Practical No: 11

Aim: - Program to simulate FTP using TCP Protocal

Line of Code: -

Tcpfile.cc

```
/* -*- 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
//
// 10Mb/s, 10ms 10Mb/s, 10ms
// n0-----------------n1-----------------n2
//
//
// - Tracing of queues and packet receptions to file
// "tcp-large-transfer.tr"
// - pcap traces also generated in the following files
// "tcp-large-transfer-$n-$i.pcap" where n and i represent node and
interface
// numbers respectively
// Usage (e.g.): ./waf --run tcp-large-transfer
#include <iostream>
#include <fstream>
#include <string>
#include "ns3/core-module.h"
#include "ns3/applications-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/netanim-module.h"

using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("TcpLargeTransfer");
// The number of bytes to send in this simulation.
static const uint32_t totalTxBytes = 2000000;
static uint32_t currentTxBytes = 0;
// Perform series of 1040 byte writes (this is a multiple of 26 since
// we want to detect data splicing in the output stream)
```

```
static const uint32_t writeSize = 1040;
uint8_t data[writeSize];
std::string animFile = "ftp-animation.xml";
// These are for starting the writing process, and handling the sending
// socket's notification upcalls (events). These two together more or less
// implement a sending "Application", although not a proper
ns3::Application
// subclass.
void StartFlow (Ptr<Socket>, Ipv4Address, uint16_t);
void WriteUntilBufferFull (Ptr<Socket>, uint32_t);
static void
CwndTracer (uint32_t oldval, uint32_t newval)
{
NS_LOG_INFO ("Moving cwnd from " << oldval << " to " << newval);
}
int main (int argc, char *argv[])
{
// Users may find it convenient to turn on explicit debugging
// for selected modules; the below lines suggest how to do this
// LogComponentEnable("TcpL4Protocol", LOG_LEVEL_ALL);
// LogComponentEnable("TcpSocketImpl", LOG_LEVEL_ALL);
// LogComponentEnable("PacketSink", LOG_LEVEL_ALL);
// LogComponentEnable("TcpLargeTransfer", LOG_LEVEL_ALL);
CommandLine cmd (__FILE__);
cmd.Parse (argc, argv);
// initialize the tx buffer.
for(uint32_t i = 0; i < writeSize; ++i)
{
char m = toascii (97 + i % 26);
data[i] = m;
}
// Here, we will explicitly create three nodes. The first container
contains
// nodes 0 and 1 from the diagram above, and the second one contains nodes
// 1 and 2. This reflects the channel connectivity, and will be used to
// install the network interfaces and connect them with a channel.
NodeContainer n0n1;
n0n1.Create (2);
NodeContainer n1n2;
n1n2.Add (n0n1.Get (1));
n1n2.Create (1);
// We create the channels first without any IP addressing information
// First make and configure the helper, so that it will put the appropriate
// attributes on the network interfaces and channels we are about to
install.
PointToPointHelper p2p;
p2p.SetDeviceAttribute ("DataRate", DataRateValue (DataRate (10000000)));
p2p.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (10)));
// And then install devices and channels connecting our topology.
NetDeviceContainer dev0 = p2p.Install (n0n1);
NetDeviceContainer dev1 = p2p.Install (n1n2);
// Now add ip/tcp stack to all nodes.
InternetStackHelper internet;
internet.InstallAll ();
// Later, we add IP addresses.
Ipv4AddressHelper ipv4;
```

```
ipv4.SetBase ("10.1.3.0", "255.255.255.0");
ipv4.Assign (dev0);
ipv4.SetBase ("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer ipInterfs = ipv4.Assign (dev1);
// and setup ip routing tables to get total ip-level connectivity.
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
////////////////////////////////////////////////////////////////////
// Simulation 1
//
// Send 2000000 bytes over a connection to server port 50000 at time 0
// Should observe SYN exchange, a lot of data segments and ACKS, and FIN
// exchange. FIN exchange isn't quite compliant with TCP spec (see release
// notes for more info)
//
////////////////////////////////////////////////////////////////////
uint16_t servPort = 50000;
// Create a packet sink to receive these packets on n2...
PacketSinkHelper sink ("ns3::TcpSocketFactory",
InetSocketAddress (Ipv4Address::GetAny (), servPort));
ApplicationContainer apps = sink.Install (n1n2.Get (1));
apps.Start (Seconds (0.0));
apps.Stop (Seconds (3.0));
// Create a source to send packets from n0. Instead of a full Application
// and the helper APIs you might see in other example files, this example
// will use sockets directly and register some socket callbacks as a
sending
// "Application".
// Create and bind the socket...
Ptr<Socket> localSocket =
Socket::CreateSocket (n0n1.Get (0), TcpSocketFactory::GetTypeId ());
localSocket->Bind ();
// Trace changes to the congestion window
Config::ConnectWithoutContext
("/NodeList/0/$ns3::TcpL4Protocol/SocketList/0/CongestionWindow",
MakeCallback (&CwndTracer));
// ...and schedule the sending "Application"; This is similar to what an
// ns3::Application subclass would do internally.
Simulator::ScheduleNow (&StartFlow, localSocket,
ipInterfs.GetAddress (1), servPort);
// One can toggle the comment for the following line on or off to see the
// effects of finite send buffer modelling. One can also change the size of
// said buffer.
//localSocket->SetAttribute("SndBufSize", UintegerValue(4096));
//Ask for ASCII and pcap traces of network traffic
AsciiTraceHelper ascii;
p2p.EnableAsciiAll (ascii.CreateFileStream ("tcp-large-transfer.tr"));
p2p.EnablePcapAll ("tcp-large-transfer");
// Create the animation object and configure for specified output
AnimationInterface anim ("tcpfile.xml");
// Finally, set up the simulator to run. The 1000 second hard limit is a
// failsafe in case some change above causes the simulation to never end
Simulator::Stop (Seconds (1000));
Simulator::Run ();
Simulator::Destroy ();
}
//begin implementation of sending "Application"
```

```
void StartFlow (Ptr<Socket> localSocket,
Ipv4Address servAddress,
uint16_t servPort)
{
NS_LOG_LOGIC ("Starting flow at time " << Simulator::Now ().GetSeconds ());
localSocket->Connect (InetSocketAddress (servAddress, servPort)); //connect
// tell the tcp implementation to call WriteUntilBufferFull again
// if we blocked and new tx buffer space becomes available
localSocket->SetSendCallback (MakeCallback (&WriteUntilBufferFull));
WriteUntilBufferFull (localSocket, localSocket->GetTxAvailable ());
}
void WriteUntilBufferFull (Ptr<Socket> localSocket, uint32_t txSpace)
{
while (currentTxBytes < totalTxBytes && localSocket->GetTxAvailable () > 0)
{
uint32_t left = totalTxBytes - currentTxBytes;
uint32_t dataOffset = currentTxBytes % writeSize;
uint32_t toWrite = writeSize - dataOffset;
toWrite = std::min (toWrite, left);
toWrite = std::min (toWrite, localSocket->GetTxAvailable ());
int amountSent = localSocket->Send (&data[dataOffset], toWrite, 0);
if(amountSent < 0)
{
// we will be called again when new tx space becomes available.
return;
}
currentTxBytes += amountSent;
}
if (currentTxBytes >= totalTxBytes)
{
localSocket->Close ();
}
}
```

Program: -

```
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ make clean
rm -f qrc_resources.cpp qrc_qtpropertybrowser.cpp
rm -f moc_predefs.h
rm -f moc_animatorscene.cpp moc_animpacket.cpp moc_netanim.cpp moc_animatormode.cpp moc_statsmo
ertybrowser.cpp moc_qtpropertymanager.cpp moc_qtpropertybrowserutils_p.cpp moc_qtpropertybrowse
ditorfactory.cpp moc_qtbuttonpropertybrowser.cpp moc_animpropertybrowser.cpp moc_filepathmanage
moc_packetsmode.cpp moc_table.cpp moc_qcustomplot.cpp
rm -f qttreepropertybrowser.moc qtpropertymanager.moc qteditorfactory.moc
rm -f main.o log.o fatal-error.o fatal-impl.o logqt.o resizeableitem.o animnode.o animatorscene
 animxmlparser.o animatorview.o animlink.o animresource.o statsview.o statsmode.o routingxmlpar
 flowmonxmlparser.o flowmonstatsscene.o textbubble.o qtvariantproperty.o qttreepropertybrowser.
qtpropertybrowser.o qtgroupboxpropertybrowser.o qteditorfactory.o qtbuttonpropertybrowser.o ani
ctory.o fileedit.o packetsmode.o packetsview.o packetsscene.o graphpacket.o table.o countertabl
ropertybrowser.o moc_animatorscene.o moc_animpacket.o moc_netanim.o moc_animatormode.o moc_stat
propertybrowser.o moc_filepathmanager.o moc_fileeditfactory.o moc_fileedit.o moc_packetsmode.o
rm -f *~ core *.core
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ qmake NetAnim.pro
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ make
```

```
    NS_LOG_INFO ("Run Simulation.");
    AnimationInterface anim("assign.xml");
     Simulator::Run ();
    Simulator::Destroy ();
    NS_LOG_INFO ("Done.");
  }


    #include <fstream>
    #include "ns3/core-module.h"
    #include "ns3/csma-module.h"
    #include "ns3/applications-module.h"
    #include "ns3/internet-module.h"
    #include "ns3/network-module.h"
    #include "ns3/netanim-module.h"
```

```
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/ns-3.33$ cd ..
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33$ cd netanim-3.108/
```

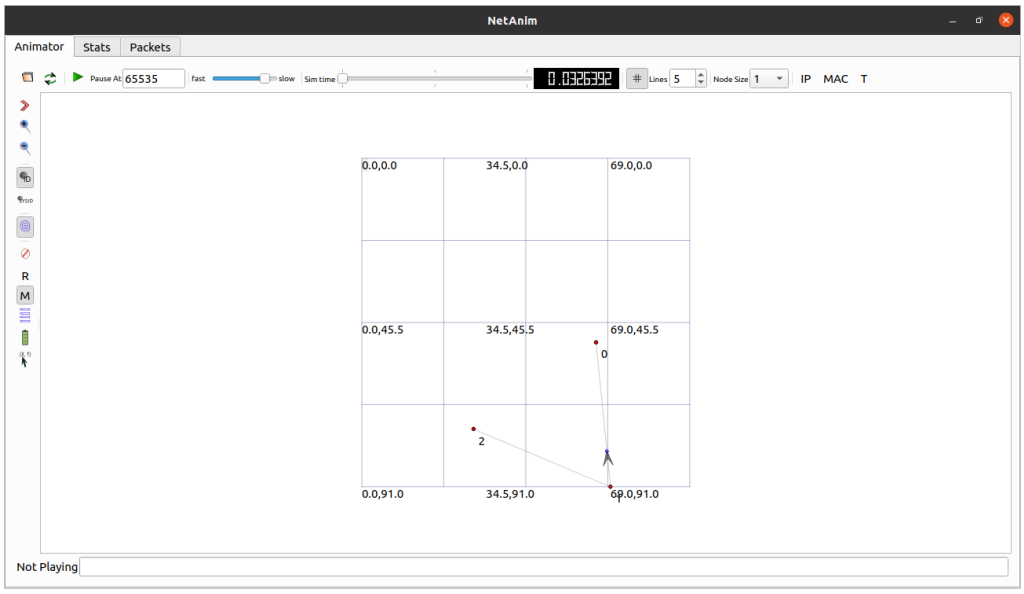
```
mca@mca-To-be-filled-by-O-E-M:~$ cd workspace/
mca@mca-To-be-filled-by-O-E-M:~/workspace$ cd ns-allinone-3.33/
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33$ cd ns-3.33/
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/ns-3.33$ ./waf --run assign
Waf: Entering directory `/home/mca/workspace/ns-allinone-3.33/ns-3.33/build'
Waf: Leaving directory `/home/mca/workspace/ns-allinone-3.33/ns-3.33/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.182s)
```

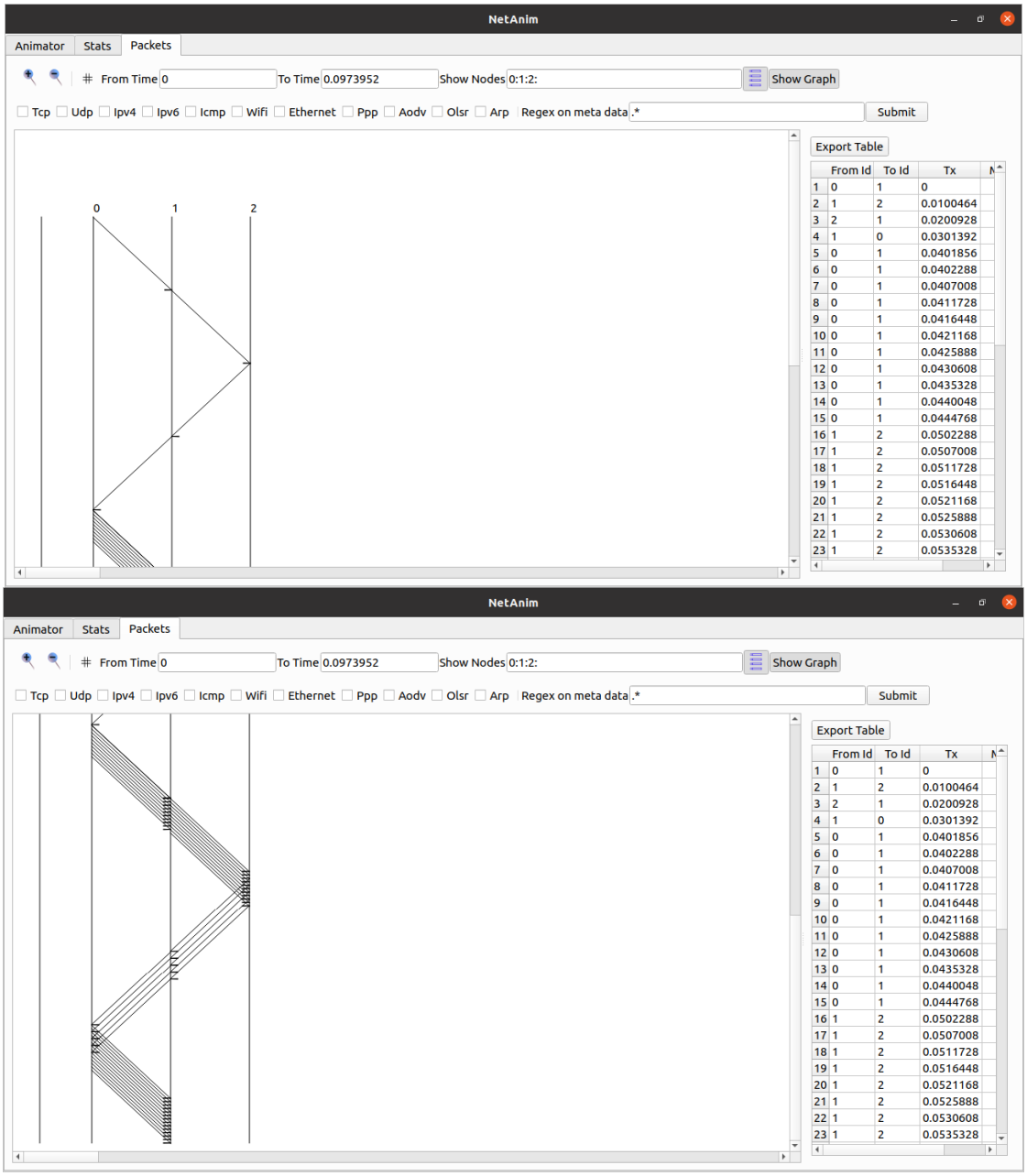
```
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/ns-3.33$ cd ..
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33$ cd netanim-3.108/
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ ./NetAnim
```

```
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ cd ..
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33$ cd ns-3.33/
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/ns-3.33$ ./waf --run assign
Waf: Entering directory `/home/mca/workspace/ns-allinone-3.33/ns-3.33/build'
Waf: Leaving directory `/home/mca/workspace/ns-allinone-3.33/ns-3.33/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.053s)
```

