AIM: PROGRAM TO SIMULATE DHCP SERVER AND N CLIENTS.

THEORY:

Why use DHCP?

Every device on a TCP/IP-based network must have a unique unicast IP address to access the network and its resources. Without DHCP, IP addresses for new computers or computers that are moved from one subnet to another must be configured manually; IP addresses for computers that are removed from the network must be manually reclaimed.

With DHCP, this entire process is automated and managed centrally. The DHCP server maintains a pool of IP addresses and leases an address to any DHCP-enabled client when it starts up on the network. Because the IP addresses are dynamic (leased) rather than static (permanently assigned), addresses no longer in use are automatically returned to the pool for reallocation.

The network administrator establishes DHCP servers that maintain TCP/IP configuration information and provide address configuration to DHCP-enabled clients in the form of a lease offer. The DHCP server stores the configuration information in a database that includes:

- Valid TCP/IP configuration parameters for all clients on the network.
- Valid IP addresses, maintained in a pool for assignment to clients, as well as excluded addresses.
- Reserved IP addresses associated with particular DHCP clients. This allows consistent assignment of a single IP address to a single DHCP client.
- The lease duration, or the length of time for which the IP address can be used before a lease renewal is required.

A DHCP-enabled client, upon accepting a lease offer, receives:

- A valid IP address for the subnet to which it is connecting.
- Requested DHCP options, which are additional parameters that a DHCP server is configured to assign to clients. Some examples of DHCP options are Router (default gateway), DNS Servers, and DNS Domain Name.

Benefits of DHCP:

DHCP provides the following benefits.

- **Reliable IP address configuration**: DHCP minimizes configuration errors caused by manual IP address configuration, such as typographical errors, or address conflicts caused by the assignment of an IP address to more than one computer at the same time.
- **Reduced network administration**: DHCP includes the following features to reduce network administration:
- Centralized and automated TCP/IP configuration.
- The ability to define TCP/IP configurations from a central location.
- The ability to assign a full range of additional TCP/IP configuration values by means of DHCP options.

- The efficient handling of IP address changes for clients that must be updated frequently, such as those for portable devices that move to different locations on a wireless network.
- The forwarding of initial DHCP messages by using a DHCP relay agent, which eliminates the need for a DHCP server on every subnet.

DHCP Server:

A DHCP Server is a network server that automatically provides and assigns IP addresses, default gateways and other network parameters to client devices. It relies on the standard protocol known as Dynamic Host Configuration Protocol or DHCP to respond to broadcast queries by clients.

A DHCP server automatically sends the required network parameters for clients to properly communicate on the network. Without it, the network administrator has to manually set up every client that joins the network, which can be cumbersome, especially in large networks. DHCP servers usually assign each client with a unique dynamic IP address, which changes when the client's lease for that IP address has expired.

DHCP Client:

The DHCP protocol has two functions with regard to the client. It delivers sufficient information to clients for them to establish an endpoint for network communications, and it supplies other parameters needed by system- and application-level software.

To perform the first function, the DHCP protocol supplies an IP address valid for the network attached to the client's hardware interface. The right to use this IP address is given to the client for a finite period of time, called a lease. This differs from the traditional static configuration. If the client wants to use the IP address for a period of time longer than the original lease, it must periodically negotiate a lease extension with the server through DHCP. When the client no longer needs the IP address, the user of the machine can relinquish its lease, returning it to the pool of available IP addresses. Otherwise, the IP address is reclaimed automatically when its lease expires.

The implementation of the client side of the DHCP protocol on Solaris must meet several criteria. Bootstrapping a Sun workstation is a complex process because of the number and diversity of services that must be configured and invoked. Any DHCP solution must coexist with other methods already in use, particularly the Reverse Address Resolution Protocol (RARP) and static configuration. It must know that the superuser can change the address of a network interface after the workstation has booted. It must be able to configure multiple interfaces. And it must respond to human control and be able to report on the status and provide the statistics of the protocol.

