

PERCOBAAN 6

SIMULASI MANET (Mobile Adhoc Network)

6.1 Tujuan :

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu :

- Memahami berbagai jenis mobility model
- Memahami pergerakan node di ns-3

6.2 Peralatan :

- 1 PC dilengkapi dengan OS Ubuntu 18.04
- Software NS-3 versi 3.29
- Software bonnmotion

6.3 Teori :

Model mobilitas adalah salah satu parameter penting perlu diperhatikan saat ingin membangun sebuah simulasi dalam lingkungan MANET (Rohankar, 2012). Model mobilitas adalah sebuah teknik dalam simulasi yang menentukan pola pergerakan node, bagaimana letak posisi node, percepatan dan perubahan kecepatan node setiap waktu. Maka dari itu, pemilihan model mobilitas dapat menjadi pengaruh besar terhadap performa dari routing protokol. Berikut adalah model mobilitas:

1. Random Waypoint Mobility Model

Node pada model mobilitas Random Waypoint disebarkan secara acak pada simulasi. Masing-masing node bergerak secara independen dan terpisah dari node yang lain. Pertama yang dilakukan adalah setiap node memilih target destinasi secara acak kemudian memilih kecepatan pergerakan. Kecepatan perpindahan dipilih secara acak sesuai dengan interval. Setelah sampai di destinasi, node kemudian berhenti untuk waktu yang sudah ditentukan (pause time) sebelum kemudian mulai melakukan prosedur yang sama lagi.

2. Random Walk Mobility Model

Node pada model mobilitas Random Walk bergerak dari lokasi awal menuju lokasi yang baru dengan menentukan kecepatan dan arah yang akan dituju secara acak.

Selain kecepatan yang sudah ditentukan dengan interval, arah destinasi juga sudah ditentukan dengan interval. Destinasi baru akan ditentukan setelah node telah bergerak sesuai waktu yang sudah ditentukan atau jarak yang sudah dilalui. Secara sederhana, model mobilitas Random Walk adalah model mobilitas Random Waypoint tanpa adanya pause time.

3. Random Direction Mobility Model

Model mobilitas Random Direction dikembangkan dengan tujuan untuk mengurangi masalah density wave yang terjadi pada model mobilitas Random Waypoint. Density wave adalah penumpukan node yang terjadi pada satu sisi field simulasi. Pada Random Direction, node memilih arah pergerakan secara acak kemudian bergerak menuju arah tersebut. Berbeda dengan Random Waypoint, node baru akan berhenti dengan waktu yang telah ditentukan (pause time) saat mencapai batas field simulasi. Setelah itu, node akan memilih arah baru dan bergerak sehingga node dapat tersebar secara merata.

6.4 Prosedur Percobaan

1. Download bonnmotion: bonnmotion-2.1a.zip
2. Instalasi bonnmotion di ubuntu:


```
# apt-get update
# apt-get install openjdk-6-jdk openjdk-6-jre
# java -version
# cd bonnmotion-2.1a
# ./install
```
3. Copy file bonnmotion-ns2-example.cc ke folder scratch:


```
# cp src/mobility/examples/bonnmotion-ns2-example.cc scratch/
```
4. Lakukan modifikasi pada program berikut:


```
# gedit scracth/bonnmotion-ns2-example.cc
```

Coding 1: bonnmotion

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
2 /*
3  * Copyright 2012 Eric Gamess
4  *
5  * This program is free software; you can redistribute it and/or modify
6  * it under the terms of the GNU General Public License version 2 as
7  * published by the Free Software Foundation;
```

