

PERCOBAAN 1

INSTALASI NETWORK SIMULATOR 3 (NS-3)

1.1 Tujuan :

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu :

- Mengerti software – software pendukung yang perlu diinstal untuk dapat menggunakan NS-3.
- Melakukan proses instalasi NS-3.
- Memahami proses dan cara penggunaan Tracemetrics untuk melihat trace output.

1.2 Peralatan :

- 1 PC dilengkapi dengan OS Ubuntu 16.04
- Software NS-3 versi 3.25
- Software Tracemetrics
- Koneksi Internet

1.3 Teori :

1.3.1 Network Simulator 3 (NS-3)

NS-3 adalah perangkat lunak yang dikembangkan untuk menyediakan platform simulasi jaringan extensible yang bersifat open, untuk keperluan penelitian dan pendidikan. NS-3 memberikan pemodelan-pemodelan tentang bagaimana packet data networks bekerja dan berjalan, dan menyediakan simulation engine bagi pengguna untuk melakukan eksperimen simulasi. Beberapa alasan untuk menggunakan NS-3 adalah untuk melakukan studi yang tidak mungkin dilakukan dengan real systems, untuk mempelajari perilaku sistem dalam lingkungan reproducible yang bisa dikontrol, dan belajar tentang bagaimana jaringan bekerja. Pengguna akan memperhatikan bahwa model yang tersedia di dalam NS-3 berfokus pada memodelkan bagaimana protokol Internet dan jaringan bekerja, tapi NS-3 tidak

terbatas pada sistem internet, beberapa pengguna menggunakan NS-3 untuk memodelkan sistem berbasis non-Internet.

NS-3 mempunyai kelebihan tersendiri dibandingkan simulator jaringan lainnya, yaitu memiliki beberapa script yang dapat dikembangkan sesuai kebutuhan jaringan yang diinginkan. Ditulis dengan bahasa C++ dan Python. Namun, script yang disajikan hanya berupa script dasar, sehingga pengguna perlu mengembangkan sendiri sesuai kebutuhan jaringan.

Secara struktur bahasa, C++ pada NS3 memang terlihat lebih rumit daripada bahasa TCL yang digunakan pada NS2. Namun NS3 lebih mudah digunakan karena sudah berisi pustaka (library) yang bisa kita panggil untuk memenuhi kebutuhan kita. Infrastruktur NS3 sendiri mendukung pengembangan model Simulasi yang mendekati kondisi real dan memiliki karakteristik mendekati emulator Jaringan. Selain dapat digunakan sebagai Simulator untuk Jaringan berbasis IP, NS3 juga dapat digunakan sebagai Simulator Jaringan berbasis non-IP.

1.3.2 Konsep Dasar NS-3

- **Node**

Dalam jaringan internet, sebuah perangkat komputasi atau komputer yang terhubung pada suatu jaringan biasa disebut *host* atau *end system*. Karena NS-3 adalah sebuah network simulator, bukan secara spesifik internet simulator, jadi disini tidak menggunakan istilah host yang sangat berkaitan erat dengan internet dan protokolnya. Disini kita menggunakan istilah yang banyak digunakan simulator lain pada umumnya, *The Node*. Pada NS-3, suatu perangkat jaringan sederhana adalah sebuah node. Abstraksi ini digambarkan oleh simulator NS-3 dengan kelas Node pada bahasa pemrograman C++ nya. Kelas ini menangani metode untuk memanager representasi dari sebuah perangkat jaringan pada simulasi.

Pada NS-3 dapat diartikan bahwa node sebagai sebuah komputer dimana nantinya akan ditambahkan suatu fungsionalitas oleh

penggunanya. Penambahan yang sering dilakukan yaitu penambahan seperti aplikasi, protokol internet dan beberapa penambahan fungsionalitas lainnya yang dapat membuat sebuah komputer berfungsi dengan penuh.

- **Application**

Secara khusus, suatu perangkat lunak komputer terbagi dalam dua kelas yang luas. *System Software* atau sistem perangkat lunak yang mengorganisir berbagai macam resource komputer seperti memori, siklus prosesor, disk, dan lain sebagainya, berdasarkan suatu model komputasi. *System software* biasanya tidak menggunakan resource yang ada untuk menyelesaikan suatu tugas yang secara langsung berkaitan dengan user. Seorang user biasanya akan menjalankan suatu *application* yang membutuhkan dan menggunakan resource yang diatur oleh *system software* untuk mencapai suatu goal tertentu.

Pembeda antara *system software* dan *application software* adalah level privilege yang berubah saat menjalankan sistem. Pada NS-3 tidak ada konsep sesungguhnya untuk pengoperasian sistem yang khususnya pada sistem perbedaan level privilegenya. NS-3 mempunyai ide untuk aplikasi. Yaitu sebuah aplikasi perangkat lunak yang berjalan pada suatu komputer untuk menjalankan tugas di “dunia nyata”. Aplikasi NS-3 berjalan pada node NS-3 untuk menjalankan simulasi di dunia simulasi.

