetChannelAttribute("Delay", StringValue("2ms"));
        p2p.SetQueue("ns3::DropTailQueue", "MaxSize", StringValue("10p"));
        devices[0] = p2p.Install(workstations.Get(0), routers.Get(0));

        // Add packet loss on A->R1
        Ptr<RateErrorModel> errorModelA = CreateObject<RateErrorModel>();
        errorModelA->SetAttribute("ErrorRate", DoubleValue(0.01));
        devices[0].Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModelA));

        // B -> R2
        p2p.SetDeviceAttribute("DataRate", StringValue("500Kbps"));
        p2p.SetChannelAttribute("Delay", StringValue("3ms"));
        p2p.SetQueue("ns3::DropTailQueue", "MaxSize", StringValue("20p"));
        devices[1] = p2p.Install(workstations.Get(1), routers.Get(1));

        // Add packet loss on B->R2
        Ptr<RateErrorModel> errorModelB = CreateObject<RateErrorModel>();
        errorModelB->SetAttribute("ErrorRate", DoubleValue(0.05));
        devices[1].Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModelB));

        // C -> R3
        p2p.SetDeviceAttribute("DataRate", StringValue("500Kbps"));
        p2p.SetChannelAttribute("Delay", StringValue("4ms"));
        p2p.SetQueue("ns3::DropTailQueue", "MaxSize", StringValue("6p"));
        devices[2] = p2p.Install(workstations.Get(2), routers.Get(2));

        // Add packet loss on C->R3
        Ptr<RateErrorModel> errorModelC = CreateObject<RateErrorModel>();
        errorModelC->SetAttribute("ErrorRate", DoubleValue(0.03));
        devices[2].Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModelC));

        // D -> R4
        p2p.SetDeviceAttribute("DataRate", StringValue("500Kbps"));
        p2p.SetChannelAttribute("Delay", StringValue("5ms"));
        p2p.SetQueue("ns3::DropTailQueue", "MaxSize", StringValue("3p"));
        devices[3] = p2p.Install(workstations.Get(3), routers.Get(3));

        // Add packet loss on D->R4
        Ptr<RateErrorModel> errorModelD = CreateObject<RateErrorModel>();
        errorModelD->SetAttribute("ErrorRate", DoubleValue(0.02));
        devices[3].Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModelD));

        p2p.SetDeviceAttribute("DataRate", StringValue("500Kbps"));
        p2p.SetChannelAttribute("Delay", StringValue("6ms"));
        p2p.SetQueue("ns3::DropTailQueue", "MaxSize", StringValue("10p"));
        devices[4] = p2p.Install(workstations.Get(4), routers.Get(1));

        // Add packet loss on E->R5
        Ptr<RateErrorModel> errorModelE = CreateObject<RateErrorModel>();
        errorModelE->SetAttribute("ErrorRate", DoubleValue(0.04));
        devices[4].Get(1)->SetAttribute("ReceiveErrorModel", PointerValue(errorModelE));

        // F -> R6
        p2p.SetDeviceAttribute("DataRate", StringValue("500Kbps"));
        p2p.SetChannelAttribute("Delay", StringValue("7ms"));
        p2p.SetQueue("ns3::DropTailQueue", "MaxSize", StringValue("12p"));
        devices[5] = p2p.Install(workstations.Get(5), routers.Get(0));
        //..
        p2p.SetDeviceAttribute("DataRate", StringValue("1000Kbps"));
        p2p.SetChannelAttribute("Delay", StringValue("3ms"));
        p2p.SetQueue("ns3::DropTailQueue", "MaxSize", StringValue("200p"));
        devices[1] = p2p.Install(routers.Get(3), routers.Get(1));
```

Thank You!