```

8  *
9  * This program is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13 *
14 * You should have received a copy of the GNU General Public License
15 * along with this program; if not, write to the Free Software
16 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
17 *
18 */
19
20 /*
21 * Example program using a ns-2-formatted mobility trace generated
22 * by the BonnMotion mobility framework.
23 *
24 * With the provided tracefile (bonnmotion.ns_movements), the movements of
25 * one node are simulated for 1000 seconds. There are a few other files
26 * that provide metadata about the mobility:
27 * - src/mobility/examples/bonnmotion.ns_params
28 * - src/mobility/examples/bonnmotion.params
29 *
30 * These files are documented in the BonnMotion documentation.
31 *
32 * It is important to remember that the trace file dictates how many nodes
33 * (in this case, one) and how long (in this case, 1000 seconds) that the
34 * ns-3 program should use. If you want to change the mobility pattern,
35 * number of nodes, or duration, you need to use BonnMotion or another
36 * tool to generate a new trace.
37 *
38 * Finally, note that you can visualize this program using the pyviz
39 * visualizer:
40 * ./waf --run bonnmotion-ns2-example --vis
41 */
42
43 #include "ns3/core-module.h"
44 #include "ns3/mobility-module.h"
45 //additional animation
46 #include "ns3/ns2-mobility-helper.h"
47 #include "ns3/netanim-module.h"
48
49 using namespace ns3;
50
51 void showPosition (Ptr<Node> node, double deltaTime)
52 {
53   uint32_t nodeId = node->GetId ();
54   Ptr<MobilityModel> mobModel = node->GetObject<MobilityModel> ();
55   Vector3D pos = mobModel->GetPosition ();
56   Vector3D speed = mobModel->GetVelocity ();
57   std::cout << "At " << Simulator::Now ().GetSeconds () << " node " << nodeId
58   << ": Position(" << pos.x << ", " << pos.y << ", " << pos.z
59   << "); Speed(" << speed.x << ", " << speed.y << ", " << speed.z
60   << ")" << std::endl;
61   Simulator::Schedule (Seconds (deltaTime), &showPosition, node, deltaTime);
62 }

```

```

62 int main (int argc, char *argv[])
63 {
64     std::cout.precision (2);
65     std::cout.setf (std::ios::fixed);
66
67     double deltaTime = 100;
68     std::string traceFile = "src/mobility/examples/bonnmotion.ns_movements";
69
70     CommandLine cmd (__FILE__);
71     cmd.AddValue ("traceFile", "Ns2 movement trace file", traceFile);
72     cmd.AddValue ("deltaTime", "time interval (s) between updates (default 100)",
73     deltaTime);
74
75     cmd.Parse (argc, argv);
76
77     Ptr<Node> n0 = CreateObject<Node> ();
78
79     Ns2MobilityHelper ns2 = Ns2MobilityHelper (traceFile);
80     ns2.Install ();
81
82     Simulator::Schedule (Seconds (0.0), &showPosition, n0, deltaTime);
83
84     //Animation configuration lines
85     AnimationInterface anim ("bonnmotion.xml");
86     anim.SetMobilityPollInterval (Seconds(1));
87     anim.EnablePacketMetadata(true);
88     //end of animation
89
90
91
92     Simulator::Stop (Seconds (1000.0));
93     Simulator::Run ();
94     Simulator::Destroy ();
95     return 0;
96 }

```

5. Jalankan netanim:

```
# cd /home/zenhadi/ns-allinone-3.29/netanim-3.108
```

```
# ./NetAnim => pilih file bonnmotion.xml
```

6. Hasilnya berupa data 1 buah node yang bergerak dengan Random Waypoint Mobility.



Integrasi bonnmotion dengan ns-3

1. Buat movement di bonnmotion berupa Random Waypoint Mobility (RWP)

```
# cd /home/zenhadi/bonnmotion/bonnmotion-3.0.1/bin
# ./bm -f rwp1 RandomWaypoint -n 100 -d 1000 -x 100 -y 100 -h 2 -p 0 -R 0
# ./bm NSFile -f rwp1
```

Akan terdapat 2 file:

- a. rwp1.ns_params
- b. rwp1.ns_movements

2. Copy-kan kedua file tersebut di folder berikut:

```
std::string traceFile = "src/mobility/examples/bonnmotion.ns_movements";
# cp rwp1* /home/zenhadi/ns-allinone-3.29/ns-3.29/src/mobility/examples/
```

3. Modifikasi “**coding 1**” diatas di bonnmotion-ns2-example.cc

