

PERCOBAAN 3

SIMULASI WIFI (Wireless AP-Client)

4.1 Tujuan :

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu :

- Mendesain dan memprogram jaringan WIFI menggunakan NS-3.
- Memahami dan membandingkan hasil trace output jaringan WIFI pada NS-3 menggunakan Tracemetrics.

4.2 Peralatan :

- 1 PC dilengkapi dengan OS Ubuntu 22.04
- Software NS-3 versi 3.36
- Software Tracemetrics

4.3 Teori :

Dalam simulasi Wireless Fidelity atau Wifi kita perlu melihat terlebih dahulu parameter yang digunakan untuk standar Wifi itu sendiri. Salah satu yang paling krusial adalah penggunaan Standar 802.11. Kemudian pada simulasi Wifi ini terdapat dua jenis node yaitu Station (Sta) dan Access Point (AP).

Selain itu, pada simulasi Wifi ini objek helper yang digunakan adalah WifiHelper dan WifiMacHelper yang dibuat untuk mengatur beberapa atribut lain.

Sama halnya seperti pada simulasi sebelumnya, pada simulasi Wifi ini output visualnya dapat dilihat pada Netanim. Skrip konfigurasi Netanim untuk simulasi ini adalah seperti yang ditunjukkan di bawah ini.

```
// Animation configuration lines
AnimationInterface anim ("mythird.xml");
anim.EnablePacketMetadata(true);
// End of animation configuration
```

Sedangkan, untuk melihat trace outputnya, masih sama yaitu dengan menggunakan Tracemetrics, dimana skrip konfigurasi ascii tracing yang perlu ditambahkan sebelumnya adalah seperti berikut.

```
//Ascii Format Tracing
AsciiTraceHelper ascii;
phy.EnableAsciiAll(ascii.CreateFileStream("mythird.tr"));
```

4.4 Prosedur Percobaan :

1. Buka direktori *ns-allinone-3.36.1/ns-3.36.1/scratch* melalui terminal.
Copykan dari *examples/tutorial/third.cc*
Namun gabungan antara wireless dan wireline. Edit program agar sesuai dengan dibawah agar full wireless.
2. Tuliskan perintah gedit dengan nama file *third.cc* seperti berikut.

gedit third.cc
3. Tuliskan script di bawah ini, dimana script ini merupakan simulasi sederhana Wifi empat node.

```
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/ssid.h"

// Default Network Topology
//
//   Wifi 10.1.3.0
//   *      *      *      AP
//   *      *      *      *
//   |      |      |      |      10.1.1.0
//   n5     n6     n7     n0 ----- n1     n2     n3     n4
//                               point-to-point | | | |
//                                           =====
//                                           LAN 10.1.2.0

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("ThirdScriptExample");

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

```

bool verbose = true;
uint32_t nCdma = 3;
uint32_t nWifi = 3;
bool tracing = false;

CommandLine cmd (__FILE__);
cmd.AddValue ("nCdma", "Number of \"extra\" CSMA
nodes/devices", nCdma);
cmd.AddValue ("nWifi", "Number of wifi STA devices", nWifi);
cmd.AddValue ("verbose", "Tell echo applications to log if
true", verbose);
cmd.AddValue ("tracing", "Enable pcap tracing", tracing);

cmd.Parse (argc,argv);

// The underlying restriction of 18 is due to the grid position
// allocator's configuration; the grid layout will exceed the
// bounding box if more than 18 nodes are provided.
if (nWifi > 18)
{
    std::cout << "nWifi should be 18 or less; otherwise grid
layout exceeds the bounding box" << std::endl;
    return 1;
}

if (verbose)
{
    LogComponentEnable ("UdpEchoClientApplication",
LOG_LEVEL_INFO);
    LogComponentEnable ("UdpEchoServerApplication",
LOG_LEVEL_INFO);
}

NodeContainer p2pNodes;
p2pNodes.Create (2);

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

NetDeviceContainer p2pDevices;
p2pDevices = pointToPoint.Install (p2pNodes);

NodeContainer csmaNodes;
csmaNodes.Add (p2pNodes.Get (1));
csmaNodes.Create (nCdma);

CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds
(6560)));

NetDeviceContainer csmaDevices;
csmaDevices = csma.Install (csmaNodes);

```

```

NodeContainer wifiStaNodes;
wifiStaNodes.Create (nWifi);
NodeContainer wifiApNode = p2pNodes.Get (0);

YansWifiChannelHelper channel = YansWifiChannelHelper::Default
();
YansWifiPhyHelper phy;
phy.SetChannel (channel.Create ());

WifiMacHelper mac;
Ssid ssid = Ssid ("ns-3-ssid");

WifiHelper wifi;

NetDeviceContainer staDevices;
mac.SetType ("ns3::StaWifiMac",
            "Ssid", SsidValue (ssid),
            "ActiveProbing", BooleanValue (false));
staDevices = wifi.Install (phy, mac, wifiStaNodes);

NetDeviceContainer apDevices;
mac.SetType ("ns3::ApWifiMac",
            "Ssid", SsidValue (ssid));
apDevices = wifi.Install (phy, mac, wifiApNode);

MobilityHelper mobility;

mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                               "MinX", DoubleValue (0.0),
                               "MinY", DoubleValue (0.0),
                               "DeltaX", DoubleValue (5.0),
                               "DeltaY", DoubleValue (10.0),
                               "GridWidth", UIntegerValue (3),
                               "LayoutType", StringValue
("RowFirst"));

mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                           "Bounds", RectangleValue (Rectangle
(-50, 50, -50, 50)));
mobility.Install (wifiStaNodes);

mobility.SetMobilityModel
("ns3::ConstantPositionMobilityModel");
mobility.Install (wifiApNode);

InternetStackHelper stack;
stack.Install (csmaNodes);
stack.Install (wifiApNode);
stack.Install (wifiStaNodes);

Ipv4AddressHelper address;

address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("10.1.2.0", "255.255.255.0");

```

```

Ipv4InterfaceContainer csmaInterfaces;
csmaInterfaces = address.Assign (csmaDevices);

address.SetBase ("10.1.3.0", "255.255.255.0");
address.Assign (staDevices);
address.Assign (apDevices);

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install
(csmaNodes.Get (nCsmas));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress
(nCsmas), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds
(1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps =
echoClient.Install (wifiStaNodes.Get (nWifi - 1));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

Simulator::Stop (Seconds (10.0));

if (tracing)
{
    phy.SetPcapDataLinkType
(WifiPhyHelper::DLT_IEEE802_11_RADIO);
    pointToPoint.EnablePcapAll ("third");
    phy.EnablePcap ("third", apDevices.Get (0));
    csma.EnablePcap ("third", csmaDevices.Get (0), true);
}

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

```

4. Buka direktori *ns-allinone-3.36.1/ns-3.36.1* dan jalankan program tersebut menggunakan *ns3* dengan menuliskan perintah seperti di bawah ini.

```

zenhadi@zenhadi-vm:~/Downloads/ns-allinone-3.36.1/ns-3.36.1$ ./ns3 run scratch/third.cc
[ 0%] Building CXX object scratch/CMakeFiles/scratch_third.dir/third.cc.o
[ 0%] Linking CXX executable ../../build/scratch/ns3.36.1-third-default
At time +2s client sent 1024 bytes to 10.1.2.4 port 9
At time +2.01624s server received 1024 bytes from 10.1.3.3 port 49153
At time +2.01624s server sent 1024 bytes to 10.1.3.3 port 49153
At time +2.02849s client received 1024 bytes from 10.1.2.4 port 9

```

5. Tambahkan kode berikut agar bisa ditampilkan animasinya:

Bagian header:

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

Bagian Tengah:

```
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");  
mobility.Install (wifiApNode);  
mobility.Install (csmaNodes); ✓
```

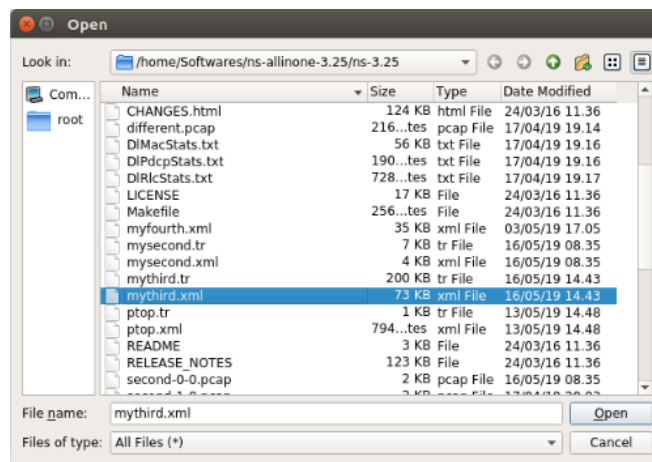
Bagian bawah:

```
// Animation configuration lines  
AnimationInterface anim ("mythird0.xml");  
anim.EnablePacketMetadata(true);  
  
// End of animation configuration  
  
// Animation configuration lines  
AnimationInterface anim ("mythird0.xml");  
anim.EnablePacketMetadata(true);  
// End of animation configuration  
  
//Ascii Format Tracing  
AsciiTraceHelper ascii;  
phy.EnableAsciiAll(ascii.CreateFileStream("mythird0.tr"));
```

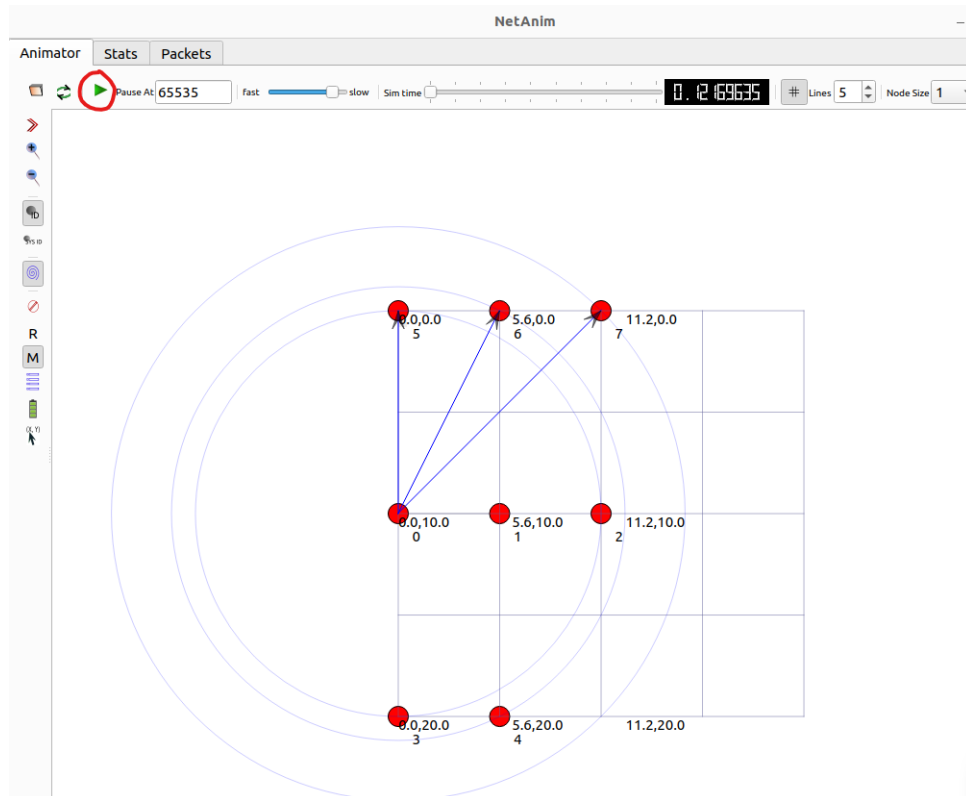
6. Kemudian, buka direktori *ns-allinone-3.36.1/netanim-3.108* buka Netanim dengan menuliskan command berikut.

