

```
1 /* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
2 /*
3  * This program is free software; you can redistribute it and/or modify
4  * it under the terms of the GNU General Public License version 2 as
5  * published by the Free Software Foundation;
6  *
7  * This program is distributed in the hope that it will be useful,
8  * but WITHOUT ANY WARRANTY; without even the implied warranty of
9  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
10 * GNU General Public License for more details.
11 *
12 * You should have received a copy of the GNU General Public License
13 * along with this program; if not, write to the Free Software
14 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
15 */
16
17 #include "ns3/core-module.h"
18 #include "ns3/network-module.h"
19 #include "ns3/internet-module.h"
20 #include "ns3/point-to-point-module.h"
21 #include "ns3/applications-module.h"
22
23 //NetAnim
24
25 #include "ns3/netanim-module.h"
26 #include "ns3/mobility-module.h"
27
28 // Default Network Topology
29 //
30 //      10.1.1.0
31 // n0 ----- n1
32 // point-to-point
33 //
34
35 using namespace ns3;
36
37 NS_LOG_COMPONENT_DEFINE("First");
38
39 int
40 main (int argc, char *argv[])
41 {
42     CommandLine cmd ("");
```

Open ▾



first.cc

~/ns-allinone-3.35/ns-3.35/scratch

```
55
56 NetDeviceContainer devices;
57 devices = pointToPoint.Install (nodes);
58
59 InternetStackHelper stack;
60 stack.Install (nodes);
61
62 Ipv4AddressHelper address;
63 address.SetBase ("10.1.1.0", "255.255.255.0");
64
65 Ipv4InterfaceContainer interfaces = address.Assign (devices);
66
67 UdpEchoServerHelper echoServer (9);
68
69 ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
70 serverApps.Start (Seconds (1.0));
71 serverApps.Stop (Seconds (10.0));
72
73 UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
74 echoClient.SetAttribute ("MaxPacketSize", UintegerValue (1));
75 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
76 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
77
78 ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
79 clientApps.Start (Seconds (1.0));
80 clientApps.Stop (Seconds (10.0));
81
82 MobilityHelper mobility;
83 mobility.SetMobilityModel ("ns3::RandomWaypointMobilityModel");
84 mobility.Install(nodes);
85
86 AnimationInterface anim("first.xml");
87 AnimationInterface::SetConstantPosition(nodes.Get(0),10,25);
88 AnimationInterface::SetConstantPosition(nodes.Get(1),40,25);
89 anim.EnablePacketMetadata(true);
90 pointToPoint.EnablePcapAll("first");
91
92
93 Simulator::Run ();
94 Simulator::Destroy ();
95 return 0;
96 }
```



```
5 published by the Free Software Foundation;
6 *
7 * This program is distributed in the hope that it will be useful,
8 * but WITHOUT ANY WARRANTY; without even the implied warranty of
9 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
10 * GNU General Public License for more details.
11 *
12 * You should have received a copy of the GNU General Public License
13 * along with this program; if not, write to the Free Software
14 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
15 */
16
17 #include "ns3/core-module.h"
18 #include "ns3/network-module.h"
19 #include "ns3/csma-module.h"
20 #include "ns3/internet-module.h"
21 #include "ns3/point-to-point-module.h"
22 #include "ns3/applications-module.h"
23 #include "ns3/ipv4-global-routing-module.h"
24 #include "ns3/netanim-module.h"
25 #include "ns3/mobility-module.h"
26
27 // Default Network Topology
28 //
29 //      10.1.1.0
30 // n0 ----- n1   n2   n3   n4
31 // point-to-point |   |   |
32 //                =====
33 //                LAN 10.1.2.0
34
35
36 using namespace ns3;
37
38 NS_LOG_COMPONENT_DEFINE ("SecondScriptExample");
39
40 int
41 main (int argc, char *argv[])
```



```
86 address.SetBase ("10.1.1.0", "255.255.255.0");
87 Ipv4InterfaceContainer p2pInterfaces;
88 p2pInterfaces = address.Assign (p2pDevices);
89
90 address.SetBase ("10.1.2.0", "255.255.255.0");
91 Ipv4InterfaceContainer csmaInterfaces;
92 csmaInterfaces = address.Assign (csmaDevices);
93
94 UdpEchoServerHelper echoServer (9);
95
96 ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsm));
97 serverApps.Start (Seconds (1.0));
98 serverApps.Stop (Seconds (10.0));
99
100 UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsm), 9);
101 echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
102 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
103 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
104
105 ApplicationContainer clientApps = echoClient.Install (p2pNodes.Get (0));
106 clientApps.Start (Seconds (2.0));
107 clientApps.Stop (Seconds (10.0));
108
109 Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
110
111 pointToPoint.EnablePcapAll ("p2p");
112 csma.EnablePcap ("csma", csmaDevices.Get (1), true);
113
114 //For Net Anim
115 MobilityHelper mobility;
116 mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
117 mobility.Install(p2pNodes);
118 mobility.Install(csmaNodes);
119 AnimationInterface anim("second.xml");
120 AnimationInterface::SetConstantPosition(p2pNodes.Get(0),10,25);
121 AnimationInterface::SetConstantPosition(p2pNodes.Get(1),20,25);
122 AnimationInterface::SetConstantPosition(csmaNodes.Get(1),40,25);
123 AnimationInterface::SetConstantPosition(csmaNodes.Get(2),50,25);
124 AnimationInterface::SetConstantPosition(csmaNodes.Get(3),60,25);
125 anim.EnablePacketMetadata(true);
126
```

