NETWORKING MINI PROJECT

Roll No: C22019 **Name:** Rushikesh Chapke

Write a program to demonstrate Hybrid Topology in ns3

- Hybrid topology refers to a network configuration that combines multiple types of network topologies. It is a combination of two or more basic network topologies, such as star, ring, bus, mesh, or tree topologies. The aim of using a hybrid topology is to leverage the advantages of different topologies and create a network that suits specific requirements.
- In a hybrid topology, different sections of the network may be connected using different topologies, resulting in a more flexible and scalable network infrastructure. For example, a network might have a star topology in one area, a ring topology in another, and a bus topology connecting these sections. This allows for efficient use of resources, better fault tolerance, improved performance, and increased network capacity.
- Some of the benefits of a hybrid topology include:
- Scalability: Hybrid topologies can be easily expanded by adding or modifying specific sections of the network without affecting the entire infrastructure. This flexibility allows for growth and adaptation to changing network needs.
- Fault tolerance: By combining different topologies, a hybrid network can provide redundancy and fault tolerance. If one section of the network fails, the remaining sections can continue to function, minimizing downtime and improving reliability.
- Performance optimization: Different topologies excel in different aspects. By combining them strategically, a hybrid topology can optimize network performance. For example, a star topology can provide centralized management and easy connectivity, while a mesh topology can offer robust communication between nodes.
- Segmentation: Hybrid topologies allow for network segmentation, enabling specific sections of the network to have their own topologies suited to their requirements. This can enhance security, isolate network issues, and improve overall network management.

- Hybrid topologies are commonly used in large-scale networks, such as enterprise networks, data centers, and telecommunications infrastructure.

They provide a versatile and adaptable network architecture that can meet diverse needs and support various applications and services.

Code:

```
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/ssid.h"
#include "ns3/netanim-module.h"
#include "ns3/mobility-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("ThirdScriptExample");

int
main (int argc, char *argv[])
{
 bool verbose = true;
 uint32_t nCsma = 3;
 uint32_t nWifi = 3;
 bool tracing = false;

 CommandLine cmd (__FILE__);
 cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);
 cmd.AddValue ("nWifi", "Number of wifi STA devices", nWifi);
 cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);
 cmd.AddValue ("tracing", "Enable pcap tracing", tracing);

 cmd.Parse (argc,argv);

 // The underlying restriction of 18 is due to the grid position
 // allocator's configuration; the grid layout will exceed the
 // bounding box if more than 18 nodes are provided.
 if (nWifi > 18)
   {
    std::cout << "nWifi should be 18 or less; otherwise grid layout exceeds the bounding box" <<
std::endl;
```

```
    return 1;
  }

if (verbose)
  {
    LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
    LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
  }

NodeContainer p2pNodes;
p2pNodes.Create (2);

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 (nCsma);

CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

NetDeviceContainer csmaDevices;
csmaDevices = csma.Install (csmaNodes);

NodeContainer wifiStaNodes;
wifiStaNodes.Create (nWifi);
NodeContainer wifiApNode = p2pNodes.Get (0);

YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
YansWifiPhyHelper phy;
phy.SetChannel (channel.Create ());

WifiHelper wifi;
wifi.SetRemoteStationManager ("ns3::AarfWifiManager");

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);

MobilityHelper mobility;

mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                   "MinX", DoubleValue (0.0),
                   "MinY", DoubleValue (0.0),
                   "DeltaX", DoubleValue (5.0),
                   "DeltaY", DoubleValue (10.0),
                   "GridWidth", UintegerValue (3),
                   "LayoutType", StringValue ("RowFirst"));

mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                 "Bounds", RectangleValue (Rectangle (-50, 50, -50, 50)));
mobility.Install (wifiStaNodes);

mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (wifiApNode);

InternetStackHelper stack;
stack.Install (csmaNodes);
stack.Install (wifiApNode);
stack.Install (wifiStaNodes);

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");
address.Assign (staDevices);
address.Assign (apDevices);

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

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

ApplicationContainer clientApps =
  echoClient.Install (wifiStaNodes.Get (nWifi - 1));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));


mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(csmaNodes);

AnimationInterface anim("ram.xml");
AnimationInterface::SetConstantPosition (csmaNodes.Get(0),10,25);
AnimationInterface::SetConstantPosition (csmaNodes.Get(1),40,25);
anim.EnablePacketMetadata(true);

PcapHelper pcapHelper;
Ptr<PcapFileWrapper> file = pcapHelper.CreateFile ("ram.pcap", std::ios::out, PcapHelper::DLT_PPP);


Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

Simulator::Stop (Seconds (10.0));

if (tracing)
  {
    phy.SetPcapDataLinkType (WifiPhyHelper::DLT_IEEE802_11_RADIO);
    pointToPoint.EnablePcapAll ("ram");
    phy.EnablePcap ("ram", apDevices.Get (0));
    csma.EnablePcap ("ram", csmaDevices.Get (0), true);
  }
```