OUTPUT: -

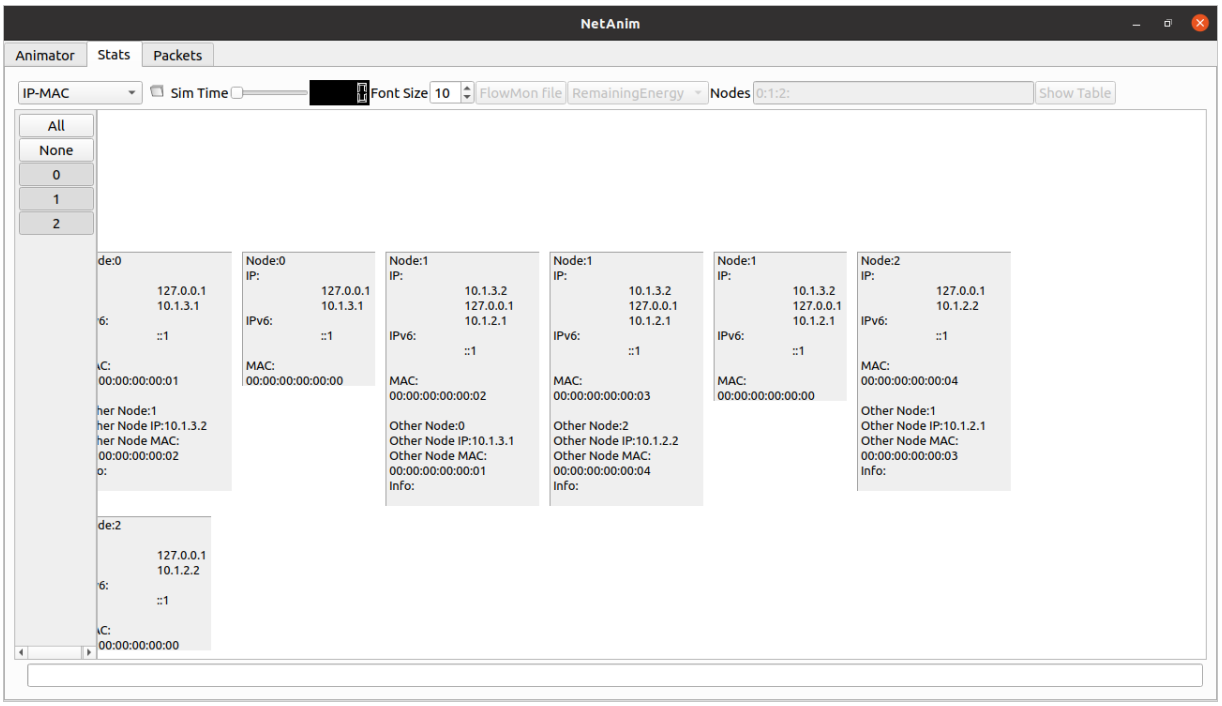
Name:Mansingh Yadav **Roll No. 61**


NetAnim
Animator
Stats
Packets
Pause At
65535
fast
slow
Sim time
Lines
5
Node Size
1
IP
MAC
T
0.0,0.0
34.5,0.0
69.0,0.0
0.0,45.5
34.5,45.5
69.0,45.5
0
2
0.0,91.0
34.5,91.0
69.0,91.0
Not Playing


NetAnim
Animator
Stats
Packets
From Time 0
To Time 0.0973952
Show Nodes 0:1:2:
Show Graph
Tcp Udp Ipv4 Ipv6 Icmp Wifi Ethernet Ppp Aodv Olsr Arp
Regex on meta data .*
Submit
Export Table
0
1
2
From Id To Id Tx
1 0 1 0
2 1 2 0.0100464
3 2 1 0.0200928
4 1 0 0.0301392
5 0 1 0.0401856
6 0 1 0.0402288
7 0 1 0.0407008
8 0 1 0.0411728
9 0 1 0.0416448
10 0 1 0.0421168
11 0 1 0.0425888
12 0 1 0.0430608
13 0 1 0.0435328
14 0 1 0.0440048
15 0 1 0.0444768
16 1 2 0.0502288
17 1 2 0.0507008
18 1 2 0.0511728
19 1 2 0.0516448
20 1 2 0.0521168
21 1 2 0.0525888
22 1 2 0.0530608
23 1 2 0.0535328

NetAnim
Animator
Stats
Packets
From Time 0
To Time 0.0973952
Show Nodes 0:1:2:
Show Graph
Tcp Udp Ipv4 Ipv6 Icmp Wifi Ethernet Ppp Aodv Olsr Arp
Regex on meta data .*
Submit
Export Table
From Id To Id Tx
1 0 1 0
2 1 2 0.0100464
3 2 1 0.0200928
4 1 0 0.0301392
5 0 1 0.0401856
6 0 1 0.0402288
7 0 1 0.0407008
8 0 1 0.0411728
9 0 1 0.0416448
10 0 1 0.0421168
11 0 1 0.0425888
12 0 1 0.0430608
13 0 1 0.0435328
14 0 1 0.0440048
15 0 1 0.0444768
16 1 2 0.0502288
17 1 2 0.0507008
18 1 2 0.0511728
19 1 2 0.0516448
20 1 2 0.0521168
21 1 2 0.0525888
22 1 2 0.0530608
23 1 2 0.0535328

```
mca@mca-To-be-filled-by-O-E-M: ~/Workspace/ns-allinone-3.33/netanim-3.108
mca@mca-To-be-filled-by-O-E-M:~/Workspace/ns-allinone-3.33/ns-3.33$ ./waf --run tcpfile
Waf: Entering directory `/home/mca/Workspace/ns-allinone-3.33/ns-3.33/build'
Waf: Leaving directory `/home/mca/Workspace/ns-allinone-3.33/ns-3.33/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.517s)
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
mca@mca-To-be-filled-by-O-E-M:~/Workspace/ns-allinone-3.33/ns-3.33$ cd ..
mca@mca-To-be-filled-by-O-E-M:~/Workspace/ns-allinone-3.33$ cd netanim-3.108/
mca@mca-To-be-filled-by-O-E-M:~/Workspace/ns-allinone-3.33/netanim-3.108$ ./NetAnim
```



Practical No: 12

Aim: -Program to assign IPv4 Addresses in NS3

Line of Code:-

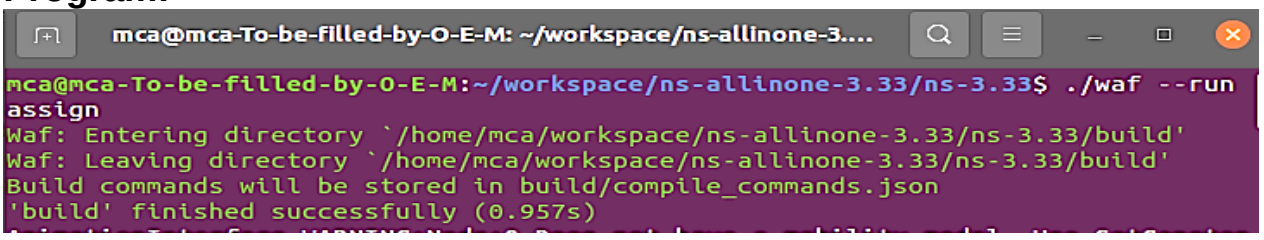
Assign.cc

```
/* -*- 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
   //       |               |
   //       =================
    //              LAN
   //
   // - UDP flows from n0 to n1
    #include <fstream>
    #include "ns3/core-module.h"
    #include "ns3/csma-module.h"
    #include "ns3/applications-module.h"
    #include "ns3/internet-module.h"
    #include "ns3/network-module.h"
    #include "ns3/netanim-module.h"
      using namespace ns3;
       NS_LOG_COMPONENT_DEFINE ("SocketOptionsIpv4");
       void ReceivePacket (Ptr<Socket> socket)
    {
     NS_LOG_INFO ("Received one packet!");
    Ptr<Packet> packet = socket->Recv ();
     SocketIpTosTag tosTag;
      if (packet->RemovePacketTag (tosTag))
       {
         NS_LOG_INFO (" TOS = " << (uint32_t)tosTag.GetTos ());
       }
     SocketIpTtlTag ttlTag;
     if (packet->RemovePacketTag (ttlTag))
       {
         NS_LOG_INFO (" TTL = " << (uint32_t)ttlTag.GetTtl ());
       }
```

```
   }
     static void SendPacket (Ptr<Socket> socket, uint32_t pktSize,
                             uint32_t pktCount, Time pktInterval )
    {
     if (pktCount > 0)
        {
          socket->Send (Create<Packet> (pktSize));
          Simulator::Schedule (pktInterval, &SendPacket,
                               socket, pktSize,pktCount - 1, pktInterval);
        }
     else
       {
         socket->Close ();
       }
   }
      int
   main (int argc, char *argv[])
   {
    // Allow the user to override any of the defaults and the above Bind()
at
   // run-time, via command-line arguments
    uint32_t packetSize = 1024;
     uint32_t packetCount = 10;
     double packetInterval = 1.0;
         //Socket options for IPv4, currently TOS, TTL, RECVTOS, and
RECVTTL
     uint32_t ipTos = 0;
     bool ipRecvTos = true;
     uint32_t ipTtl = 0;
     bool ipRecvTtl = true;

     CommandLine cmd;
     cmd.AddValue ("PacketSize", "Packet size in bytes", packetSize);
     cmd.AddValue ("PacketCount", "Number of packets to send",
packetCount);
     cmd.AddValue ("Interval", "Interval between packets", packetInterval);
     cmd.AddValue ("IP_TOS", "IP_TOS", ipTos);
      cmd.AddValue ("IP_RECVTOS", "IP_RECVTOS", ipRecvTos);
     cmd.AddValue ("IP_TTL", "IP_TTL", ipTtl);
     cmd.AddValue ("IP_RECVTTL", "IP_RECVTTL", ipRecvTtl);
    cmd.Parse (argc, argv);

     NS_LOG_INFO ("Create nodes.");
     NodeContainer n;
     n.Create (2);
     InternetStackHelper internet;
    internet.Install (n);
    Address serverAddress;
      NS_LOG_INFO ("Create channels.");
    CsmaHelper csma;
    csma.SetChannelAttribute ("DataRate", DataRateValue (DataRate
(5000000)));
    csma.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (2)));
   csma.SetDeviceAttribute ("Mtu", UintegerValue (1400));
   NetDeviceContainer d = csma.Install (n);
       NS_LOG_INFO ("Assign IP Addresses.");
```

```
    Ipv4AddressHelper ipv4;
   ipv4.SetBase ("10.1.1.0", "255.255.255.0");
     Ipv4InterfaceContainer i = ipv4.Assign (d);
    serverAddress = Address(i.GetAddress (1));
   NS_LOG_INFO ("Create sockets.");
    //Receiver socket on n1
    TypeId tid = TypeId::LookupByName ("ns3::UdpSocketFactory");
    Ptr<Socket> recvSink = Socket::CreateSocket (n.Get (1), tid);
    InetSocketAddress local = InetSocketAddress (Ipv4Address::GetAny (),
4477);
     recvSink->SetIpRecvTos (ipRecvTos);
    recvSink->SetIpRecvTtl (ipRecvTtl);
    recvSink->Bind (local);
    recvSink->SetRecvCallback (MakeCallback (&ReceivePacket));

     //Sender socket on n0
     Ptr<Socket> source = Socket::CreateSocket (n.Get (0), tid);
     InetSocketAddress remote = InetSocketAddress (i.GetAddress (1), 4477);
       //Set socket options, it is also possible to set the options after
the socket has been created/connected.
     if (ipTos > 0)
       {
       source->SetIpTos (ipTos);
     }
    if (ipTtl > 0)
     {
         source->SetIpTtl (ipTtl);
      }
    source->Connect (remote);
      AsciiTraceHelper ascii;
    csma.EnableAsciiAll (ascii.CreateFileStream ("socket-options-
ipv4.tr"));
    csma.EnablePcapAll ("socket-options-ipv4", false);

   //Schedule SendPacket
    Time interPacketInterval = Seconds (packetInterval);
   Simulator::ScheduleWithContext (source->GetNode ()->GetId (),
                                   Seconds (1.0), &SendPacket,
                                    source, packetSize, packetCount,
interPacketInterval);
      NS_LOG_INFO ("Run Simulation.");
    AnimationInterface anim("assign.xml");
     Simulator::Run ();
    Simulator::Destroy ();
    NS_LOG_INFO ("Done.");
  }
```

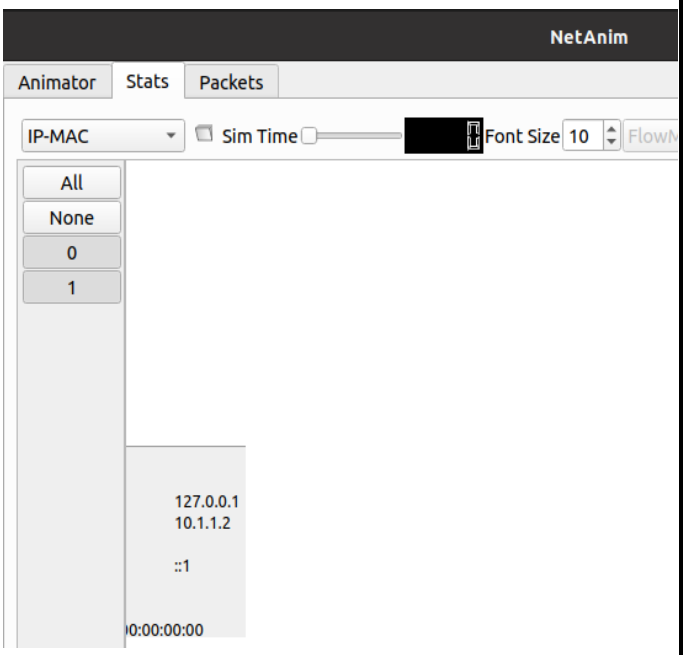
Program: -

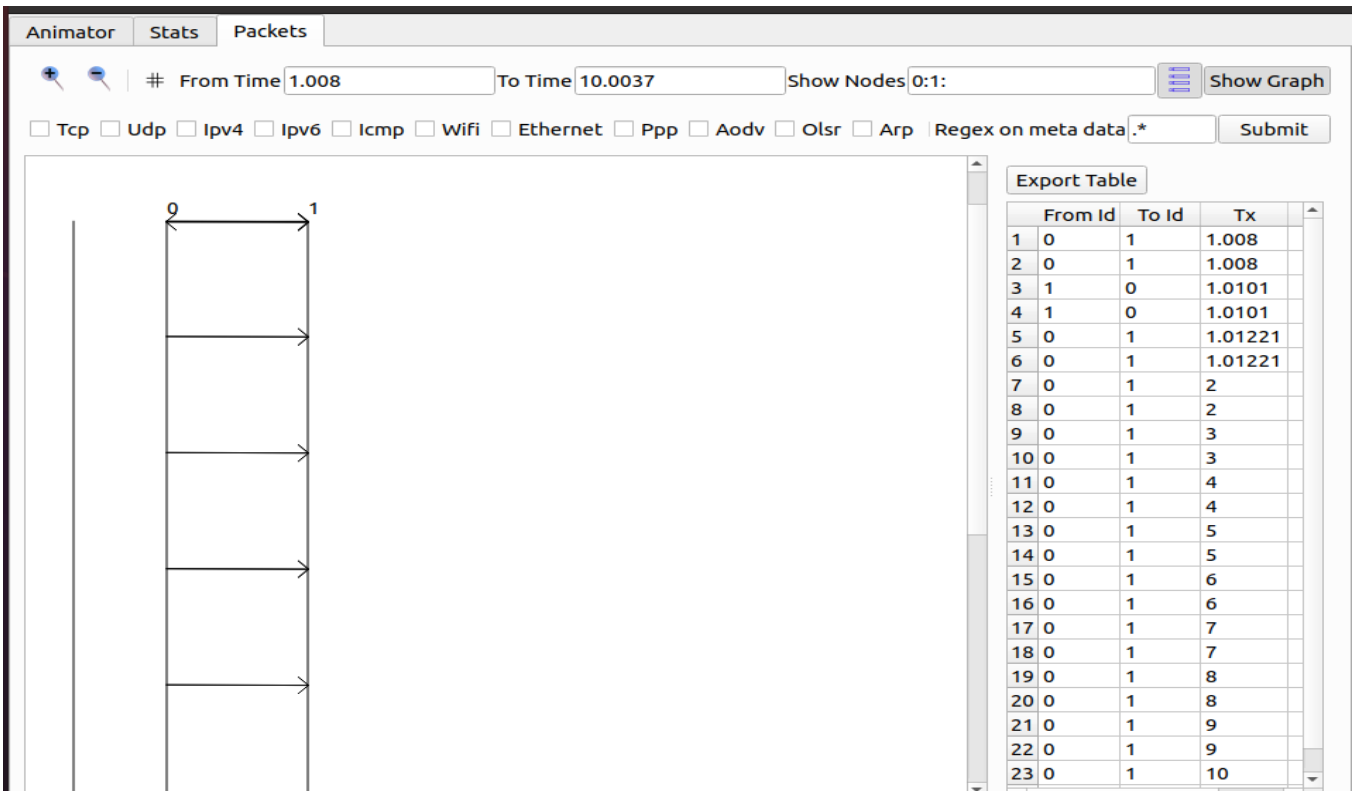


```
thmanager.o moc_fileeditfactory.o moc_fileedit.o moc_packetsmode.o moc_table.o m
oc_qcustomplot.o
rm -f *~ core *.core
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ qmake
NetAnim.pro
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ make
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -fPIC -DNS3_LOG_ENABLE -DQT_NO_DEBUG -DQT
_PRINTSUPPORT_LIB -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I. -Iqtpropertybr
owser/src -isystem /usr/include/x86_64-linux-gnu/qt5 -isystem /usr/include/x86_6
4-linux-gnu/qt5/QtPrintSupport -isystem /usr/include/x86_64-linux-gnu/qt5/QtWidg
ets -isystem /usr/include/x86_64-linux-gnu/qt5/QtGui -isystem /usr/include/x86_6
4-linux-gnu/qt5/QtCore -I. -I/usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++ -o
main.o main.cpp
```

```
wser.o moc_filepathmanager.o moc_fileeditfactory.o moc_fileedit.o moc_packetsmod
e.o moc_table.o moc_qcustomplot.o   /usr/lib/x86_64-linux-gnu/libQt5PrintSupport
.so /usr/lib/x86_64-linux-gnu/libQt5Widgets.so /usr/lib/x86_64-linux-gnu/libQt5G
ui.so /usr/lib/x86_64-linux-gnu/libQt5Core.so /usr/lib/x86_64-linux-gnu/libGL.so
 -lpthread
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ ./run
assign
bash: ./run: No such file or directory
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ ./NetA
nim
```