<u>**SOURCE CODE:**</u>

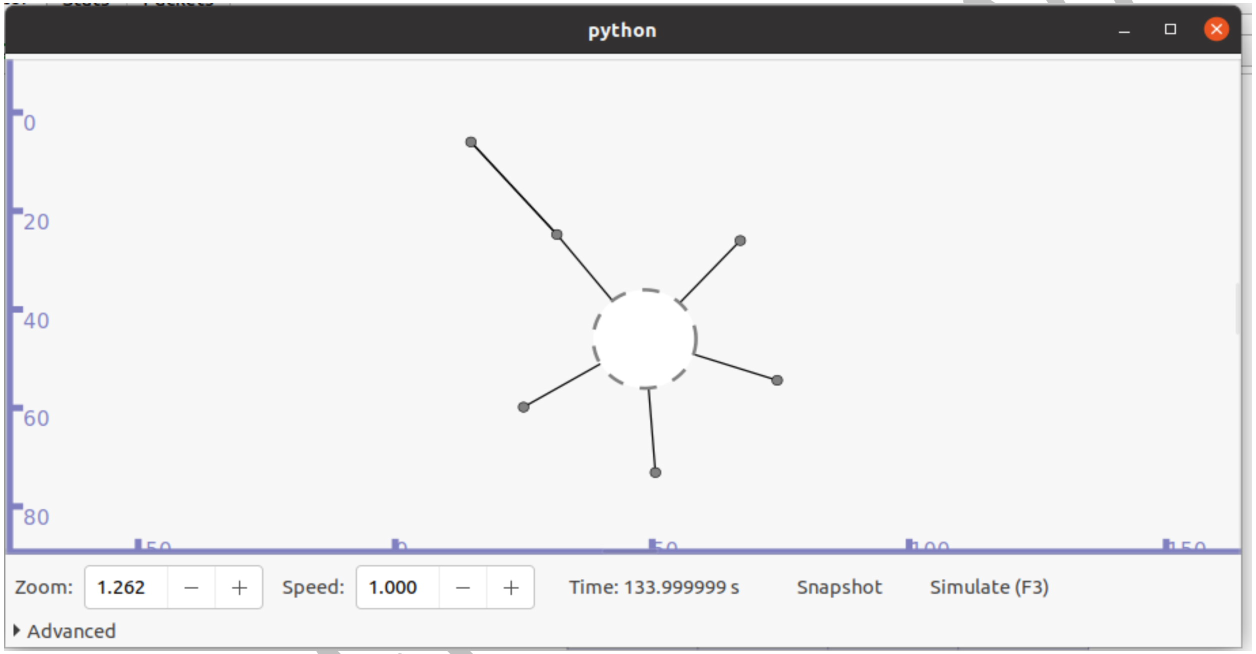
```
/*
*               Network layout:
 *
```

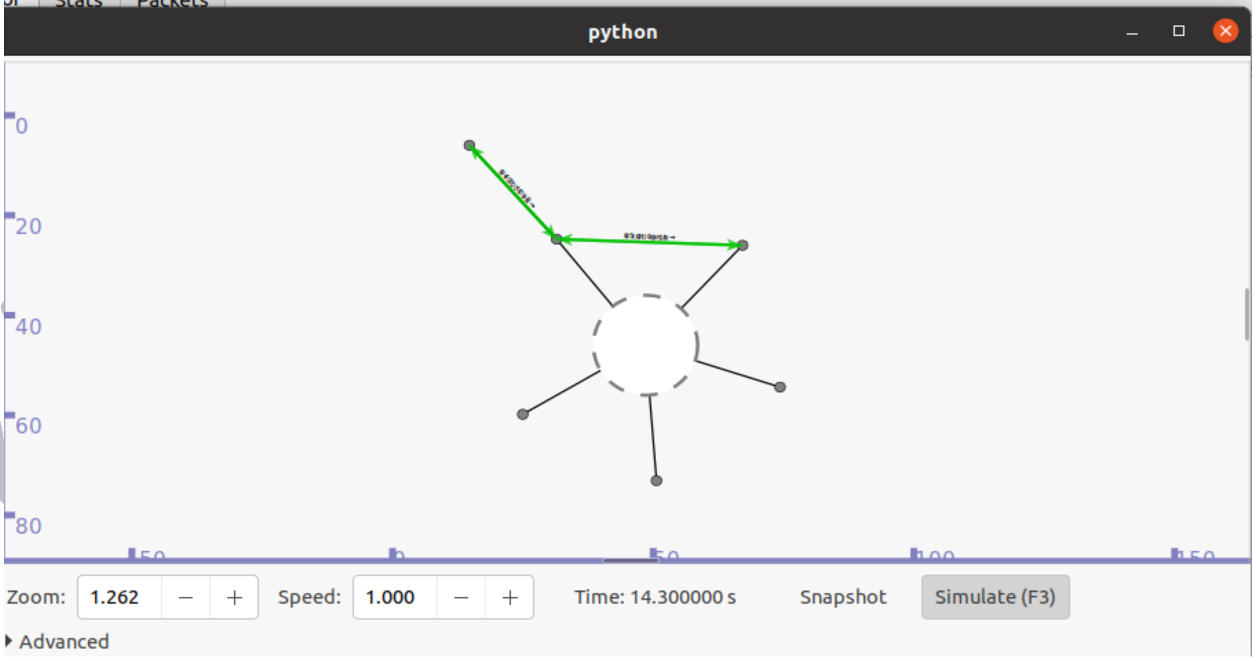
```
*               R0 is a DHCP server. The DHCP server announced R1 as the default
                router.
*               Nodes N1 will send UDP Echo packets to node A.
 *
 *
*               ┌-------------------------------------------------┐
*               │ DHCP Clients                                    │
*               │                                  172.30.0.14    │
*               │                                  DHCP static    │
*               │   ┌──────┐       ┌──────┐       ┌──────┐        │
*               │   │  N0  │       │  N1  │       │  N2  │        │
                ┌──────┐
*               │   └──────┘       └──────┘       └──────┘        │       ┌────│
                A   │
*               │      │              │              │          │       │
                └──────┘
*               └-------│--------------│---------------│---------┘       │
                172.30.1.2
*               DHCP Server           │              │               │
                │
*               ┌──────┐       │              │              │      ┌──────┐
                │
*               │  R0  │───────┴──────────────┴──────────────┴──────│  R1
                │────┘
*               └──────┘
                └──────┘172.30.1.1
*               172.30.0.12                                   172.30.0.17
 *
*               Things to notice:
*               1) The routes in A are manually set to have R1 as the default
                router,
*               just because using a dynamic outing in this example is an
                overkill.
*               2) R1's address is set statically though the DHCP server helper
                interface.  *    This is useful to prevent address conflicts
                with the dynamic pool.
*               Not necessary if the DHCP pool is not conflicting with static
                addresses.  * 3) N2 has a dynamically-assigned, static address
                (i.e., a fixed address assigned via DHCP).
 *
 */
```

