Aim : Write a program for the implementation of ftp using tcp bulk transfer.

Description : In this program, we implement FTP using TCP bulk transfer .There is no single way to exchange bulk data using TCP. It is a dynamic process that depends on many factors, some of which we can control (e.g., send and receive buffer sizes) and some of which we have no control over (e.g., network congestion, implementation features). Fundamental to the efficient transfer of bulk data is TCP's sliding window protocol.

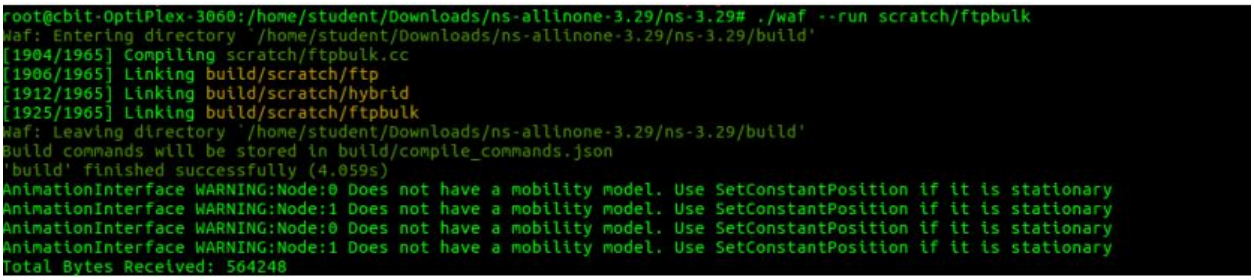
Code :

```
#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" #include
 "ns3/netanim-module.h"

 using namespace
 ns3;

 NS_LOG_COMPONENT_DEFINE
 ("TcpBulkSendExample");

 int main (int argc, char *argv[]) {

   bool tracing = false;
   uint32_t maxBytes =
   0;

   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);
```

```
NS_LOG_INFO ("Create nodes.");
NodeContainer nodes;
nodes.Create (2);

NS_LOG_INFO ("Create
channels.");

PointToPointHelper
pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue
("500Kbps")); pointToPoint.SetChannelAttribute ("Delay", StringValue
("5ms"));

NetDeviceContainer devices; devices
= pointToPoint.Install (nodes);


InternetStackHelper
internet; internet.Install
(nodes);


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.");


uint16_t port = 9; // well-known echo port number


BulkSendHelper source
("ns3::TcpSocketFactory",
                InetSocketAddress (i.GetAddress (1),
```

```
                port));

  source.SetAttribute ("MaxBytes", UintegerValue (maxBytes));
ApplicationContainer sourceApps = source.Install (nodes.Get
(0)); sourceApps.Start (Seconds (0.0)); sourceApps.Stop (Seconds
(10.0));


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));


if
(tracing)
  {
  AsciiTraceHelper ascii; pointToPoint.EnableAsciiAll
  (ascii.CreateFileStream ("tcp-bulk-send.tr")); pointToPoint.EnablePcapAll
  ("tcp-bulk-send", false); }

NS_LOG_INFO ("Run
Simulation."); Simulator::Stop
(Seconds (10.0));
AnimationInterface
anim("bulktcp.xml"); 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:



Conclusion :

from the above program we come to know about how to implement ftp using tcp bulk transfer.