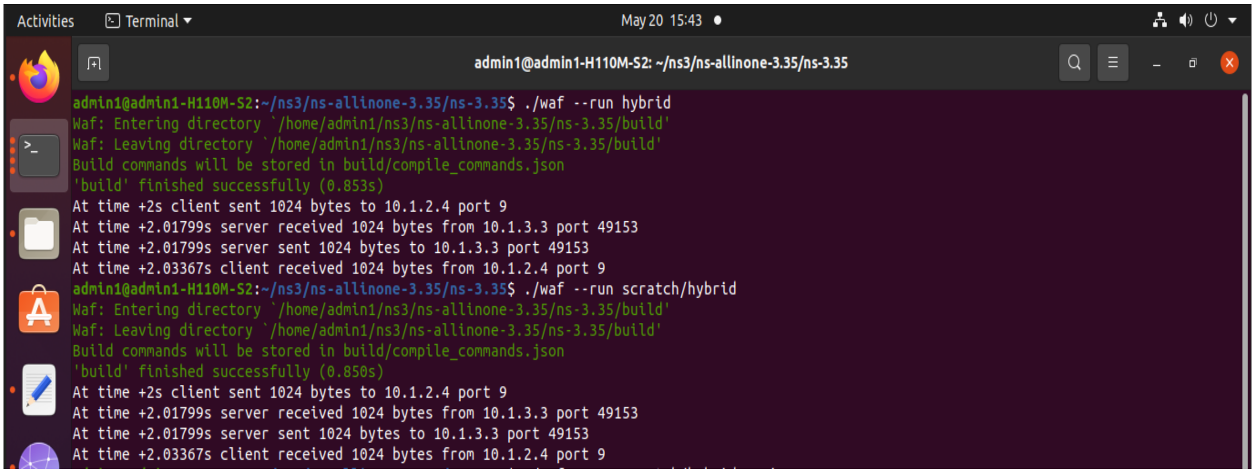
```
 Simulator::Run ();
 Simulator::Destroy ();
 return 0;
}
```

Once the code has been saved the file needs to be build using the command:

“./waf –run filename”

After successful build the file needs to be run using the following command:

“./waf –run scratch/filename”



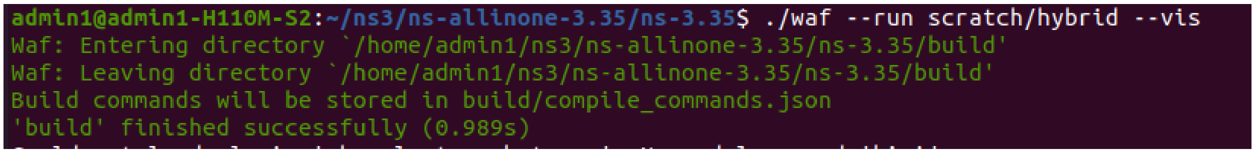
```
admin1@admin1-H110M-S2:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run hybrid
Waf: Entering directory `/home/admin1/ns3/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/admin1/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.853s)
At time +2s client sent 1024 bytes to 10.1.2.4 port 9
At time +2.01799s server received 1024 bytes from 10.1.3.3 port 49153
At time +2.01799s server sent 1024 bytes to 10.1.3.3 port 49153
At time +2.03367s client received 1024 bytes from 10.1.2.4 port 9
admin1@admin1-H110M-S2:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/hybrid
Waf: Entering directory `/home/admin1/ns3/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/admin1/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.850s)
At time +2s client sent 1024 bytes to 10.1.2.4 port 9
At time +2.01799s server received 1024 bytes from 10.1.3.3 port 49153
At time +2.01799s server sent 1024 bytes to 10.1.3.3 port 49153
At time +2.03367s client received 1024 bytes from 10.1.2.4 port 9
```

In order to activate the pyvis the following command needs to be entered

“./waf –run scratch/filename --vis”