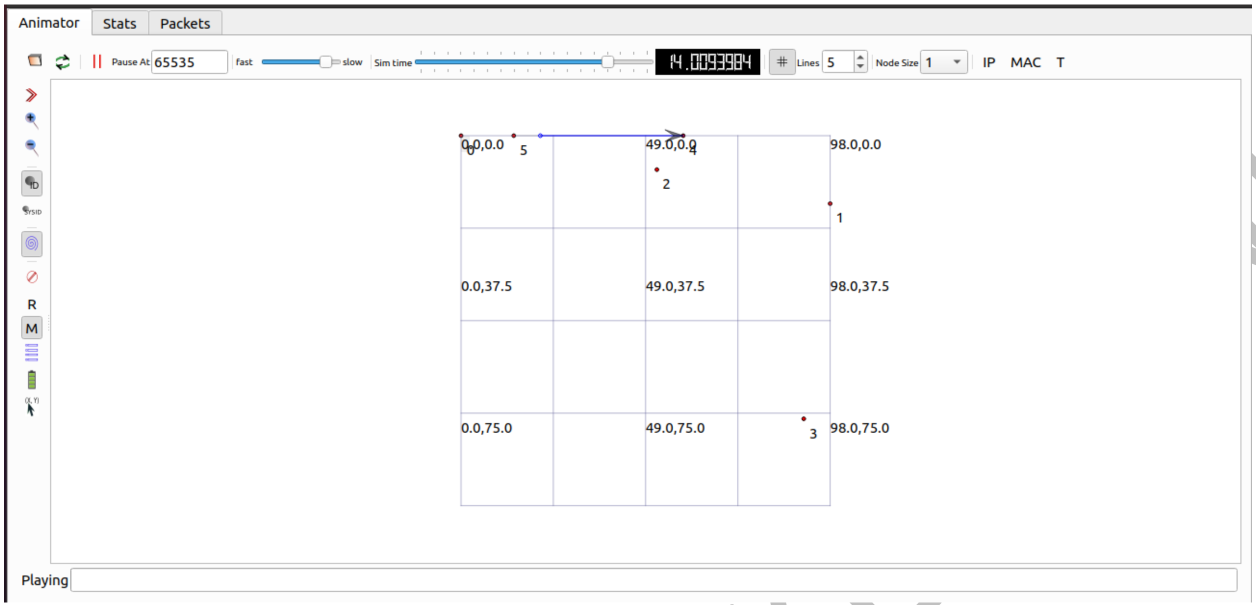
```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-apps-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
```