Output: Netanim Screen:





	From Id	To Id	Tx
1	0	1	1.008
2	0	1	1.008
3	1	0	1.0101
4	1	0	1.0101
5	0	1	1.01221
6	0	1	1.01221
7	0	1	2
8	0	1	2
9	0	1	3
10	0	1	3
11	0	1	4
12	0	1	4
13	0	1	5
14	0	1	5
15	0	1	6
16	0	1	6
17	0	1	7
18	0	1	7
19	0	1	8
20	0	1	8
21	0	1	9
22	0	1	9
23	0	1	10

Practical No: 13

Aim: - Animate a simple network using NetAnim in Network Simulator

Line Of Code: -

Wifinodes1.cc

```
/* -*- 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/point-to-point-module.h"
#include "ns3/csma-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/internet-module.h"
#include "ns3/netanim-module.h"
#include "ns3/basic-energy-source.h"
#include "ns3/simple-device-energy-model.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/ssid.h"
#include "ns3/wifi-radio-energy-model.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("WirelessAnimationExample");
int
main (int argc, char *argv[])
{
  uint32_t nWifi = 20;
  CommandLine cmd (__FILE__);
  cmd.AddValue ("nWifi", "Number of wifi STA devices", nWifi);
  cmd.Parse (argc,argv);
  NodeContainer allNodes;
  NodeContainer wifiStaNodes;
  wifiStaNodes.Create (nWifi);
  allNodes.Add (wifiStaNodes);
  NodeContainer wifiApNode ;
  wifiApNode.Create (1);
  allNodes.Add (wifiApNode);
  YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
  YansWifiPhyHelper phy;
```

```
  phy.SetChannel (channel.Create ());

  WifiHelper wifi;
  WifiMacHelper mac;
  Ssid ssid = Ssid ("ns-3-ssid");
  mac.SetType ("ns3::StaWifiMac",
               "Ssid", SsidValue (ssid),
               "ActiveProbing", BooleanValue (false));

  NetDeviceContainer staDevices;
  staDevices = wifi.Install (phy, mac, wifiStaNodes);
  mac.SetType ("ns3::ApWifiMac",
               "Ssid", SsidValue (ssid));

  NetDeviceContainer apDevices;
  apDevices = wifi.Install (phy, mac, wifiApNode);
  NodeContainer p2pNodes;
  p2pNodes.Add (wifiApNode);
  p2pNodes.Create (1);
  allNodes.Add (p2pNodes.Get (1));

  PointToPointHelper pointToPoint;
  pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
  pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

  NetDeviceContainer p2pDevices;
  p2pDevices = pointToPoint.Install (p2pNodes);

  NodeContainer csmaNodes;
  csmaNodes.Add (p2pNodes.Get (1));
  csmaNodes.Create (1);
  allNodes.Add (csmaNodes.Get (1));
  CsmaHelper csma;
  csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
  csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

  NetDeviceContainer csmaDevices;
  csmaDevices = csma.Install (csmaNodes);
  MobilityHelper mobility;
  mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                                 "MinX", DoubleValue (10.0),
                                 "MinY", DoubleValue (10.0),
                                 "DeltaX", DoubleValue (5.0),
                                 "DeltaY", DoubleValue (2.0),
                                 "GridWidth", UintegerValue (5),
                                 "LayoutType", StringValue ("RowFirst"));
  mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                    "Bounds", RectangleValue (Rectangle (-50, 50, -25,
50)));
  mobility.Install (wifiStaNodes);
  mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
  mobility.Install (wifiApNode);
  AnimationInterface::SetConstantPosition (p2pNodes.Get (1), 10, 30);
  AnimationInterface::SetConstantPosition (csmaNodes.Get (1), 10, 33);
  Ptr<BasicEnergySource> energySource = CreateObject<BasicEnergySource>();
```

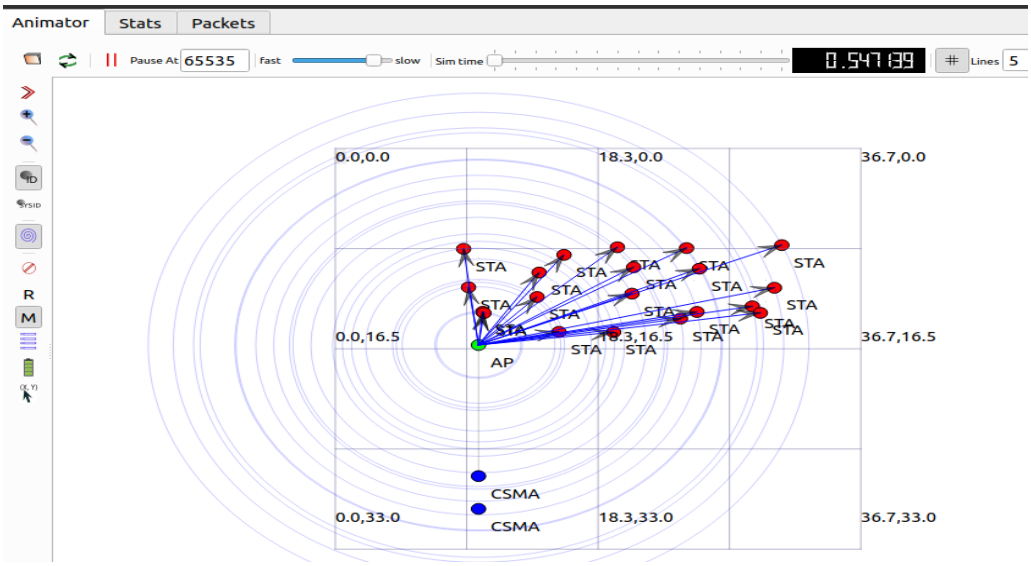
```
  Ptr<WifiRadioEnergyModel> energyModel =
CreateObject<WifiRadioEnergyModel>();
  energySource->SetInitialEnergy (300);
  energyModel->SetEnergySource (energySource);
  energySource->AppendDeviceEnergyModel (energyModel);

  // aggregate energy source to node
  wifiApNode.Get (0)->AggregateObject (energySource);
  // Install internet stack
  InternetStackHelper stack;
  stack.Install (allNodes);
  // Install Ipv4 addresses

  Ipv4AddressHelper address;
  address.SetBase ("10.1.1.0", "255.255.255.0");
  Ipv4InterfaceContainer p2pInterfaces;
  p2pInterfaces = address.Assign (p2pDevices);
  address.SetBase ("10.1.2.0", "255.255.255.0");
  Ipv4InterfaceContainer csmaInterfaces;
  csmaInterfaces = address.Assign (csmaDevices);
  address.SetBase ("10.1.3.0", "255.255.255.0");
  Ipv4InterfaceContainer staInterfaces;
  staInterfaces = address.Assign (staDevices);
  Ipv4InterfaceContainer apInterface;
  apInterface = address.Assign (apDevices);
  csma.EnablePcapAll ("wifinodes-echo", false);

  UdpEchoServerHelper echoServer (9);
  ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (1));
  serverApps.Start (Seconds (1.0));
  serverApps.Stop (Seconds (15.0));
  UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (1), 9);
  echoClient.SetAttribute ("MaxPackets", UintegerValue (10));
  echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.)));
  echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
  ApplicationContainer clientApps = echoClient.Install (wifiStaNodes);
  clientApps.Start (Seconds (2.0));
  clientApps.Stop (Seconds (15.0));

  Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
  Simulator::Stop (Seconds (15.0));
  AnimationInterface anim ("wireless-animation.xml"); // Mandatory
  for (uint32_t i = 0; i < wifiStaNodes.GetN (); ++i)
    {
      anim.UpdateNodeDescription (wifiStaNodes.Get (i), "STA"); // Optional
      anim.UpdateNodeColor (wifiStaNodes.Get (i), 255, 0, 0); // Optional
    }
  for (uint32_t i = 0; i < wifiApNode.GetN (); ++i)
    {
      anim.UpdateNodeDescription (wifiApNode.Get (i), "AP"); // Optional
      anim.UpdateNodeColor (wifiApNode.Get (i), 0, 255, 0); // Optional
    }
  for (uint32_t i = 0; i < csmaNodes.GetN (); ++i)
    {
      anim.UpdateNodeDescription (csmaNodes.Get (i), "CSMA"); // Optional
      anim.UpdateNodeColor (csmaNodes.Get (i), 0, 0, 255); // Optional
```

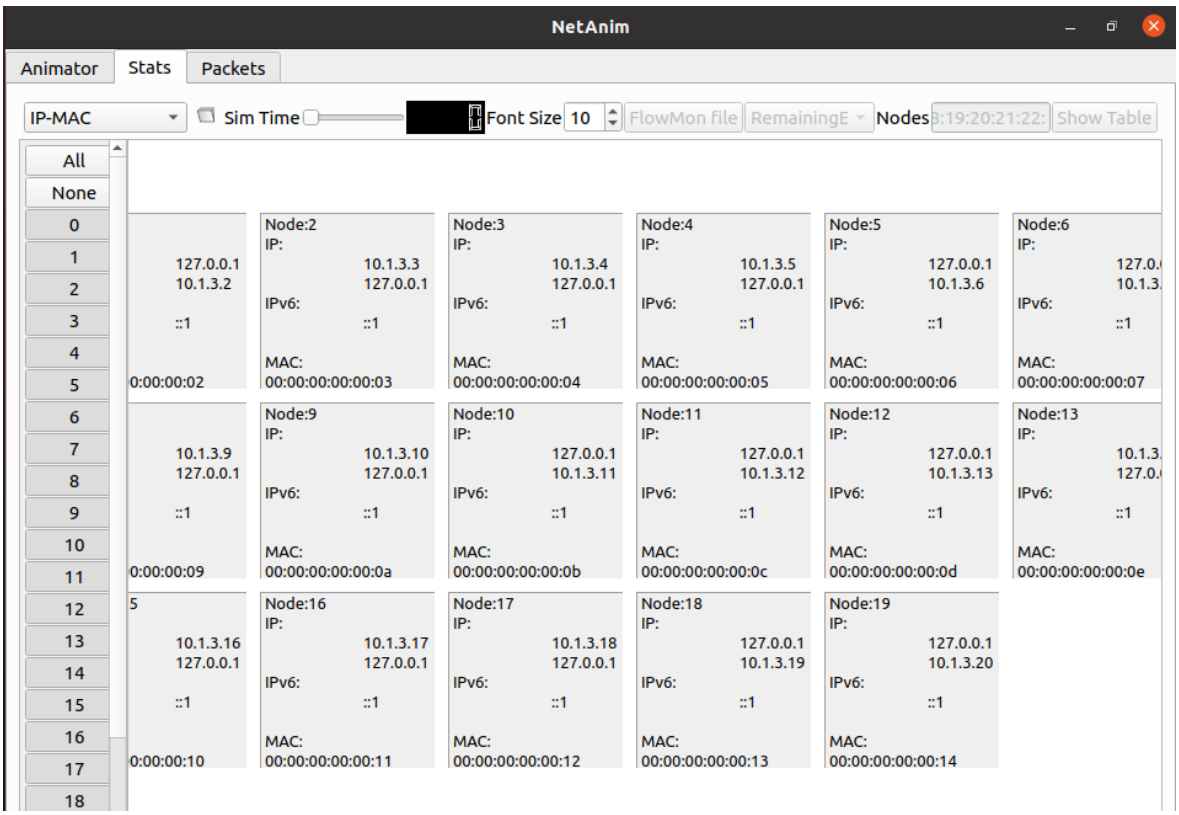
```
    }
//anim.EnablePacketMetadata (); // Optional
//anim.EnableIpv4RouteTracking ("routingtable-wireless1.xml", Seconds (0),
Seconds (5), Seconds (0.25)); //Optional
//anim.EnableWifiMacCounters (Seconds (0), Seconds (10)); //Optional
// anim.EnableWifiPhyCounters (Seconds (0), Seconds (10)); //Optional
 //AnimationInterface anim ("routingtable-wireless1.xml");
  Simulator::Run ();
  Simulator::Destroy ();
  return 0;
}
```

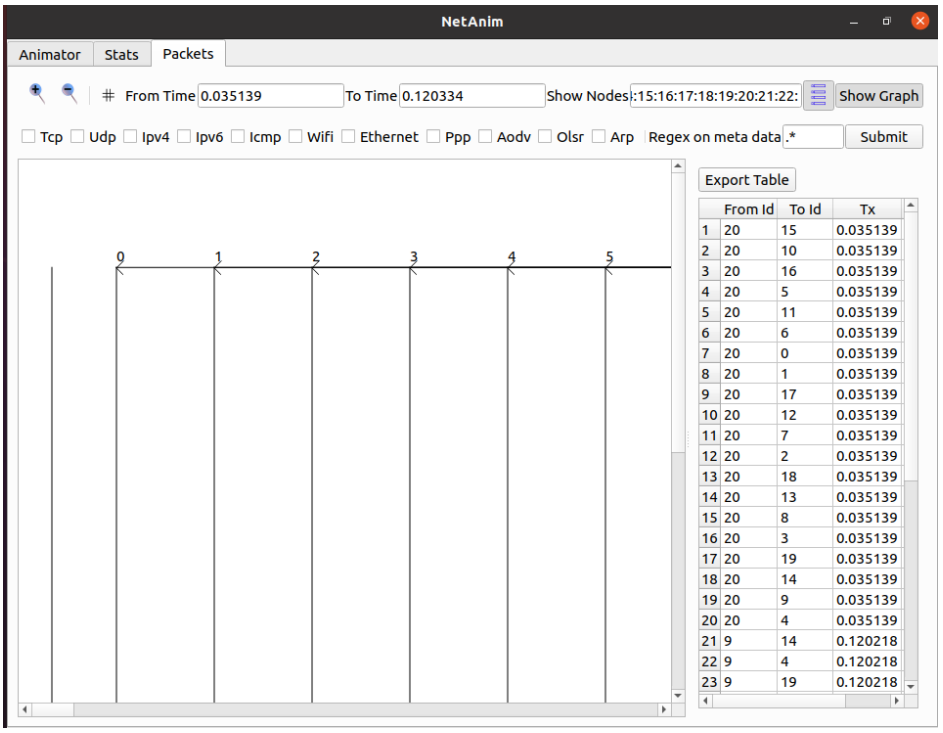
Program:-

```
mca@mca-To-be-filled-by-O-E-M: ~/workspace/ns-allinone-...
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/ns-3.33$ ./waf --run
 wifinodes1
Waf: Entering directory `/home/mca/workspace/ns-allinone-3.33/ns-3.33/build'
[2026/2105] Compiling scratch/wifinodes1.cc
[2027/2105] Compiling scratch/first.cc
[2028/2105] Compiling scratch/scratch-simulator.cc
[2029/2105] Compiling scratch/subdir/scratch-simulator-subdir.cc
[2060/2105] Compiling scratch/assign.cc
[2061/2105] Linking build/scratch/scratch-simulator
[2062/2105] Linking build/scratch/subdir/subdir
[2063/2105] Linking build/scratch/first
[2064/2105] Linking build/scratch/wifinodes1
[2065/2105] Linking build/scratch/assign
Waf: Leaving directory `/home/mca/workspace/ns-allinone-3.33/ns-3.33/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (8.028s)
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/ns-3.33$ cd ..
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33$ cd netanim-3.108
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ make
clean
rm -f qrc_resources.cpp qrc_qtpropertybrowser.cpp
propertybrowser.o moc_filepathmanager.o moc_fileeditfactory.o moc_fileedit.o mo
c_packetsmode.o moc_table.o moc_qcustomplot.o    /usr/lib/x86_64-linux-gnu/libQt
5PrintSupport.so /usr/lib/x86_64-linux-gnu/libQt5Widgets.so /usr/lib/x86_64-lin
ux-gnu/libQt5Gui.so /usr/lib/x86_64-linux-gnu/libQt5Core.so /usr/lib/x86_64-lin
ux-gnu/libGL.so -lpthread
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.33/netanim-3.108$ ./Net
Anim
```