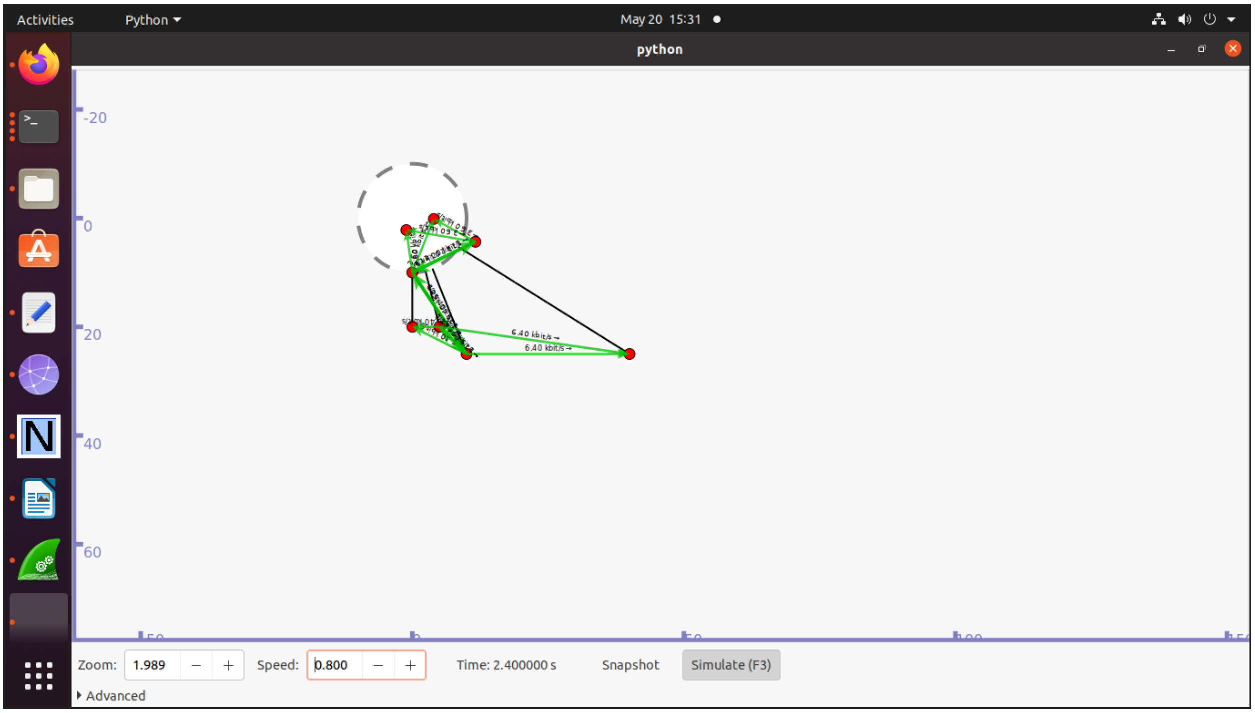
NETWORKING MINI PROJECT

Roll No: C22019 **Name:** Rushikesh Chapke



```
admin1@admin1-H110M-S2:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/hybrid --vis
Waf: Entering directory `/home/admin1/ns3/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/admin1/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.989s)
```

The following screen will be visible



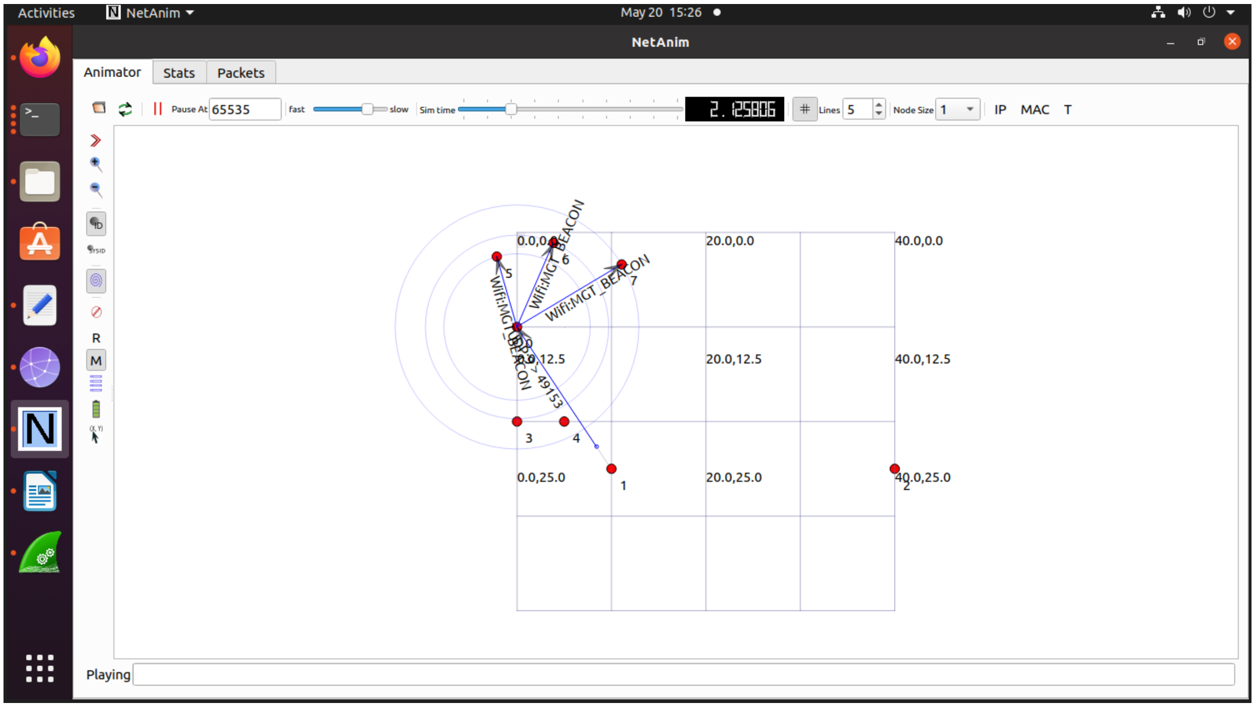
Zoom: 1.989 Speed: 0.800 Time: 2.400000 s