```
zenhadi@zenhadi-vm:~/Downloads/ns-allinone-3.36.1/netanim-3.108$ ./NetAnim
```

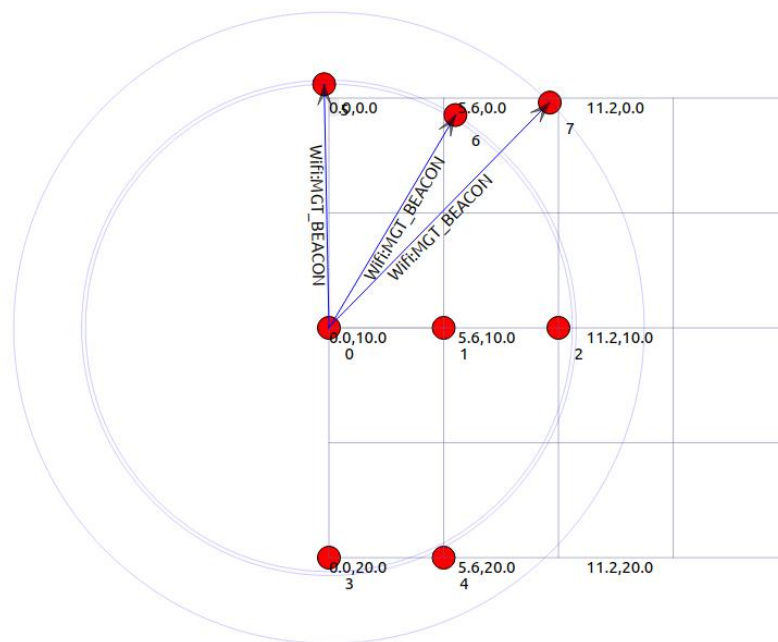
7. Buka file *mythird.xml* dengan cara klik icon Open File di pojok kiri atas dan akan muncul tampilan berikut.



8. Jalankan simulasi tersebut dengan klik icon Play berwarna hijau di pojok kiri atas.



9. Ketika dijalankan, node – node yang ditampilkan seperti yang ditunjukkan pada gambar di bawah ini.



10. Tambahkan kode berikut untuk tracing file pada sistem komunikasi yang sudah dibangun.

```
//Ascii Format Tracing
AsciiTraceHelper ascii;
phy.EnableAsciiAll(ascii.CreateFileStream("mythird0.tr"));

//Ascii Format Tracing
AsciiTraceHelper ascii;
phy.EnableAsciiAll(ascii.CreateFileStream("mythird0.tr"));

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

11. Untuk mengetahui trace outputnya, buka direktori *tracemetrics-1.4.0*.

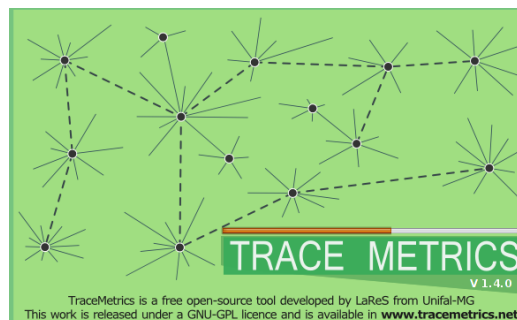
Gunakan *tracemetrics-1.3.0* jika masih gagal dengan versi 1.4.0.

<https://stackoverflow.com/questions/65045027/exception-when-trying-to-run-tracemetrics-with-ns3>

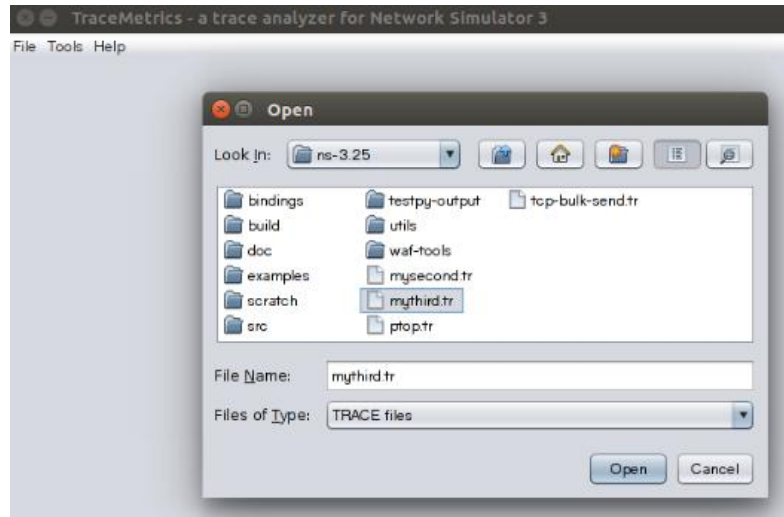
Kemudian, buka file *tracemetrics.jar* melalui terminal dengan menuliskan perintah berikut.

```
root@dina-Inspiron-5458: /home/Softwares/tracemetrics-1.4.0# java -jar "tracemetrics.jar"
```

