

NS3 AND CREATING CUSTOM TOPOLOGY

Akanksha and Sumit



INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI



Index

- Introduction to NS-3 - Installation
- First example-
Scratch-simulator.cc
- Second example- First.cc
- Different ways to trace the results

What is NS3



- ns-3 is a discrete-event network simulator
- provides models of how packet data networks work and perform
- provides a simulation engine for users to conduct simulation experiments
- perform studies that are more difficult or not possible to perform with real systems, to study system behavior in a highly controlled, reproducible environment, and to learn about how networks work

Installation of NS-3

Prerequisites

Tool	Version check command
g++	<code>\$ g++ --version</code>
clang++	<code>\$ clang++ --version</code>
python3	<code>\$ python3 -V</code>
cmake	<code>\$ cmake --version</code>

Installation of NS-3

- <https://www.nsnam.org/docs/release/3.42/installation/html/quick-start.html>

```
$ wget
```

```
https://www.nsnam.org/releases/ns-allinone-3.42.tar.bz2
```

```
$ tar xjf ns-allinone-3.42.tar.bz2
```

```
$ cd ns-allinone-3.42/ns-3.42
```

```
$ ./ns3 configure --enable-examples --enable-tests
```

```
$ ./ns3 build
```

Once build successfully. Test the software.

```
$ ./test.py
```

Once test is completed. Run the script.

```
$ ./ns3 run scratch/scratch-simulator.cc
```

It should print **Scratch simulator** in the terminal.

Introduction to NS-3



Some important folders:

- Scratch
- Src
- Doc
- Examples

Few terminology

1. Node - Computing device that connects to a network
2. Application - basic abstraction for a user program that generates some activity to be simulated
3. Channel - A channel represents the communication medium that links network devices together. Eg PointToPointChannel and WifiChannel
4. Net Device - net device in NS-3 is equivalent to a network interface card (NIC) in real-world systems. It connects a node to a channel and simulates the network protocols at the data link layer.
5. Topology helpers - topology helper objects that combine those many distinct operations into an easy to use model for your convenience

First script:
Scratch-Simulator.cc

```
17 #include "ns3/core-module.h"
```

Include the files

```
18
```

```
19 using namespace ns3;
```

The ns-3 project is implemented in a C++ namespace called ns3. This groups all ns-3-related declarations in a scope outside the global namespace

```
20
```

```
21 NS_LOG_COMPONENT_DEFINE ("ScratchSimulator");
```

```
22
```

```
23 int
```

```
24 main (int argc, char *argv[])
```

```
25 {
```

```
26     NS_LOG_UNCOND ("Scratch Simulator");
```

```
27
```

```
28     Simulator::Run ();
```

```
29     Simulator::Destroy ();
```

```
30 }
```

A logging component called ScratchSimulator that allows you to enable and disable console message logging by reference to the name.

Log the associated message unconditionally

run the simulation using the global

function Simulator::Run.

All that remains is to clean up

Second Example:
First.cc

```

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 // Default Network Topology
24 //
25 //      10.1.1.0
26 // n0 ----- n1
27 //      point-to-point
28 //
29
30 using namespace ns3;
31
32 NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
33
34 int
35 main (int argc, char *argv[])
36 {
37     CommandLine cmd (__FILE__);
38     cmd.Parse (argc, argv);
39
40     Time::SetResolution (Time::NS);
41     LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
42     LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
43
44     NodeContainer nodes;
45     nodes.Create (2);
46
47     PointToPointHelper pointToPoint;
48     pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
49     pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
50
51     NetDeviceContainer devices;
52     devices = pointToPoint.Install (nodes);
53

```

Sets time resolution to one nsec. i.e., the smallest time value that can be represented.

The next two lines of the script are used to enable two logging components that are built into the Echo Client and Echo Server applications

create the ns-3 Node objects and define the value to 2 nodes.