Output: Netanim Screen:-







	From Id	To Id	Tx
1	20	15	0.035139
2	20	10	0.035139
3	20	16	0.035139
4	20	5	0.035139
5	20	11	0.035139
6	20	6	0.035139
7	20	0	0.035139
8	20	1	0.035139
9	20	17	0.035139
10	20	12	0.035139
11	20	7	0.035139
12	20	2	0.035139
13	20	18	0.035139
14	20	13	0.035139
15	20	8	0.035139
16	20	3	0.035139
17	20	19	0.035139
18	20	14	0.035139
19	20	9	0.035139
20	20	4	0.035139
21	9	14	0.120218
22	9	4	0.120218
23	9	19	0.120218

Practical No: 14

Aim: - Animate Three way handshake for TCP connection using NetAnim

Line Of Code: -

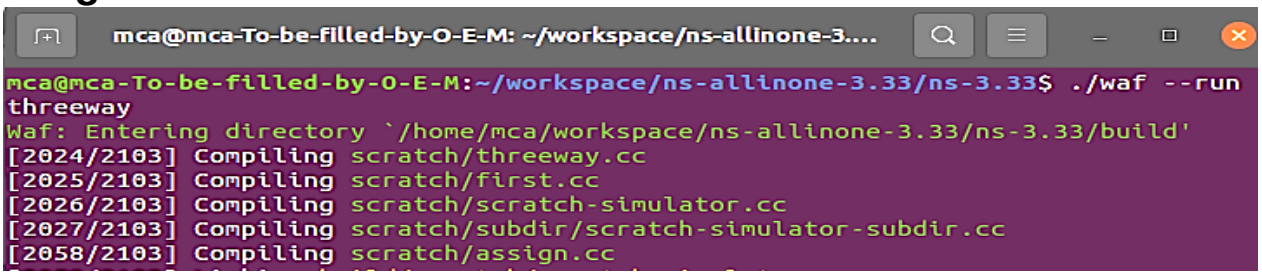
Threeway.cc

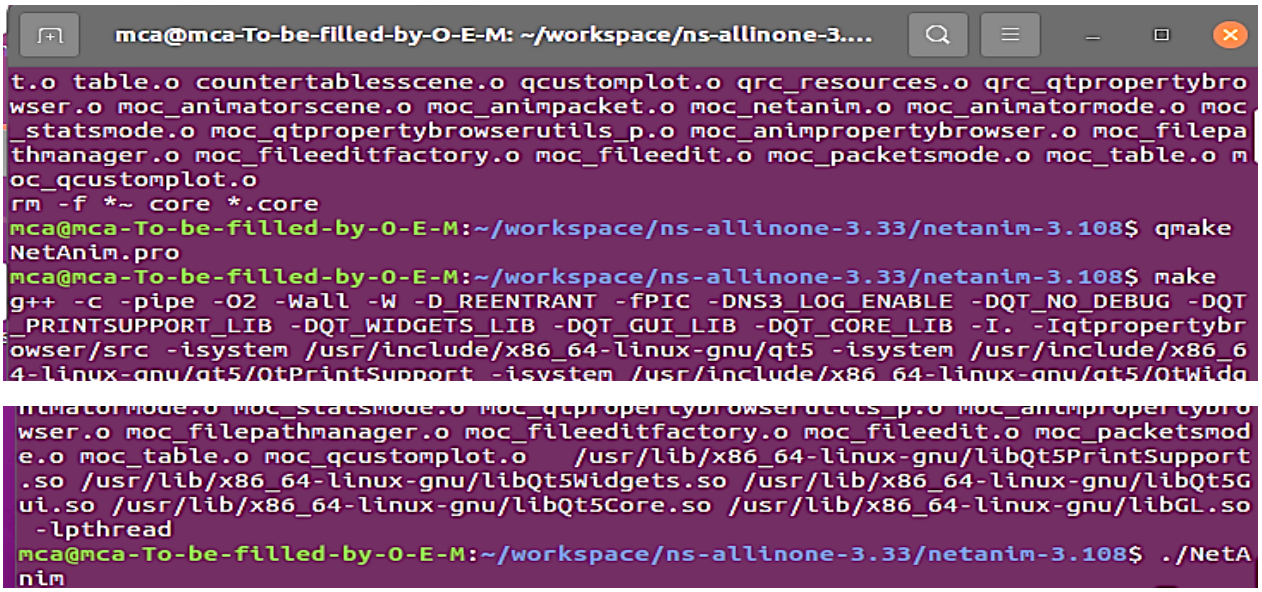
```
#include <iostream>
#include <fstream>
#include <string>
#include <cassert>
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/netanim-module.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("TcpServer");
int
main (int argc, char *argv[])
{
// Users may find it convenient to turn on explicit debugging
// for selected modules; the below lines suggest how to do this
//LogComponentEnable ("TcpServer", LOG_LEVEL_INFO);
//LogComponentEnable ("TcpL4Protocol", LOG_LEVEL_ALL);
//LogComponentEnable ("TcpSocketImpl", LOG_LEVEL_ALL);
//LogComponentEnable ("PacketSink", LOG_LEVEL_ALL);
// Set up some default values for the simulation.
Config::SetDefault ("ns3::OnOffApplication::PacketSize",
UintegerValue (250));
Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue
("5kb/s"));
uint32_t N = 5; //number of nodes in the star
// Allow the user to override any of the defaults and the above
// Config::SetDefault()s at run-time, via command-line arguments
CommandLine cmd;
cmd.AddValue ("nNodes", "Number of nodes to place in the star", N);
cmd.Parse (argc, argv);
// Here, we will create N nodes in a star.
NS_LOG_INFO ("Create nodes.");
NodeContainer serverNode;
NodeContainer clientNodes;
serverNode.Create (1);
clientNodes.Create (N-1);
NodeContainer allNodes = NodeContainer (serverNode, clientNodes);
// Install network stacks on the nodes
InternetStackHelper internet;
internet.Install (allNodes);
std::vector<NodeContainer> nodeAdjacencyList (N-1);
for(uint32_t i=0; i<nodeAdjacencyList.size (); ++i)
{
nodeAdjacencyList[i] = NodeContainer (serverNode, clientNodes.Get
```

```
(i));
}
NS_LOG_INFO ("Create channels.");
PointToPointHelper p2p;
p2p.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
p2p.SetChannelAttribute ("Delay", StringValue ("2ms"));
std::vector<NetDeviceContainer> deviceAdjacencyList (N-1);
for(uint32_t i=0; i<deviceAdjacencyList.size (); ++i)
{
deviceAdjacencyList[i] = p2p.Install (nodeAdjacencyList[i]);
//Animating The Network
}
// Later, we add IP addresses.
NS_LOG_INFO ("Assign IP Addresses.");
Ipv4AddressHelper ipv4;
std::vector<Ipv4InterfaceContainer> interfaceAdjacencyList (N-1);
for(uint32_t i=0; i<interfaceAdjacencyList.size (); ++i)
{
std::ostringstream subnet;
subnet<<"10.1."<<i+1<<".0";
ipv4.SetBase (subnet.str ().c_str (), "255.255.255.0");
interfaceAdjacencyList[i] = ipv4.Assign (deviceAdjacencyList[i]);
}
//Turn on global static routing
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
// Create a packet sink on the star "hub" to receive these packets
uint16_t port = 50000;
Address sinkLocalAddress (InetSocketAddress (Ipv4Address::GetAny(), port));
PacketSinkHelper sinkHelper ("ns3::TcpSocketFactory",
sinkLocalAddress);
ApplicationContainer sinkApp = sinkHelper.Install (serverNode);
sinkApp.Start (Seconds (1.0));
sinkApp.Stop (Seconds (10.0));
// Create the OnOff applications to send TCP to the server
OnOffHelper clientHelper ("ns3::TcpSocketFactory", Address ());
clientHelper.SetAttribute ("OnTime", StringValue
("ns3::ConstantRandomVariable[Constant=1]"));
clientHelper.SetAttribute ("OffTime", StringValue
("ns3::ConstantRandomVariable[Constant=0]"));
//normally wouldn't need a loop here but the server IP address is different
ApplicationContainer clientApps;
for(uint32_t i=0; i<clientNodes.GetN (); ++i)
{
AddressValue remoteAddress
(InetSocketAddress (interfaceAdjacencyList[i].GetAddress (0), port));
clientHelper.SetAttribute ("Remote", remoteAddress);
clientApps.Add (clientHelper.Install (clientNodes.Get (i)));
}
clientApps.Start (Seconds (1.0));
clientApps.Stop (Seconds (10.0));
//configure tracing
AsciiTraceHelper ascii;
p2p.EnableAsciiAll (ascii.CreateFileStream ("tcp-star-server.tr"));
p2p.EnablePcapAll ("tcp-star-server");
NS_LOG_INFO ("Run Simulation.");
AnimationInterface anim ("Anim.xml");
```

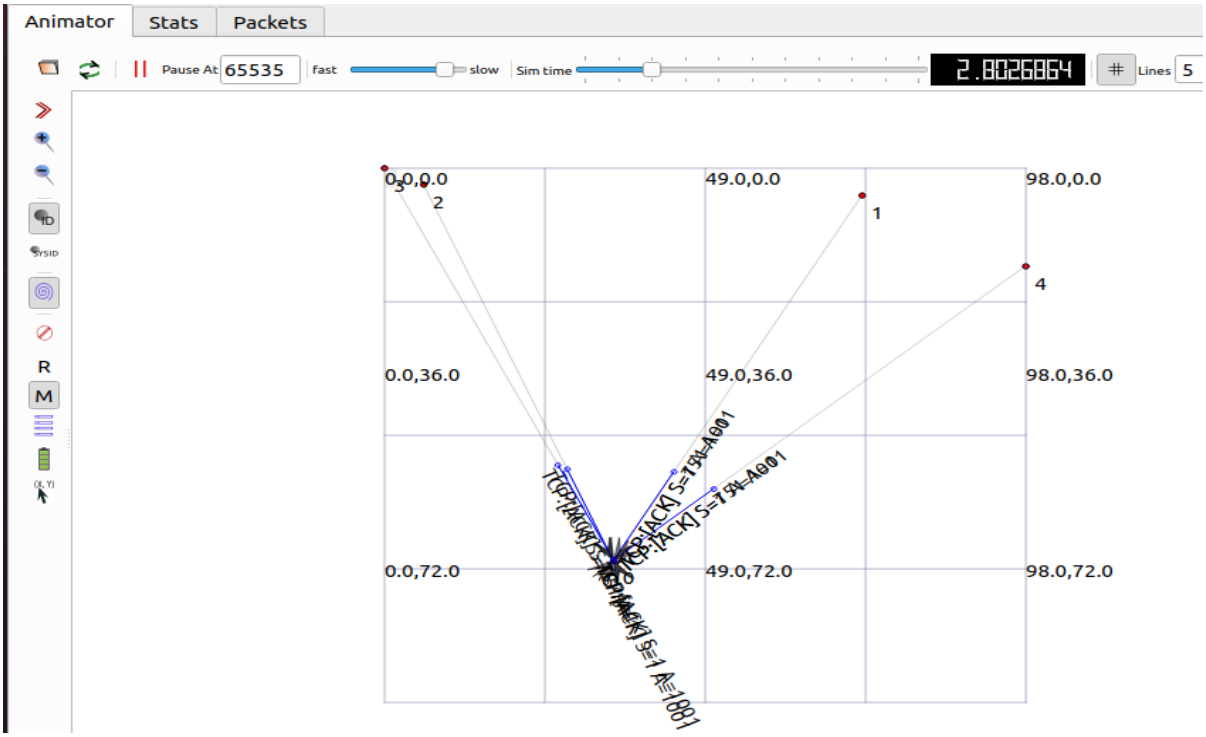
```
anim.EnablePacketMetadata (true);
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
return 0;
}
```

