

PERCOBAAN 4

KONFIGURASI MOBILITAS NODE DI JARINGAN NIRKABEL MENGUNAKAN NS-2

4.1. Tujuan:

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu:

- Membuat pemrograman simulasi jaringan nirkabel dengan posisi node yang random
- Membuat pemrograman simulasi jaringan dengan mobilitas node menggunakan kecepatan tertentu
- Menjalankan program Tcl pada NS2
- Menampilkan hasil simulasi dengan nam, trace file dan xgraph

4.2. Peralatan:

- OS Ubuntu 10.04
- Simulator NS-2.34 ke atas

4.3. Teori

4.3.1. Mobile Node

Sebuah jaringan nirkabel tersusun atas beberapa mobile node yang tersebar di sebuah bidang observasi. Node-node tersebut bergerak dengan kecepatan tertentu dari sebuah posisi ke posisi tujuan yang telah ditentukan ataupun secara random. Masing-masing node dapat berkomunikasi satu dengan yang lain secara nirkabel, dan dapat saling mengirimkan paket informasi.

Pada NS-2, sebuah mobile node tersusun dari class MobileNode pada C++ yang terikat dengan class Node/MobileNode pada OTcl. Secara hirarki pada C++, class MobileNode diturunkan dari class Node, sedangkan pada OTcl, class Node/MobileNode berada di level tertinggi. Program C++ pada gambar 4.1 terdiri dari mobile-node.h dan mobile-node.cc, menunjukkan deklarasi sebuah MobileNode.

Sebuah mobile node reguler berdasarkan konfigurasi dan mekanisme paketyang di-forward, dibedakan menjadi 3 macam:

1. *Source Node* – Pada node ini, paket-paket dibangkitkan oleh sebuah aplikasi. Paket akan dikirimkan ke destination address melalui address classifier. Destination address bisa tidak sama dengan address node sendiri. Address classifier memforward ke destination address melalui routing agent dengan perintah “default_target_”

```
//~ns/common/mobile-node.h
1 class MobileNode : public Node
2 {
3 protected:
4     double X_,Y_,Z_;           //the current location
5     double speed_;            //in meter/s
6     double dX_,dY_,dZ_;       //Unit
7     double destX_,destY_;     //the destination
8     event po_intr_;
9     double position_update_time;
10    double position_update_interval_; //Update interval
11    PositionHandler pos_handle_; //For random-motion only
12 private:
13    int random_motion_; //Are we running random motion?
14    Topography *T_; //Define an area for moving
15    LIST_ENTRY(MobileNode) link_; //A global list of mobile nodes
16 };

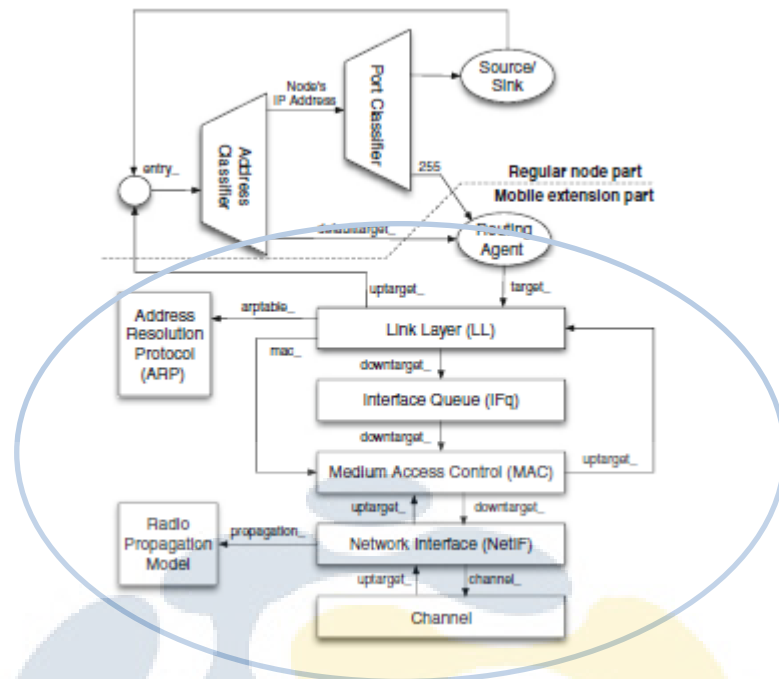
//~ns/common/mobile-node.cc
17 static class MobileNodeClass : public TclClass {
18 public:
19     MobileNodeClass() : TclClass("Node/MobileNode") {}
20     TclObject* create(int, const char*const*) {
21         return (new MobileNode);
22     }
23 } class_mobilenode;
```