Physical layer:
constructing a point to point link

Create ns-3 device objects and add P2P channel between the nodes

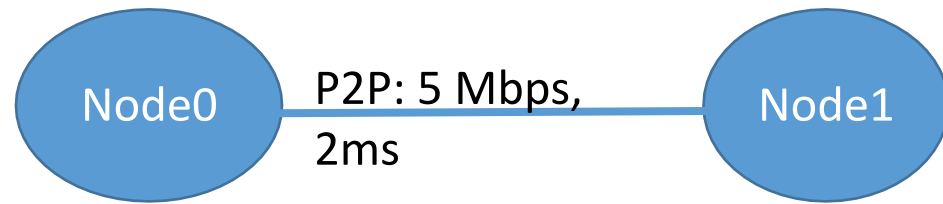
The InternetStackHelper is a topology helper that is to internet stacks what the PointToPointHelper is to point-to-point net devices.

```
51 NetDeviceContainer devices;  
52 devices = pointToPoint.Install (nodes);  
53  
54 InternetStackHelper stack;  
55 stack.Install (nodes);  
56  
57 Ipv4AddressHelper address;  
58 address.SetBase ("10.1.1.0", "255.255.255.0");  
59  
60 Ipv4InterfaceContainer interfaces = address.Assign (devices);  
61  
62 UdpEchoServerHelper echoServer (9);  
63  
64 ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));  
65 serverApps.Start (Seconds (1.0));  
66 serverApps.Stop (Seconds (10.0));  
67  
68 UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);  
69 echoClient.SetAttribute ("MaxPackets", UintegerValue (1));  
70 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));  
71 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));  
72  
73 ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));  
74 clientApps.Start (Seconds (2.0));  
75 clientApps.Stop (Seconds (10.0));  
76  
77 Simulator::Run ();  
78 Simulator::Destroy ();  
79 return 0;  
80 }
```

associate the devices with IP addresses

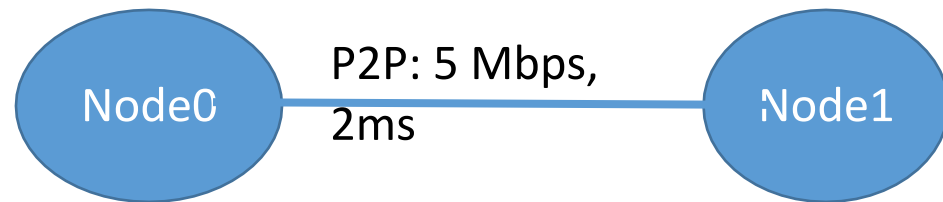
set up a UDP echo server application on one of the nodes at port no. 9

set up a UDP echo client application on one of the nodes 1



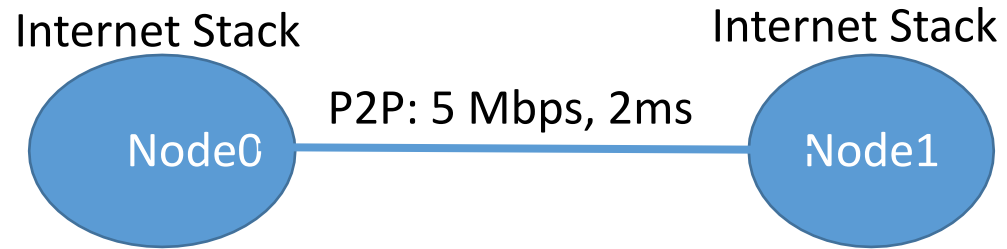
```
NodeContainer nodes;  
nodes.Create (2);
```

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

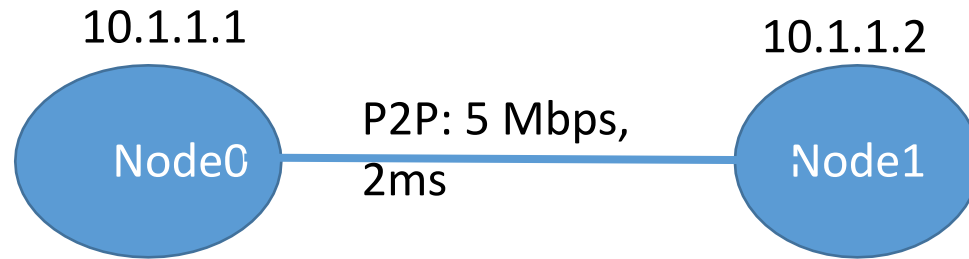


```
NetDeviceContainer devices;  
devices = pointToPoint.Install (nodes);
```

Physical layer

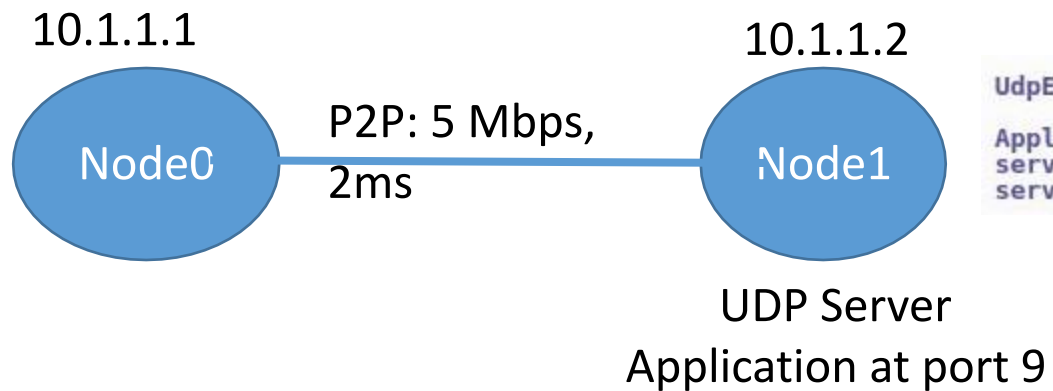


```
InternetStackHelper stack;  
stack.Install (nodes);
```