In order to activate the pyvis the following command needs to be entered

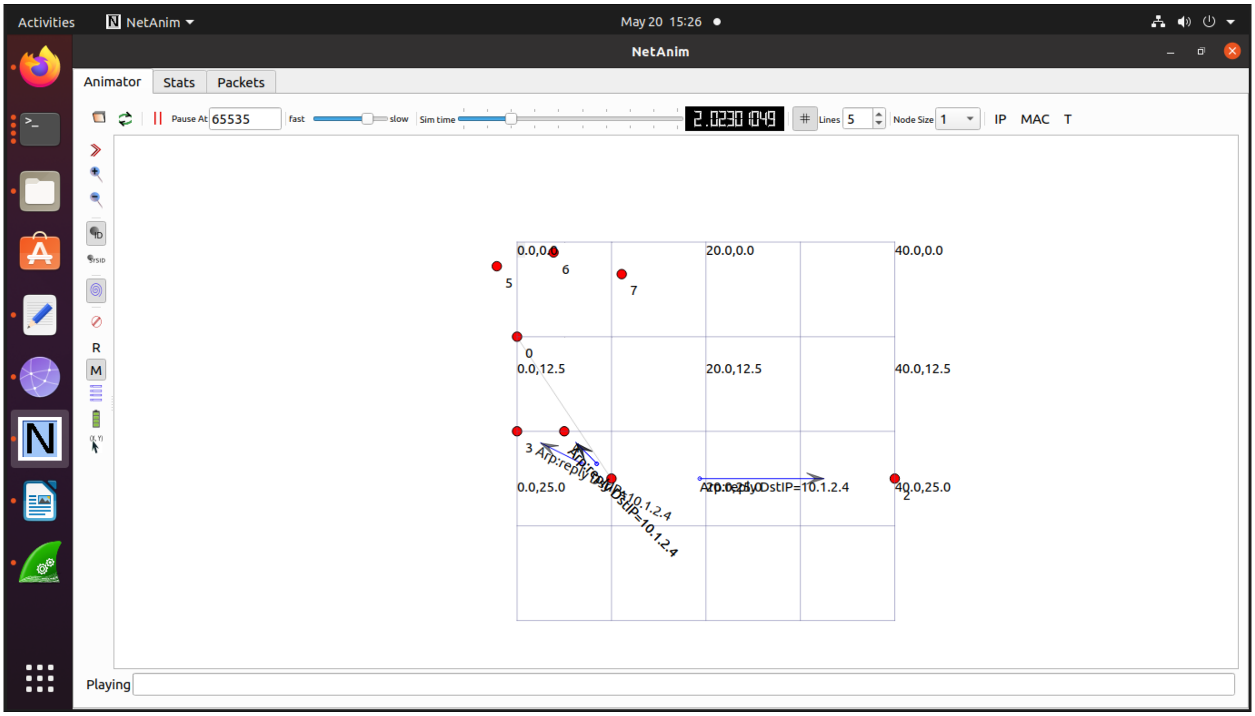
“./NetAnim”