Gambar 4.1. Deklarasi Mobile Node pada C++

2. *Destination Node* – Paket yang diterima pada destination node ini melewati link layer, atas perintah address classifier. Di sini, destination address akan sama dengan address node. Paket-paket selanjutnya diarahkan ke port classifier yang akan memforward paket tersebut ke aplikasi sink yang tersedia.
3. *Forwarding Node* – Paket-paket dilewatkan melalui link layer ke node entry dan address classifier, namun karena header dari paket tidak sama dengan address node, makam paket diforwardkan kembali melalui routing agent.

4.3.2. Ekstensi dari Mobile Node

Ekstensi dari sebuah mobile node seperti ditunjukkan pada bagian lingkaran biru di gambar 4.2 bagian bawah. Komponen-komponennya dispesifikasikan dengan opsi-opsi seperti pada Tabel 4.1.



Gambar 4.2. Ekstensi dari Mobile Node

Tabel 4.1. Komponen dan Opsi Ekstensi Mobile Node

Opsi	instvar	Deskripsi
-adhocRouting	routingAgent_	Protocol ruting (mis: AODV,DSDV)
-llType	lltype_	Tipe link layer (mis: LL)
-macType	macType_	Tipe MAC (Mac/802_11)
-ifqType	ifqType_	Tipe ifq Queue/DropTail/PriQueue
-ifqLen	ifqlen_	Ukuran buffer untuk paket (mis: 5 paket)
-antType	antType_	Tipe Antena (mis:Antenna/OmniAntenna)
-propType	propType_	Tipe propagasi radio (mis: Propagation/TwoRayGround)
-phyType	phyType_	Tipe Network interface (Phy/WirelessPhy)
-channelType	channelType_	Tipe kanal (Channel/WirelessChannel)
-topoInstance	topoInstance_	Instance oTcl yang mengidentifikasi topografi (new Topology)
-agenTrace	agentTrace_	Meng-ON/OFF kan agent Trace
-routerTrace	routerTrace_	Meng-ON/OFF kan router trace
-macTrace	macTrace_	Meng-ON/OFF kan MAC trace
-movementTrace	movementTrace_	Meng-ON/OFF kan movement trace

Untuk membentuk sebuah mobile node, dilakukan 2 proses:

1. Konfigurasi Mobile Node

Pada proses ini informasi konfigurasi pada **instvar** yang ada di dalam obyek Simulator akan disimpan menggunakan prosedur instproc berikut:

```
$ns node-config -<option><value>
```

Adapun \$ns adalah nama obyek Simulator, sedangkan <option> dan <value> nya dinyatakan pada tabel 4.1.

Contoh konfigurasi mobile node diberikan sebagai berikut:

```
$ns node-config      -adhocRouting AODV \  
                    -llType LL \  
                    -macType Mac/802_11 \  
                    -ifqType Queue/DropTail/PriQueue \  
                    -ifqLen 50 \  
                    -antType Antenna/OmniAntenna \  
                    -propType Propagation/TwoRayGround \  
                    -phyType Phy/WirelessPhy \  
                    -channelType Channel/WirelessChannel \  
                    -topoInstance $topo \  
                    -agentTrace ON \  
                    -routerTrace ON \  
                    -macTrace OFF \  
                    -movementTrace OFF
```

Konfigurasi di atas menjelaskan bahwa protocol routing yang digunakan adalah tipe AODV, dengan link layer bertipe LL, protocol Mac nya adalah IEEE 802.11. Setiap Mobile node memiliki buffer yang dapat menyimpan hingga 50 paket. Antrian layanan menggunakan priority queue, tipe antenna adalah omni directional dengan propagasi model two ray. Tipe physical layer dan kanal kedua-duanya adalah wireless. Agent Trace dan router Trace dibuat ON, sedangkan Mac Trace dan movement Trace dibuat OFF.