```
double deltaTime = 100;
//std::string traceFile = "src/mobility/examples/bonnmotion.ns_movements";
std::string traceFile = "src/mobility/examples/rwp1.ns_movements";

CommandLine cmd;
cmd.AddValue ("traceFile", "Ns2 movement trace file", traceFile);
cmd.AddValue ("deltaTime", "time interval (s) between updates (default 100)", deltaTime);
cmd.Parse (argc, argv);

//Ptr<Node> n0 = CreateObject<Node> ();
NodeContainer nodes;
nodes.Create (100);

Ns2MobilityHelper ns2 = Ns2MobilityHelper (traceFile);
ns2.Install ();

//Simulator::Schedule (Seconds (0.0), &showPosition, n0, deltaTime);
//Simulator::Schedule (Seconds (0.0), &showPosition, nodes, deltaTime);

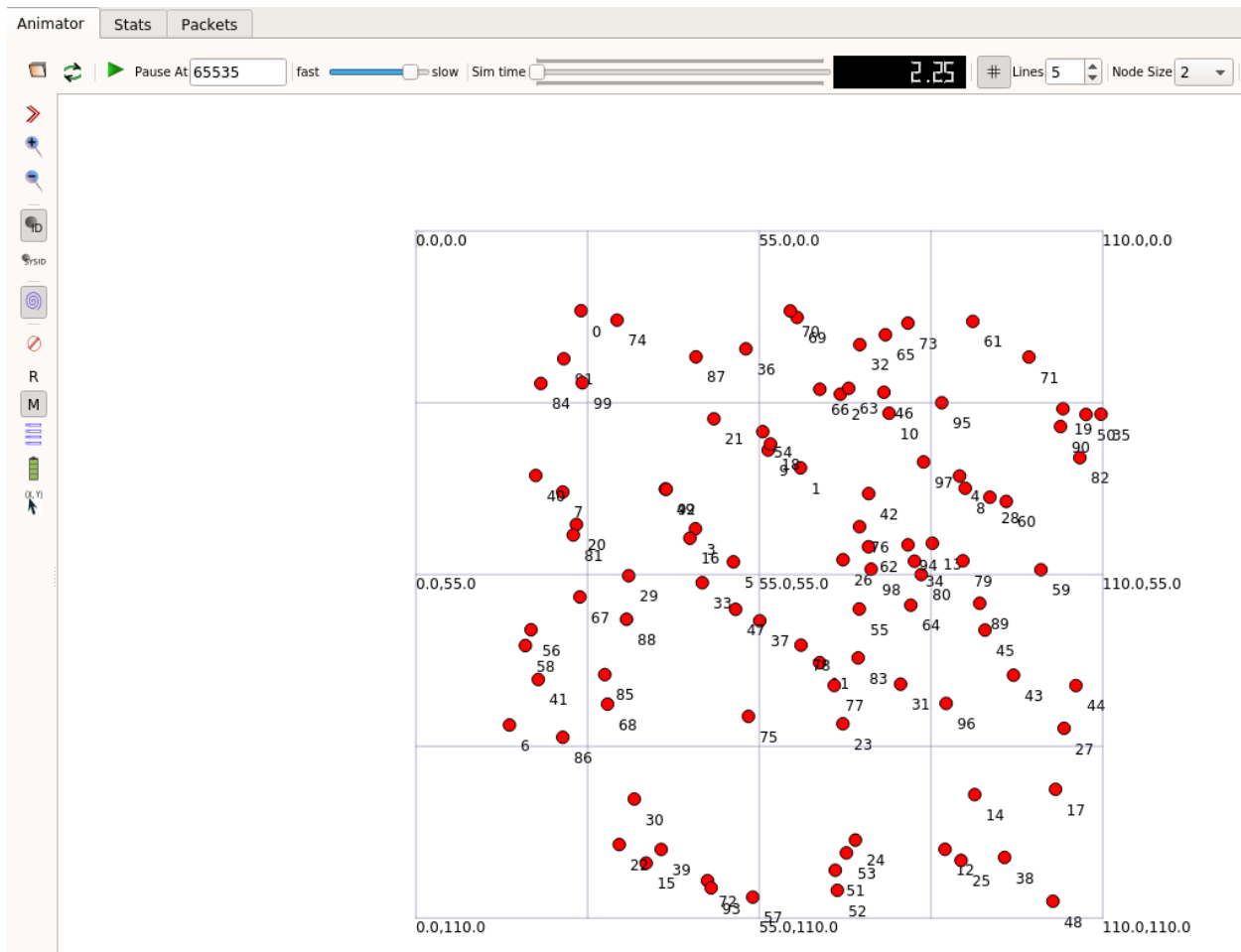
//Animation configuration lines
AnimationInterface anim ("bonnmotion.xml");
anim.SetMobilityPollInterval (Seconds(1));
anim.EnablePacketMetadata(true);
```

Keterangan:

Tambahkan node sebanyak 100 seperti di bonnmotion.

Di perintah ns2.install, secara otomatis akan mengintegrasikan mobility dengan node yang telah dibuat.

4. Jalankan di netanim



Buatlah Mobility model yang lainnya:

1. Random Waypoint Mobility (RWP)
2. Gauss-Markov Model
3. Random Walk
4. Nomadic Community Model
5. Reference Point Group Mobility (RPGM)

Contoh generate mobility model:

A. RANDOMWAYPOINT

#!/bm -hm RandomWaypoint

App:

-D print stack trace

Scenario:

-a <attractor parameters (if applicable for model)>

-c [use circular shape (if applicable for model)]

-d <scenario duration>

-i <number of seconds to skip>

-n <number of nodes>

-x <width of simulation area>

-y <height of simulation area>

-R <random seed>

RandomSpeedBase:

-h <max. speed>

-l <min. speed>

-p <max. pause time>

RandomWaypoint:

-o <dimension: 1: x only, 2: x or y, 3: x and y>

B. Nomadic:

-a <average no. of nodes per group>

-r <max. distance to group center [m]>

-s <group size standard deviation>

-c <reference point max. pause>

C. RPGM:

-a <average no. of nodes per group>

-c <group change probability>

-r <max. distance to group center>

-s <group size standard deviation>

D. DisasterArea:

-a <average no. of nodes per group>

-b <catastrophe area (can be used multiple times for several catastrophe areas)>

-c <group change probability>

-e <max catastrophe areas>

-r <max. distance to group center>

-s <group size standard deviation>

-O <obstacle for only one group (specified in last param)>

-w <write vis. info to file & show movements>

-v <write vis. info to file & show vis.graph>

-K <do not knock over pedestrians - no ambulances in areas beside APP>

E. RandomWalk:

-t x (mode: [T]ime. Nodes will walk until time interval x is up)

OR (do not use both -t and -s)

-s x (mode: Di[s]tance. Nodes will walk until distance x has been covered)

F. ProbRandomWalk:

-t <time interval to advance by>

COMMAND:

RWP: ./bm -f rwp1 RandomWaypoint -n 10 -d 100 -x 100 -y 100 -h 5 -p 5 -R 0
dimana

h	maxspeed
p	pause
R	to get the same random

GaussMarkov : ./bm -f gmv1 GaussMarkov -n 10 -d 100 -x 100 -y 100 -h 5 -p 5

RPGM : ./bm -f rpgm1 RPGM -n 10 -d 100 -x 100 -y 100 -h 5 -p 5 -a 3 -c 0 -r 2

Static: ./bm -f static1 Static -n 10 -d 100 -x 80 -y 80

REFERENSI:

1. https://www.nsnam.org/doxygen/classns3_1_1_ns2_mobility_helper.html#aafed6f99e77c1fc74192da4d54e0eae7, mobility helper class
2. <https://www.nsnam.org/docs/release/3.30/models/singlehtml/index.html#document-mobility>, ns-3 model library
3. <https://www.nsnam.org/docs/models/html/animation.html>, animation netanim
4. https://groups.google.com/g/ns-3-users/c/Z1UnbY_P7rc, mobility movements ns2 to ns3