NETWORK SIMULATION LAB

WEEK - 09

DATE:

NAME:Vivekananda Shonti

ROLL NO: 19R21A05H2

PROBLEM STATEMENT: 09

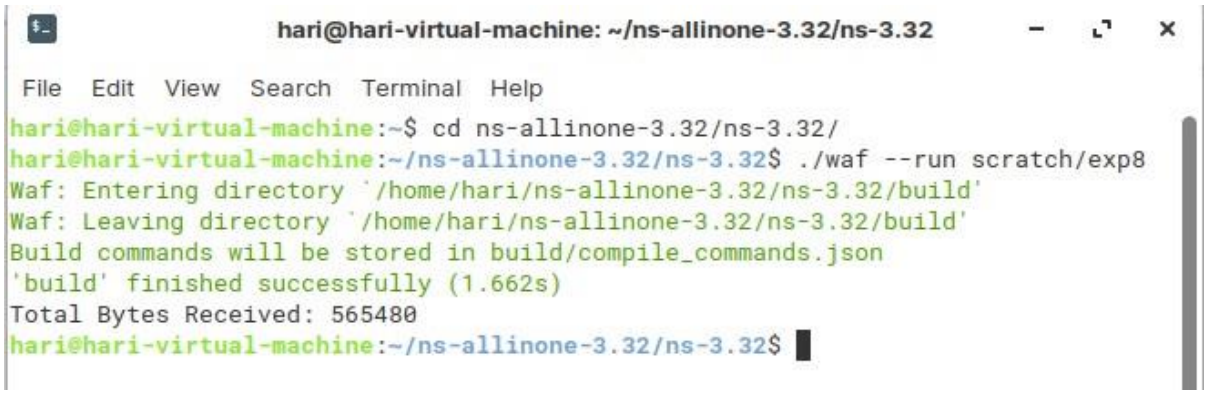
Program in NS3 to implement FTP using TCP bulk transfer.

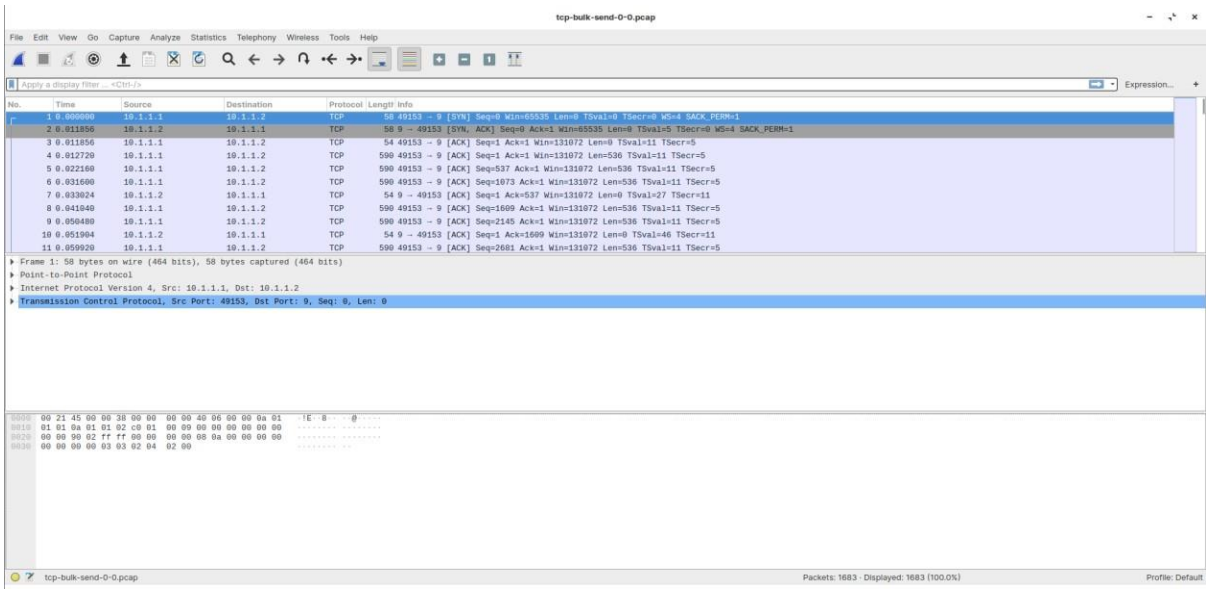
```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License version 2 as
* published by the Free Software Foundation;
*
* This program is distributed in the hope 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
*/
// Network topology
//
// n0 ------------ n1
// 500 Kbps
// 5 ms
//
// - Flow from n0 to n1 using BulkSendApplication.
// - Tracing of queues and packet receptions to file "tcp-bulk-send.tr"
// and pcap tracing available when tracing is turned on.
#include <string>
#include <fstream>
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/internet-module.h"
#include "ns3/applications-module.h"
#include "ns3/network-module.h"
#include "ns3/packet-sink.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("TcpBulkSendExample");
int
main (int argc, char *argv[])
{
//make boolean value of tracing as true
bool tracing = true;
uint32_t maxBytes = 0;
//
// Allow the user to override any of the defaults at
// run-time, via command-line arguments
//
```