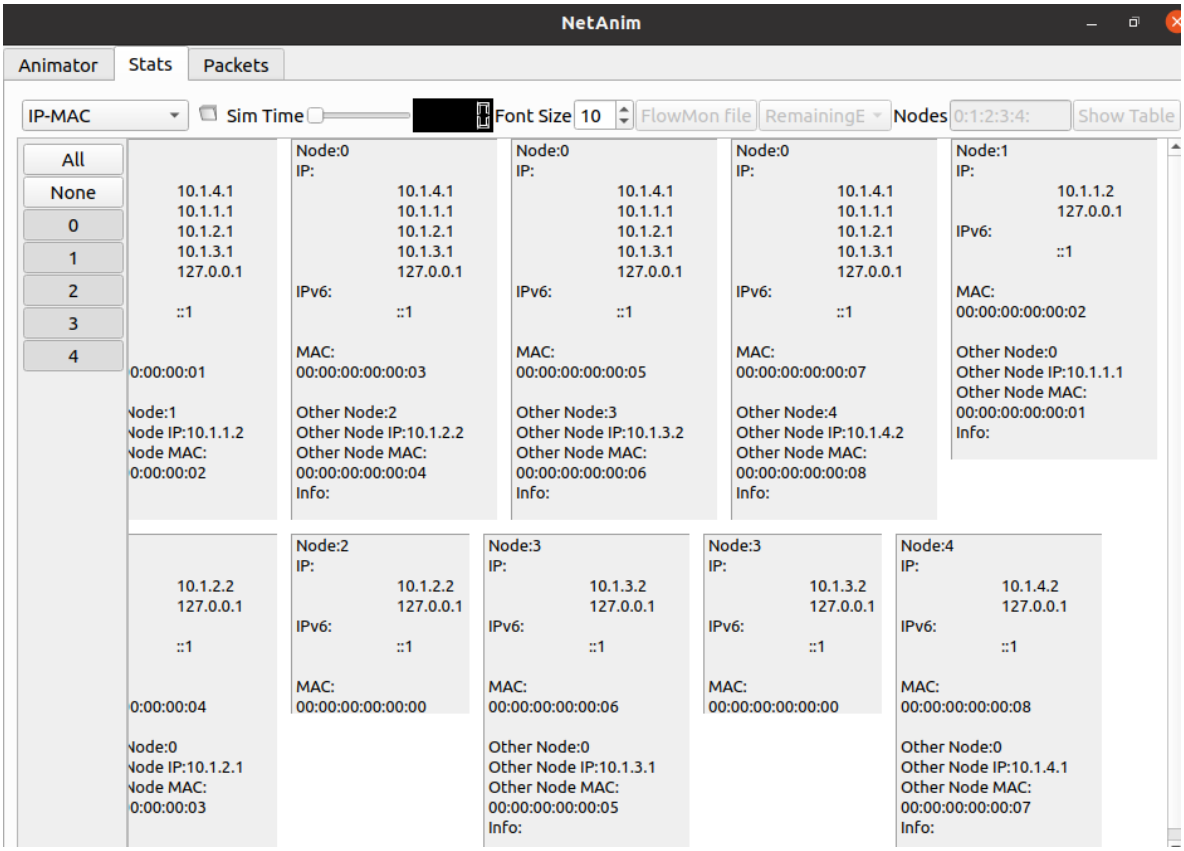
Program: -



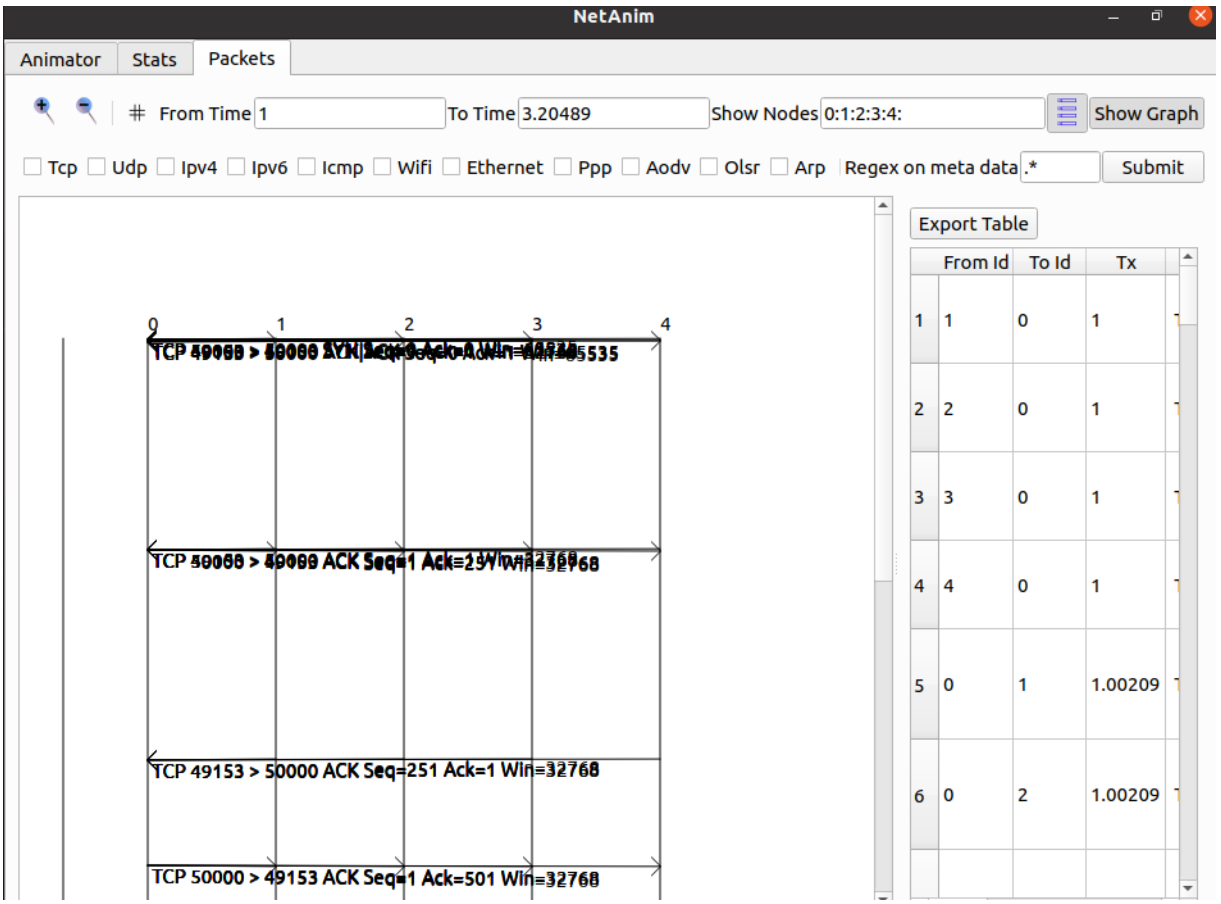


Output: Netanim Screen:-





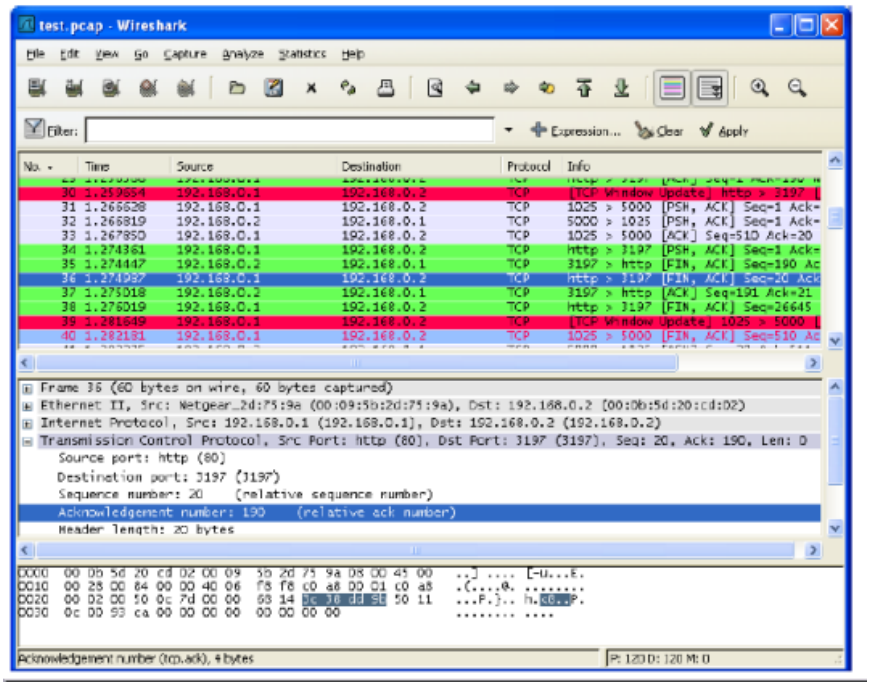
Name:Mansingh Yadav **Roll No. 61**

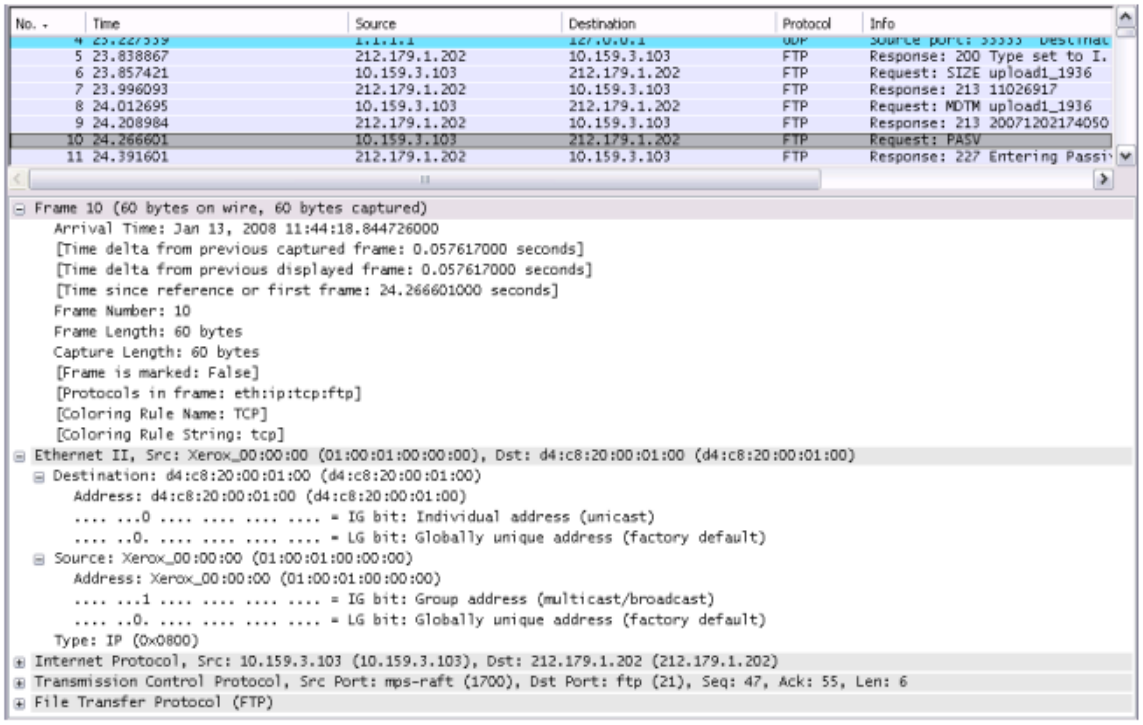

NetAnim
Animator
Stats
Packets
From Time 1
To Time 3.20489
Show Nodes 0:1:2:3:4:
Show Graph
Tcp Udp Ipv4 Ipv6 Icmp Wifi Ethernet Ppp Aodv Olsr Arp
Regex on meta data .*
Submit
Export Table
From Id
To Id
Tx
1 1 0 1
2 2 0 1
3 3 0 1
4 4 0 1
5 0 1 1.00209
6 0 2 1.00209
0
1
2
3
4
TCP 49153 > 50000 ACK Seq=251 Ack=1 Win=32768
TCP 50000 > 49153 ACK Seq=1 Ack=501 Win=32768