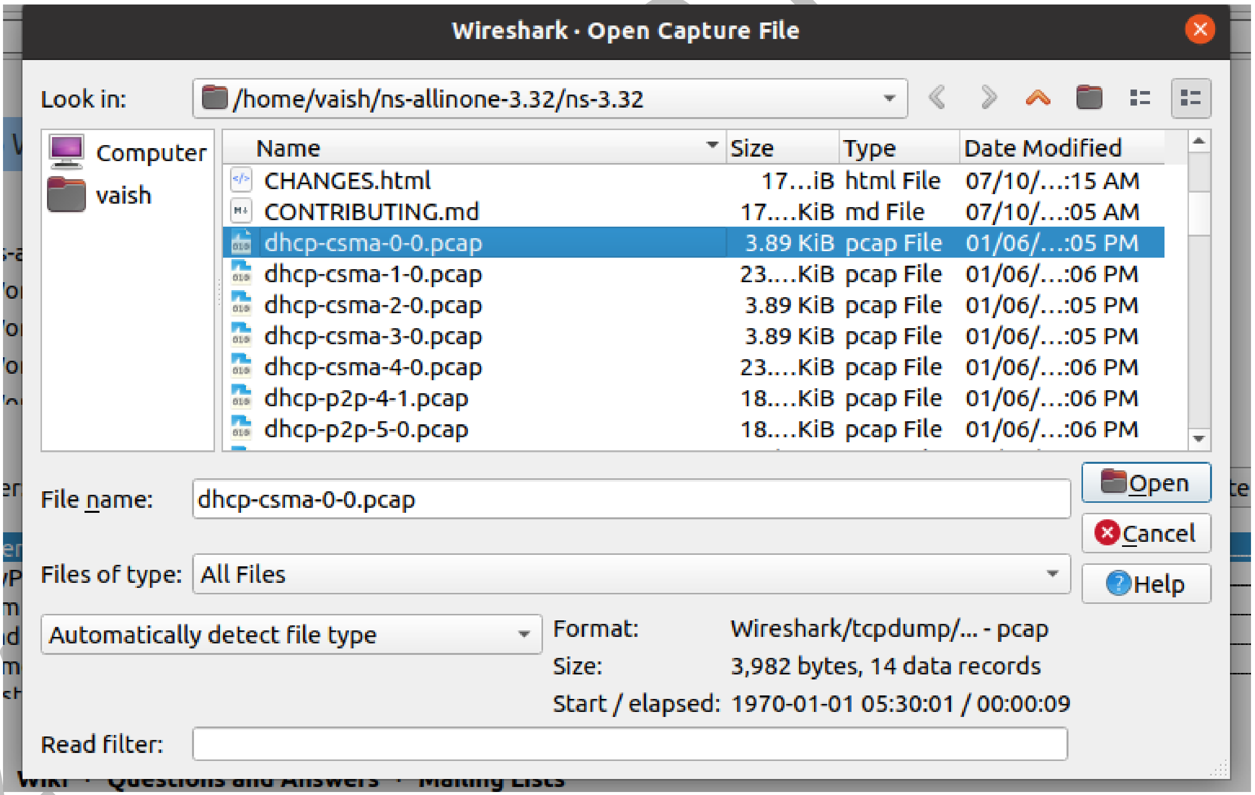
```
#include "ns3/netanim-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("DhcpExample");

int main (int argc, char *argv[])
{
  CommandLine cmd (__FILE__);

  bool verbose = true;
  bool tracing = true;
  cmd.AddValue ("verbose", "turn on the logs", verbose);
  cmd.AddValue ("tracing", "turn on the tracing", tracing);

  cmd.Parse (argc, argv);

  // GlobalValue::Bind ("ChecksumEnabled", BooleanValue (true));

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

  Time stopTime = Seconds (20);

  NS_LOG_INFO ("Create nodes.");
  NodeContainer nodes;
  NodeContainer router;
  nodes.Create (3);
  router.Create (2);

  NodeContainer net (nodes, router);

  NS_LOG_INFO ("Create channels.");
  CsmaHelper csma;
  csma.SetChannelAttribute ("DataRate", StringValue ("5Mbps"));
  csma.SetChannelAttribute ("Delay", StringValue ("2ms"));
  csma.SetDeviceAttribute ("Mtu", UintegerValue (1500));
  NetDeviceContainer devNet = csma.Install (net);

  NodeContainer p2pNodes;
  p2pNodes.Add (net.Get (4));
  p2pNodes.Create (1);

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

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

  InternetStackHelper tcpip;
  tcpip.Install (nodes);
  tcpip.Install (router);
  tcpip.Install (p2pNodes.Get (1));

  Ipv4AddressHelper address;
  address.SetBase ("172.30.1.0", "255.255.255.0");
  Ipv4InterfaceContainer p2pInterfaces;
  p2pInterfaces = address.Assign (p2pDevices);

  // manually add a routing entry because we don't want to add a dynamic routing
  Ipv4StaticRoutingHelper ipv4RoutingHelper;
  Ptr<Ipv4> ipv4Ptr = p2pNodes.Get (1)->GetObject<Ipv4> ();
  Ptr<Ipv4StaticRouting> staticRoutingA = ipv4RoutingHelper.GetStaticRouting (ipv4Ptr);
  staticRoutingA->AddNetworkRouteTo (Ipv4Address ("172.30.0.0"), Ipv4Mask ("/24"),
                             Ipv4Address ("172.30.1.1"), 1);

  NS_LOG_INFO ("Setup the IP addresses and create DHCP applications.");
  DhcpHelper dhcpHelper;

  // The router must have a fixed IP.
  Ipv4InterfaceContainer fixedNodes = dhcpHelper.InstallFixedAddress (devNet.Get (4), Ipv4Address