Open



third.cc

~/ns-allinone-3.35/ns-3.35/scratch

```

11 *
12 * You should have received a copy of the GNU General Public License
13 * along with this program; if not, write to the Free Software
14 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
15 */
16
17 #include "ns3/core-module.h"
18 #include "ns3/point-to-point-module.h"
19 #include "ns3/network-module.h"
20 #include "ns3/applications-module.h"
21 #include "ns3/mobility-module.h"
22 #include "ns3/csma-module.h"
23 #include "ns3/internet-module.h"
24 #include "ns3/yans-wifi-helper.h"
25 #include "ns3/ssid.h"
26
27 //netanimation
28 #include "ns3/netanim-module.h"
29
30 // Default Network Topology
31 //
32 //   Wifi 10.1.3.0
33 //
34 //   *       *       *       *
35 //   |       |       |       |   10.1.1.0
36 //   n5      n6      n7      n0  ----- n1   n2   n3   n4
37 //                                     point-to-point |   |   |   |
38 //                                                     =====
39 //                                                     LAN 10.1.2.0
40
41 using namespace ns3;
42
43 NS_LOG_COMPONENT_DEFINE ("ns3-netanim");
44
45 int
46 main (int argc, char *argv[])

```


Open ▾



*third.cc

~/ns-allinone-3.35/ns-3.35/scratch

```
153 csmaInterfaces = address.Assign (csmaDevices);
154
155 address.SetBase ("10.1.3.0", "255.255.255.0");
156 address.Assign (staDevices);
157 address.Assign (apDevices);
158
159 UdpEchoServerHelper echoServer (9);
160
161 ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));
162 serverApps.Start (Seconds (1.0));
163 serverApps.Stop (Seconds (10.0));
164
165 UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsma), 9);
166 echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
167 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
168 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
169
170 ApplicationContainer clientApps = echoClient.Install (wifiStaNodes.Get (nWifi - 1));
171 clientApps.Start (Seconds (2.0));
172 clientApps.Stop (Seconds (10.0));
173 AnimationInterface anim("hybridWifi.xml");
174 AnimationInterface::SetConstantPosition(p2pNodes.Get(0),10,25);
175 AnimationInterface::SetConstantPosition(csmaNodes.Get(1),40,25);
176 anim.EnablePacketMetadata(true);
177 pointToPoint.EnablePcapAll("hybridwifi");
178
179 Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
180
181 Simulator::Stop (Seconds (10.0));
182
183 if (tracing == true)
184 {
185     phy.SetPcapDataLinkType (WifiPhyHelper::DLT_IEEE802_11_RADIO);
186     pointToPoint.EnablePcapAll ("third");
187     phy.EnablePcap ("third", apDevices.Get (0));
188 }
```

```
12 * You should have received a copy of the GNU General Public License
13 * along with this program; if not, write to the Free Software
14 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
15 */
16
17 // Network topology
18 //
19 //      n0      n1      n2      n3
20 //      |        |        |        |
21 //      +-----+-----+-----+
22 //              LAN
23 //
24 // - UDP flows from n0 to n1 and back
25 // - DropTail queues
26 // - Tracing of queues and packet receptions to file "udp-echo.tr"
27
28 #include <fstream>
29 #include "ns3/core-module.h"
30 #include "ns3/core-module.h"
31 #include "ns3/core-module.h"
32 #include "ns3/core-module.h"
33
34 #include "ns3/netanim-module.h"
35 #include "ns3/mobility-module.h"
36
37 using namespace ns3;
38
39 NS_LOG_COMPONENT_DEFINE ("UdpEchoExample");
40
41 int
42 main (int argc, char *argv[])
43 {
44 //
45 // Users may find it convenient to turn on explicit debugging
```