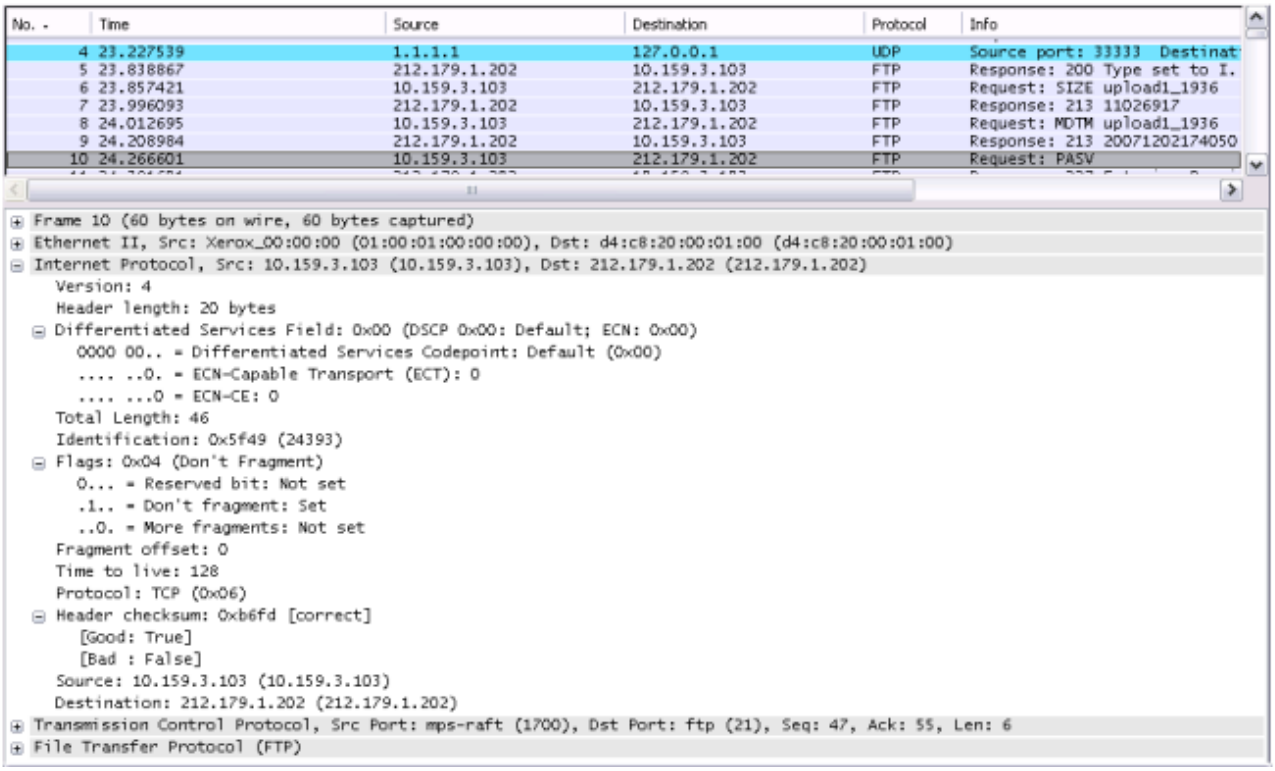
Practical No. 15

Aim:- Analyze the network traffic using WireShark

Output :-

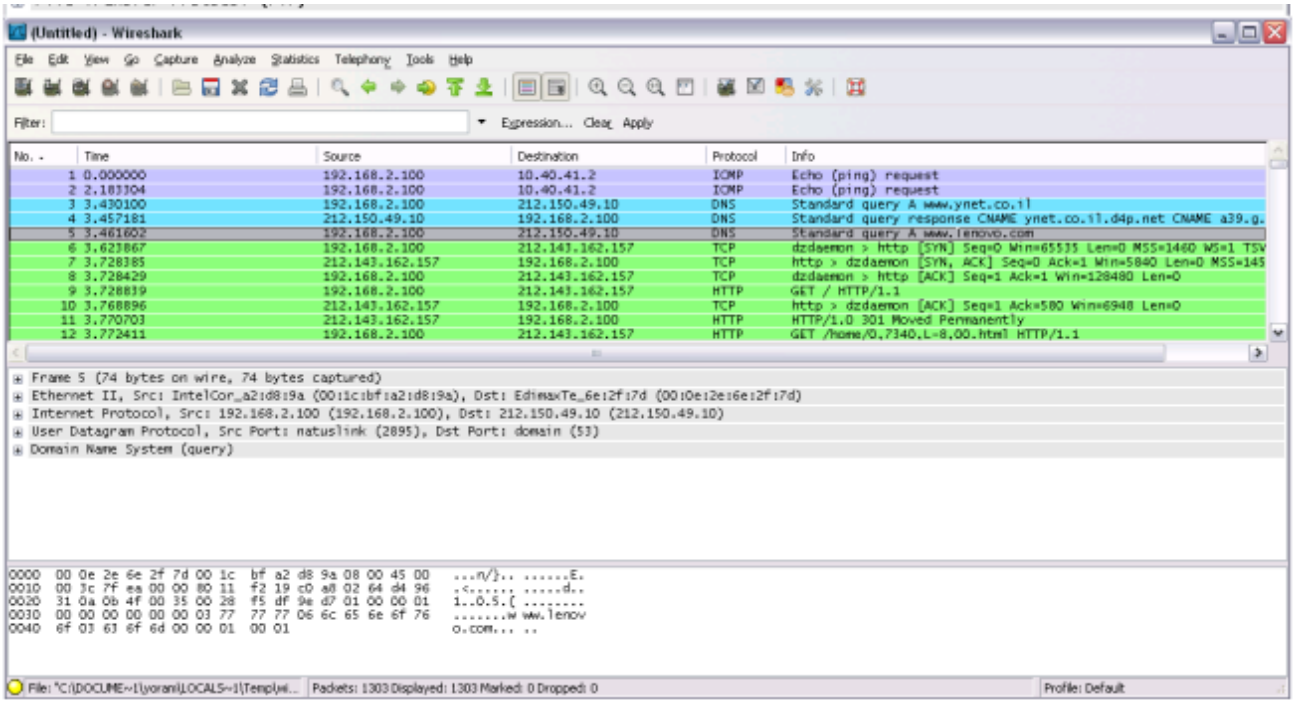


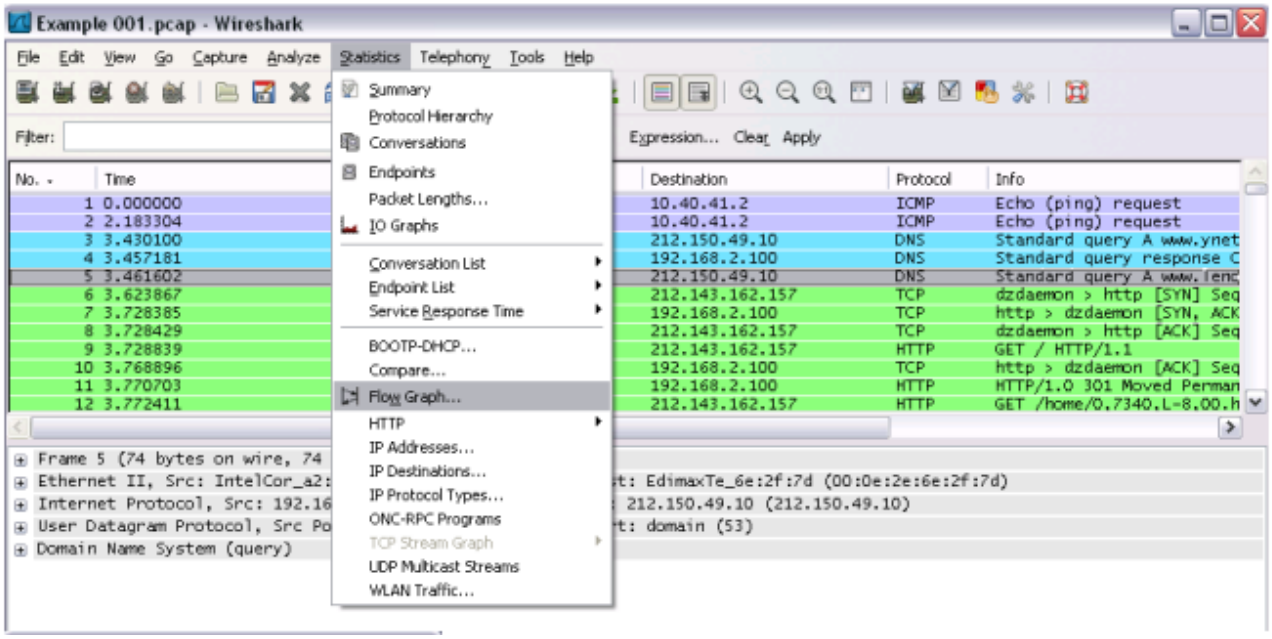


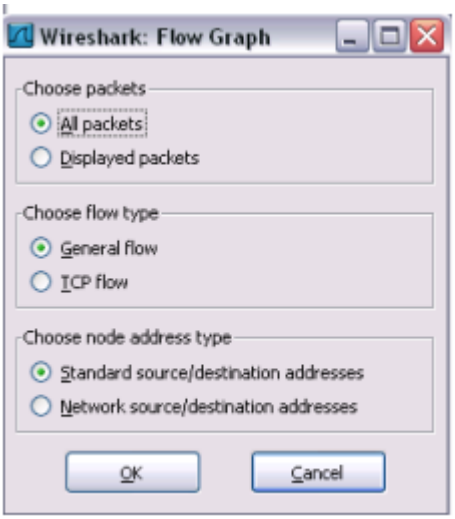


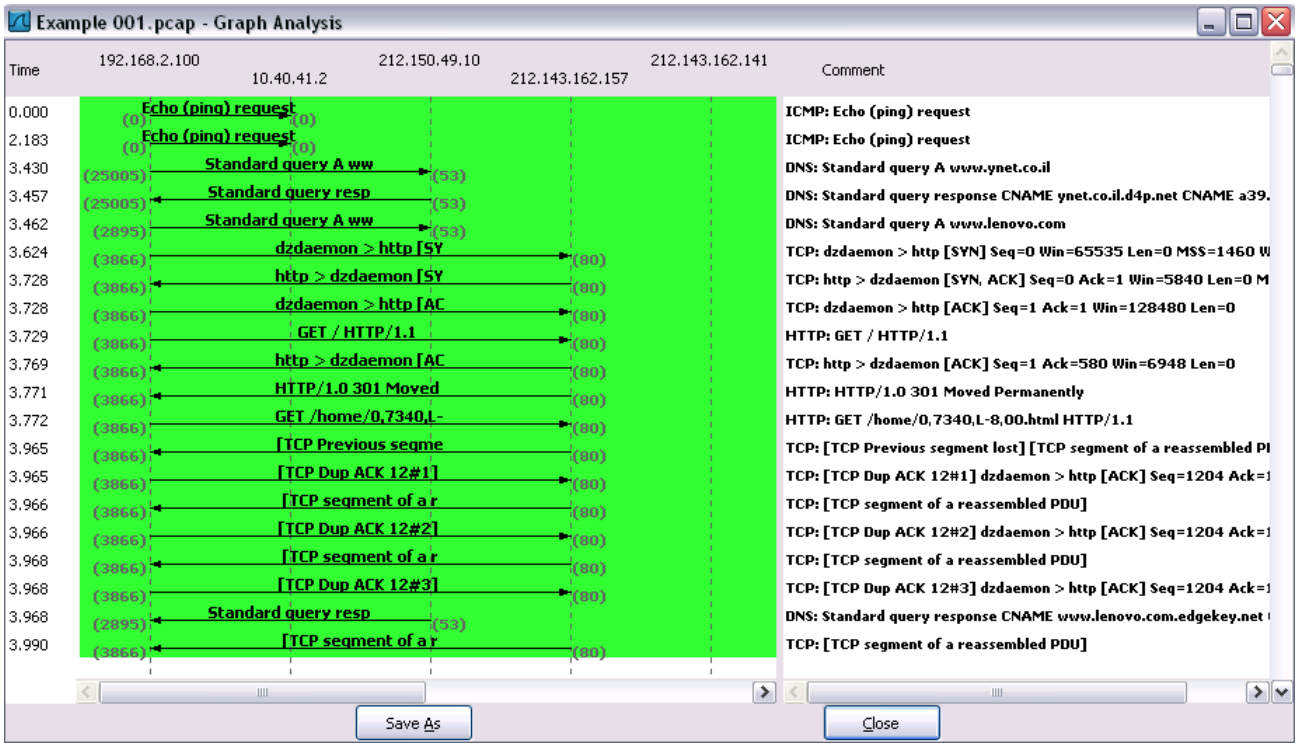
No.	Time	Source	Destination	Protocol	Info
4	23.227539	1.1.1.1	127.0.0.1	UDP	Source port: 33333 Destinat
5	23.838867	212.179.1.202	10.159.3.103	FTP	Response: 200 Type set to I.
6	23.857421	10.159.3.103	212.179.1.202	FTP	Request: SIZE upload1_1936
7	23.996093	212.179.1.202	10.159.3.103	FTP	Response: 213 11026917
8	24.012695	10.159.3.103	212.179.1.202	FTP	Request: MDTM upload1_1936
9	24.208984	212.179.1.202	10.159.3.103	FTP	Response: 213 20071202174050
10	24.266601	10.159.3.103	212.179.1.202	FTP	Request: PASV

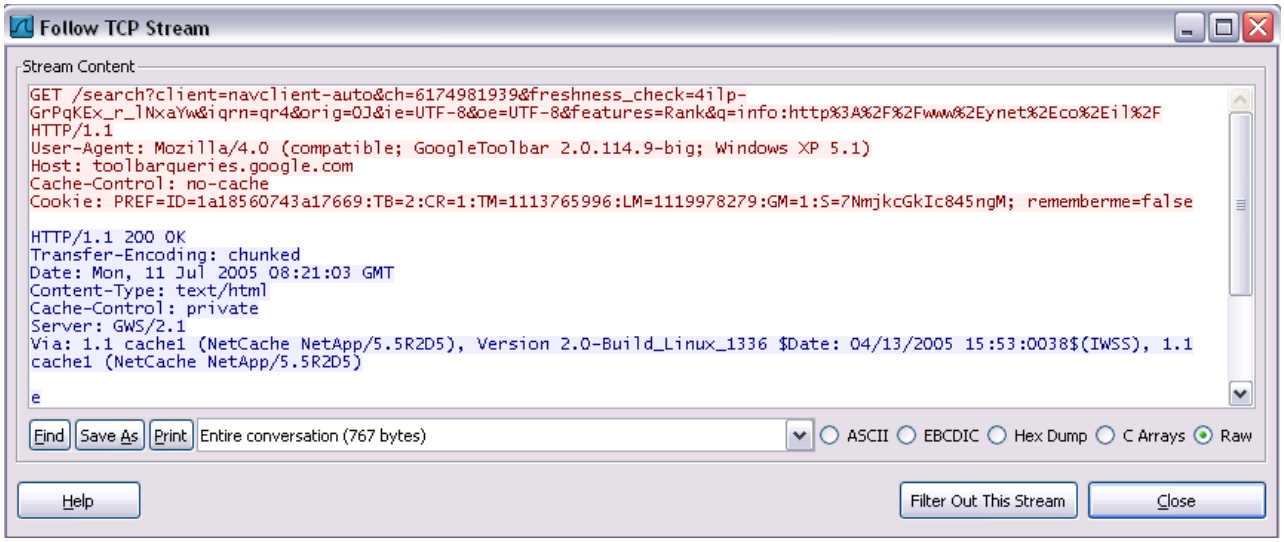
```
⊞ Frame 10 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: Xerox_00:00:00 (01:00:01:00:00:00), Dst: d4:c8:20:00:01:00 (d4:c8:20:00:01:00)
⊟ Internet Protocol, Src: 10.159.3.103 (10.159.3.103), Dst: 212.179.1.202 (212.179.1.202)
    Version: 4
    Header length: 20 bytes
  ⊟ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..0. = ECN-Capable Transport (ECT): 0
      .... ...0 = ECN-CE: 0
    Total Length: 46
    Identification: 0x5f49 (24393)
  ⊟ Flags: 0x04 (Don't Fragment)
      0... = Reserved bit: Not set
      .1.. = Don't fragment: Set
      ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
  ⊟ Header checksum: 0xb6fd [correct]
      [Good: True]
      [Bad : False]
    Source: 10.159.3.103 (10.159.3.103)
    Destination: 212.179.1.202 (212.179.1.202)
⊞ Transmission Control Protocol, Src Port: mps-raft (1700), Dst Port: ftp (21), Seq: 47, Ack: 55, Len: 6
⊞ File Transfer Protocol (FTP)
```

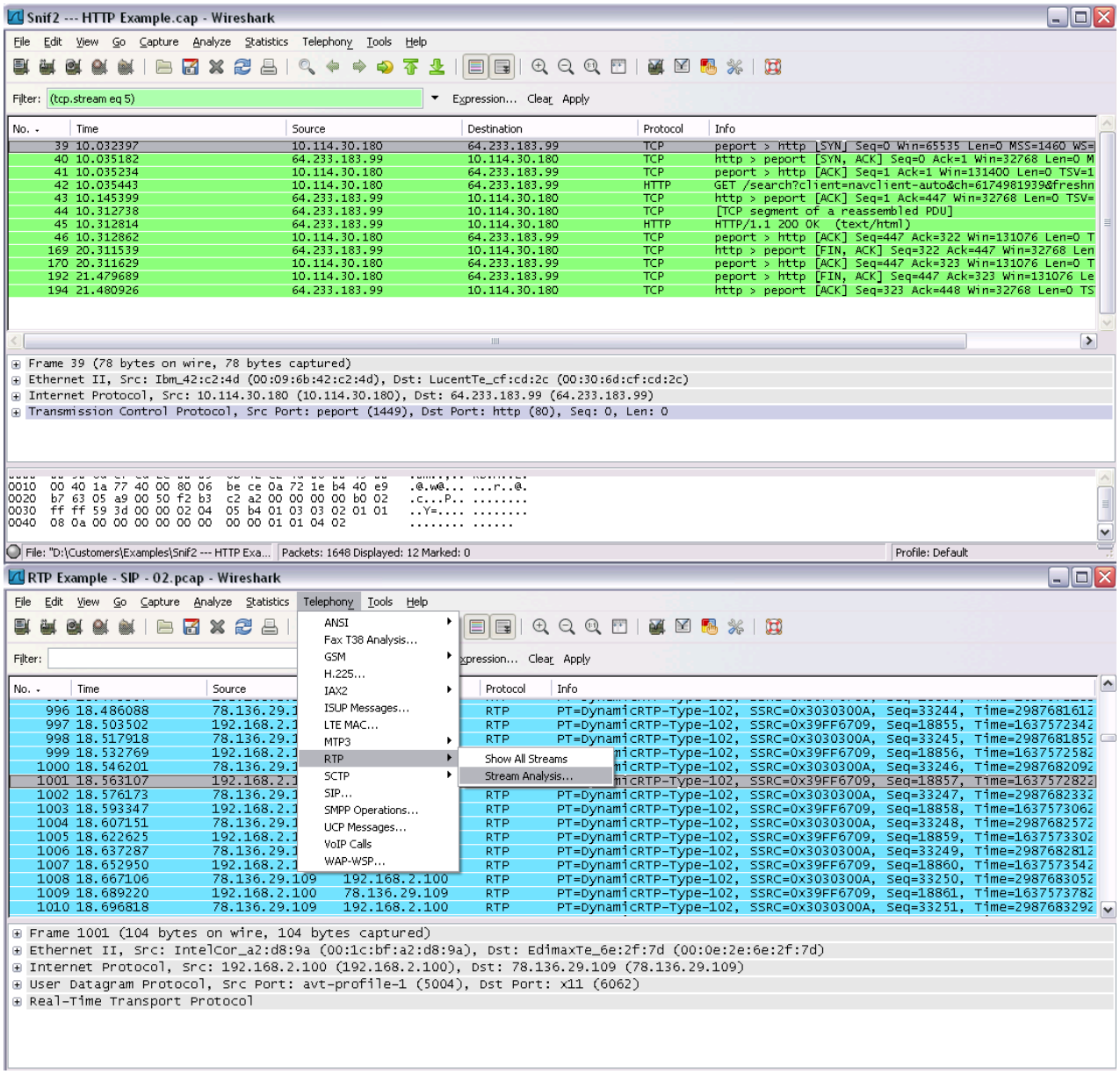


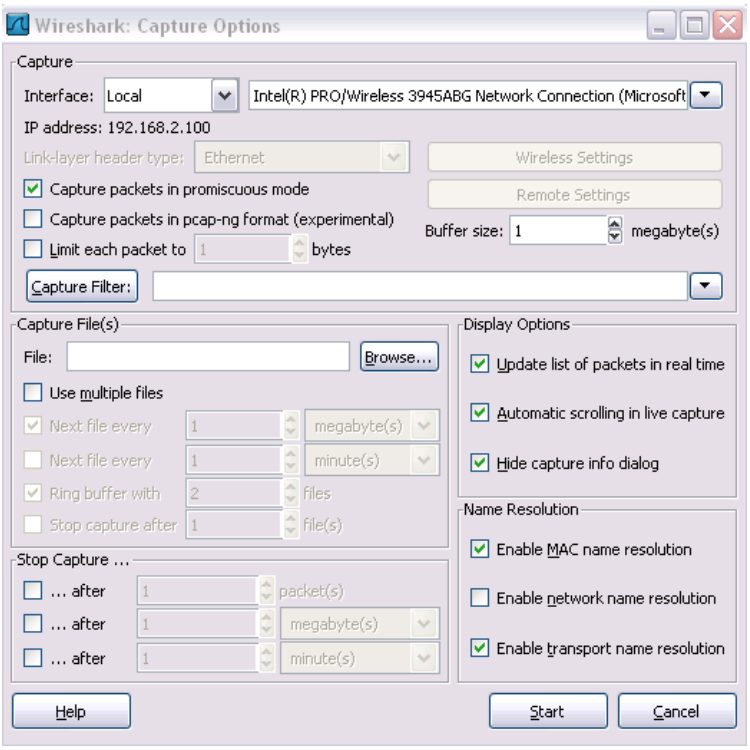

Example 001.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Summary
Protocol Hierarchy
Conversations
Endpoints
Packet Lengths...
IO Graphs
Conversation List
Endpoint List
Service Response Time
BOOTP-DHCP...
Compare...
Flow Graph...
HTTP
IP Addresses...
IP Destinations...
IP Protocol Types...
ONC-RPC Programs
TCP Stream Graph
UDP Multicast Streams
WLAN Traffic...
Filter:
Expression... Clear Apply
No. Time Destination Protocol Info
1 0.000000 10.40.41.2 ICMP Echo (ping) request
2 2.183304 10.40.41.2 ICMP Echo (ping) request
3 3.430100 212.150.49.10 DNS Standard query A www.ynet
4 3.457181 192.168.2.100 DNS Standard query response C
5 3.461602 212.150.49.10 DNS Standard query A www.fenc
6 3.623867 212.143.162.157 TCP dzdaemon > http [SYN] Seq
7 3.728385 192.168.2.100 TCP http > dzdaemon [SYN, ACK
8 3.728429 212.143.162.157 TCP dzdaemon > http [ACK] Seq
9 3.728839 212.143.162.157 HTTP GET / HTTP/1.1
10 3.768896 192.168.2.100 TCP http > dzdaemon [ACK] Seq
11 3.770703 192.168.2.100 HTTP HTTP/1.0 301 Moved Perman
12 3.772411 212.143.162.157 HTTP GET /home/0,7340,L-8,00.h
Frame 5 (74 bytes on wire, 74
Ethernet II, Src: IntelCor_a2:
Internet Protocol, Src: 192.16
User Datagram Protocol, Src Po
Domain Name System (query)
t: EdimaxTe_6e:2f:7d (00:0e:2e:6e:2f:7d)
212.150.49.10 (212.150.49.10)
t: domain (53)