NETWORKING MINI PROJECT

Roll No: C22019 **Name:** Rushikesh Chapke



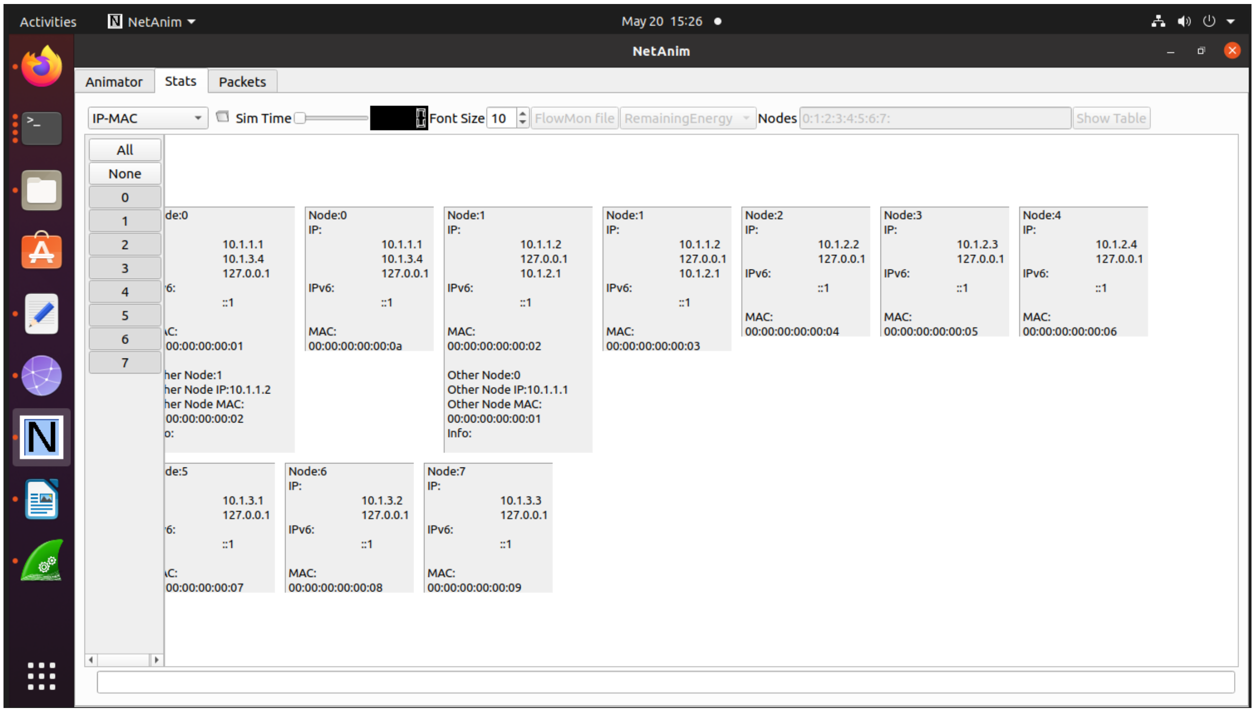
Pause At 65535 Lines 5 Node Size 1



Pause At 65535 Lines 5 Node Size 1

NETWORKING MINI PROJECT

Roll No: C22019 **Name:** Rushikesh Chapke



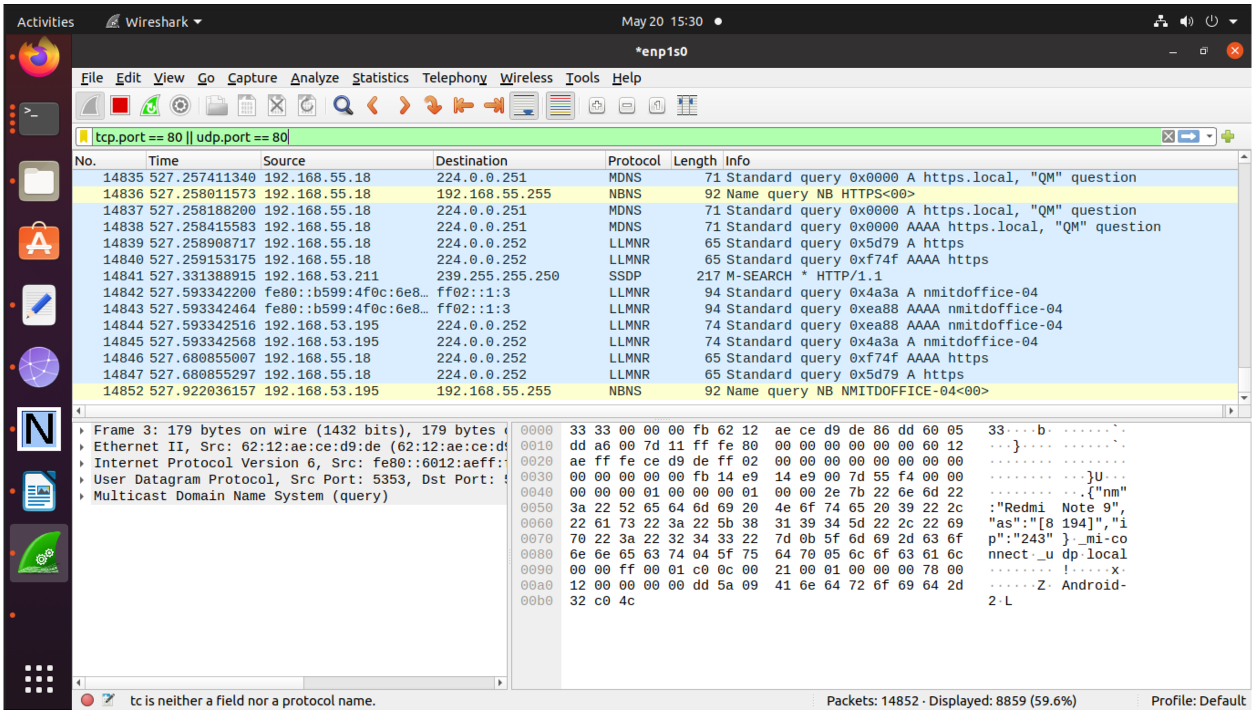
IP-MAC □ Sim Time Font Size 10 Nodes 0:1:2:3:4:5:6:7:

```
de:0                 Node:0               Node:1               Node:1               Node:2               Node:3               Node:4
                     IP:                  IP:                  IP:                  IP:                  IP:                  IP:
      10.1.1.1             10.1.1.1             10.1.1.2             10.1.1.2             10.1.2.2             10.1.2.3             10.1.2.4
      10.1.3.4             10.1.3.4             127.0.0.1            127.0.0.1            127.0.0.1            127.0.0.1            127.0.0.1
      127.0.0.1            127.0.0.1            10.1.2.1             10.1.2.1             IPv6:                IPv6:                IPv6:
6:                   IPv6:                IPv6:                IPv6:                      ::1                  ::1                  ::1
      ::1                  ::1                  ::1                  ::1
                                                                                        MAC:                 MAC:                 MAC:
C:                   MAC:                 MAC:                 MAC:                 00:00:00:00:00:04    00:00:00:00:00:05    00:00:00:00:00:06
00:00:00:00:01       00:00:00:00:00:0a    00:00:00:00:00:02    00:00:00:00:00:03

her Node:1                                Other Node:0
her Node IP:10.1.1.2                      Other Node IP:10.1.1.1
her Node MAC:                             Other Node MAC:
00:00:00:00:02                            00:00:00:00:00:01
o:                                        Info:

de:5                 Node:6               Node:7
                     IP:                  IP:
      10.1.3.1             10.1.3.2             10.1.3.3
      127.0.0.1            127.0.0.1            127.0.0.1
6:                   IPv6:                IPv6:
      ::1                  ::1                  ::1

C:                   MAC:                 MAC:
00:00:00:00:07       00:00:00:00:00:08    00:00:00:00:00:09
```