Prosedur instproc untuk konfigurasi mobile node yang lengkap ada di //~ns/lib/nc-lib.tcl, yaitu Simulator instproc node-config {}

2. Konstruksi Mobile Node

Konstruksi sebuah mobile node dibuat dengan perintah oTcl sebagai berikut:

```
$ns node
```

Perintah membuat sebuah node diawali dengan class node pada instproc node di `//~ns/lib/nc-lib.tcl`, yang akan memanggil instproc `create-wireless-node` arg {} dan `create-node-instance` arg {}. Sebagian dari pemrograman oTcl tersebut ditunjukkan pada gambar 4.3.

```
//~ns/tcl/lib/ns-lib.tcl
1 Simulator instproc node args {
2 $self instvar Node_ routingAgent_ wiredRouting_
  satNodeType_
3 if { [info exists routingAgent_] && ($routingAgent_
  != "") } {
4 set node [eval $self create-wireless-node $args]
5 ...
6 }
7 return $node
8 }
9 ...
10 return $node
11 }

//~ns/tcl/lib/ns-lib.tcl
1 Simulator instproc create-wireless-node args {
2 ...
3 set node [eval $self create-node-instance $args]
4 switch -exact $routingAgent_ {
5 AODV {
6 set ragent [$self create-aodv-agent $node]
7 }
8 DSR {
9 $self at 0.0 ``$node start-dsr"
10 }
11 ...
12 default {
13 eval $node addr $args
14 puts "Wrong node routing agent!"
15 exit
16 }
17 }
18 $node add-interface $chan $propInstance_ $llType_
  $macType_ $ifqType_ $ifqlen_ $phyType_ $antType_
  $topoInstance_
19 $node attach $ragent [Node set ragent_port_]
20 $node topography $topoInstance_
21 return $node
22 }
23 Simulator instproc create-node-instance args {
24 set nodeclass Node/MobileNode
25 return [eval new $nodeclass $args]
26 }
```

Gambar 4.3. Instproc `create-wireless-node` dan turunannya pada `nslib.tcl`

Prosedur-prosedur instproc yang terlibat di dalam pembentukan wireless node tersebut berasal dari class-class yang berbeda, dimana hubungan antar kelas dan instproc-nya ditunjukkan pada gambar 4.4.

Class	Instproc
Simulator	create-node-instant create-aodv-agent
Node/Mobile Node	add-interface
Node	attach
Mobile Node	topography

Gambar 4.4. Hubungan antar kelas dan instproc pada pembentukan wireless-node

Penjelasan masing-masing instproc di atas adalah sebagai berikut:

1. Instproc create-aodv-agent milik class Simulator, untuk membangkitkan agen ruting AODV
2. Instproc add-interface pada class Node/MobileNode digunakan untuk input agen ruting
3. Instproc attach pada class Node digunakan untuk meletakkan agen ruting ke obyek node
4. Instproc topography pada class MobileNode di C++ digunakan untuk menyatakan ukuran luasan topografi dari obyek Mobile Node.