("172.30.0.17"), Ipv4Mask ("/24"));
  // Not really necessary, IP forwarding is enabled by default in IPv4.
  fixedNodes.Get (0).first->SetAttribute ("IpForward", BooleanValue (true));

  // DHCP server
  ApplicationContainer dhcpServerApp = dhcpHelper.InstallDhcpServer (devNet.Get (3), Ipv4Address
("172.30.0.12"),
                                          Ipv4Address ("172.30.0.0"), Ipv4Mask ("/24"),
                                          Ipv4Address ("172.30.0.10"), Ipv4Address ("172.30.0.15"),
                                          Ipv4Address ("172.30.0.17"));

  // This is just to show how it can be done.
  DynamicCast<DhcpServer> (dhcpServerApp.Get (0))->AddStaticDhcpEntry (devNet.Get (2)->GetAddress
(), Ipv4Address ("172.30.0.14"));

  dhcpServerApp.Start (Seconds (0.0));
  dhcpServerApp.Stop (stopTime);

  // DHCP clients
  NetDeviceContainer dhcpClientNetDevs;
  dhcpClientNetDevs.Add (devNet.Get (0));
  dhcpClientNetDevs.Add (devNet.Get (1));
  dhcpClientNetDevs.Add (devNet.Get (2));

  ApplicationContainer dhcpClients = dhcpHelper.InstallDhcpClient (dhcpClientNetDevs);
  dhcpClients.Start (Seconds (1.0));
  dhcpClients.Stop (stopTime);

  UdpEchoServerHelper echoServer (9);

  ApplicationContainer serverApps = echoServer.Install (p2pNodes.Get (1));
  serverApps.Start (Seconds (0.0));
  serverApps.Stop (stopTime);

  UdpEchoClientHelper echoClient (p2pInterfaces.GetAddress (1), 9);
  echoClient.SetAttribute ("MaxPackets", UintegerValue (100));
  echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
  echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

  ApplicationContainer clientApps = echoClient.Install (nodes.Get (1));
  clientApps.Start (Seconds (10.0));
  clientApps.Stop (stopTime);

  Simulator::Stop (stopTime + Seconds (10.0));

  if (tracing)
    {
      csma.EnablePcapAll ("dhcp-csma");
      pointToPoint.EnablePcapAll ("dhcp-p2p");
    }
  AnimationInterface anim ("narender-dhcp.xml");
  NS_LOG_INFO ("Run Simulation.");
  Simulator::Run ();
  Simulator::Destroy ();
  NS_LOG_INFO ("Done.");
}
```