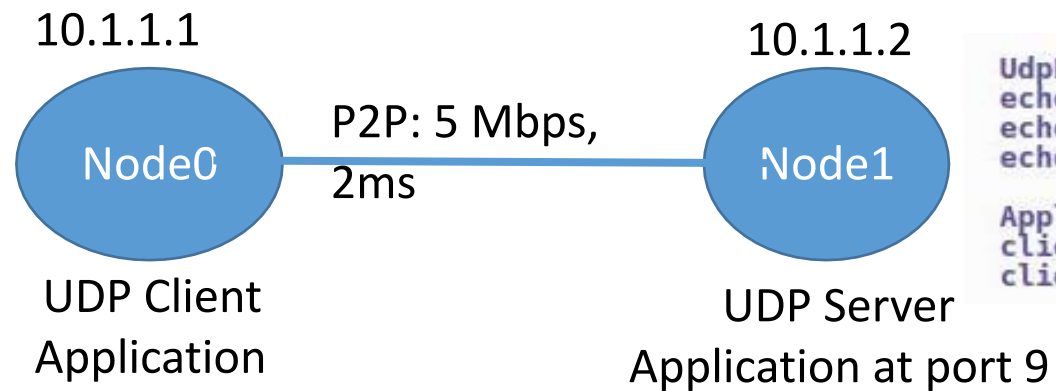


```
Ipv4AddressHelper address;  
address.SetBase ("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer interfaces = address.Assign (devices);
```

Internet stack helper



```
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));
```

Application
layer

Result

```
[2701/2801] Linking build/server/first
Waf: Leaving directory `/home/akshita/ns-allinone-3.30.1/ns-allinone-3.30.1/ns-3
.30.1/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.532s)
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
akshita@akshita-HP-Laptop:~/ns-allinone-3.30.1/ns-allinone-3.30.1/ns-3.30.1$
```

Quick exercises:

- Change the delay of P2P link from 2 msec to 5 msec?

At time 2s client sent 1024 bytes to 10.1.1.2 port 9

At time 2.00669s server received 1024 bytes from 10.1.1.1 port 49153

At time 2.00669s server sent 1024 bytes to 10.1.1.1 port 49153

At time 2.01337s client received 1024 bytes from 10.1.1.2 port 9

- Change the data rate of the P2P link from 5 Mbps to 10 Mbps?

At time 2s client sent 1024 bytes to 10.1.1.2 port 9

At time 2.00284s server received 1024 bytes from 10.1.1.1 port 49153

At time 2.00284s server sent 1024 bytes to 10.1.1.1 port 49153

At time 2.00569s client received 1024 bytes from 10.1.1.2 port 9

- Change the data rate of the P2P link from 5 Mbps to 10 Mbps, and increase the packet size to 2048?

At time 2s client sent 2048 bytes to 10.1.1.2 port 9

At time 2.00536s server received 2048 bytes from 10.1.1.1 port 49153

At time 2.00536s server sent 2048 bytes to 10.1.1.1 port 49153

At time 2.01072s client received 2048 bytes from 10.1.1.2 port 9

- Change the server port number from 9 to 21?