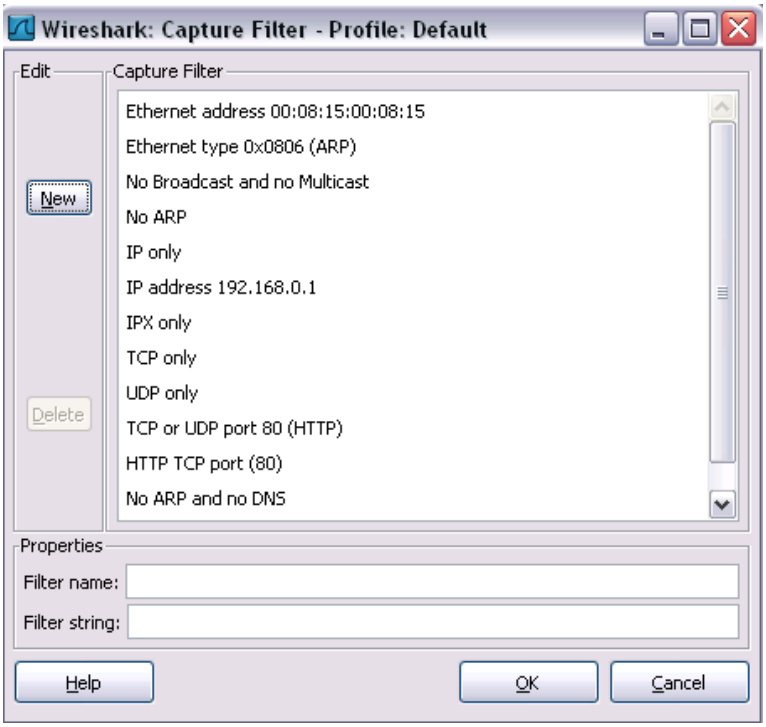

Wireshark: Flow Graph
Choose packets
All packets
Displayed packets
Choose flow type
General flow
TCP flow
Choose node address type
Standard source/destination addresses
Network source/destination addresses
OK
Cancel

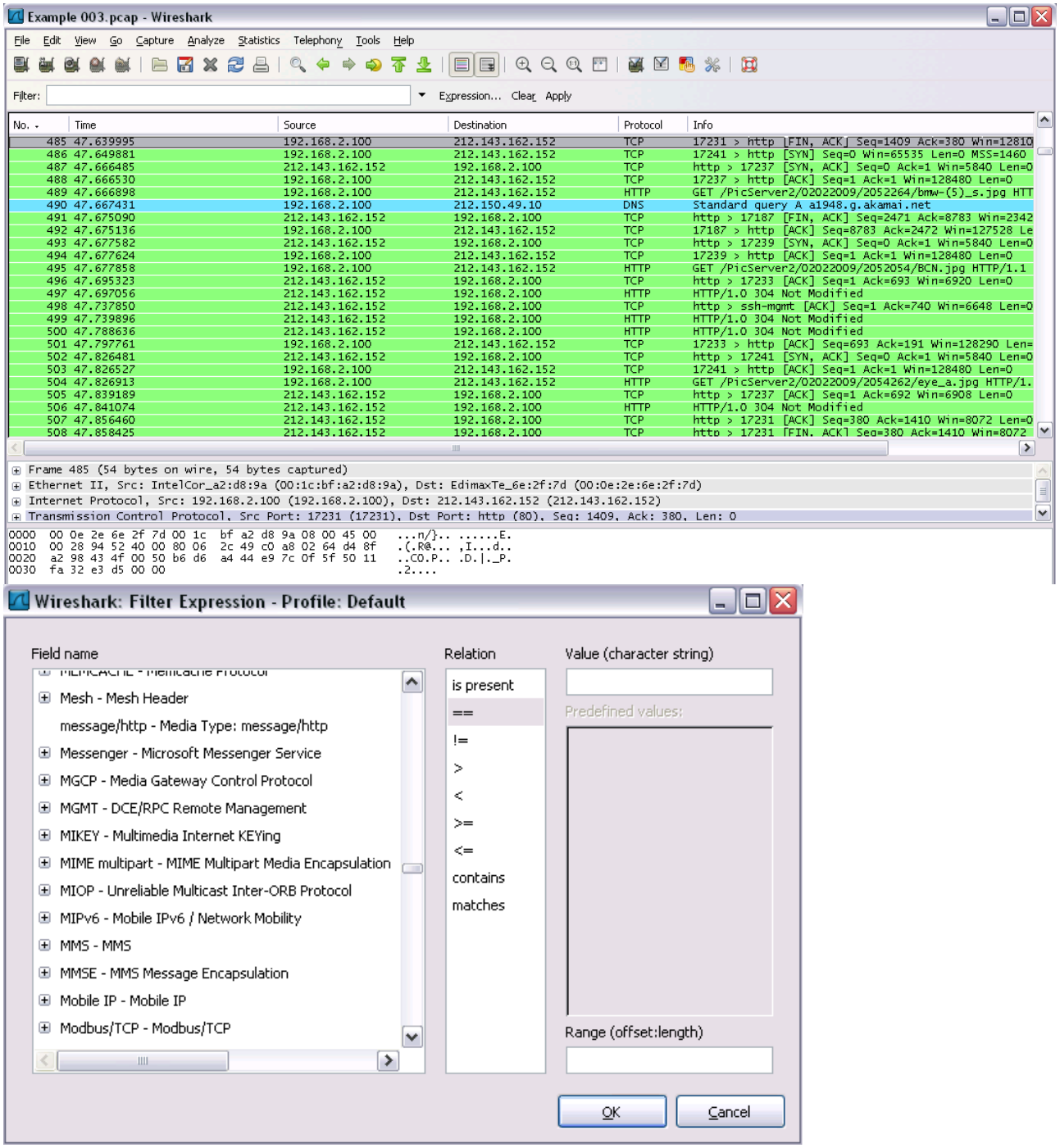

Example 001.pcap - Graph Analysis
Time
192.168.2.100
10.40.41.2
212.150.49.10
212.143.162.157
212.143.162.141
Comment
0.000 Echo (ping) request (0) (0) ICMP: Echo (ping) request
2.183 Echo (ping) request (0) (0) ICMP: Echo (ping) request
3.430 Standard query A ww (25005) (53) DNS: Standard query A www.ynet.co.il
3.457 Standard query resp (25005) (53) DNS: Standard query response CNAME ynet.co.il.d4p.net CNAME a39.
3.462 Standard query A ww (2895) (53) DNS: Standard query A www.lenovo.com
3.624 dzdaemon > http [SY (3866) (80) TCP: dzdaemon > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 W
3.728 http > dzdaemon [SY (3866) (80) TCP: http > dzdaemon [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 M
3.728 dzdaemon > http [AC (3866) (80) TCP: dzdaemon > http [ACK] Seq=1 Ack=1 Win=128480 Len=0
3.729 GET / HTTP/1.1 (3866) (80) HTTP: GET / HTTP/1.1
3.769 http > dzdaemon [AC (3866) (80) TCP: http > dzdaemon [ACK] Seq=1 Ack=580 Win=6948 Len=0
3.771 HTTP/1.0 301 Moved (3866) (80) HTTP: HTTP/1.0 301 Moved Permanently
3.772 GET /home/0,7340,L- (3866) (80) HTTP: GET /home/0,7340,L-8,00.html HTTP/1.1
3.965 [TCP Previous segme (3866) (80) TCP: [TCP Previous segment lost] [TCP segment of a reassembled PI
3.965 [TCP Dup ACK 12#1] (3866) (80) TCP: [TCP Dup ACK 12#1] dzdaemon > http [ACK] Seq=1204 Ack=1
3.966 [TCP segment of a r (3866) (80) TCP: [TCP segment of a reassembled PDU]
3.966 [TCP Dup ACK 12#2] (3866) (80) TCP: [TCP Dup ACK 12#2] dzdaemon > http [ACK] Seq=1204 Ack=1
3.968 [TCP segment of a r (3866) (80) TCP: [TCP segment of a reassembled PDU]
3.968 [TCP Dup ACK 12#3] (3866) (80) TCP: [TCP Dup ACK 12#3] dzdaemon > http [ACK] Seq=1204 Ack=1
3.968 Standard query resp (2895) (53) DNS: Standard query response CNAME www.lenovo.com.edgekey.net
3.990 [TCP segment of a r (3866) (80) TCP: [TCP segment of a reassembled PDU]
Save As
Close


Follow TCP Stream
Stream Content
GET /search?client=navclient-auto&ch=6174981939&freshness_check=4ilp-GrPqKEx_r_lNxaYw&iqrn=qr4&orig=OJ&ie=UTF-8&oe=UTF-8&features=Rank&q=info:http%3A%2F%2Fwww%2Eynet%2Eco%2Eil%2F HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; GoogleToolbar 2.0.114.9-big; Windows XP 5.1)
Host: toolbarqueries.google.com
Cache-Control: no-cache
Cookie: PREF=ID=1a18560743a17669:TB=2:CR=1:TM=1113765996:LM=1119978279:GM=1:S=7NmjkcGkIc845ngM; rememberme=false
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Date: Mon, 11 Jul 2005 08:21:03 GMT
Content-Type: text/html
Cache-Control: private
Server: GWS/2.1
Via: 1.1 cache1 (NetCache NetApp/5.5R2D5), Version 2.0-Build_Linux_1336 $Date: 04/13/2005 15:53:0038$(IWSS), 1.1 cache1 (NetCache NetApp/5.5R2D5)
e
Find
Save As
Print
Entire conversation (767 bytes)
ASCII
EBCDIC
Hex Dump
C Arrays
Raw
Help
Filter Out This Stream
Close


Snif2 --- HTTP Example.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: (tcp.stream eq 5)
Expression... Clear Apply
No. Time Source Destination Protocol Info
39 10.032397 10.114.30.180 64.233.183.99 TCP peport > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=
40 10.035182 64.233.183.99 10.114.30.180 TCP http > peport [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 M
41 10.035234 10.114.30.180 64.233.183.99 TCP peport > http [ACK] Seq=1 Ack=1 Win=131400 Len=0 TSV=1
42 10.035443 10.114.30.180 64.233.183.99 HTTP GET /search?client=navclient-auto&ch=6174981939&freshn
43 10.145399 64.233.183.99 10.114.30.180 TCP http > peport [ACK] Seq=1 Ack=447 Win=32768 Len=0 TSV=
44 10.312738 64.233.183.99 10.114.30.180 TCP [TCP segment of a reassembled PDU]
45 10.312814 64.233.183.99 10.114.30.180 HTTP HTTP/1.1 200 OK (text/html)
46 10.312862 10.114.30.180 64.233.183.99 TCP peport > http [ACK] Seq=447 Ack=322 Win=131076 Len=0 T
169 20.311539 64.233.183.99 10.114.30.180 TCP http > peport [FIN, ACK] Seq=322 Ack=447 Win=32768 Len
170 20.311629 10.114.30.180 64.233.183.99 TCP peport > http [ACK] Seq=447 Ack=323 Win=131076 Len=0 T
192 21.479689 10.114.30.180 64.233.183.99 TCP peport > http [FIN, ACK] Seq=447 Ack=323 Win=131076 Le
194 21.480926 64.233.183.99 10.114.30.180 TCP http > peport [ACK] Seq=323 Ack=448 Win=32768 Len=0 TS
Frame 39 (78 bytes on wire, 78 bytes captured)
Ethernet II, Src: Ibm_42:c2:4d (00:09:6b:42:c2:4d), Dst: LucentTe_cf:cd:2c (00:30:6d:cf:cd:2c)
Internet Protocol, Src: 10.114.30.180 (10.114.30.180), Dst: 64.233.183.99 (64.233.183.99)
Transmission Control Protocol, Src Port: peport (1449), Dst Port: http (80), Seq: 0, Len: 0
0010 00 40 1a 77 40 00 80 06 be ce 0a 72 1e b4 40 e9 .@.w@... ...r..@.
0020 b7 63 05 a9 00 50 f2 b3 c2 a2 00 00 00 00 b0 02 .c...P.. ....
0030 ff ff 59 3d 00 00 02 04 05 b4 01 03 03 02 01 01 ..Y=.... ....
0040 08 0a 00 00 00 00 00 00 00 00 01 01 04 02 ....
File: "D:\Customers\Examples\Snif2 --- HTTP Exa... Packets: 1648 Displayed: 12 Marked: 0 Profile: Default
RTP Example - SIP - 02.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
ANSI
Fax T38 Analysis...
GSM
H.225...
IAX2
ISUP Messages...
LTE MAC...
MTP3
RTP
SCTP
SIP...
SMPP Operations...
UCP Messages...
VoIP Calls
WAP-WSP...
Show All Streams
Stream Analysis...
Filter:
Clear Apply
No. Time Source Destination Protocol Info
996 18.486088 78.136.29.1 RTP PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33244, Time=298768161
997 18.503502 192.168.2.1 RTP PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18855, Time=163757234
998 18.517918 78.136.29.1 RTP PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33245, Time=298768185
999 18.532769 192.168.2.1 RTP PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18856, Time=163757258
1000 18.546201 78.136.29.1 RTP PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33246, Time=298768209
1001 18.563107 192.168.2.1 RTP PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18857, Time=163757282
1002 18.576173 78.136.29.1 RTP PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33247, Time=298768233
1003 18.593347 192.168.2.1 RTP PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18858, Time=163757306
1004 18.607151 78.136.29.1 RTP PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33248, Time=298768257
1005 18.622625 192.168.2.1 RTP PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18859, Time=163757330
1006 18.637287 78.136.29.1 RTP PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33249, Time=298768281
1007 18.652950 192.168.2.1 RTP PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18860, Time=163757354
1008 18.667106 78.136.29.109 192.168.2.100 RTP PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33250, Time=298768305
1009 18.689220 192.168.2.100 78.136.29.109 RTP PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18861, Time=163757378
1010 18.696818 78.136.29.109 192.168.2.100 RTP PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33251, Time=298768329
Frame 1001 (104 bytes on wire, 104 bytes captured)
Ethernet II, Src: IntelCor_a2:d8:9a (00:1c:bf:a2:d8:9a), Dst: EdimaxTe_6e:2f:7d (00:0e:2e:6e:2f:7d)
Internet Protocol, Src: 192.168.2.100 (192.168.2.100), Dst: 78.136.29.109 (78.136.29.109)
User Datagram Protocol, Src Port: avt-profile-1 (5004), Dst Port: x11 (6062)
Real-Time Transport Protocol


Wireshark: Capture Options
Capture
Interface: Local
Intel(R) PRO/Wireless 3945ABG Network Connection (Microsoft
IP address: 192.168.2.100
Link-layer header type: Ethernet
Wireless Settings
Capture packets in promiscuous mode
Remote Settings
Capture packets in pcap-ng format (experimental)
Buffer size: 1 megabyte(s)
Limit each packet to 1 bytes
Capture Filter:
Capture File(s)
File:
Browse...
Use multiple files
Next file every 1 megabyte(s)
Next file every 1 minute(s)
Ring buffer with 2 files
Stop capture after 1 file(s)
Display Options
Update list of packets in real time
Automatic scrolling in live capture
Hide capture info dialog
Name Resolution
Enable MAC name resolution
Enable network name resolution
Enable transport name resolution
Stop Capture ...
... after 1 packet(s)
... after 1 megabyte(s)
... after 1 minute(s)
Help
Start
Cancel


Wireshark: Capture Filter - Profile: Default
Edit
Capture Filter
New
Delete
Ethernet address 00:08:15:00:08:15
Ethernet type 0x0806 (ARP)
No Broadcast and no Multicast
No ARP
IP only
IP address 192.168.0.1
IPX only
TCP only
UDP only
TCP or UDP port 80 (HTTP)
HTTP TCP port (80)
No ARP and no DNS
Properties
Filter name:
Filter string:
Help
OK
Cancel