OUTPUT:





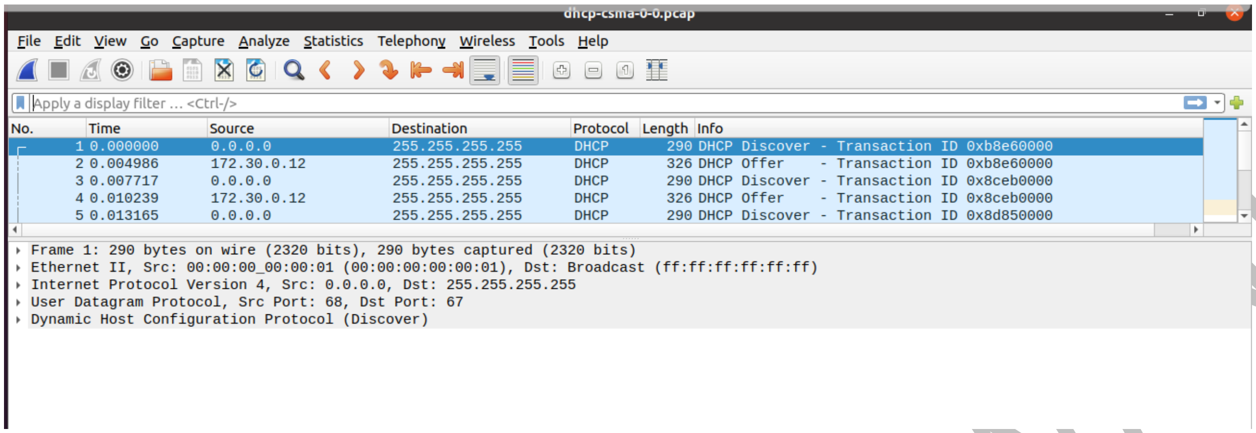




Look in: /home/vaish/ns-allinone-3.32/ns-3.32

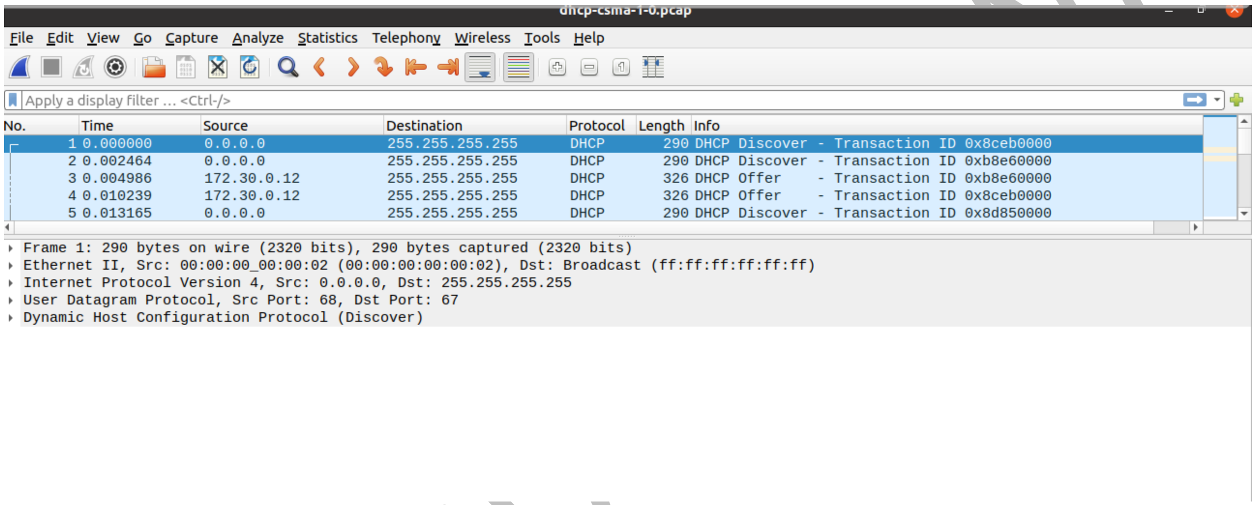
Name	Size	Type	Date Modified
CHANGES.html	17…iB	html File	07/10/…:15 AM
CONTRIBUTING.md	17….KiB	md File	07/10/…:05 AM
dhcp-csma-0-0.pcap	3.89 KiB	pcap File	01/06/…:05 PM
dhcp-csma-1-0.pcap	23….KiB	pcap File	01/06/…:06 PM
dhcp-csma-2-0.pcap	3.89 KiB	pcap File	01/06/…:05 PM
dhcp-csma-3-0.pcap	3.89 KiB	pcap File	01/06/…:05 PM
dhcp-csma-4-0.pcap	23….KiB	pcap File	01/06/…:06 PM
dhcp-p2p-4-1.pcap	18….KiB	pcap File	01/06/…:06 PM
dhcp-p2p-5-0.pcap	18….KiB	pcap File	01/06/…:06 PM

File name: dhcp-csma-0-0.pcap
Files of type: All Files
Automatically detect file type
Format: Wireshark/tcpdump/… - pcap
Size: 3,982 bytes, 14 data records
Start / elapsed: 1970-01-01 05:30:01 / 00:00:09
Read filter:


dhcp-csma-0-0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

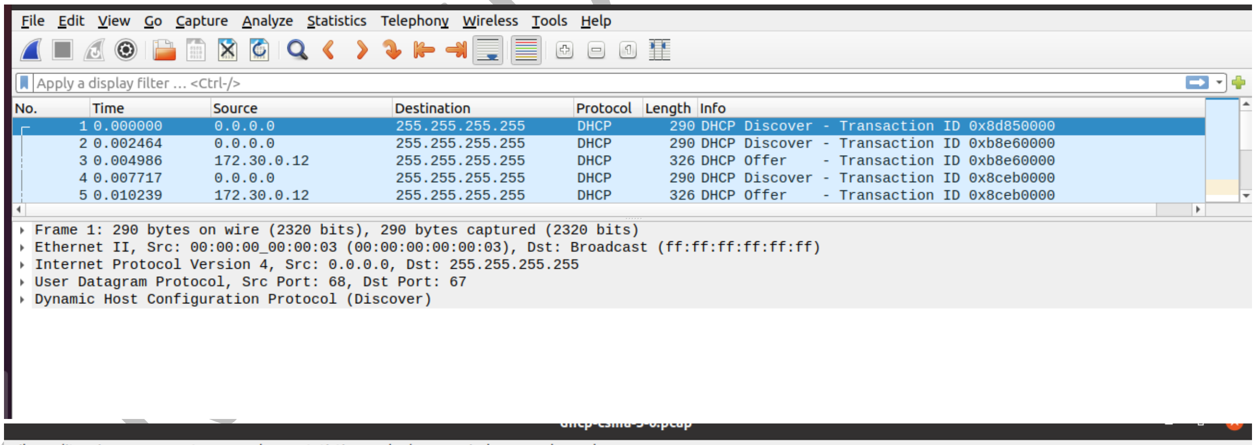
```
No.  Time      Source       Destination      Protocol Length Info
   1 0.000000  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0xb8e60000
   2 0.004986  172.30.0.12  255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0xb8e60000
   3 0.007717  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8ceb0000
   4 0.010239  172.30.0.12  255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0x8ceb0000
   5 0.013165  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8d850000
```

```
› Frame 1: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits)
› Ethernet II, Src: 00:00:00_00:00:01 (00:00:00:00:00:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
› User Datagram Protocol, Src Port: 68, Dst Port: 67
› Dynamic Host Configuration Protocol (Discover)
```