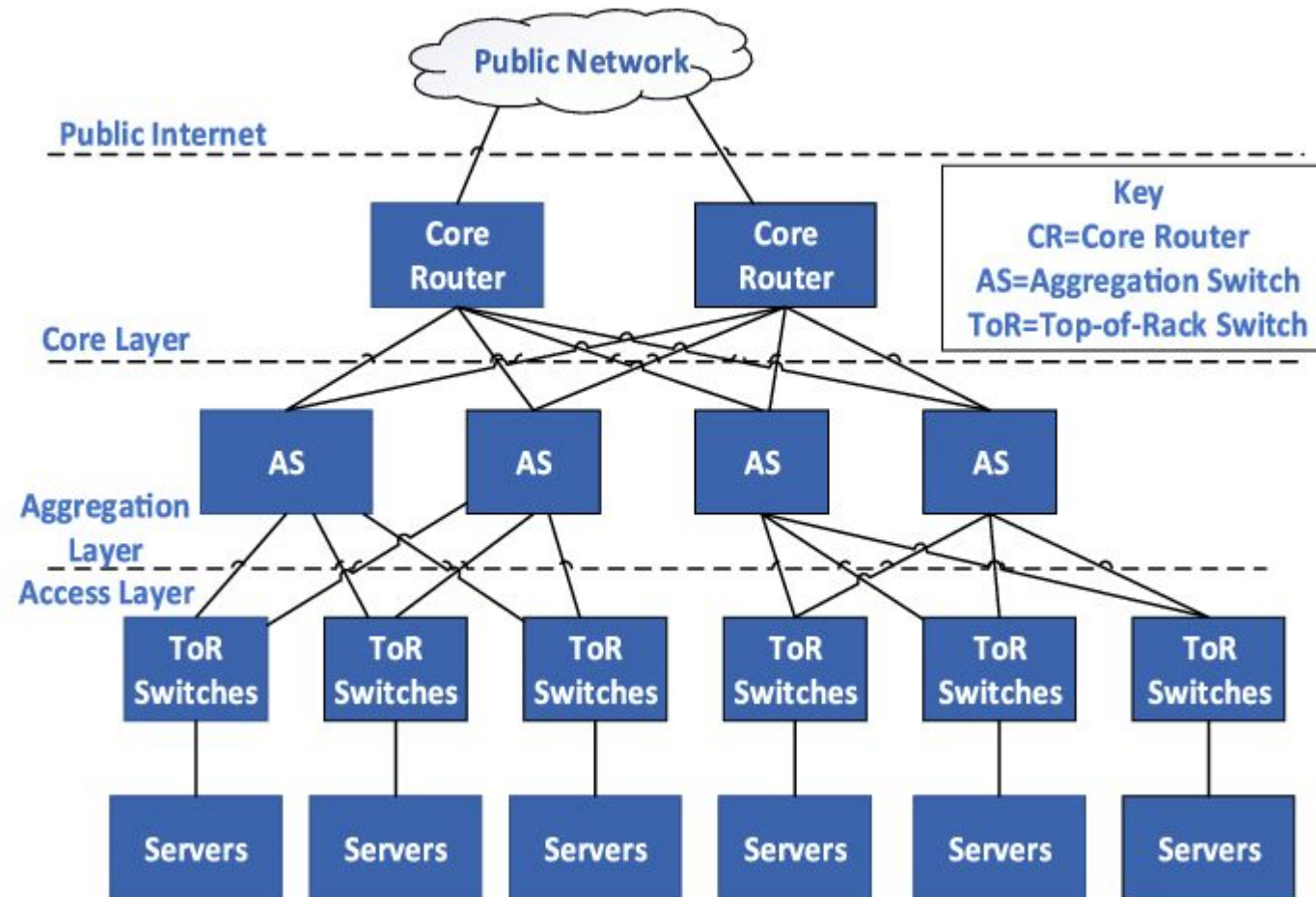
At time 2s client sent 1024 bytes to 10.1.1.2 port 21

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 21

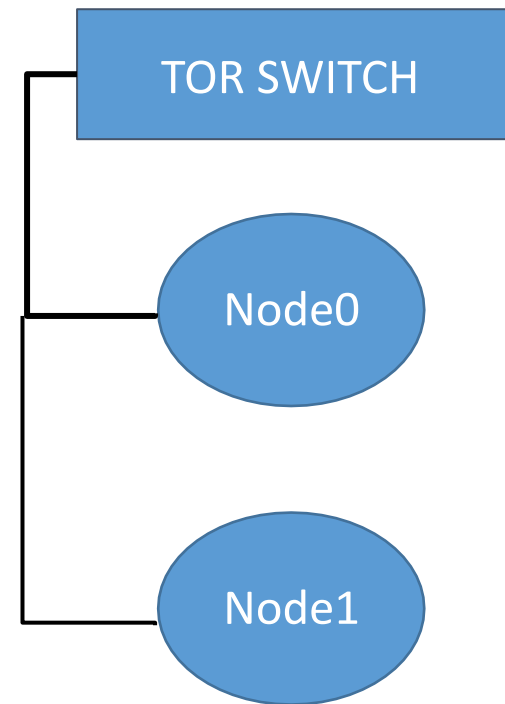
Datacenter Network Topology



Creating Custom Topology



Making a custom topology:



TOR = Top of the rack switch

SERVER RACK IN A DATACENTER

Creating Custom Topology



```
// Create 3 nodes: 1 TOR switch and 2 hosts (representing a rack)
NodeContainer hosts;
hosts.Create(2); // 2 hosts

NodeContainer torSwitch;
torSwitch.Create(1); // 1 TOR switch

// Log information about the topology
NS_LOG_INFO("Creating a topology with 1 ToR switch and 2 hosts");
NS_LOG_INFO("Host 1 connected to ToR switch.");
NS_LOG_INFO("Host 2 connected to ToR switch.");

// Create point-to-point links between TOR switch and each host
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute("DataRate", StringValue("10Gbps"));
pointToPoint.SetChannelAttribute("Delay", StringValue("1ms"));
```