4.3.3. *Mobilitas Node*

Salah satu contoh syntax Tcl untuk simulasi wireless node tunjukkan pada gambar 4.5, dimana konfigurasi mobilitas node terdiri dari 4 tahap:

1. Membangun topologi
Pada program di atas, baris 1 membangun obyek topologi, dan baris 2 menyatakan wilayah dimana Mobile Node akan melakukan mobilitas
2. Konfigurasi GOD (akan dijelaskan kemudian)
Baris 3 membangun GOD dan menginformasikan bahwa simulasi akan dijalankan dengan **nn** buah node .
3. Inisialisasi lokasi
Baris 8 sampai 13 menyatakan lokasi masing-masing node dengan koordinat 3 dimensi
4. Spesifikasi pola mobilitas

Baris 6 menyatakan bahwa mobilitas node pada simulasi ini adalah random deterministic, dan baris 14 sampai 16 menyatakan pergerakan node menggunakan instproc setdest {...} dari class MobileNode dengan sintaks:

\$node setdest <dest_x><dest_y><speed>

```
//~ns/tcl/ex/simple-wireless.tcl
1 set topo [new Topography]
2 $topo load_flatgrid 500 500
3 create-god $val(nn)
4 for {set i 0} {$i < $val(mn)} {incr i} {
5 set node_($i) [$ns_ node]
6 $node_($i) random-motion 0;# disable random
  motion
7 }
8 $node_(0) set X_ 5.0
9 $node_(0) set Y_ 2.0
10 $node_(0) set Z_ 0.0
11 $node_(1) set X_ 390.0
12 $node_(1) set Y_ 385.0
13 $node_(1) set Z_ 0.0
14 $ns_ at 50.0 "$node_(1) setdest 25.0 20.0 15.0"
15 $ns_ at 10.0 "$node_(0) setdest 20.0 18.0 1.0"
16 $ns_ at 100.0 "$node_(1) setdest 490.0 480.0 15.0"
```

Gambar 4.5. Contoh konfigurasi mobilitas dari mobile node

GOD (*General Operation Director*) adalah entitas untuk mengenali semua informasi environment simulasi. Pada jaringan mobile, GOD menghitung pergerakan semua node dan jarak dalam hop) antar node sebelum simulasi dijalankan. GOD adalah mandatory object pada simulasi jaringan mobile. Pada fase konfigurasi jaringan, obyek MAC akan membaca jumlah node dari obyek GOD. Jika obyek GOD ini tidak diinisialisasi, maka pesan run-time error akan muncul di layar.

4.4. Prosedur Percobaan.

4.4.1. Membangun simulasi jaringan Nirkabel dengan Lokasi Dinamis

Skenario pertama adalah membangun jaringan nirkabel pengaturan topologi random pada luasan observasi tertentu.

Sebuah jaringan wireless terdiri dari 4 node mobile(n0, n1, n2 dan n3)., dimana ke empat node tersebut diletakkan pada koordinat lokasi random pada luasan 500x500m². Node-node n0,n1 dan n2 mengirim paket FTP melalui jalur TCP ke node n3, dimulai pada t=3 dan berakhir pada t=20. Output FTP dari ketiga sumber node tersebut akan ditampilkan pada out0.tr, out1.tr dan out2.tr dan digambarkan grafiknya menggunakan xgraph. Output simulasi ditampilkan

dalam file *dina1.nam*. Node-node berpindah ke posisi tertentu, dengan speed tertentu yang telah ditetapkan pada $t=5$.

Langkah-langkah Pemrograman Tcl

1. Buka file tcl baru dengan perintah gedit. Misalkan nama file adalah *dina1.tcl*, maka
gedit dina1.tcl
2. Pada baris teratas dari file tersebut, definisikan beberapa parameter untuk komponen jaringan sebagai berikut:

```
# Define Options
#
set val (chan)      Channel/WirelessChannel      ;#Channel Type
set val(prop)      Propagation/TwoRayGround    ;#Radio propagation model
set val(netif)     Phy/WirelessPhy             ;#network interface type
set val(mac)       MAC/802_11                  ;#MAC type
set val(ifq)       Queue/DropTail/Priqueue     ;#interface queue type
set val(ll)        LL                           ;#link layer type
set val(ant)       Antenna/OmniAntenna        ;#Antenna model
set val(ifqlen)    50                           ;#max packet in ifq
set val(n)         4                             ;#number of mobile node
set val(rp)        DSDV                         ;#routing protocol
set val(x)         500
set val(y)         500
```