tcp.port == 80 || udp.port == 80

```
No.   Time          Source                 Destination       Protocol Length Info
14835 527.257411340 192.168.55.18          224.0.0.251       MDNS       71 Standard query 0x0000 A https.local, "QM" question
14836 527.258011573 192.168.55.18          192.168.55.255    NBNS       92 Name query NB HTTPS<00>
14837 527.258188200 192.168.55.18          224.0.0.251       MDNS       71 Standard query 0x0000 A https.local, "QM" question
14838 527.258415583 192.168.55.18          224.0.0.251       MDNS       71 Standard query 0x0000 AAAA https.local, "QM" question
14839 527.258908717 192.168.55.18          224.0.0.252       LLMNR      65 Standard query 0x5d79 A https
14840 527.259153175 192.168.55.18          224.0.0.252       LLMNR      65 Standard query 0xf74f AAAA https
14841 527.331388915 192.168.53.211         239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
14842 527.593342200 fe80::b599:4f0c:6e8…   ff02::1:3         LLMNR      94 Standard query 0x4a3a A nmitdoffice-04
14843 527.593342464 fe80::b599:4f0c:6e8…   ff02::1:3         LLMNR      94 Standard query 0xea88 AAAA nmitdoffice-04
14844 527.593342516 192.168.53.195         224.0.0.252       LLMNR      74 Standard query 0xea88 AAAA nmitdoffice-04
14845 527.593342568 192.168.53.195         224.0.0.252       LLMNR      74 Standard query 0x4a3a A nmitdoffice-04
14846 527.680855007 192.168.55.18          224.0.0.252       LLMNR      65 Standard query 0xf74f AAAA https
14847 527.680855297 192.168.55.18          224.0.0.252       LLMNR      65 Standard query 0x5d79 A https
14852 527.922036157 192.168.53.195         192.168.55.255    NBNS       92 Name query NB NMITDOFFICE-04<00>
```