Open



udp-echo.cc

~/ns-allinone-3.35/ns-3.35/scratch

```

104 // Create a UdpEchoServer application on node one.
105 //
106 uint16_t port = 9; // well-known echo port number
107 UdpEchoServerHelper server (port);
108 ApplicationContainer apps = server.Install (n.Get (1));
109 apps.Start (Seconds (1.0));
110 apps.Stop (Seconds (10.0));
111
112 //
113 // Create a UdpEchoClient application to send UDP datagrams from node zero to
114 // node one.
115 //
116 uint32_t packetSize = 1024;
117 uint32_t maxPacketCount = 1;
118 Time interPacketInterval = Seconds (1.);
119 UdpEchoClientHelper client (serverAddress, port);
120 client.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));
121 client.SetAttribute ("Interval", TimeValue (interPacketInterval));
122 client.SetAttribute ("PacketSize", UintegerValue (packetSize));
123 apps = client.Install (n.Get (0));
124 apps.Start (Seconds (1.0));
125 apps.Stop (Seconds (10.0));
126
127 AnimationInterface anim("udpbus.xml");
128 AnimationInterface::SetConstantPosition (n.Get(0),10,25);
129 AnimationInterface::SetConstantPosition (n.Get(1),40,25);
130 AnimationInterface::SetConstantPosition (n.Get(2),20,25);
131 AnimationInterface::SetConstantPosition (n.Get(3),33,25);
132 anim.EnablePacketMetadata(true);
133 csma.EnablePcapAll("udpbus");
134
135 #if 0
136 //
137 // Users may find it convenient to initialize echo packets with actual data;
138 // the below lines suggest how to do this
139 //
140 client.SetFill (apps.Get (0), "Hello World");
141
142 client.SetFill (apps.Get (0), 0x5, 1024);

