**Program to simulate FTP using TCP**

Code :

#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[])

{

bool tracing = false;

uint32\_t maxBytes = 0;

//

// Allow the user to override any of the defaults at

// run-time, via command-line arguments

//

CommandLine cmd;

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

//

if (tracing)

{

AsciiTraceHelper ascii;

pointToPoint.EnableAsciiAll (ascii.CreateFileStream ("tcp-bulk-send.tr"));

pointToPoint.EnablePcapAll ("tcp-bulk-send", false);

}

//

// 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;

return 0;

}

OUTPUT/SCREENSHOT :