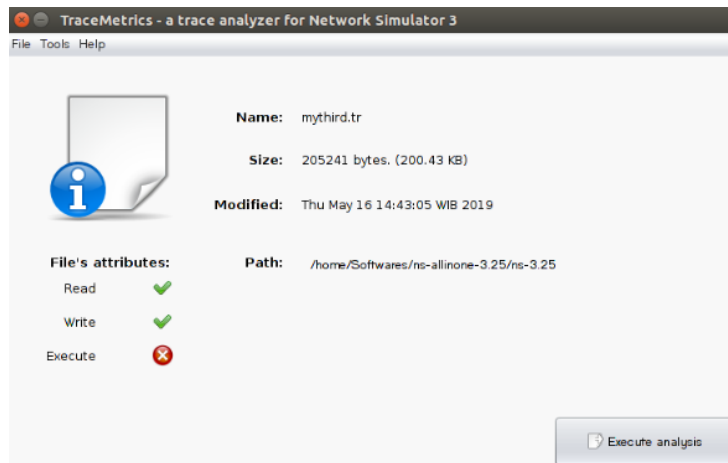
Maka, akan muncul tampilan seperti di bawah ini.



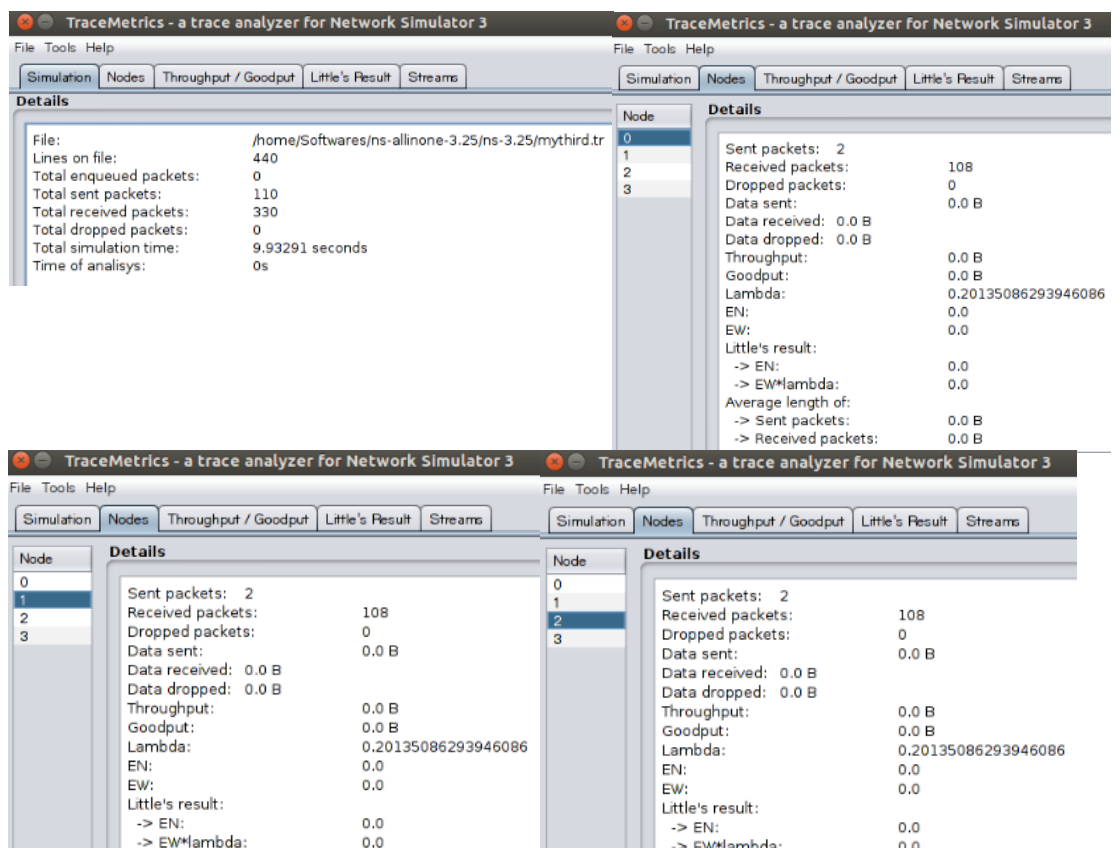
12. Buka file *mythird.tr* dengan cara klik File → Choose File.