```


Open ▾



udp-client-server.cc

~/ns-allinone-3.35/ns-3.35/scratch

```
0
1 // Network topology
2 //
3 //      n0      n1
4 //      |       |
5 //      =====
6 //      LAN (CSMA)
7 //
8 // - UDP flow from n0 to n1 of 1024 byte packets at intervals of 50 ms
9 // - maximum of 320 packets sent (or limited by simulation duration)
10 // - option to use IPV4 or IPV6 addressing
11 // - option to disable logging statements
12
13 #include <fstream>
14 #include "ns3/core-module.h"
15 #include "ns3/csma-module.h"
16 #include "ns3/applications-module.h"
17 #include "ns3/internet-module.h"
18
19 #include "ns3/netanim-module.h"
20 #include "ns3/mobility-module.h"
21
22 using namespace ns3;
23
24 NS_LOG_COMPONENT_DEFINE ("UdpClientServerExample");
25
26 int
27 main (int argc, char *argv[])
28 {
29     // Declare variables used in command-line arguments
30     bool useV6 = false;
```

```
80 if (useV6 == false)
81 {
82     Ipv4AddressHelper ipv4;
83     ipv4.SetBase ("10.1.1.0", "255.255.255.0");
84     Ipv4InterfaceContainer i = ipv4.Assign (d);
85     serverAddress = Address (i.GetAddress (1));
86 }
87 else
88 {
89     Ipv6AddressHelper ipv6;
90     ipv6.SetBase ("2001:0000:f00d:cafe::", Ipv6Prefix (64));
91     Ipv6InterfaceContainer i6 = ipv6.Assign (d);
92     serverAddress = Address (i6.GetAddress (1,1));
93 }
94
95 NS_LOG_INFO ("Create UdpServer application on node 1.");
96 uint16_t port = 4000;
97 UdpServerHelper server (port);
98 ApplicationContainer apps = server.Install (n.Get (1));
99 apps.Start (Seconds (1.0));
100 apps.Stop (Seconds (10.0));
101
102 NS_LOG_INFO ("Create UdpClient application on node 0 to send to node 1.");
103 uint32_t MaxPacketSize = 1024;
104 Time interPacketInterval = Seconds (0.05);
105 uint32_t maxPacketCount = 300;
106 UdpClientHelper client (serverAddress, port);
107 client.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));
108 client.SetAttribute ("Interval", TimeValue (interPacketInterval));
109 client.SetAttribute ("PacketSize", UintegerValue (MaxPacketSize));
110 apps = client.Install (n.Get (0));
111 apps.Start (Seconds (1.0));
112 apps.Stop (Seconds (10.0));
113
114 AnimationInterface anim ("udp-cs.xml");
115 AnimationInterface::SetConstantPosition (n.Get (0), 10, 25);
116 AnimationInterface::SetConstantPosition (n.Get (1), 40, 25);
117 anim.EnablePacketMetadata (true);
118 csma.EnablePcapAll ("udp-cs");
119
120 NS_LOG_INFO ("Run the simulation");
```


Open

dhcp-example.cc

~/ns-allinone-3.35/ns-3.35/scratch



41 * DHCP Server

42 *
43 * R0
44 *
45 * 172.30.0.12

172.30.0.17

172.30.1.1

46 *
47 * Things to notice:

48 * 1) The routes in A are manually set to have R1 as the default router,
49 * just because using a dynamic routing in this example is an overkill.

50 * 2) R1's address is set statically though the DHCP server helper interface.

51 * This is useful to prevent address conflicts with the dynamic pool.

52 * Not necessary if the DHCP pool is not conflicting with static addresses.

53 * 3) N2 has a dynamically-assigned, static address (i.e., a fixed address assigned via DHCP).

54 *
55 */

56
57 #include "ns3/core-module.h"

58 #include "ns3/internet-apps-module.h"

59 #include "ns3/csma-module.h"

60 #include "ns3/internet-module.h"

61 #include "ns3/point-to-point-module.h"

62 #include "ns3/application-module.h"

63
64 #include "ns3/netanim-module.h"

65
66 using namespace ns3;

67
68 NS_LOG_COMPONENT_DEFINE ("dhcp-example.cc");

69
70 int

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


```
6
7 // DHCP clients
8 NetDeviceContainer dhcpClientNetDevs;
9 dhcpClientNetDevs.Add (devNet.Get (0));
10 dhcpClientNetDevs.Add (devNet.Get (1));
11 dhcpClientNetDevs.Add (devNet.Get (2));
12
13 ApplicationContainer dhcpClients = dhcpHelper.InstallDhcpClient (dhcpClientNetDevs);
14 dhcpClients.Start (Seconds (1.0));
15 dhcpClients.Stop (stopTime);
16
17 UdpEchoServerHelper echoServer (9);
18
19 ApplicationContainer serverApps = echoServer.Install (p2pNodes.Get (1));
20 serverApps.Start (Seconds (0.0));
21 serverApps.Stop (stopTime);
22
23 UdpEchoClientHelper echoClient (p2pInterfaces.GetAddress (1), 9);
24 echoClient.SetAttribute ("MaxPackets", UintegerValue (100));
25 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
26 echoClient.SetAttribute ("PacketSize", UintegerValue (1000));
27
28 ApplicationContainer clientApps = echoClient.Install (nodes.Get ());
29 clientApps.Start (Seconds (1.0));
30 clientApps.Stop (stopTime);
31
32 Simulator::Stop (stopTime + Seconds ( ));
33
34 //Create animation obj and configure
35 AnimationInterface anim ("dhcp.example.xml");
36
37 if (tracing)
38 {
```