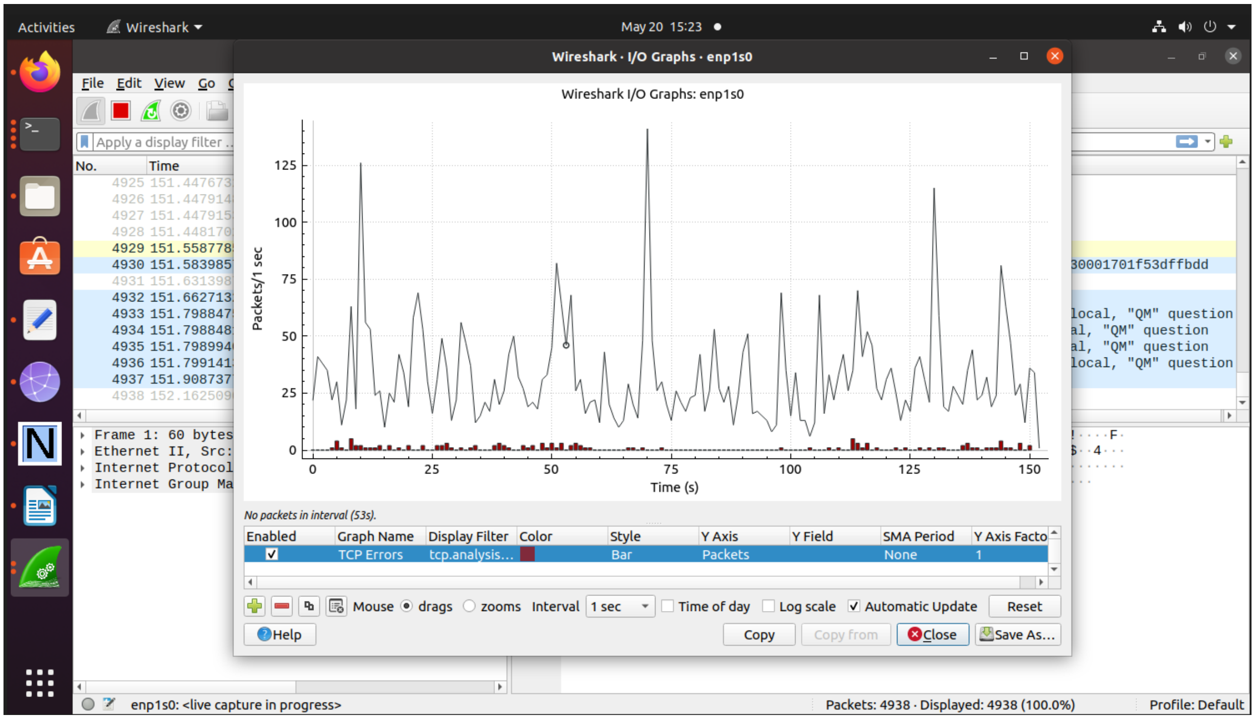
```
▸ Frame 3: 179 bytes on wire (1432 bits), 179 bytes (
▸ Ethernet II, Src: 62:12:ae:ce:d9:de (62:12:ae:ce:d9
▸ Internet Protocol Version 6, Src: fe80::6012:aeff:1
▸ User Datagram Protocol, Src Port: 5353, Dst Port: 5
▸ Multicast Domain Name System (query)
```

```
0000  33 33 00 00 00 fb 62 12  ae ce d9 de 86 dd 60 05   33····b· ······`·
0010  dd a6 00 7d 11 ff fe 80  00 00 00 00 00 00 60 12   ···}···· ······`·
0020  ae ff fe ce d9 de ff 02  00 00 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 fb 14 e9  14 e9 00 7d 55 f4 00 00   ········ ···}U···
0040  00 00 00 01 00 00 00 01  00 00 2e 7b 22 6e 6d 22   ········ ···.{"nm"
0050  3a 22 52 65 64 6d 69 20  4e 6f 74 65 20 39 22 2c   :"Redmi  Note 9",
0060  22 61 73 22 3a 22 5b 38  31 39 34 5d 22 2c 22 69   "as":"[8 194]","i
0070  70 22 3a 22 32 34 33 22  7d 0b 5f 6d 69 2d 63 6f   p":"243" }·_mi-co
0080  6e 6e 65 63 74 04 5f 75  64 70 05 6c 6f 63 61 6c   nnect·_u dp·local
0090  00 00 ff 00 01 c0 0c 00  21 00 01 00 00 00 78 00   ········ !·····x·
00a0  12 00 00 00 00 dd 5a 09  41 6e 64 72 6f 69 64 2d   ······Z· Android-
00b0  32 c0 4c                                           2·L
```

tc is neither a field nor a protocol name. Packets: 14852 · Displayed: 8859 (59.6%) Profile: Default

NETWORKING MINI PROJECT

Roll No: C22019 **Name:** Rushikesh Chapke



```
No.  Time
4925 151.447673
4926 151.447914
4927 151.447915
4928 151.448170
4929 151.558778
4930 151.583985
4931 151.631398
4932 151.662713
4933 151.798847
4934 151.798848
4935 151.798994
4936 151.799141
4937 151.908737
4938 152.162509

▸ Frame 1: 60 bytes
▸ Ethernet II, Src:
▸ Internet Protocol
▸ Internet Group Ma
```