dhcp-csma-1-0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

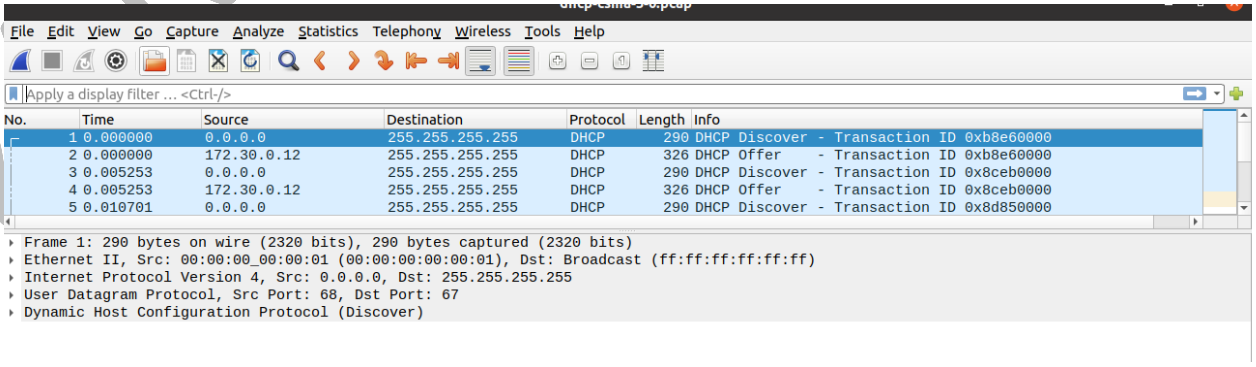
```
No.  Time      Source       Destination      Protocol Length Info
   1 0.000000  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8ceb0000
   2 0.002464  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0xb8e60000
   3 0.004986  172.30.0.12  255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0xb8e60000
   4 0.010239  172.30.0.12  255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0x8ceb0000
   5 0.013165  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8d850000
```

```
› Frame 1: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits)
› Ethernet II, Src: 00:00:00_00:00:02 (00:00:00:00:00:02), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
› User Datagram Protocol, Src Port: 68, Dst Port: 67
› Dynamic Host Configuration Protocol (Discover)
```


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

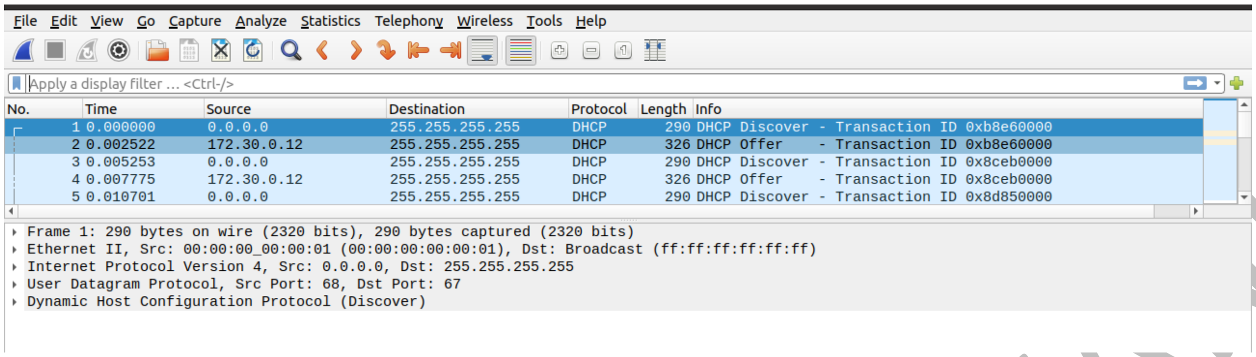
```
No.  Time      Source       Destination      Protocol Length Info
   1 0.000000  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8d850000
   2 0.002464  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0xb8e60000
   3 0.004986  172.30.0.12  255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0xb8e60000
   4 0.007717  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8ceb0000
   5 0.010239  172.30.0.12  255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0x8ceb0000
```

```
› Frame 1: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits)
› Ethernet II, Src: 00:00:00_00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
› User Datagram Protocol, Src Port: 68, Dst Port: 67
› Dynamic Host Configuration Protocol (Discover)
```


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

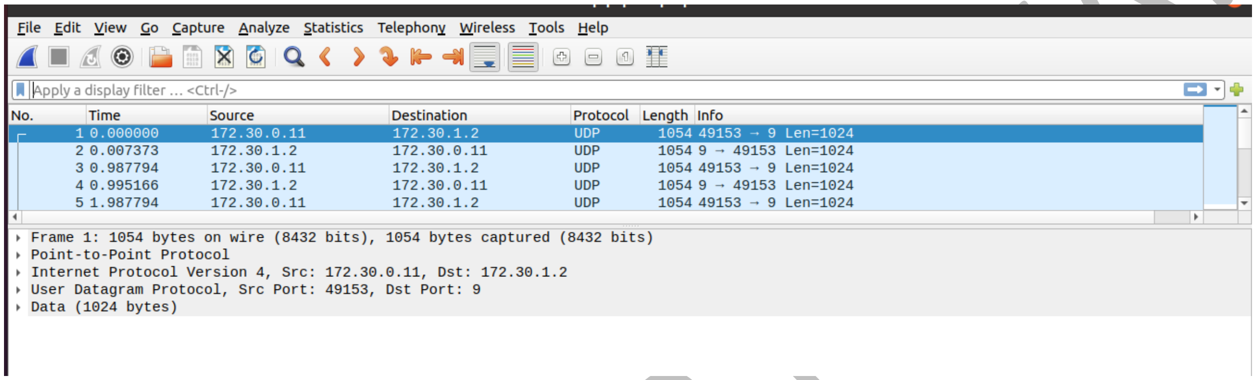
```
No.  Time      Source       Destination      Protocol Length Info
   1 0.000000  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0xb8e60000
   2 0.000000  172.30.0.12  255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0xb8e60000
   3 0.005253  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8ceb0000
   4 0.005253  172.30.0.12  255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0x8ceb0000
   5 0.010701  0.0.0.0      255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8d850000
```

```
› Frame 1: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits)
› Ethernet II, Src: 00:00:00_00:00:01 (00:00:00:00:00:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
› User Datagram Protocol, Src Port: 68, Dst Port: 67
› Dynamic Host Configuration Protocol (Discover)
```