- **Channel**

Dalam dunia nyata, seseorang dapat menghubungkan komputer ke sebuah jaringan. Seringkali media dimana data mengalir pada jaringan ini disebut *channel* atau saluran. Ketika kita menghubungkan kabel ethernet kita ke jaringan ethernet, kita menghubungkan komputer kita ke channel komunikasi ethernet. Pada simulasi NS-3, seseorang menghubungkan node ke sebuah objek yang merepresentasikan channel

komunikasi. Pada source program C++, diwakili dalam suatu kelas Channel.

Kelas channel menyediakan metode untuk manage komunikasi pada objek dan mengkoneksikan node terhadapnya. Channel ini mungkin juga harus dikhususkan oleh pengembang pada pemrograman berbasis object-oriented.

Channel mungkin merepresentasikan suatu model yang sangat simpel seperti komunikasi kabel. Secara khusus pun bisa merepresentasikan model yang kompleks seperti model ethernet switch yang luas, ataupun wireless network.

- **Net Device**

Sudah menjadi suatu kasus umum bahwa jika kita ingin menghubungkan komputer kita ke suatu jaringan, maka kita harus memiliki sebuah kabel jaringan dan sebuah *peripheral card* yang telah memiliki beberapa fungsi jaringan biasanya disebut *NIC (Network Interface Card)*. Pada masa kini, sebagian besar komputer telah dilengkapi dengan perangkat keras jaringan ini, tetapi penggunaanya jarang melihat bentuk blok ini.

Sebuah NIC tidak akan bekerja tanpa adanya *software driver* untuk mengontrol hardware ini. Pada Unix, hardware ini diklasifikasikan sebagai device, dan dikontrol menggunakan sebuah driver, dan NIC dikontrol oleh NetDevice atau network device driver. Kita mengenal NetDevice ini dengan nama yang sudah biasa kita lihat yaitu eth0 atau wlan0 (Unix atau Linux).

Pada NS-3, NetDevice ini mewakili keduanya baik software maupun hardware. NetDevice ini di install pada sebuah node untuk memungkinkan sebuah node berkomunikasi dengan node lainnya pada simulasi melalui channel. Seperti pada kehidupan nyata, sebuah node dapat terkoneksi ke beberapa channel dengan menggunakan banyak NetDevice. NetDevice ini direpresentasikan oleh kelas NetDevice pada source C++ nya.

- **Topology Helper**

Pada jaringan yang sesungguhnya, kita akan menemukan sebuah komputer dilengkapi dengan NIC. Pada NS-3, kita akan menemukan sebuah node yang dilengkapi dengan NetDevices. Pada simulasi jaringan yang lebih besar, kita perlu mengatur banyak koneksi antar node, Netdevice dan channel yang digunakan.

Menghubungkan NetDevice ke sebuah Node, menghubungkan NetDevice ke sebuah Channel, dan menetapkan IP dan lain sebagainya adalah hal yang biasa, NS-3 menyediakan apa yang disebut dengan *Topology Helper* untuk memudahkan simulasi. Contohnya, untuk membuat NetDevice, menambahkan alamat MAC, menginstal NetDevice ke Node, konfigurasi protokol stack nodenya, dan mengkoneksikan NetDevice ke suatu Channel. Bahkan operasi yang lebih akan diperlukan untuk menghubungkan beberapa perangkat ke beberapa channel dan untuk mengkoneksikan jaringan ke jaringan yang lebih besar. NS-3 menyediakan objek topology helper yang mengkombinasikan beberapa operasi berbeda menjadi lebih mudah untuk digunakan.

1.4 Prosedur Percobaan :

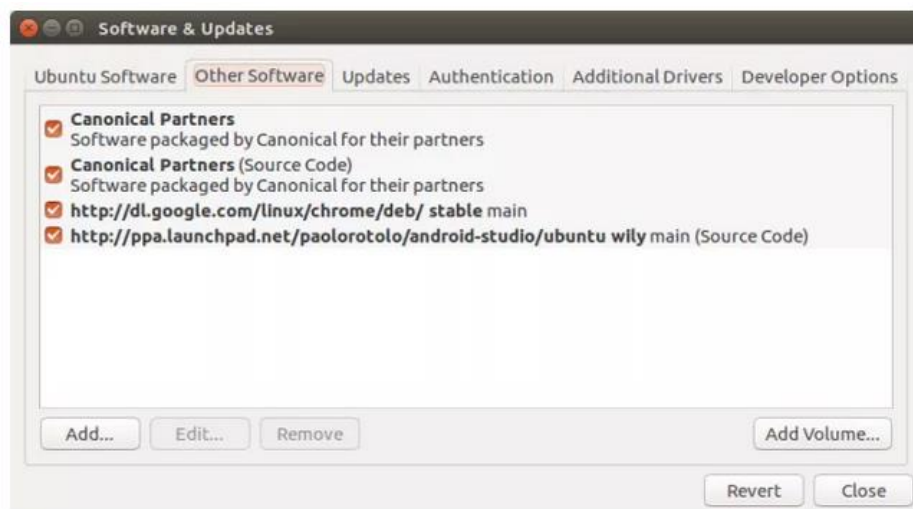
1.4.1 Instalasi NS-3

Instal packages dan library yang diperlukan sebelum mengunduh dan menginstal NS-3.