3. Create sebuah simulator dengan nama *ns_*

```
# Create simulator
set ns_ [new Simulator]
```

4. Buat variable obyek-obyek trace, yaitu **fd** yang akan dijadikan output file text dengan nama *dina1.tr*, dan **nam1** yang akan dijadikan output file visualisasi dengan nama *dina1.nam*. Nama-nama variable obyek dan file output terserah anda.

```
# Create trace object
set fd [open dina1.tr w]
$ns_ trace-all $fd

# Create a NAM trace file
set nam1 [open dina1.nam w]
$ns_ namtrace-all-wireless $nam1 $val(x) $val(y)
```

5. Create topologi jaringan dan implementasikan pada luasan yang didefinisikan

```
# Create object topology
set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)
```


6. Create GOD untuk menyimpan semua informasi obyek

```
# Create GOD untuk menyimpan semua konfigurasi
# topologi (jumlah node, table hop)
create-god $val(n)
```

7. Create API baru (untuk ns-2.34 ke atas). Terdiri dari:

- Create Channel (atau multiple channel)
- Spesifikasikan channel ke dalam konfigurasi node bersama parameter lainnya.
- Create node untuk simulasi.

```
# Create channel #1
set chan_1_ [new $val(chan)]
```

```
# Create node-config
#
$ns_ node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace ON \
                -movementTrace OFF \
                -channel $chan_1_
```

```
# Create node0 sampai node3
set node_(0) [$ns_ node]
set node_(1) [$ns_ node]
set node_(2) [$ns_ node]
set node_(3) [$ns_ node]

$node_(0) random-motion 0
$node_(1) random-motion 0
$node_(2) random-motion 0
$node_(3) random-motion 0
```

8. Tentukan posisi random awal dari masing-masing mobile node

```
# Lokasi random untuk node
for {set i 0} {$i < $val(n)} {incr i} {
    $node_($i) set X_ [expr rand()*$val(x)]
    $node_($i) set Y_ [expr rand()*$val(y)]
    $node_($i) set Z_ 0
}
```

9. Buat aliran traffic data antar 3 node awal terhadap node(3). Gunakan prosedur *attach-ftp-traffic*.

```
# Create an FTP traffic flow over TCP agent from n0 to n2 and n1 to n3
proc attach-ftp-traffic { nodea sink size burst idle rate } {
    #Get an instance of the simulator
    set ns_ [Simulator instance]

    #Create a TCP agent and attach it to the node
    set sources [new Agent/TCP]
    $ns_ attach-agent $nodea $sources

    #Create an FTP traffic agent and set its configuration parameters
    set traffic [new Application/TCP]
    $traffic set packetSize_ $size
    $traffic set burst_time_ $burst
    $traffic set idle_time_ $idle
    $traffic set rate_ $rate

    # Attach traffic source to the traffic generator
    $traffic attach-agent $sources
    #Connect the source and the sink
    $ns_ connect $sources $sink
    return $traffic
}
```

10. Definisikan ukuran masing-masing node pada editor nam (ukuran 30)

```
# Define node size in nam editor
for {set i 0} {$i < $val(n)} {incr i} {
    $ns_ initial_node_pos $node_($i) 30
}
```

11. Create sink node sebagai agen Loss Monitor dan tempatkan pada node(3)

```
set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]
set sink2 [new Agent/LossMonitor]
$ns_ attach-agent $node_(3) $sink0
$ns_ attach-agent $node_(3) $sink1
$ns_ attach-agent $node_(3) $sink2
```

12. Panggil prosedur *attach-ftp-traffic* untuk mengisikan trafik pada masing-masing node(0), node(1) dan node(2), dengan parameter: node asal, node tujuan, packetSize, nurst, idle,rate.

```
set source0 [attach-ftp-traffic $node_(0) $sink0 200 2s 1s 100k]
set source1 [attach-ftp-traffic $node_(1) $sink1 200 2s 1s 200k]
set source2 [attach-ftp-traffic $node_(2) $sink2 200 2s 1s 300k]
```

