

## PERCOBAAN 3

### KONFIGURASI DASAR JARINGAN NIRKABEL DENGAN NS2

#### 3.1. Tujuan:

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu:

- Membuat pemrograman simulasi jaringan nirkabel dengan node statis dan mobile menggunakan script Tcl
- Menjalankan program Tcl pada NS2
- Menampilkan hasil simulasi dengan nam dan trace file

#### 3.2. Peralatan:

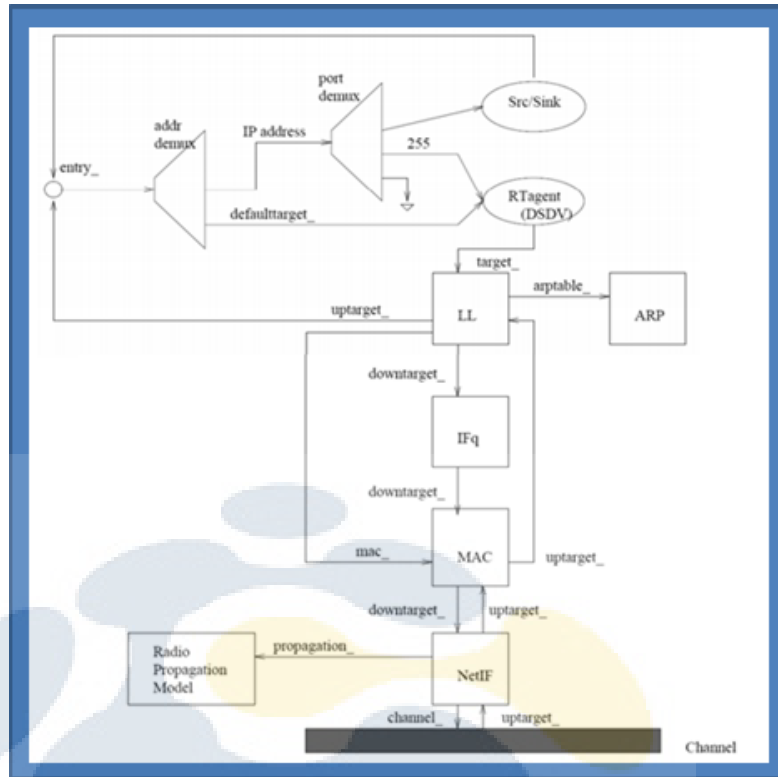
- OS Ubuntu 10.04
- Simulator NS-2.34

#### 3.3. Teori

Jaringan nirkabel terdiri dari node-node mobile, dimana node-node tersebut memiliki karakteristik sebagai berikut: adanya mekanisme pergerakan (movement), kemampuan mengirim dan menerima informasi melalui kanal secara mobile, ketergantungan kepada lingkungan nirkabelnya [Mark Greiss]. Karena itu, mengkonfigurasi jaringan nirkabel berbeda dengan mengkonfigurasi jaringan berbasis kabel. Arsitektur sebuah node mobile di NS2 merupakan kepanjangan dari arsitektur sebuah node biasa, dimana mobile node dilengkapi dengan beberapa komponen jaringan, yaitu link layer, interface queue, mac layer, network interface dengan antenna yang digunakan untuk mendefinisikan model propagasi. Meninterkoneksi komponen-komponen ini dan menghubungkannya dengan channel. Seluruh koneksi tersebut ditunjukkan pada gambar 3.1.

Secara umum komponen-komponen jaringan yang digunakan untuk mengkonfigurasi sebuah mobile node meliputi:

1. *Tipe kanal yang digunakan (chan)*. Merupakan tipe kanal untuk media pengiriman dan penerimaan paket informasi. Secara default disediakan Wireless Channel.
2. *Network Interface (Netif)*. Merupakan interface antara MAC Layer dan Physical Layer, dalam hal ini Channel. Disediakan secara default Wireless Phy.



Gambar 3.1. Arsitektur sebuah Mobile Node

3. *Model Radio Propagasi*. Tipe radio propagasi yang tersedia adalah TwoRay Ground, dimana redamannya pada jarak jauh adalah  $1/r^4$ , dengan r adalah jarak dari pemancar ke penerima.
4. *Protokol Medium Access Control (MAC)*. Ns2 menggunakan IEEE 802.11 untuk protocol MAC ini
5. *Protokol Routing adhoc*. Ada 5 routing protocol adhoc yang mensupport Ns2, yaitu DSDV (Destination Sequence Distance Vector), DSR (Dynamic Source Routing), TORA (Temporally Ordered Routing Algorithm), AODV (Adhoc On Demand Distance Vector) dan PUMA (Protocol for Unified Multicasting through Announcement).
6. *Interface Queue (ilq)*. Ns2 menyediakan PriQueue, yaitu interface yang memberikan priority untuk merutekan paket protocol, menambahkan head dari queue paket-paket tersebut. Implementasi interface queue ini ada di /ns-2/priqueue.h dan priqueue.cc.
7. *Link Layer (LL)*. Pada NS2, LL yang digunakan memiliki modul ARP (Address Resolution Protocol) yang mengkonversikan seluruh IP ke hardware Mac. Biasanya

seluruh paket outgoing dibawa ke LL oleh Routing Agent, kemudian LL membawa paket-paket tersebut ke ilq. Sedangkan pada paket incoming, mac layer membawanya ke LL dan LL melanjutkannya ke entry node. LL pada Ns-2 ada di /tcl/lan/ns-ll.tcl.