1. Cari *Software and Updates* pada Ubuntu Launcher. Klik icon *Software and Updates*.
2. Centang semua check box pada *Ubuntu Software*. Anda mungkin diminta memasukkan kata sandi untuk melakukan perubahan.



3. Klik *Other Software* dan centang semua check box pada *Other Software*.



4. Klik *Updates* dan centang semua check box pada *Updates*.



5. Klik tombol close. Kemudian, klik reload ketika Anda diminta untuk mengkonfirmasi reloading repository cache.

Setelah itu, buka Terminal dengan mencarinya di Ubuntu Launcher dan jalankan perintah - perintah berikut di Terminal.

6. Lakukan update dan upgrade sistem.

```
sudo apt-get update && sudo apt-get -y upgrade
```

7. Lakukan instalasi development libraries dan dependencies.

```
sudo apt-get -y install gcc g++ python python-dev qt4-dev-tools mercurial bzip2 cmake libc6-dev libc6-dev-i386 g++-multilib gdb valgrind gsl-bin libgsl-dev libgsl2 flex bison libfl-dev tcpdump sqlite3 libsqlite3-dev libxml2 libxml2-dev libgtk2.0-0 libgtk2.0-dev uncrustify python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev
```

8. Lakukan instalasi Wireshark.

```
sudo apt-get -y install wireshark
```

9. Instal Gnuplot, karena NS-3 memiliki dukungan bawaan untuk menghasilkan grafik menggunakan Gnuplot. Gnuplot juga memiliki antarmuka front-end grafis yang disebut PlotDrop yang mungkin berguna bagi pemula untuk membuat plot / grafik menggunakan antarmuka drag-and-drop sederhana. Untuk menginstal Gnuplot dan PlotDrop gunakan command berikut :

```
sudo apt-get -y install gnuplot plotdrop
```

10. Lakukan instalasi NS-3 Stable Branch. Untuk mengunduh dan menginstal NS-3 Stable, buat direktori baru bernama 'Software' di bawah direktori 'Home'. Untuk membuat direktori baru, lakukan dengan perintah berikut :

```
cd ~  
mkdir Softwares
```

11. Lakukan pengunduhan file ns-3.25 dan ekstrak file tersebut di dalam direktori 'Softwares' dengan perintah di bawah ini :

```
cd Softwares
wget http://www.nsnam.org/release/ns-allinone-3.25.tar.bz2
tar xvjf ns-allinone-3.25.tar.bz2
```

12. Masuklah ke dalam direktori 'ns-allinone-3.25' dengan menuliskan command seperti berikut :

```
cd ns-allinone-3.25
```

13. Lihatlah struktur dari direktori di atas menggunakan perintah 'ls'.
14. Bila proses instalasi sudah benar, maka akan muncul struktur direktori seperti di bawah ini :

```
bake          constants.py  ns-3.25      README
build.py      netanim-3.107 pybindgen-0.17.0.post49+ng0e4e3bc  util.py
```

15. Build NS-3 menggunakan command di bawah ini :

```
./build.py --enable-examples --enable-tests
```

16. Bila proses build selesai dengan sukses, maka akhirnya akan muncul tampilan seperti yang ditunjukkan di bawah ini.

```
Modules built:
antenna          aodv            applications
bridge          buildings       config-store
core            csma            csma-layout
dsv             dsr             energy
fd-net-device   flow-monitor    internet
internet-apps   lr-wpan         lte
mesh            mobility        mpi
netanim (no Python) network         nix-vector-routing
olsr            point-to-point  point-to-point-layout
propagation     sixlowpan       spectrum
stats           tap-bridge      test (no Python)
topology-read   traffic-control uan
virtual-net-device visualizer       wave
wifi            wimax

Modules not built (see ns-3 tutorial for explanation):
brite           click           openflow

Leaving directory './ns-3.25'
```


17. Di tahap ini alat waf sudah dapat digunakan. Kemudian, bangun skrip simulasi dengan menjalankan perintah berikut :

```
cd ns-3.25
./waf --run scratch/scratch-simulator
```

1.4.2 Instalasi Netanim dan Testing First File

1. Buka terminal baru, kemudian buka file NS-3 yang telah diekstrak melalui terminal.
2. Lihat isi file tersebut menggunakan perintah “ls”

