Computer Networks Lab
Nilay Ganvit - 200001053
18th March 2023

ns-3 Assignment 1
End-to-End throughput:

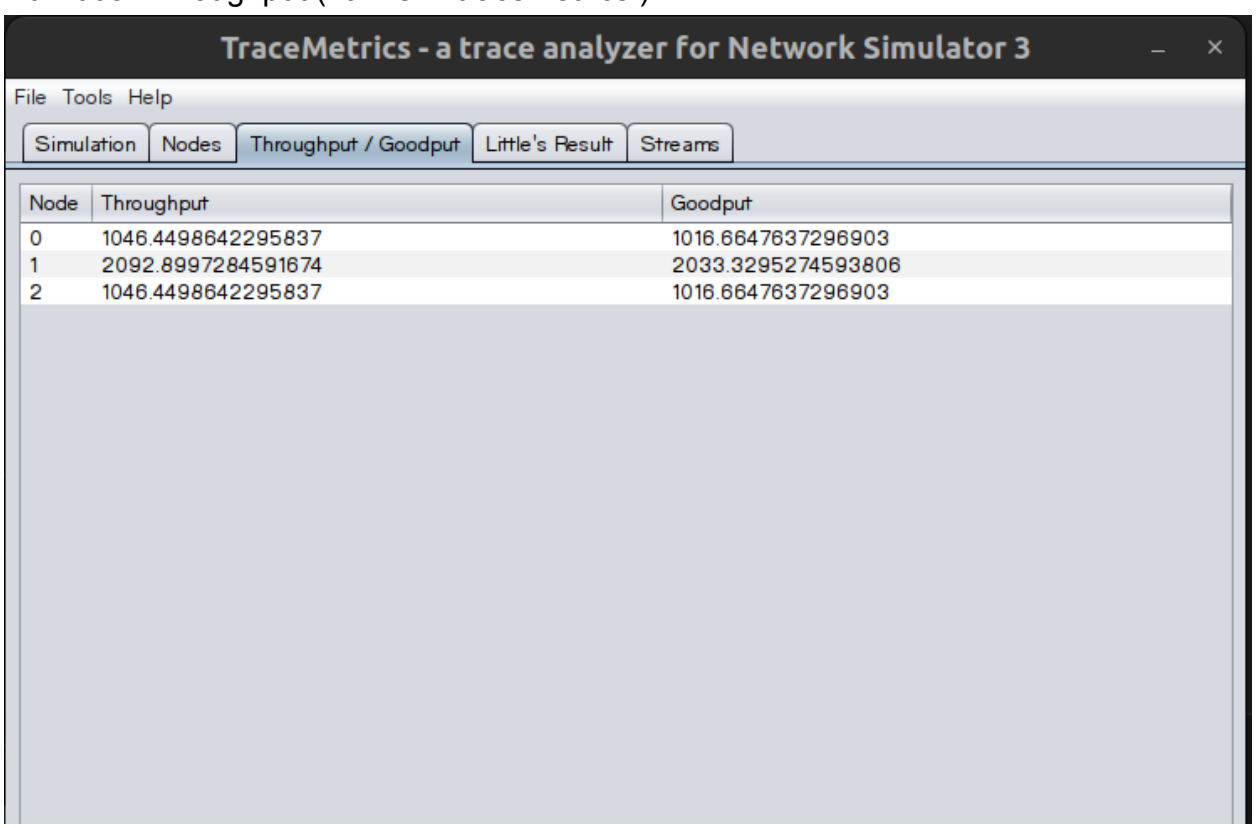
The maximum achievable end-end throughput is the capacity of the link with the minimum capacity, which is 5Mbps.
The End-to-End Delay is 2 seconds while the Total time taken including the delay for packets to be sent form the client to server and echo to return is 2.01443 seconds.
The Total data transmitted from client to the server is 2048 bytes which is then received and echo is transmitted from server to client of total size 2048 bytes which together makes up 4096 bytes.
Therefore the End-to-End throughput of one-way from client to server is

2048*8/2.01075 =8148.2034 bits/sec or 7.9572 Kbits/sec

And the End-to-End throughput of two-way communication is

4096*8/2.01812=16236.8937 bits/sec or 15.8563 Kbits/sec

Individual Throughput (.tr file in traceMetrics):



Node	Throughput	Goodput
0	1046.4498642295837	1016.6647637296903
1	2092.8997284591674	2033.3295274593806
2	1046.4498642295837	1016.6647637296903

