EXPERIMENT - 7

Computer Networks Lab

Aim

Using Free Open Source Software tools ns3, design and Implement star topology using StarHelperClass.

Syeda Reeha Quasar
14114802719
6C7

EXPERIMENT – 7

Aim:

Using Free Open Source Software tools ns3, design and Implement star topology using StarHelperClass.

Source Code:

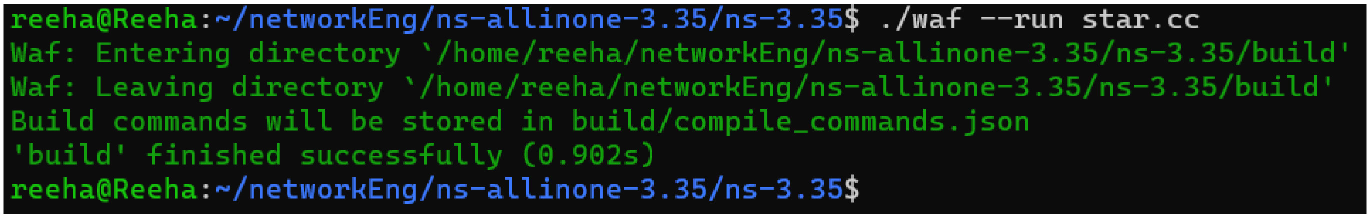
```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/netanim-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/point-to-point-layout-module.h"

// Network topology (default)
//
//       n2 n3 n4            .
//        \ | /             .
//         \|/              .
//    n1--- n0---n5          .
//         /|\              .
//        / | \             .
//       n8 n7 n6            .
//


using namespace ns3;
```

```
NS_LOG_COMPONENT_DEFINE ("Star");

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

  //
  // Set up some default values for the simulation.
  //
  Config::SetDefault ("ns3::OnOffApplication::PacketSize", UintegerValue (137));

  // ??? try and stick 15kb/s into the data rate
  Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("14kb/s"));

  //
  // Default number of nodes in the star.  Overridable by command line argument.
  //
  uint32_t nSpokes = 8;

  CommandLine cmd (__FILE__);
  cmd.AddValue ("nSpokes", "Number of nodes to place in the star", nSpokes);
  cmd.Parse (argc, argv);

  NS_LOG_INFO ("Build star topology.");
  PointToPointHelper pointToPoint;
  pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
  pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
  PointToPointStarHelper star (nSpokes, pointToPoint);
```

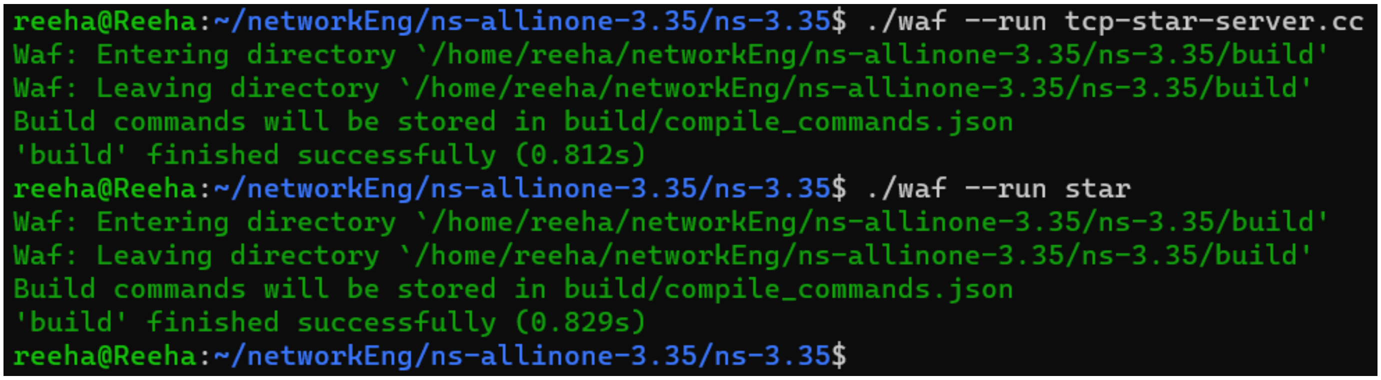
```
NS_LOG_INFO ("Install internet stack on all nodes.");
InternetStackHelper internet;
star.InstallStack (internet);

NS_LOG_INFO ("Assign IP Addresses.");
star.AssignIpv4Addresses (Ipv4AddressHelper ("10.1.1.0", "255.255.255.0"));

NS_LOG_INFO ("Create applications.");
//
// Create a packet sink on the star "hub" to receive packets.
//
uint16_t port = 50000;
Address hubLocalAddress (InetSocketAddress (Ipv4Address::GetAny (), port));
PacketSinkHelper packetSinkHelper ("ns3::TcpSocketFactory", hubLocalAddress);
ApplicationContainer hubApp = packetSinkHelper.Install (star.GetHub ());
hubApp.Start (Seconds (1.0));
hubApp.Stop (Seconds (10.0));

//
// Create OnOff applications to send TCP to the hub, one on each spoke node.
//
OnOffHelper onOffHelper ("ns3::TcpSocketFactory", Address ());
onOffHelper.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1]"));
onOffHelper.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0]"));

ApplicationContainer spokeApps;

for (uint32_t i = 0; i < star.SpokeCount (); ++i)
```

```
    {
      AddressValue remoteAddress (InetSocketAddress (star.GetHubIpv4Address (i), port));
      onOffHelper.SetAttribute ("Remote", remoteAddress);
      spokeApps.Add (onOffHelper.Install (star.GetSpokeNode (i)));
    }
  spokeApps.Start (Seconds (1.0));
  spokeApps.Stop (Seconds (10.0));

  NS_LOG_INFO ("Enable static global routing.");
  //
  // Turn on global static routing so we can actually be routed across the star.
  //
  Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

  NS_LOG_INFO ("Enable pcap tracing.");
  //
  // Do pcap tracing on all point-to-point devices on all nodes.
  //
  pointToPoint.EnablePcapAll ("star");

  NS_LOG_INFO ("Run Simulation.");
  Simulator::Run ();
  Simulator::Destroy ();
  NS_LOG_INFO ("Done.");

  return 0;
}
```

Output:



```
reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$ ./waf --run star.cc
Waf: Entering directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.902s)
reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$
```



```
reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$ ./waf --run tcp-star-server.cc
Waf: Entering directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.812s)
reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$ ./waf --run star
Waf: Entering directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.829s)
reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$
```