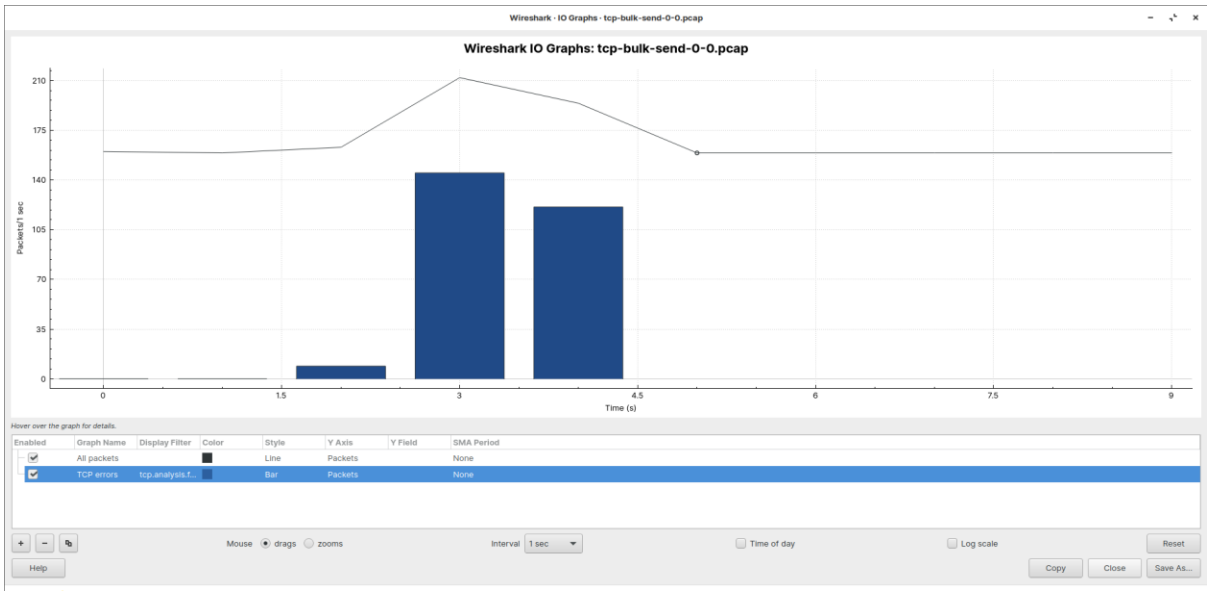
```
CommandLine cmd (__FILE__);
cmd.AddValue ("tracing", "Flag to enable/disable tracing", tracing);
cmd.AddValue ("maxBytes",
"Total number of bytes for application to send", maxBytes);
cmd.Parse (argc, argv);
//
// Explicitly create the nodes required by the topology (shown above).
//
NS_LOG_INFO ("Create nodes.");
NodeContainer nodes;
nodes.Create (2);
NS_LOG_INFO ("Create channels.");
//
// Explicitly create the point-to-point link required by the topology (shown above).
//
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("500Kbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("5ms"));
NetDeviceContainer devices;
devices = pointToPoint.Install (nodes);
//
// Install the internet stack on the nodes
//
InternetStackHelper internet;
internet.Install (nodes);
//
// We've got the "hardware" in place. Now we need to add IP addresses.
//
NS_LOG_INFO ("Assign IP Addresses.");
Ipv4AddressHelper ipv4;
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign (devices);
NS_LOG_INFO ("Create Applications.");
//
// Create a BulkSendApplication and install it on node 0
//
uint16_t port = 9; // well-known echo port number
BulkSendHelper source ("ns3::TcpSocketFactory",
InetSocketAddress (i.GetAddress (1), port));
// Set the amount of data to send in bytes. Zero is unlimited.
source.SetAttribute ("MaxBytes", UintegerValue (maxBytes));
ApplicationContainer sourceApps = source.Install (nodes.Get (0));
sourceApps.Start (Seconds (0.0));
sourceApps.Stop (Seconds (10.0));
//
// Create a PacketSinkApplication and install it on node 1
//
PacketSinkHelper sink ("ns3::TcpSocketFactory",
InetSocketAddress (Ipv4Address::GetAny (), port));
ApplicationContainer sinkApps = sink.Install (nodes.Get (1));
sinkApps.Start (Seconds (0.0));
sinkApps.Stop (Seconds (10.0));
//
// Set up tracing if enabled
//Set up tracing boolean value as true
if (tracing)
{
AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll (ascii.CreateFileStream ("tcp-bulk-send.tr"));
pointToPoint.EnablePcapAll ("tcp-bulk-send", true);
```