13. Arahkan masing-masing node ke tempat tujuan yang telah ditentukan.

```
$ns_ at 5.0 "$node_(0) setdest 400.0 100.0 15.0"  
$ns_ at 5.0 "$node_(1) setdest 40.0 200.0 15.0"  
$ns_ at 5.0 "$node_(2) setdest 200.0 375.0 25.0"  
$ns_ at 5.0 "$node_(3) setdest 250.0 350.0 25.0"
```

14. Siapkan output file untuk me-record data

```
set f0 [open out0.tr w]  
set f1 [open out1.tr w]  
set f2 [open out2.tr w]
```

15. Running simulasi transmisi paket data

```
$ns_ at 0.0 "record"  
$ns_ at 3.0 "$source0 start"  
$ns_ at 3.0 "$source1 start"  
$ns_ at 3.0 "$source2 start"  
$ns_ at 20.0 "$source0 stop"  
$ns_ at 20.0 "$source1stop"  
$ns_ at 20.0 "$source2 stop"  
$ns_ at 20.01 "finish"  
$ns_ at 20.002 "puts \"NS EXITING...\" ;$ns_ halt"
```

16. Definisikan prosedur "finish", yaitu prosedur mengakhiri simulasi. Eksekusi nam file dan xgraph untuk menampilkan grafik output simulasi

```
# Define a finish procedure  
proc finish {} {  
    global ns_ fd f0 f1 f2 nam1  
    $ns_ flush-trace  
    close $fd  
    close $f0  
    close $f1  
    close $f2  
    close $nam1  
    exec xgraph out0.tr out1.tr out2.tr -geometry 800x400 &  
    exec nam dinal.nam &  
    exit 0  
}  
puts "Starting Simulation....."  
$ns_ run
```

17. Prosedur record data

```
proc record {} {
    global sink0 sink1 sink2 f0 f1 f2
    set ns_ [Simulator instance]
    set time 0.5
    set bw0 [$sink0 set bytes_]
    set bw1 [$sink1 set bytes_]
    set bw2 [$sink2 set bytes_]
    set now [$ns_ now]
    puts $f0 "$now [expr bw0/$time*8/1000000]"
    puts $f1 "$now [expr bw0/$time*8/1000000]"
    puts $f2 "$now [expr bw0/$time*8/1000000]"
    # Reset sink bytes to 0
    $sink0 set bytes_ 0
    $sink1 set bytes_ 0
    $sink2 set bytes_ 0
    $ns_ at [expr $now+$time] "record"
}
```

18. Setelah seluruh syntax tersebut ditulis, simpan. Pada direktori dimana fileku.tcl berada, compile file tersebut dengan mengetikkan:

```
# ns dina1.tcl
```

19. Jika tidak ada kesalahan dalam penulisan, maka akan ada 2 jenis output yang bisa dilihat saat simulasi dijalankan, yaitu output out0.tr, out1.tr, out2.tr yang berisi data-data hasil simulasi dan yang ditampilkan di xgraph, dan output pada dina1.nam yang merupakan animasi simulasi.

4.4.2. Membangun simulasi jaringan Nirka bel dengan Mobilitas Dinamis

Skenario kedua adalah membuat pergerakan node yang dinamis. Pada scenario kedua ini node-node dibuat bergerak ke arah tertentu atau bergerak secara random. Kecepatan (speed) dari gerakan bisa diatur dalam satuan m/s. Dengan menggunakan topologi sebaran awal statik, akan dilakukan simulasi dengan dua jenis gerakan node: pada t=4, node bergerak ke arah koordinat yang telah ditetapkan, dan pada t=10 node bergerak ke arah random. Koordinat node adalah $n_0=(5,8,0)$, $n_1=(50,30,0)$, $n_2=(200,300,0)$, $n_3=(100,80,0)$. Luasan observasi, jumlah node dan parameter lainnya masih sama dengan scenario yang pertama.