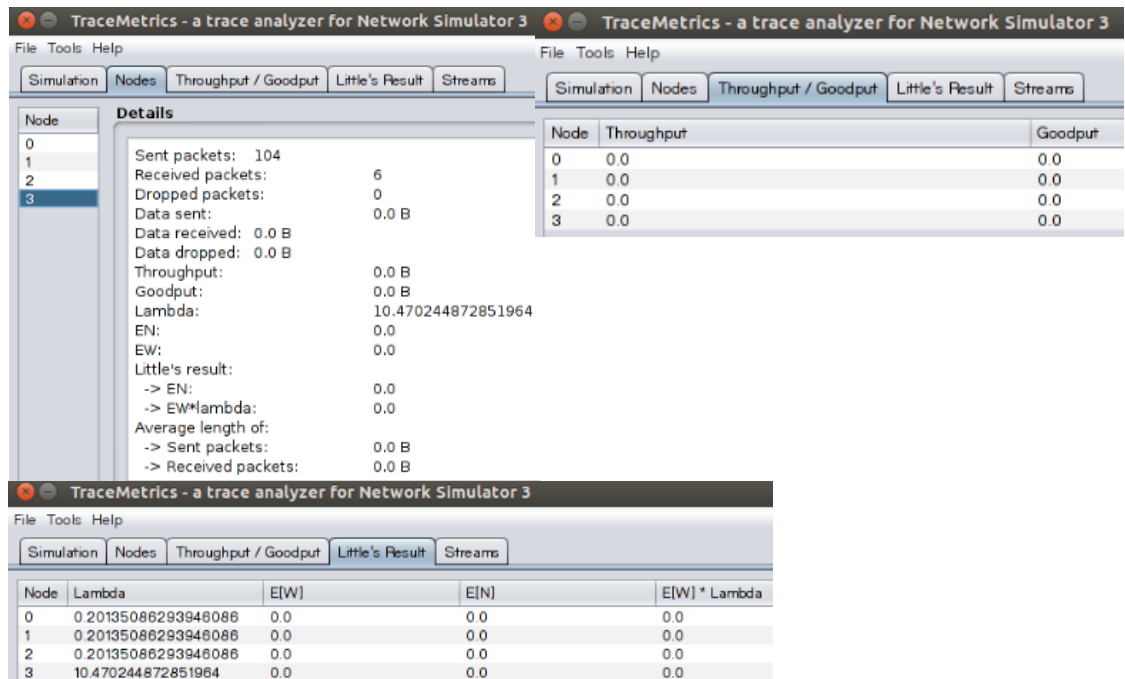


13. Setelah itu, akan muncul tampilan seperti di bawah ini. Kemudian, klik Execute analysis.



14. Hasil trace output untuk simulasi Wifi ini adalah seperti berikut.





Buat grafik excel untuk throughput, goodput dan lambda.

4.5 Analisa

- Pada percobaan ini, coba ubahlah nilai *Rectangle Value* menjadi (-100, 100, -100, 100).
- Amatilah perubahan yang terjadi.