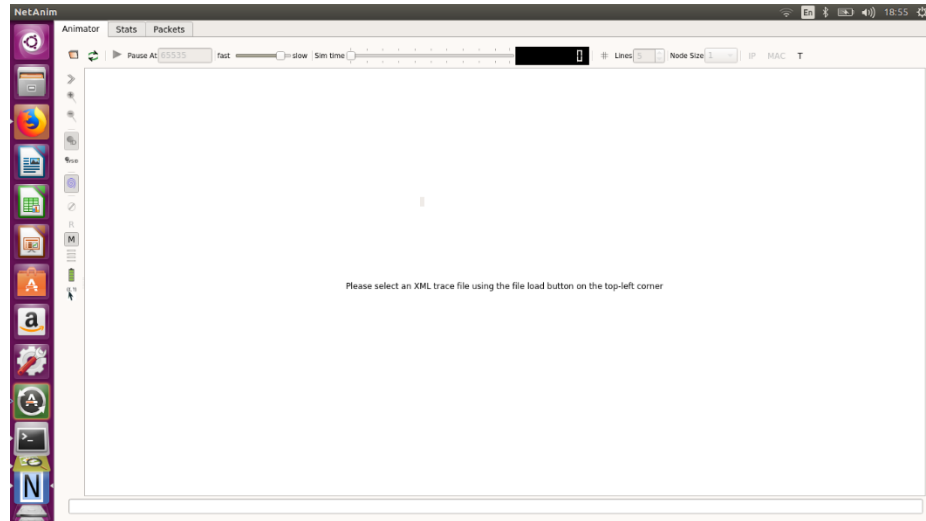
```
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25# ls
bake          constants.pyc  pybindgen-0.17.0.post49+ng0e4e3bc  util.pyc
build.py      netanim-3.107  README
constants.py  ns-3.25       util.py
```

3. Cari file ‘netanim-3.107’ dan masuk ke direktori file tersebut dengan perintah “cd netanim-3.107”
4. Lihat isi file ‘netanim-3.107’ menggunakan perintah “ls”, maka akan muncul seperti berikut.

```
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25# cd netanim-3.107/
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25/netanim-3.107# ls
abort.h                    animxmlparser.cpp          netanim-logo.png
animatorconstants.h       animxmlparser.h           NetAnim.pro
animator_fileopen.svg     assert.h                  ns-3-background.png
animator_grid.svg         battery_icon_0.png        ns-3-logo1.png
animatormode.cpp          battery_icon_1.png        ns-3-logo2.png
animatormode.h           battery_icon_2.png        packetsmode.cpp
animator_mouseposition.png battery_icon_3.png        packetsmode.h
animator_nodeid.svg       battery_icon_4.png        packetsscene.cpp
animator_nodesysid.svg    common.h                  packetsscene.h
animator_packetstats.svg  countertablesscene.cpp   packetsview.cpp
animator_pause.svg        countertablesscene.h     packetsview.h
animator_play.svg         fatal-error.cpp           qcustomplot.cpp
animator_reload.svg       fatal-error.h            qcustomplot.h
animatorscene.cpp         fatal-impl.cpp           qrc_testrc.cpp
animatorscene.h           fatal-impl.h             qtpropertybrowser
animator_showpackets.svg  flowmonstatsscene.cpp   RELEASE_NOTES
animator_step.svg         flowmonstatsscene.h     resizeableitem.cpp
animatorview.cpp          flowmonxmlparser.cpp     resizeableitem.h
animatorview.h            flowmonxmlparser.h       resources.qrc
animator_wirelesscircles.svg graphpacket.cpp           routingstatsscene.cpp
animator_zoomin.svg       graphpacket.h            routingstatsscene.h
animator_zoomout.svg      interfacestatsscene.cpp  routingxmlparser.cpp
animevent.cpp             interfacestatsscene.h    routingxmlparser.h
animevent.h               log.cpp                  statisticsconstants.h
animlink.cpp              log.h                    statsmode.cpp
animlink.h                logqt.cpp                statsmode.h
animnode.cpp              logqt.h                  statsview.cpp
animnode.h                main.cpp                 statsview.h
animpacket.cpp            Makefile                 table.cpp
animpacket.h              mode.cpp                 table.h
animpropertybrowser.cpp   mode.h                   textbubble.cpp
animpropertybrowser.h     NetAnim                  textbubble.h
animresource.cpp          netanim.cpp              timevalue.h
animresource.h            netanim.h
```

5. Ketikkan perintah “make clean”. Kemudian tuliskan perintah “qmake NetAnim.pro”.
6. Ketikkan perintah “make”.

7. Tuliskan perintah “./NetAnim” maka akan muncul tampilan berikut.



8. Untuk mencoba menjalankan first file pada netanim. Masuklah ke direktori “ns-3.25” dan kemudian masuk lagi ke direktori “scratch”.
9. Buka file “first.cc” dan edit skripnya menggunakan gedit dengan menuliskan perintah “gedit first.cc”. Maka, akan muncul tampilan seperti di bawah ini.

```
Open [?] /home/Softwares/ns-allinone-3.25/ns-3.25/scratch Save
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");

int
main (int argc, char *argv[])
{
    Time::SetResolution (Time::NS);
    LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
    LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);

    NodeContainer nodes;
    nodes.Create (2);

    PointToPointHelper pointToPoint;
```