FlowMonitor:

```
nil@y:~/ns-allinone-3.36.1/ns-3.36.1$ python3 flowmon-parse-results.py 200001053_ns3_EXP1.xml
Reading XML file  . done.
FlowID: 1 (UDP 10.1.1.1/49153 --> 10.1.2.2/9)
        TX bitrate: None
        RX bitrate: None
        Mean Delay: 6.37 ms
        Packet Loss Ratio: 0.00 %
FlowID: 2 (UDP 10.1.1.1/49154 --> 10.1.2.2/10)
        TX bitrate: None
        RX bitrate: None
        Mean Delay: 8.06 ms
        Packet Loss Ratio: 0.00 %
FlowID: 3 (UDP 10.1.2.2/9 --> 10.1.1.1/49153)
        TX bitrate: None
        RX bitrate: None
        Mean Delay: 6.37 ms
        Packet Loss Ratio: 0.00 %
FlowID: 4 (UDP 10.1.2.2/10 --> 10.1.1.1/49154)
        TX bitrate: None
        RX bitrate: None
        Mean Delay: 6.37 ms
        Packet Loss Ratio: 0.00 %
nil@y:~/ns-allinone-3.36.1/ns-3.36.1$
```

Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.1.1	10.1.2.2	UDP	1054	49153 → 9 Len=1024
2	0.001686	10.1.1.1	10.1.2.2	UDP	1054	49154 → 10 Len=1024
3	0.012745	10.1.2.2	10.1.1.1	UDP	1054	9 → 49153 Len=1024
4	0.014432	10.1.2.2	10.1.1.1	UDP	1054	10 → 49154 Len=1024

```
▸ Frame 1: 1054 bytes on wire (8432 bits), 1054 bytes captured (8432 bits)
▸ Point-to-Point Protocol
▸ Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.1.2.2
▸ User Datagram Protocol, Src Port: 49153, Dst Port: 9
▸ Data (1024 bytes)
```

```
0000  00 21 45 00 04 1c 00 00  00 00 40 11 00 00 0a 01   ·!E····· ··@·····
0010  01 01 0a 01 02 02 c0 01  00 09 04 08 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Code:

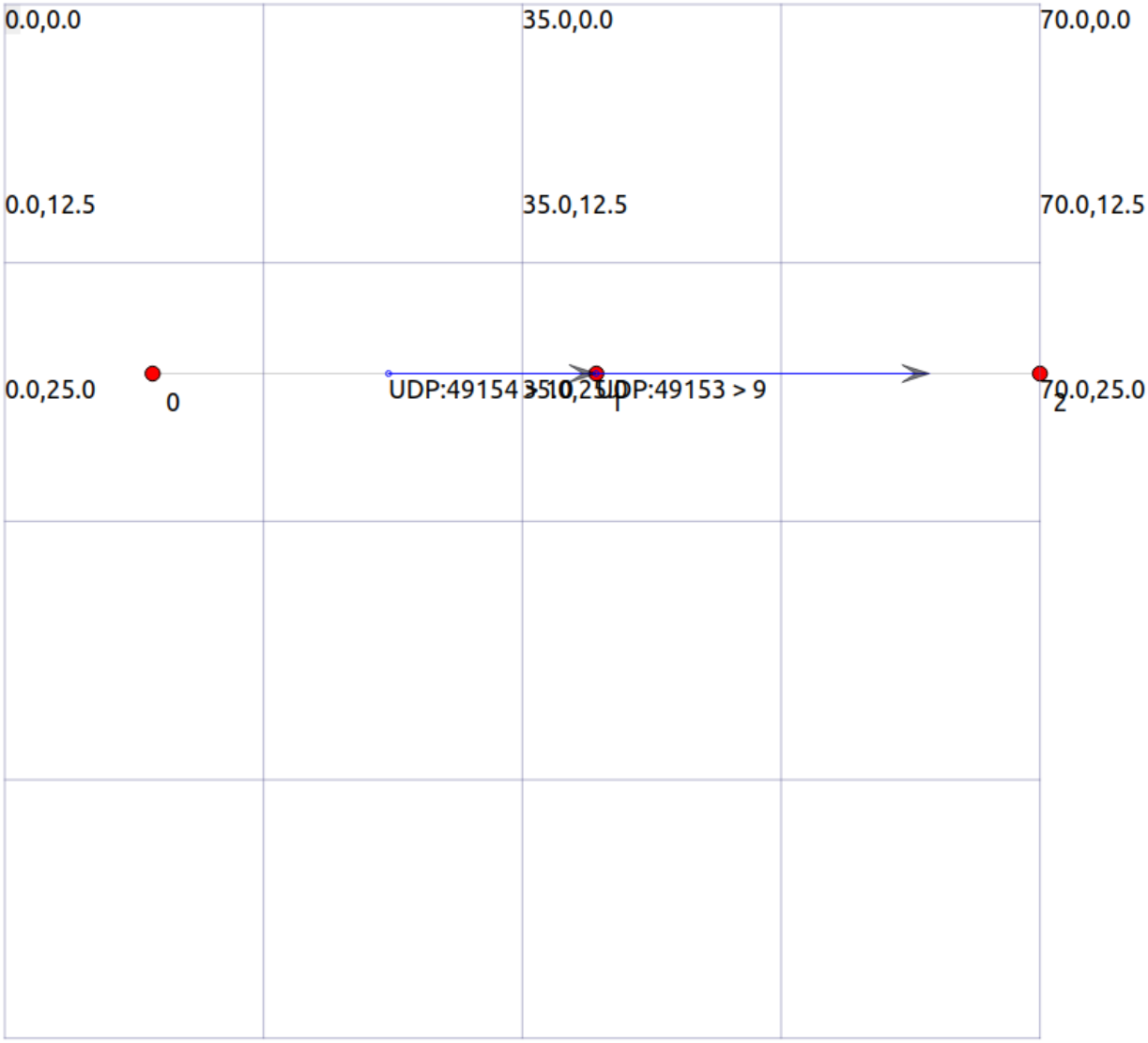
```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License version 2 as
* published by the Free Software Foundation;
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307
USA
*/