Langkah- langkah Pemrogram Tcl

1. Buka file tcl baru dengan perintah gedit.

```
# gedit dina2.tcl
```

- Langkah percobaan kedua ini sama dengan scenario pertama, hanya langkah ke-8 diganti dengan syntax ini:

Atur posisi awal node.

```
# Node initial position topology
$node_(0) set X_ 5.0
$node_(0) set Y_ 8.0
$node_(0) set Z_ 0.0
$node_(1) set X_ 50.0
$node_(1) set Y_ 30.0
$node_(1) set Z_ 0.0
$node_(2) set X_ 200.0
$node_(2) set Y_ 300.0
$node_(2) set Z_ 0.0
$node_(3) set X_ 100.0
$node_(3) set Y_ 80.0
$node_(3) set Z_ 0.0
```

- Langkah ke-13 pada scenario pertama, diganti dengan scenario berikut: pada t=4, node-node bergerak kearah koordinat tertentu dengan kecepatan tertentu.

```
$ns_ at 4.0 "$node_(0) setdest 100.0 40.0 25.0"
$ns_ at 4.0 "$node_(1) setdest 300.0 400.0 15.0"
$ns_ at 4.0 "$node_(2) setdest 60.0 100.0 25.0"
$ns_ at 4.0 "$node_(3) setdest 10.0 300.0 40.0"
```

- Dilanjutkan dengan langkah ini: pada t=10, node-node bergerak dengan arah random

```
# Definisikan Mobilitas random dengan kecepatan 40 m/s
for {set i 0} {$i < $val(n)} {incr i} {
    set xr [expr rand()*val(x)]
    set yr [expr rand()*val(y)]
    $ns_ at 10.0 "$node_($i) setdest $xr $yr 40"
}
```

- Prosedur membuat trafik hingga mengakhiri simulasi sama dengan langkah 9 sampai 18 skenario pertama, hanya diganti nama file .tr nya dan nama file .nam nya.
- Simpan konfigurasi di atas. Jalankan simulasi dengan perintah:

```
# ns dina2.tcl
```

4.5. Tugas dan Pertanyaan

1. Konfigurasilah sebuah jaringan dengan 16 node statis yang akan disimulasikan pada NS-2 dengan skenario sebagai berikut:

Posisikan masing-masing node secara grid pada luasan area yang disediakan (berbentuk matriks 4x4 dimana setiap node berada di perpotongan matriks tersebut)

Luas area 400x400 m².

Alirkan paket FTP pada jalur TCP untuk setiap pasangan node. Gunakan prosedur untuk membangkitkan trafik TCP.

Simulasikan saat paket dikirim ke masing-masing node terima pada $t=2$ detik, dan berakhir pada $t=20$ detik.

Simulasi berakhir pada $t=25$ detik.

Dapatkan trace file outputnya.

2. Konfigurasilah sebuah jaringan dengan 4 node mobile yang akan disimulasikan pada NS-2 dengan skenario sebagai berikut:

Empat node berturut-turut di posisi awal (100,150,0), (200,100,0), (300,350,0) dan (10,20,0).

Luas area 750x750 m².

Node(1) merupakan agent UDP, dengan tujuan ke node (3), sedangkan node(0) agen UDP dan node(2) adalah sink-nya. Paket yang dikirim adalah paket CBR.

Pada $t=3$ detik, node(0) berpindah ke (300,200,0) dan node(2) ke (50,50,0) dan paket CBR mulai dikirimkan.

Pada $t=5$ detik, node(1) berpindah ke (150,100,0) dan node(3) ke (350,500,0) dan paket CBR mulai dikirimkan.

Simulasikan event tersebut. Dapatkan trace file untuk event node(1) saja. Gunakan AWK untuk parsing datanya.

Last updated: 19 December 2015 by Prima Kristalina