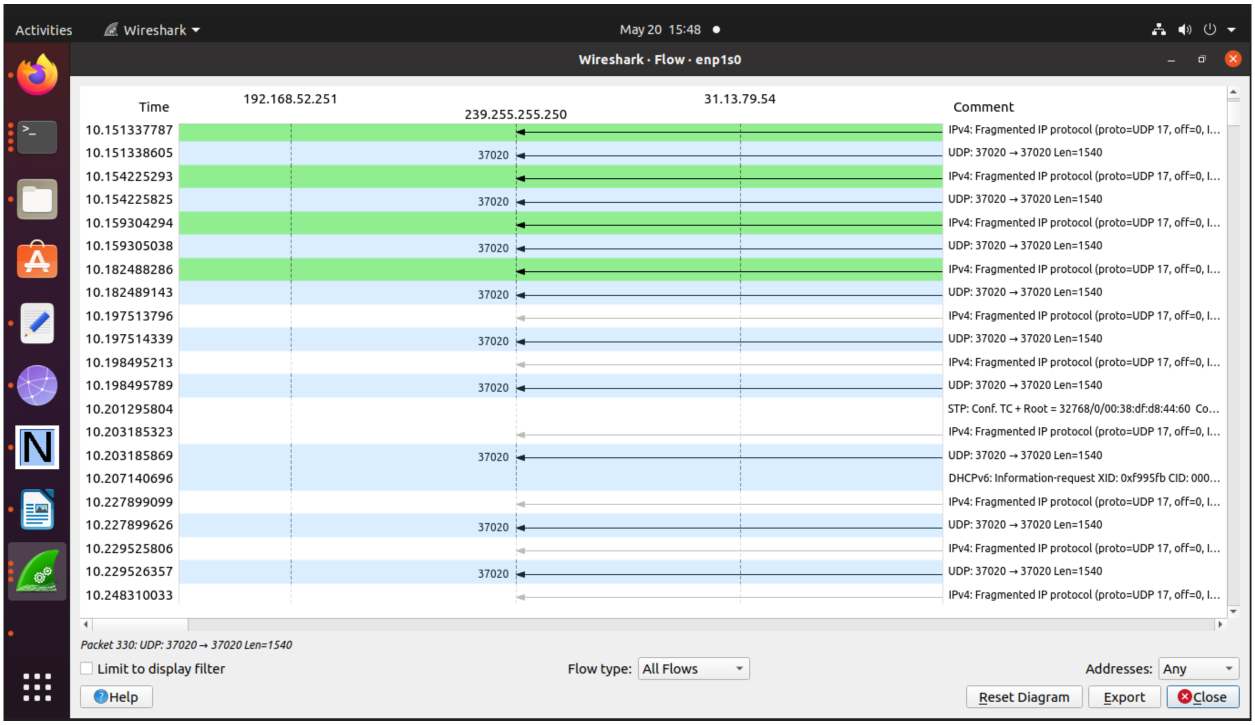
```
30001701f53dffbdd
local, "QM" question
al, "QM" question
al, "QM" question
local, "QM" question
```

No packets in interval (53s).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period	Y Axis Facto
☑	TCP Errors	tcp.analysis…	🟥	Bar	Packets		None	1

Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale ☑ Automatic Update

enp1s0: <live capture in progress> Packets: 4938 · Displayed: 4938 (100.0%) Profile: Default



Time	192.168.52.251	239.255.255.250	31.13.79.54	Comment
10.151337787				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.151338605		37020		UDP: 37020 → 37020 Len=1540
10.154225293				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.154225825		37020		UDP: 37020 → 37020 Len=1540
10.159304294				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.159305038		37020		UDP: 37020 → 37020 Len=1540
10.182488286				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.182489143		37020		UDP: 37020 → 37020 Len=1540
10.197513796				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.197514339		37020		UDP: 37020 → 37020 Len=1540
10.198495213				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.198495789		37020		UDP: 37020 → 37020 Len=1540
10.201295804				STP: Conf. TC + Root = 32768/0/00:38:df:d8:44:60 Co...
10.203185323				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.203185869		37020		UDP: 37020 → 37020 Len=1540
10.207140696				DHCPv6: Information-request XID: 0xf995fb CID: 000...
10.227899099				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.227899626		37020		UDP: 37020 → 37020 Len=1540
10.229525806				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...
10.229526357		37020		UDP: 37020 → 37020 Len=1540
10.248310033				IPv4: Fragmented IP protocol (proto=UDP 17, off=0, I...

Packet 330: UDP: 37020 → 37020 Len=1540

□ Limit to display filter Flow type: All Flows Addresses: Any