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"
#include "ns3/mobility-module.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/flow-monitor-helper.h"
#include "ns3/ipv4-flow-classifier.h"

// Default Network Topology
//
//       10.1.1.0          10.1.2.0
// n0 -------------- n1 -------------- n2
//    point-to-point    point-to-point
//

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("200001053_ns3_EXP1");
```

```
int
main (int argc, char *argv[])
{
 CommandLine cmd (__FILE__);
 cmd.Parse (argc, argv);

 Time::SetResolution (Time::NS);
 LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
 LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);

 NodeContainer nodes;
 nodes.Create (3);

 NodeContainer n0n1 = NodeContainer (nodes.Get (0), nodes.Get (1));
 NodeContainer n1n2 = NodeContainer (nodes.Get (1), nodes.Get (2));

 PointToPointHelper pointToPoint;
 pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
 // varying the latency of the link
 pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

 NetDeviceContainer devices = pointToPoint.Install (n0n1);

 PointToPointHelper pointToPoint1;
 pointToPoint1.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
 // varying the latency of the link
 pointToPoint1.SetChannelAttribute ("Delay", StringValue ("1ms"));

 NetDeviceContainer devices1 = pointToPoint1.Install (n1n2);

 InternetStackHelper stack;
 stack.Install (nodes);

 Ipv4AddressHelper address;
 address.SetBase ("10.1.1.0", "255.255.255.0");
 Ipv4InterfaceContainer interfaces = address.Assign (devices);

 address.SetBase ("10.1.2.0", "255.255.255.0");

 Ipv4InterfaceContainer interfaces1 = address.Assign (devices1);
```

```
UdpEchoServerHelper echoServer (9);

UdpEchoServerHelper echoServer1 (10);

ApplicationContainer serverApps = echoServer.Install (nodes.Get (2));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

ApplicationContainer serverApps1 = echoServer1.Install (nodes.Get (2));
serverApps1.Start (Seconds (1.0));
serverApps1.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (interfaces1.GetAddress (1), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

UdpEchoClientHelper echoClient1 (interfaces1.GetAddress (1), 10);
echoClient1.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient1.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient1.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

ApplicationContainer clientApps1 = echoClient1.Install (nodes.Get (0));
clientApps1.Start (Seconds (2.0));
clientApps1.Stop (Seconds (10.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

pointToPoint.EnablePcapAll ("200001053_ns3_EXP1");
pointToPoint1.EnablePcap ("200001053_ns3_EXP1", devices1.Get (1), true);
MobilityHelper mobility;
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (nodes);
AnimationInterface anim ("200001053_ns3_EXP1_anim.xml");
AnimationInterface::SetConstantPosition (nodes.Get (0), 10, 25);
```

```
 AnimationInterface::SetConstantPosition (nodes.Get (1), 40, 25);
 AnimationInterface::SetConstantPosition (nodes.Get (2), 70, 25);
 anim.EnablePacketMetadata (true);
 AsciiTraceHelper ascii;
 pointToPoint.EnableAsciiAll (ascii.CreateFileStream
("200001053_ns3_EXP1.tr"));
 Ptr<FlowMonitor> flowMonitor;
 FlowMonitorHelper flowHelper;
 flowMonitor = flowHelper.InstallAll ();

 // stopping application container
 double stop_time = 10.0;
 double cleanup_time = 1.0;
 clientApps.Stop (Seconds (stop_time));
 clientApps1.Stop (Seconds (stop_time));
 serverApps.Stop (Seconds (stop_time));
 serverApps1.Stop (Seconds (stop_time));
 Simulator::Stop (Seconds (stop_time + cleanup_time));
 Simulator::Run ();
 flowMonitor->SerializeToXmlFile ("200001053_ns3_EXP1.xml", true, true);
 Simulator::Destroy ();
 return 0;
}
```

NetAnim:


0.0,0.0
35.0,0.0
70.0,0.0
0.0,12.5
35.0,12.5
70.0,12.5
0.0,25.0
0
UDP:49154 > 10
UDP:49153 > 9
1
70.0,25.0
2