Creating Custom Topology



```
// Install links between the first host and TOR switch
devicesHostToTor1 = pointToPoint.Install(NodeContainer(hosts.Get(0), torSwitch.Get(0))
NS_LOG_INFO("Link established: Host 1 <--> ToR Switch");

// Install links between the second host and TOR switch
devicesHostToTor2 = pointToPoint.Install(NodeContainer(hosts.Get(1), torSwitch.Get(0))
NS_LOG_INFO("Link established: Host 2 <--> ToR Switch");
```

Creating Custom Topology



```
// Install Internet stack on all nodes
```

```
InternetStackHelper stack;  
stack.Install(hosts);  
stack.Install(torSwitch);
```

```
// Assign IP addresses
```

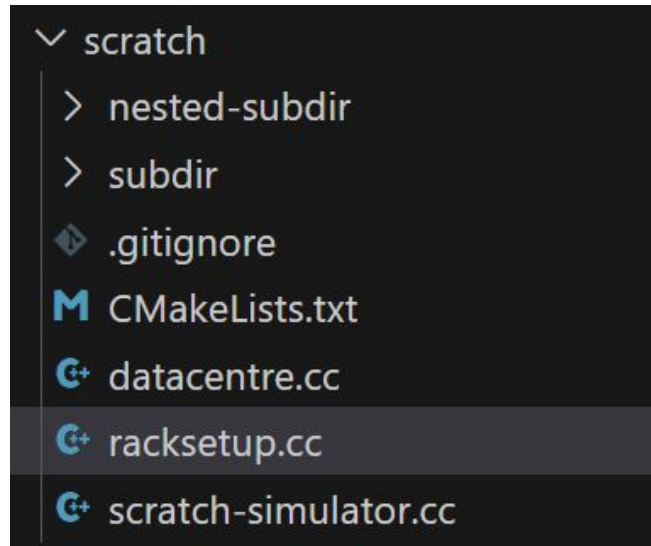
```
Ipv4AddressHelper address1, address2;  
address1.SetBase("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer interfacesHostToTor1 = address1.Assign(devicesHostToTor1);  
NS_LOG_INFO("Assigned IP to Host 1: " << interfacesHostToTor1.GetAddress(0));
```

```
address2.SetBase("10.1.2.0", "255.255.255.0");  
Ipv4InterfaceContainer interfacesHostToTor2 = address2.Assign(devicesHostToTor2);  
NS_LOG_INFO("Assigned IP to Host 2: " << interfacesHostToTor2.GetAddress(0));
```


Running Custom Topology



1. Place .cc file in scratch folder



2. Build Again using command: **./ns3 build**

3. Run code using command:

NS_LOG=DataCenterExample=info ./ns3 run scratch/racksetup.cc

DEMO



Code available at Github repo:

https://github.com/AkankshaSingal8/socket_programming



Make any Topology on your own!



You now have the tools to make and run any custom topology and network simulations at your disposal!

Use this to simulate cool networks like and visualise data transfer:

A bus topology, star topology

IIITD architecture! (at least a smaller version)



NetAnim install instructions

```
$ cd ns-allinone-3.30.1/netanim-3.108/  
$ sudo apt install qt4-default qt4-qmake  
$ qmake NetAnim.pro  
$ make  
$ ./NetAnim
```



Cont...



NetAnim (show the results through simulation)

In First.cc add this lines :

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

```
AnimationInterface anim("first.xml");  
anim.SetConstantPosition(nodes.Get(0),10.0,10.0);  
anim.SetConstantPosition(nodes.Get(1),20.0,20.0);
```