10. Edit skripnya dengan menambahkan `#include "ns3/netanim-module.h"` seperti berikut. Kemudian, klik save.

```
first.cc
/home/Softwares/ns-allinone-3.25/ns-3.25/scratch
Save

/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

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

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
```

11. Masih dalam satu file yang sama, tambahkan skrip untuk mengatur posisi node seperti di bawah ini. Dimana yang berada di dalam kotak hijau merupakan posisi node pada koordinat X, dan di dalam kotak kuning merupakan posisi node pada koordinat Y. Kemudian, simpan perubahan skrip tersebut.

```
*first.cc
/home/Softwares/ns-allinone-3.25/ns-3.25/scratch
Save

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

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

InternetStackHelper stack;
stack.Install (nodes);

Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer interfaces = address.Assign (devices);

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));

AnimationInterface anim ("anim1.xml");
anim.SetConstantPosition(nodes.Get(0), 1.0, 2.0);
anim.SetConstantPosition(nodes.Get(1), 2.0, 3.0);

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

C++ Tab Width: 8 Ln 22, Col 32 INS
```

12. Klik *Ctrl* + *C* untuk keluar dari gedit.

13. Tuliskan perintah berikut pada terminal.

```
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25/ns-3.25/scratch# cd ..
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25/ns-3.25# ./waf --run scratch/first
```

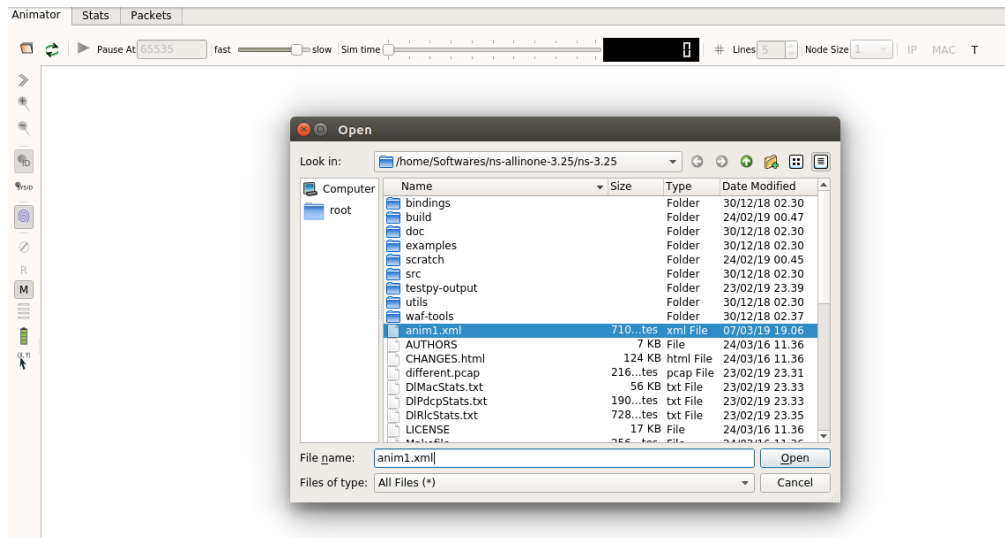
Maka, akan muncul tampilan seperti berikut.

```
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25/ns-3.25/scratch# cd ..
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25/ns-3.25# ./waf --run scratch/first
waf: Entering directory '/home/Softwares/ns-allinone-3.25/ns-3.25/build'
[ 910/2572] Compiling scratch/first.cc
[2545/2572] Linking build/scratch/first
waf: Leaving directory '/home/Softwares/ns-allinone-3.25/ns-3.25/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.362s)
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
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
```

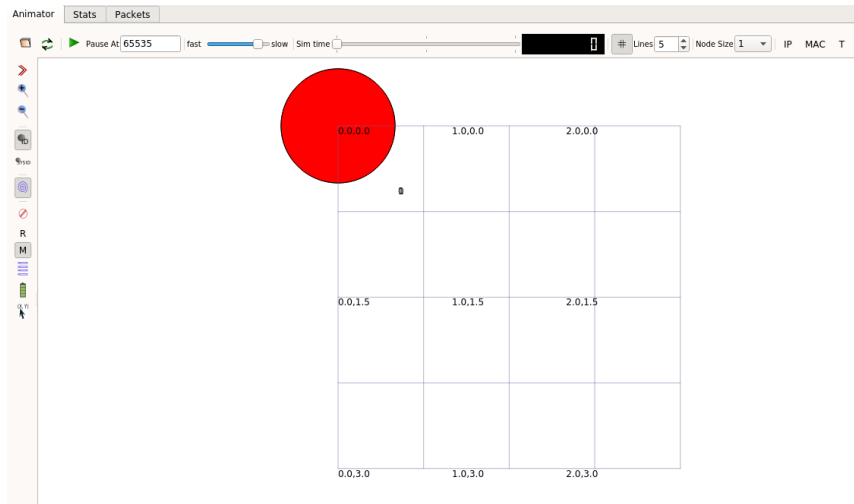
14. Run netanim dengan menuliskan command berikut pada terminal.

```
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25/ns-3.25# cd ..
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25# ls
bake          constants.pyc  pybindgen-0.17.0.post49+ng0e4e3bc  util.pyc
build.py      netanim-3.107  README
constants.py  ns-3.25       util.py
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25# cd netanim-3.107/
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25/netanim-3.107# ./NetAnim
```