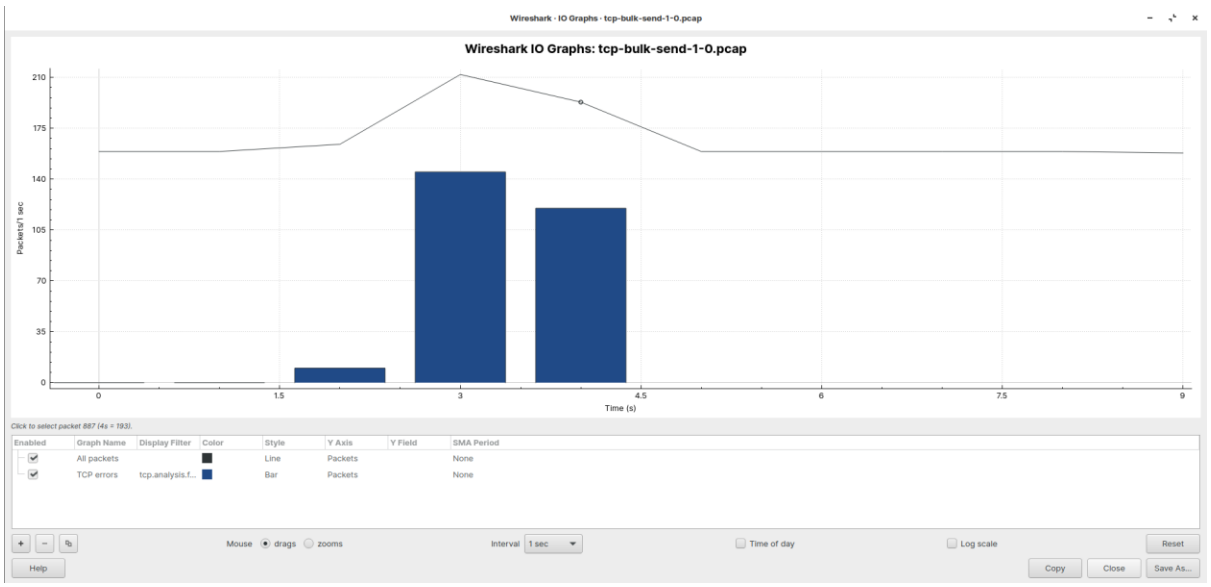
```
}
//
// Now, do the actual simulation.
//
NS_LOG_INFO ("Run Simulation.");
Simulator::Stop (Seconds (10.0));
Simulator::Run  ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
Ptr<PacketSink> sink1 = DynamicCast<PacketSink> (sinkApps.Get (0));
std::cout << "Total Bytes Received: " << sink1->GetTotalRx () << std::endl;
}
```

OUTPUT:






Wireshark · IO Graphs · tcp-bulk-send-0-0.pcap
Wireshark IO Graphs: tcp-bulk-send-0-0.pcap
Packets/1 sec
Time (s)
Hover over the graph for details.
Enabled
Graph Name
Display Filter
Color
Style
Y Axis
Y Field
SMA Period
All packets
Line
Packets
None
TCP errors
tcp.analysis.f...
Bar
Packets
None
Mouse
drags
zooms
Interval
1 sec
Time of day
Log scale
Reset
Help
Copy
Close
Save As...


Wireshark · IO Graphs · tcp-bulk-send-1-0.pcap
Wireshark IO Graphs: tcp-bulk-send-1-0.pcap
Packets/1 sec
Time (s)
Click to select packet 887 (4s = 193).
Enabled
Graph Name
Display Filter
Color
Style
Y Axis
Y Field
SMA Period
All packets
Line
Packets
None
TCP errors
tcp.analysis.f...
Bar
Packets
None
Mouse
drags
zooms
Interval
1 sec
Time of day
Log scale
Reset
Help
Copy
Close
Save As...