Animator

Stats

Packets



Pause At 65535

fast



slow

Sim time



Lines

5



Node Size

1



IP

MAC

T

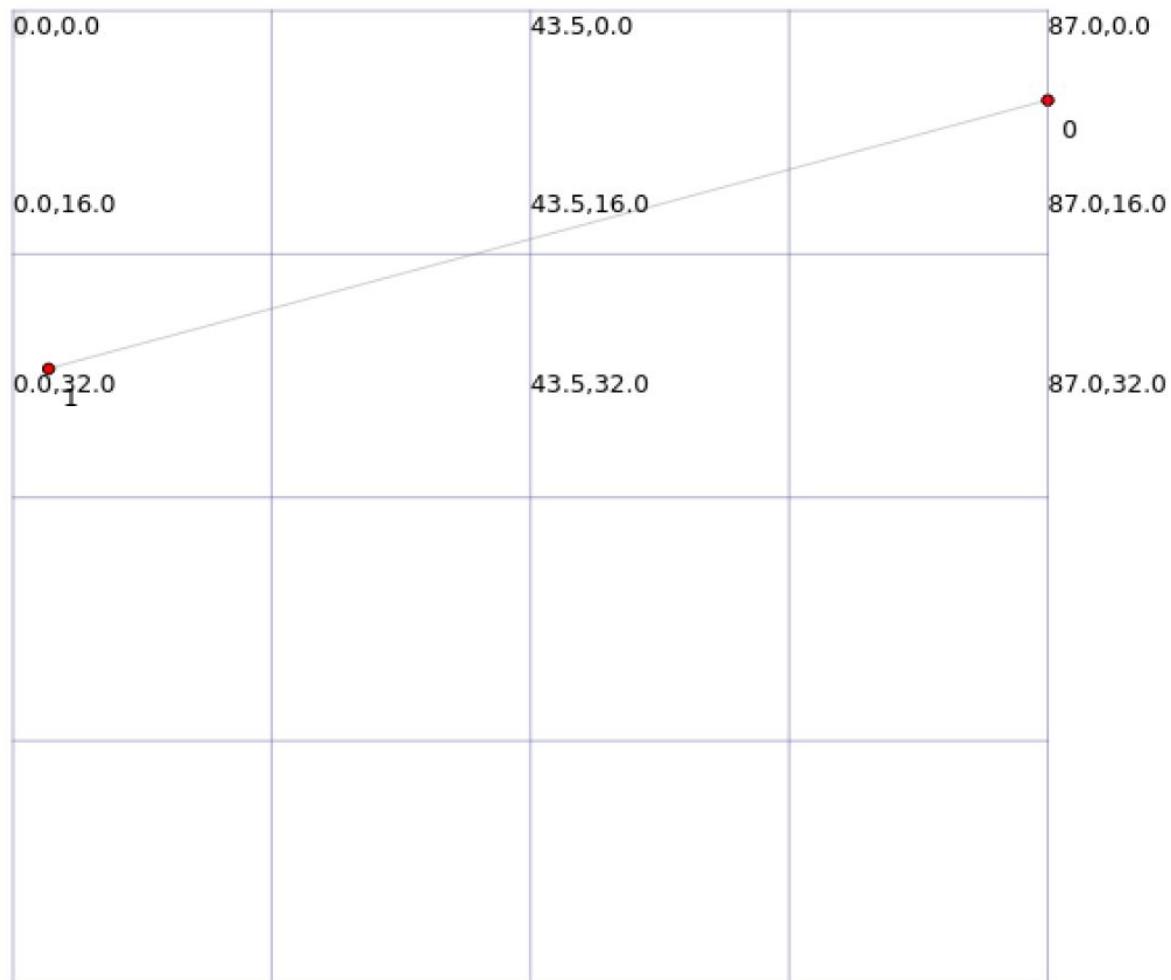


R

M



(X, Y)