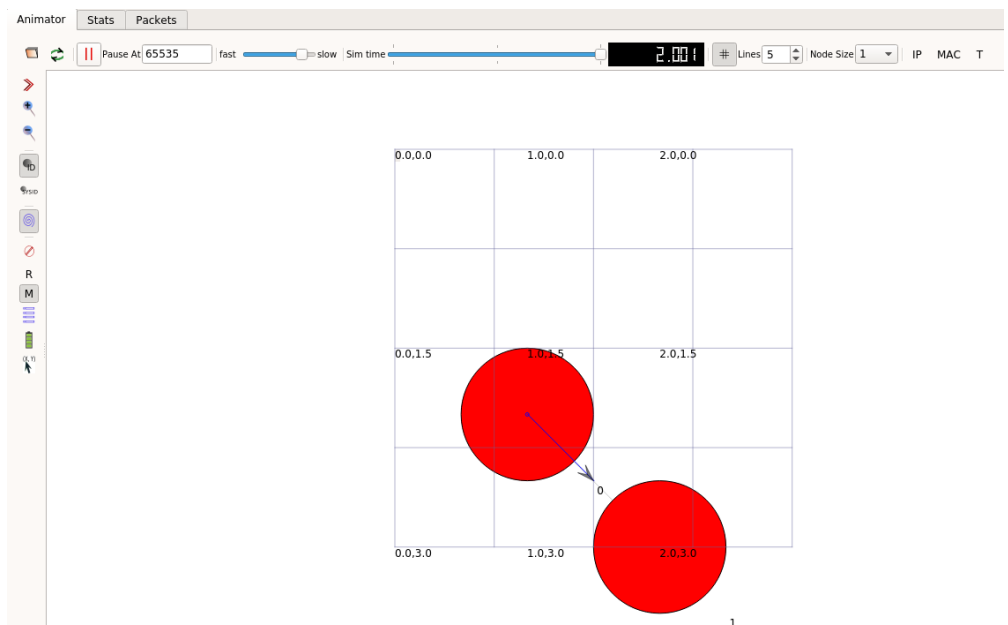
15. Buka file dengan nama *anim1.xml*



Maka, akan muncul tampilan seperti di bawah ini.



16. Coba run simulasinya dengan klik tombol play, hingga node berjalan.



1.4.3 Instalasi Tracemetrics

1. Instal JDK dengan menuliskan command berikut.

```
sudo apt install default-jdk
```

2. Copy file *tcp-bulk-send.cc* dari direktori *ns-allinone-3.25/ns-3.25/examples/tcp* dan paste file tersebut di direktori *ns-allinone-3.25/ns-3.25/scratch*.

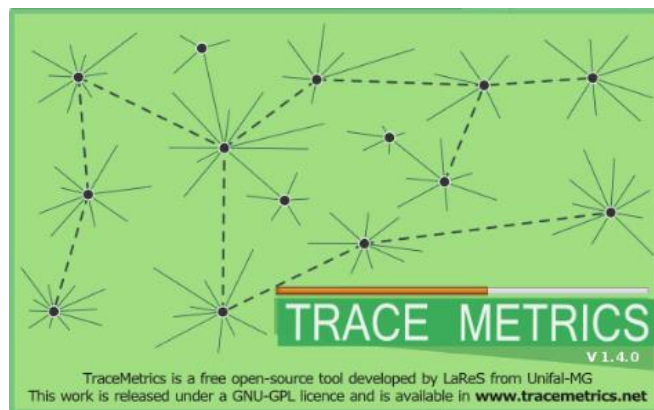
3. Buka file direktori *ns-allinone-3.25/ns-3.25* melalui terminal, kemudian jalankan file *tcp-bulk-send.cc* dengan menuliskan perintah seperti di bawah ini.

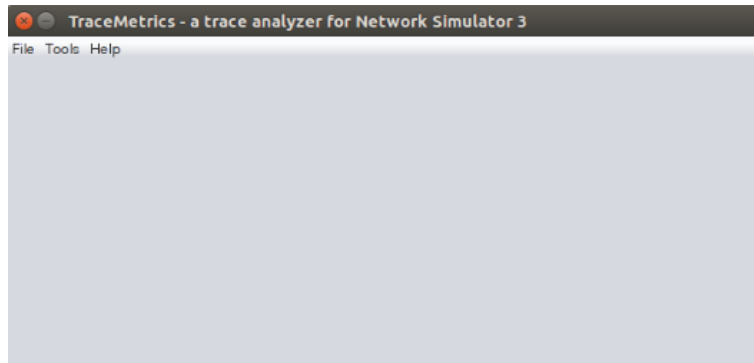
```
root@dina-Inspiron-5458:/home/Softwares/ns-allinone-3.25/ns-3.25# ./waf --run "s
cratch/tcp-bulk-send --tracing=true"
```

