

LAB NO.:01

Name of Experiment:

Simple Client-Server communication (NS-3warmup)

Objectives:

1. To create a simple topology of two nodes (Node1 and Node2) separated by a point to point link.
2. To setup a UDP client on one node (Node-1) and a UDP server on another (Node-2). Let it be of a fixed data rate 1.
3. To start the Client application and measure end to end throughput whilst varying the latency.

Source Code:

```
/* -*- 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/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"

// Default Network Topology
//
// 10.1.1.0
// n0 ----- n1
```

```

// point-to-point
//

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");

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

    Time::SetResolution (Time::NS);
    LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
    LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);

    NodeContainer nodes;
    nodes.Create (2);

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

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

    InternetStackHelper stack;
    stack.Install (nodes);

    Ipv4AddressHelper address;
    address.SetBase ("10.1.1.0", "255.255.255.0");

    Ipv4InterfaceContainer interfaces = address.Assign (devices);

    UdpEchoServerHelper echoServer (9);

    ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
    serverApps.Start (Seconds (1.0));
    serverApps.Stop (Seconds (10.0));

    UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
    echoClient.SetAttribute ("MaxPackets", UIntegerValue (1));
    echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
    echoClient.SetAttribute ("PacketSize", UIntegerValue (1024));

```

```
ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

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

Output:

A terminal window with a dark background and light green text. The prompt is 'raisa@raisa-HP-Pavilion-Laptop-15-cc0xx: ~/repos/ns-allinone-3.31/ns-3.31'. The command './waf --run scratch/first' is entered. The output shows Waf entering and leaving a build directory, followed by a successful build. Then, four lines of network traffic logs are displayed, showing a client sending 1024 bytes to a server and the server responding. The terminal ends with the prompt 'raisa@raisa-HP-Pavilion-Laptop-15-cc0xx:~/repos/ns-allinone-3.31/ns-3.31\$' and a cursor.

```
raisa@raisa-HP-Pavilion-Laptop-15-cc0xx: ~/repos/ns-allinone-3.31/ns-3.31$ ./waf --run scratch/first
Waf: Entering directory `/home/raisa/repos/ns-allinone-3.31/ns-3.31/build'
Waf: Leaving directory `/home/raisa/repos/ns-allinone-3.31/ns-3.31/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.930s)
At time 2s client sent 1024 bytes to 10.1.1.2 port 9
At time 2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time 2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time 2.00737s client received 1024 bytes from 10.1.1.2 port 9
raisa@raisa-HP-Pavilion-Laptop-15-cc0xx:~/repos/ns-allinone-3.31/ns-3.31$
```

Conclusion: Here in this experiment, Basic client server paradigm, reading of pcap traces are found and it's the basic of NS-3 simulation and it was successfully done.