Parsing complete:Click Play

Animator

Stats

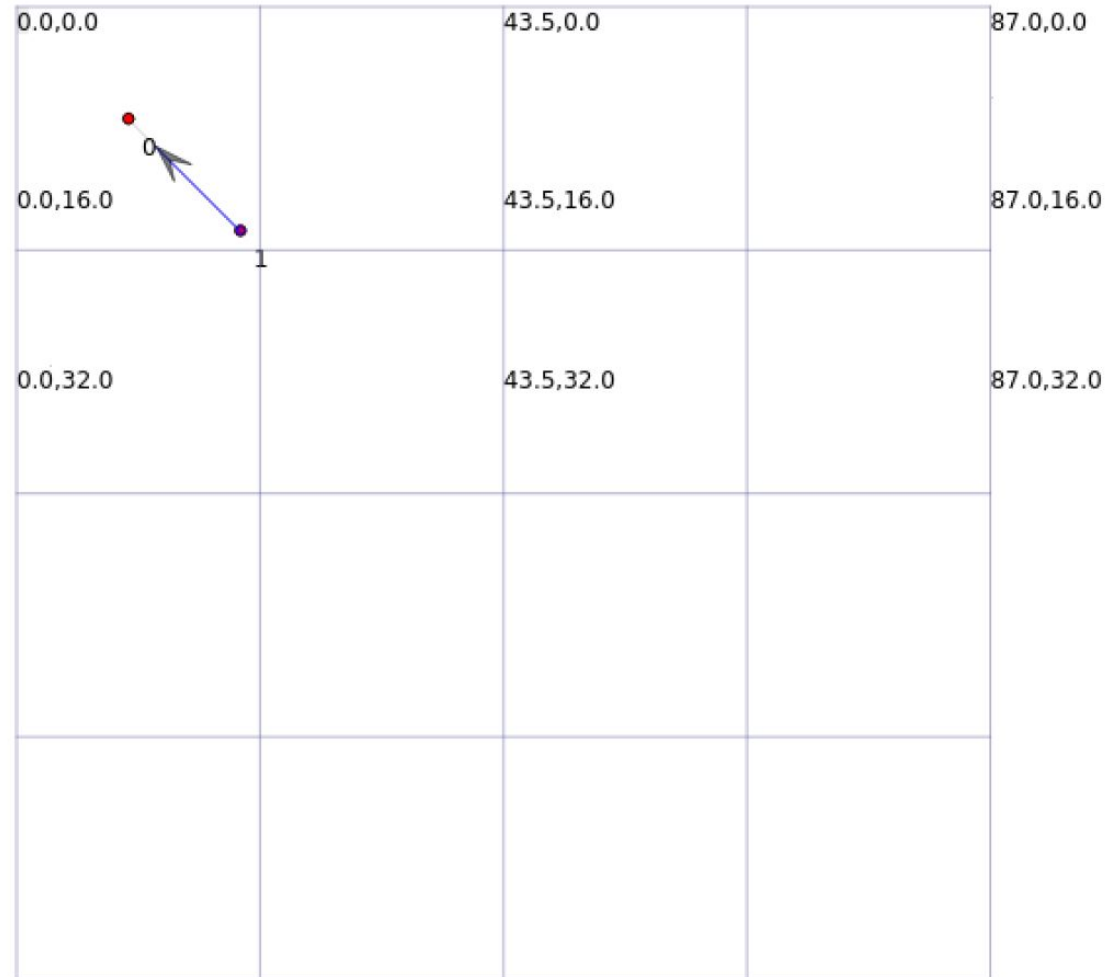
Packets



 Pause At
 fast  slow
 Sim time 
2.0051864

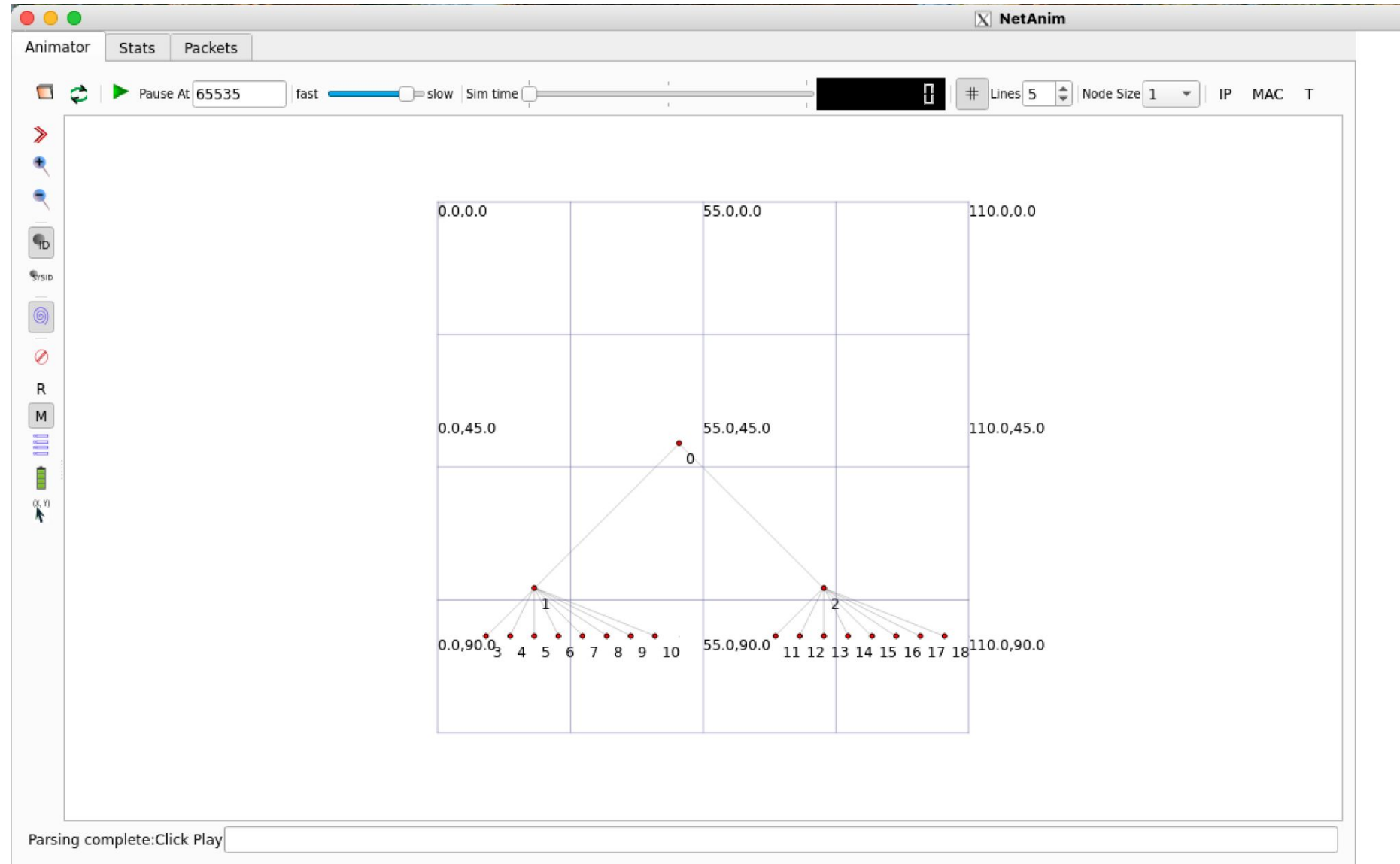
 Lines
 Node Size
 IP MAC T

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 



Playing

Exercise :