4. Jika proses running berhasil, carilah file *tcp-bulk-send.tr* pada direktori *ns-allinone-3.25/ns-3.25*.
5. Jika Anda menemukan file *tcp-bulk-send.tr*, maka trace file Anda siap dianalisis. *tcp-bulk-send.tr* atau file dengan format *.tr* adalah trace file yang akan dianalisis oleh Tracemetrics.
6. Download Tracemetrics pada alamat <http://sourceforge.net/projects/tracemetrics/>.
7. Copy file yang telah didownload pada direktori */home/Tracemetrics*.
8. Ekstrak file tracemetrics yang telah didownload pada folder tersebut.
9. Buka tracemetrics melalui terminal dengan menuliskan command berikut.

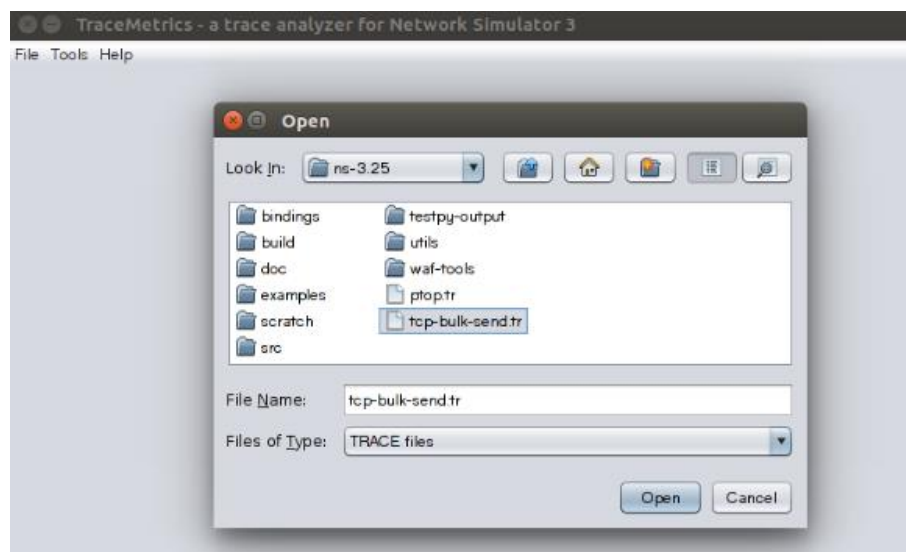
```
root@dina-Inspiron-5458:/home/Softwares/tracemetrics-1.4.0# java -jar "tracemetr
ics.jar"
```

10. Setelah tracemetrics terbuka, maka akan muncul tampilan seperti di bawah ini.

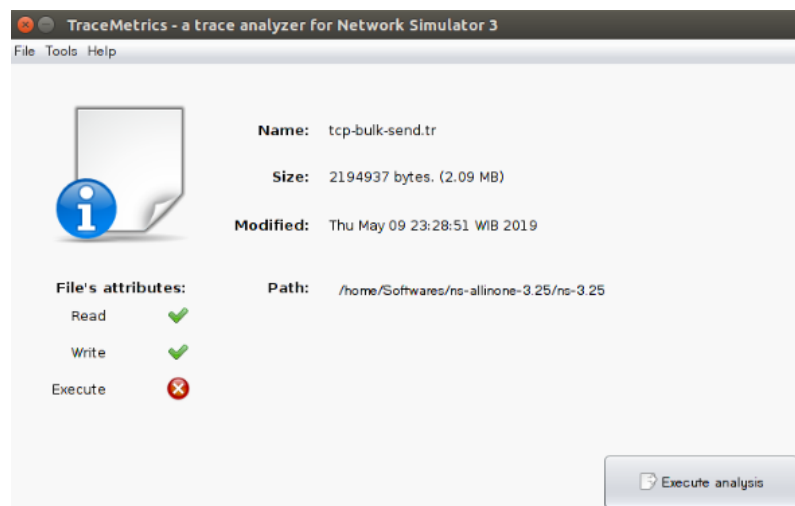




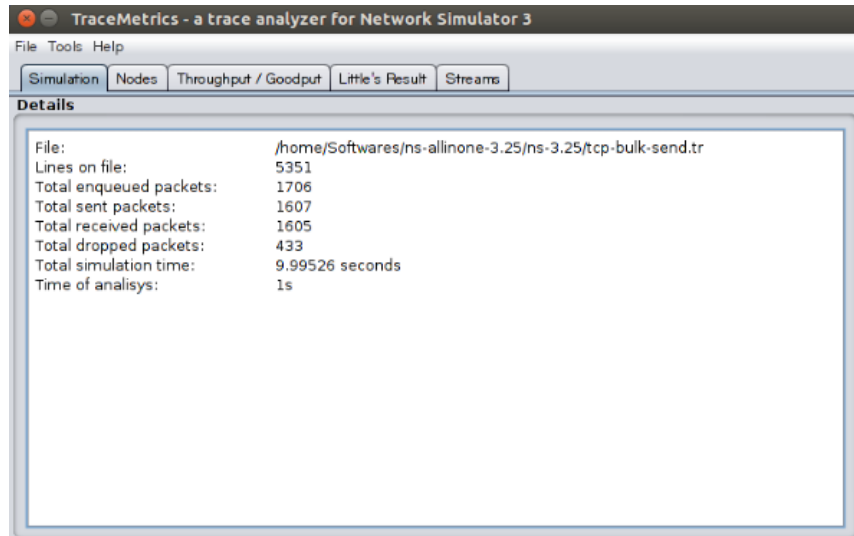
11. Untuk mencoba menjalankan file *tcp-bulk-send.tr*, klik file → choose file dan cari file *tcp-bulk-send.tr* seperti berikut.



