#include "ns3/core-module.h"  
#include "ns3/network-module.h"  
#include "ns3/csma-module.h"  
#include "ns3/applications-module.h"  
#include "ns3/internet-apps-module.h"  
#include "ns3/internet-module.h"  
  
using namespace ns3;  
  
NS\_LOG\_COMPONENT\_DEFINE ("CsmaPingExample");  
  
static void PingRtt (std::string context, Time rtt)  
{  
  std::cout << context << " " << rtt << std::endl;  
}  
  
int  
main (int argc, char \*argv[])  
{  
  
  CommandLine cmd;  
  cmd.Parse (argc, argv);  
  
  // Here, we will explicitly create six nodes.  
  NS\_LOG\_INFO ("Create nodes.");  
  NodeContainer c;  
  c.Create (6);  
  
  // connect all our nodes to a shared channel.  
  NS\_LOG\_INFO ("Build Topology.");  
  CsmaHelper csma;  
  csma.SetChannelAttribute ("DataRate", DataRateValue (DataRate ("1Mbps")));  
  csma.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (0.2)));  
  NetDeviceContainer devs = csma.Install (c);  
  
  // add an ip stack to all nodes.  
  NS\_LOG\_INFO ("Add ip stack.");  
  InternetStackHelper ipStack;  
  ipStack.Install (c);  
  
  // assign ip addresses  
  NS\_LOG\_INFO ("Assign ip addresses.");  
  Ipv4AddressHelper ip;  
  ip.SetBase ("192.168.1.0", "255.255.255.0");  
  Ipv4InterfaceContainer addresses = ip.Assign (devs);  
  
// Create an OnOff application to send UDP datagrams from node zero to //node 1.  
  uint16\_t port = 9;     
  
  OnOffHelper onoff ("ns3::UdpSocketFactory",   
         Address (InetSocketAddress (addresses.GetAddress (2), port)));  
  onoff.SetConstantRate (DataRate ("5Mb/s"));  
  
  ApplicationContainer app = onoff.Install (c.Get (0));  
  app.Start (Seconds (6.0));  
  app.Stop (Seconds (10.0));  
  
// Create an packet sink to receive these packets  
  PacketSinkHelper sink ("ns3::UdpSocketFactory",  
            Address (InetSocketAddress (Ipv4Address::GetAny (), port)));  
  app = sink.Install (c.Get (2));  
  app.Start (Seconds (0.0));  
  
  NS\_LOG\_INFO ("Create pinger");  
  V4PingHelper ping = V4PingHelper (addresses.GetAddress (2));  
  NodeContainer pingers;  
  pingers.Add (c.Get (0));  
  pingers.Add (c.Get (1));  
  
  ApplicationContainer apps;  
  apps = ping.Install (pingers);  
  apps.Start (Seconds (1.0));  
  apps.Stop (Seconds (5.0));  
  
  
  // finally, print the ping rtts.  
  Config::Connect ("/NodeList/\*/ApplicationList/\*/$ns3::V4Ping/Rtt",  
                   MakeCallback (&PingRtt));  
  
   NS\_LOG\_INFO ("Run Simulation.");  
  
  AsciiTraceHelper ascii;  
  csma.EnableAsciiAll (ascii.CreateFileStream ("[ping1.tr](http://ping1.tr)"));  
  
  Simulator::Run ();  
  Simulator::Destroy ();  
  NS\_LOG\_INFO ("Done.");  
}