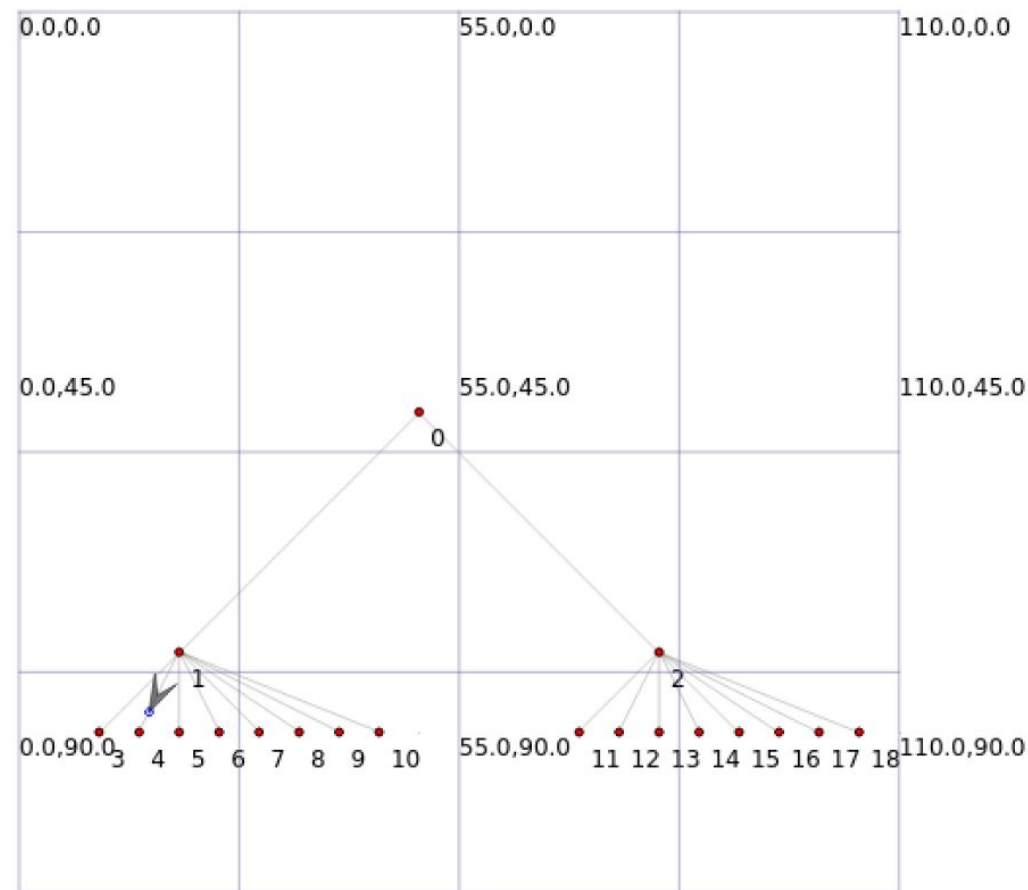


Animator

Stats

Packets

Pause At 65535 fast slow Sim time 2.00375253 # Lines 5 Node Size 1 IP MAC T



Playing

Resources



Building topologies:

<https://www.nsnam.org/docs/tutorial/html/building-topologies.html>

NS3 TUTORIALS:

<https://www.nsnam.org/docs/tutorial/html/index.html#>