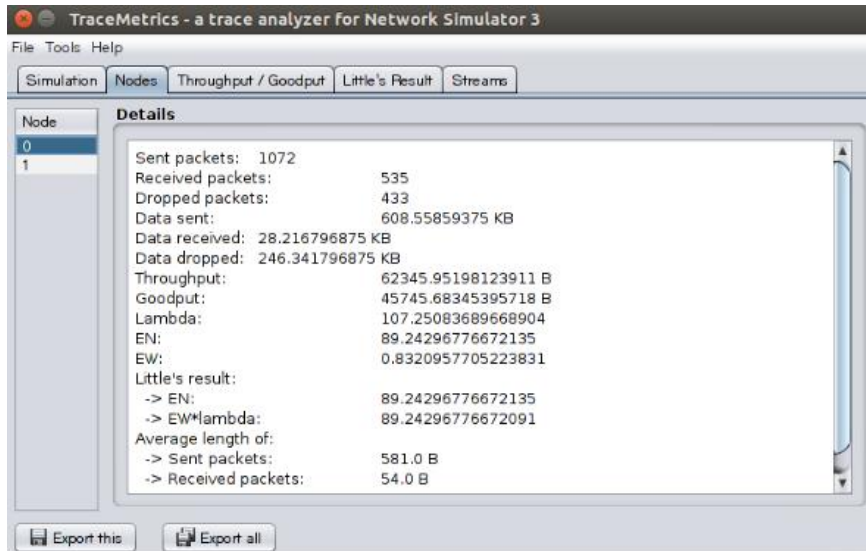
12. Setelah klik open, maka akan muncul tampilan seperti di bawah ini.



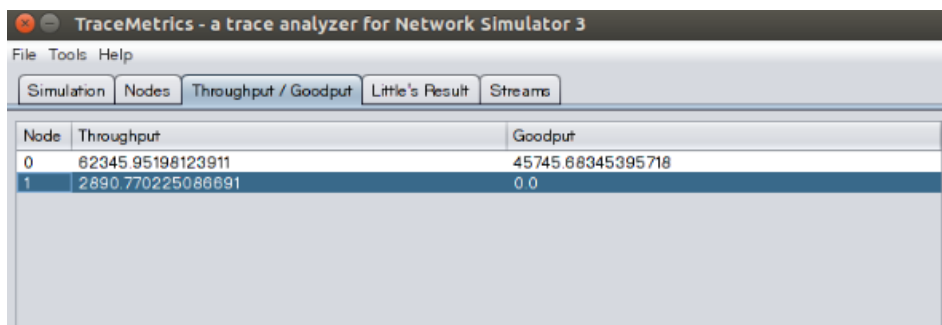
13. Klik Execute analysis, maka akan muncul tampilan berikut. Dimana, pada tab Simulation ini menunjukkan parameter simulasi umum.



Untuk tab Nodes menampilkan beberapa statistic simpul node dari dua node yang terlibat dalam simulasi ini.



Tab berikutnya adalah tab Throughput / Goodput, dimana tab ini menampilkan throughput dan goodput dari dua node yang terlibat dalam simulasi ini.



Tab selanjutnya adalah tab Little's Result, dimana tab ini menampilkan besar lambda dan hal - hal terkait lainnya dari dua node yang terlibat dalam simulasi ini.

Node	Lambda	E[W]	E[N]	E[W] * Lambda
0	107.25083689668904	0.8320957705223831	89.24296776672135	89.24296776672091
1	53.525371025866264	0.0	0.0	0.0

Sedangkan, untuk tab yang terakhir adalah tab Streams. Tab ini menampilkan statistik aliran data dari semua aliran yang terlibat dalam skenario jaringan dua node tersebut.

Stream	IPs	Ports	Number of packets	Average Delay	Delay variance	Average PDV	PDV variance	Average IPDV	IPDV variance
Stream 0	10.1.1.2 --> 10.1.1.1	9 --> 49153	535	0.0058676186915887635	2.3285043229737743E-11	7.618691590341779E-6	2.32415197844484E-11	-1.0861423221007548E-7	1.9681086845348262E-11
Stream 1	10.1.1.1 --> 10.1.1.2	49153 --> 9	1070	0.0067926242990654605					