Examples:

- Capture only traffic to or from IP address 172.18.5.4
  - **host 172.18.5.4**
- Capture traffic to or from a range of IP addresses
  - **net 192.168.0.0/24**
  - **net 192.168.0.0 mask 255.255.255.0**
- Capture traffic from a range of IP addresses
  - **srcnet 192.168.0.0/24**

- **srcnet 192.168.0.0 mask 255.255.255.0**

■Capture traffic to a range of IP addresses

- **dstnet 192.168.0.0/24**
- **dstnet 192.168.0.0 mask 255.255.255.0**

■Capture only DNS (port 53) traffic

- port 53

■Capture non-HTTP and non-SMTP traffic on your server

- **host www.example.com and not (port 80 or port 25)**
- **host www.example.com and not port 80 and not port 25**

Capture except all ARP and DNS traffic

- **port not 53 and not arp**

■Capture traffic within a range of ports

- **(tcp[2:2] > 1500 and tcp[2:2] < 1550) or (tcp[4:2] > 1500 and tcp[4:2] < 1550)**
- **tcpportrange1501-1549**

■Capture only Ethernet type EAPOL

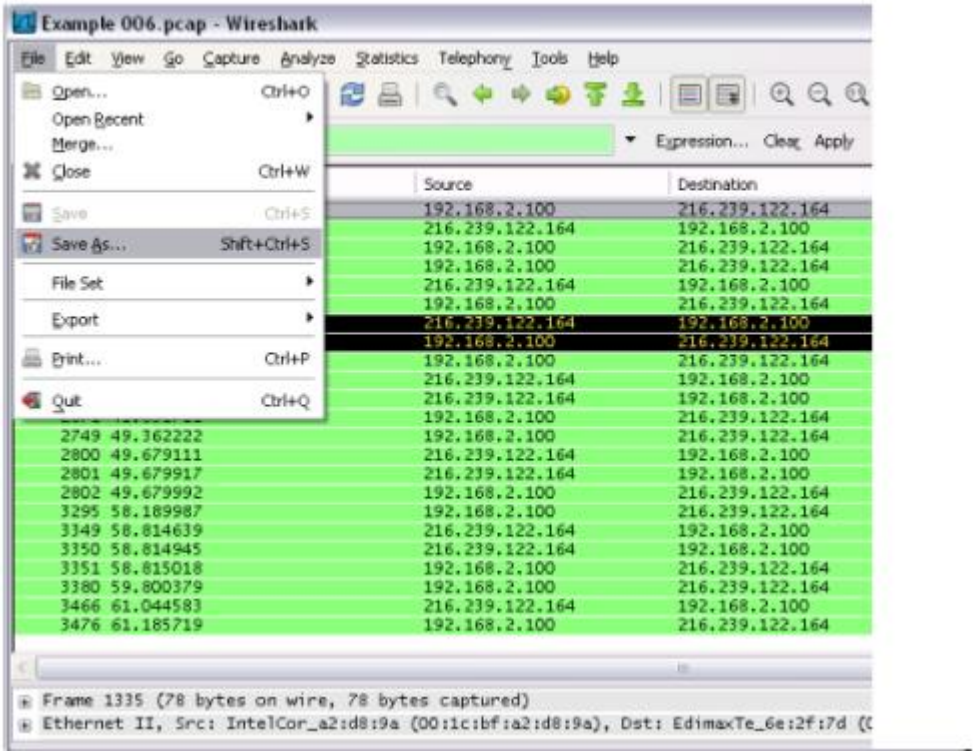
- ether proto 0x888e

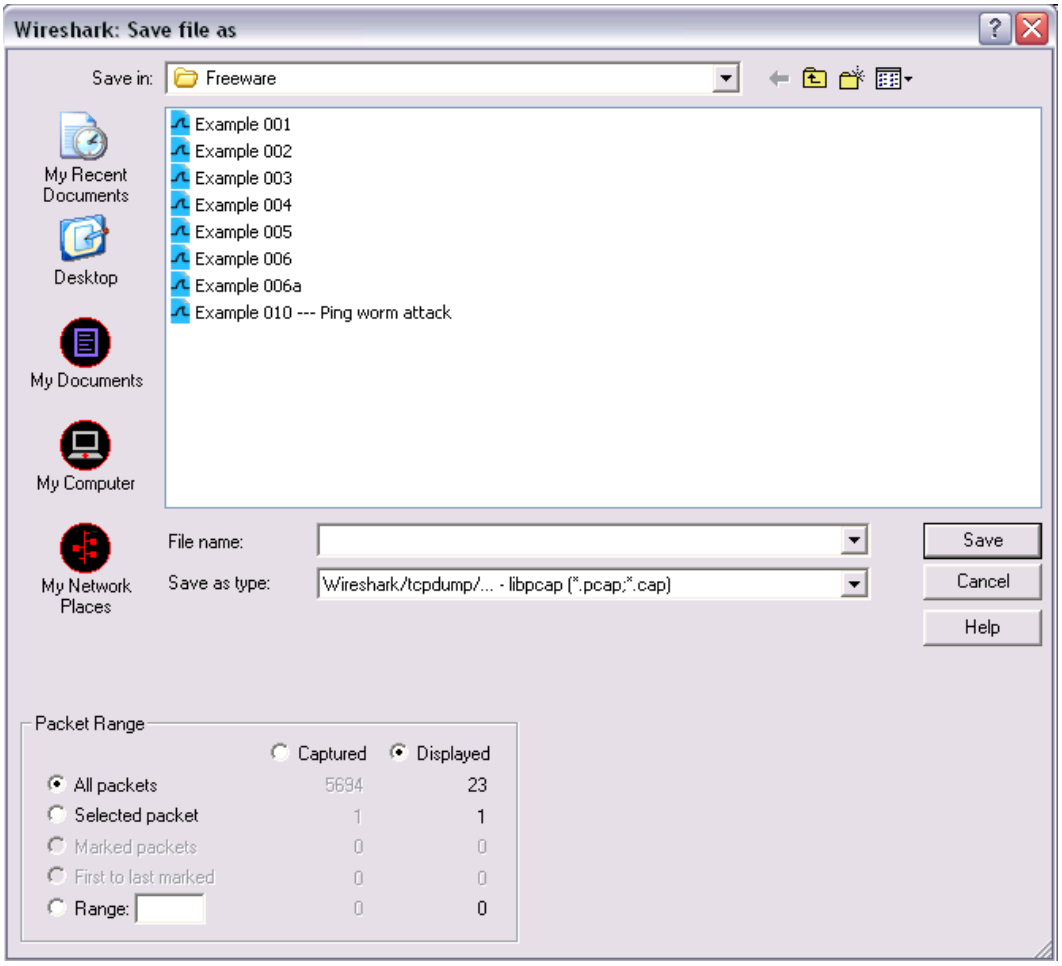
■Capture only IP traffic

(the shortest filter, but sometimes very useful to get rid of lower layer protocolslike ARP and STP)

- ip

■Capture only unicast traffic


Example 006.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Open... Ctrl+O
Open Recent
Merge...
Close Ctrl+W
Save Ctrl+S
Save As... Shift+Ctrl+S
File Set
Export
Print... Ctrl+P
Quit Ctrl+Q
Expression... Clear Apply
Source
Destination
192.168.2.100 216.239.122.164
216.239.122.164 192.168.2.100
2749 49.362222
2800 49.679111
2801 49.679917
2802 49.679992
3295 58.189987
3349 58.814639
3350 58.814945
3351 58.815018
3380 59.800379
3466 61.044583
3476 61.185719
Frame 1335 (78 bytes on wire, 78 bytes captured)
Ethernet II, Src: IntelCor_a2:d8:9a (00:1c:bf:a2:d8:9a), Dst: EdimaxTe_6e:2f:7d


Wireshark: Save file as
Save in: Freeware
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
Example 001
Example 002
Example 003
Example 004
Example 005
Example 006
Example 006a
Example 010 --- Ping worm attack
File name:
Save as type: Wireshark/tcpdump/... - libpcap (*.pcap;*.cap)
Save
Cancel
Help
Packet Range
Captured
Displayed
All packets 5694 23
Selected packet 1 1
Marked packets 0 0
First to last marked 0 0
Range: 0 0

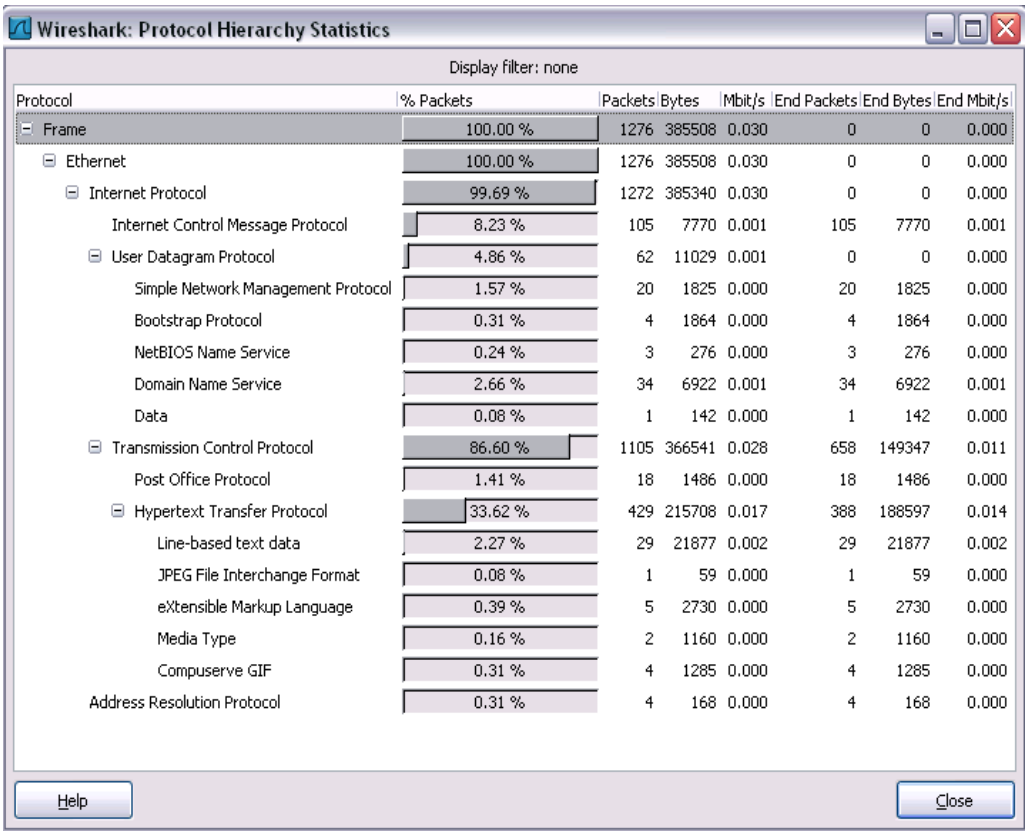
No.	Time	Time Variation	Source	Destination	Protocol	Info
1	0	0	192.168.2.100	216.239.122.164	TCP	27837 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1 TSV=0 TSER=0
2	0.226724	0.226724	216.239.122.164	192.168.2.100	TCP	http > 27837 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1380
3	0.226772	4.8E-05	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.227146	0.227098	192.168.2.100	216.239.122.164	HTTP	GET /\/b.jpg HTTP/1.1
5	0.700674	0.473576	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
6	0.883533	0.409957	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=649 Ack=767 Win=64769 Len=0
7	1.161312	0.751355	216.239.122.164	192.168.2.100	HTTP	[TCP Retransmission] HTTP/1.1 200 OK (JPEG JFIF image)
8	1.161361	0.410006	192.168.2.100	216.239.122.164	TCP	[TCP Dup ACK 6#1] 27837 > http [ACK] Seq=649 Ack=767 Win=64769 Len=0
9	16.211468	15.801462	192.168.2.100	216.239.122.164	HTTP	GET /\/b.jpg HTTP/1.1
10	16.452024	0.650562	216.239.122.164	192.168.2.100	TCP	[TCP segment of a reassembled PDU]
11	16.452343	15.801781	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
12	16.452417	0.650636	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=1539 Ack=1533 Win=65535 Len=0
13	24.122928	23.472292	192.168.2.100	216.239.122.164	HTTP	GET /\/b.jpg HTTP/1.1
14	24.439817	0.967525	216.239.122.164	192.168.2.100	TCP	[TCP segment of a reassembled PDU]
15	24.440623	23.473098	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
16	24.440698	0.9676	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=2384 Ack=2299 Win=64769 Len=0
17	32.950693	31.983093	192.168.2.100	216.239.122.164	HTTP	GET /\/b.jpg HTTP/1.1
18	33.575345	1.592252	216.239.122.164	192.168.2.100	TCP	[TCP segment of a reassembled PDU]
19	33.575651	31.983399	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
20	33.575724	1.592325	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=3269 Ack=3065 Win=65535 Len=0
21	34.561085	32.96876	192.168.2.100	216.239.122.164	HTTP	GET /b.gif HTTP/1.1
22	35.805269	2.836529	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (GIF89a)
23	35.946425	33.109896	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=4080 Ack=3567 Win=66033 Len=0

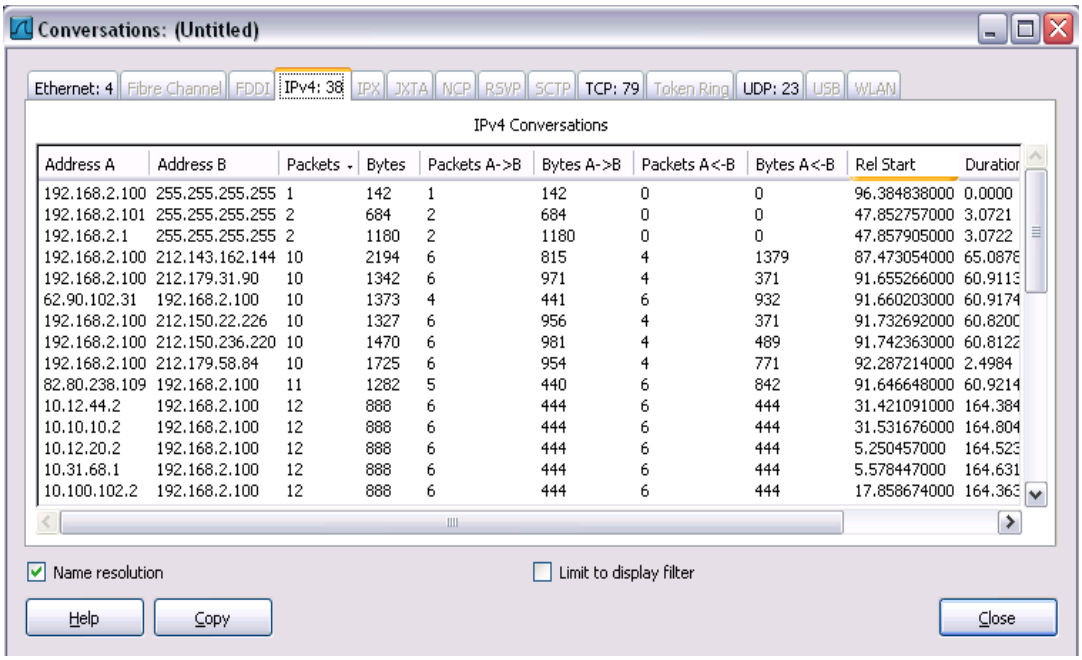
MCAL 26 Skill based Lab course Networking with Linux Lab

Practical No. 16

Aim: - Analyze the performance parameters of network using Wire Shark

Output: -





Name: Mansingh Yadav **Roll No. 61**

	A	B	C	D	E	F	G	H	I	J	K	L
1	Address A	Address B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration	bps A->B	bps A<-B
2	10.10.10.1	10.159.3.103	2	124	0	0	2	124	0	42.0039	N/A	23.62
3	1.1.1.1	10.159.3.103	2	120	0	0	2	120	24.49414	1.1885	N/A	807.76
4	1.1.1.1	127.0.0.1	4	248	4	248	0	0	10.713867	14.9814	132.43	N/A
5	10.159.3.103	212.179.1.202	491	458158	185	10643	306	447515	23.216796	15.4082	5525.89	232351.54