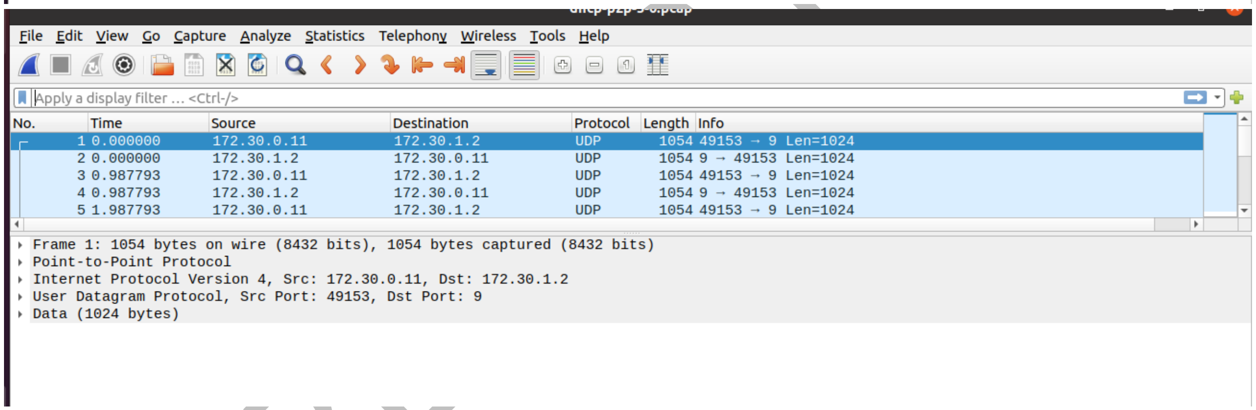
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time      Source        Destination      Protocol Length Info
   1 0.000000  0.0.0.0       255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0xb8e60000
   2 0.002522  172.30.0.12   255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0xb8e60000
   3 0.005253  0.0.0.0       255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8ceb0000
   4 0.007775  172.30.0.12   255.255.255.255  DHCP     326 DHCP Offer    - Transaction ID 0x8ceb0000
   5 0.010701  0.0.0.0       255.255.255.255  DHCP     290 DHCP Discover - Transaction ID 0x8d850000

Frame 1: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits)
Ethernet II, Src: 00:00:00_00:00:01 (00:00:00:00:00:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Discover)
```



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time      Source        Destination   Protocol Length Info
   1 0.000000  172.30.0.11   172.30.1.2    UDP      1054 49153 → 9 Len=1024
   2 0.007373  172.30.1.2    172.30.0.11   UDP      1054 9 → 49153 Len=1024
   3 0.987794  172.30.0.11   172.30.1.2    UDP      1054 49153 → 9 Len=1024
   4 0.995166  172.30.1.2    172.30.0.11   UDP      1054 9 → 49153 Len=1024
   5 1.987794  172.30.0.11   172.30.1.2    UDP      1054 49153 → 9 Len=1024

Frame 1: 1054 bytes on wire (8432 bits), 1054 bytes captured (8432 bits)
Point-to-Point Protocol
Internet Protocol Version 4, Src: 172.30.0.11, Dst: 172.30.1.2
User Datagram Protocol, Src Port: 49153, Dst Port: 9
Data (1024 bytes)
```



dhcp-p2p-3-0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time      Source        Destination   Protocol Length Info
   1 0.000000  172.30.0.11   172.30.1.2    UDP      1054 49153 → 9 Len=1024
   2 0.000000  172.30.1.2    172.30.0.11   UDP      1054 9 → 49153 Len=1024
   3 0.987793  172.30.0.11   172.30.1.2    UDP      1054 49153 → 9 Len=1024
   4 0.987793  172.30.1.2    172.30.0.11   UDP      1054 9 → 49153 Len=1024
   5 1.987793  172.30.0.11   172.30.1.2    UDP      1054 49153 → 9 Len=1024

Frame 1: 1054 bytes on wire (8432 bits), 1054 bytes captured (8432 bits)
Point-to-Point Protocol
Internet Protocol Version 4, Src: 172.30.0.11, Dst: 172.30.1.2
User Datagram Protocol, Src Port: 49153, Dst Port: 9
Data (1024 bytes)
```

CONCLUSION:

From this practical, I have understood the implementation of dhcp server in ns3 simulator.