8. *Topografi Obyek (topo)*. Komponen ini mendefinisikan topologi dari jaringan nirkabel yang didisain, yaitu berbentuk topografi.
9. *Tipe Antenna (ant)*. Tipe antenna yang disediakan adalah omnidirectional antenna, yaitu antenna yang memiliki arah pancar 360o dengan gain = 1. Definisi antenna ini ada di ns2/antenna.cc dan antenna.h

Perbedaan lain dari konfigurasi jaringan nirkabel dengan jaringan berbasis kabel adalah adanya *GOD (General Operations Director)*. GOD adalah obyek yang digunakan untuk menyimpan informasi global tentang kondisi lingkungan, jaringan atau node. Biasanya obyek GOD menyimpan jumlah total mobile node dan table dari jumlah hop terpendek yang diperlukan dari node asal ke tujuan.

### **3.4. Prosedur Percobaan.**

#### **3.4.1. Membangun simulasi jaringan Nirkabel Statik**

Skenario pertama adalah membangun jaringan nirkabel dengan node-node statis, dengan aturan sebagai berikut:

Sebuah jaringan wireless terdiri dari 4 node statis (n0, n1, n2 dan n3). Ke empat node tersebut diletakkan pada koordinat sbb: n0 =(10,10,0), n1=(10,400,0), n2=(400,10,0) dan n3=(400,400,0). Node-node tersebut disebar pada luasan 500x500 m<sup>2</sup>. Node n0 mengirim paket FTP melalui jalur TCP ke node n2, dan node n1 juga mengirim paket FTP ke node n3. Jaringan akan disimulasikan sbb:

Pengiriman paket FTP n0 – n2 mulai pada detik ke 1 dan paket FTP n1 – n3 pada detik ke 2.

Pengiriman paket selesai pada detik ke 15.00

Simulasi selesai pada detik ke 15.001.

#### **Langkah-langkah Pemrogram Tcl**

1. Buka file tcl baru dengan perintah gedit. Misalkan nama file adalah fileku.tcl, maka **# gedit statik.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 **tracefd** yang akan dijadikan output file text dengan nama *statik.tr*, dan **nam1** yang akan dijadikan output file visualisasi dengan nama *statik.nam*. Nama-nama variable obyek dan file output terserah anda.

```

# Create trace object
set tracefd [open statik.tr w]
$ns_ trace-all $tracefd

# Create a NAM trace file
set nam1 [open statik.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
for {set i 0} {$i < $val(n)} {incr i} {
    set node_($i) [$ns_ node]
}

$node_(0) color blue
$node_(1) color blue
$node_(2) color blue
$node_(3) color blue
```

8. Tentukan posisi dari masing-masing mobile node

```
# Node position static (fixed) topology
$node_(0) set X_ 10.0
$node_(0) set Y_ 10.0
$node_(0) set Z_ 0.0
$node_(1) set X_ 10.0
$node_(1) set Y_ 400.0
$node_(1) set Z_ 0.0
$node_(2) set X_ 400.0
$node_(2) set Y_ 10.0
$node_(2) set Z_ 0.0
$node_(3) set X_ 400.0
$node_(3) set Y_ 400.0
$node_(3) set Z_ 0.0
```

9. Buat aliran traffic data antar 2 node. Set koneksi TCP, alirkan trafik FTP di atasnya.

```
# Create a TCP flow from n0 to n2 and n1 to n3
set tcp1 [new Agent/TCP]
$tcp1 set class_ 2
set sink1 [new Agent/TCPSink]
set tcp2 [new Agent/TCP]
$ns_ attach-agent $node_(0) $tcp1
$ns_ attach-agent $node_(2) $sink1
$tcp2 set class_ 2
set sink2 [new Agent/TCPSink]
$ns_ attach-agent $node_(1) $tcp2
$ns_ attach-agent $node_(3) $sink2
$ns_ connect $tcp1 $sink1
$ns_ connect $tcp2 $sink2

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ns_ at 1.0 "$ftp1 start"
$ns_ at 2.0 "$ftp2 start"
$ns_ at 14.0 "$ftp1 stop"
$ns_ at 14.0 "$ftp2 stop"
```

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. Atur akhir simulasi

```
for {set i 0} {$i < $val(n)} {incr i} {
    $ns_ at 15.0 "$node_($i) reset"
}
$ns_ at 15.001 "stop"
$ns_ at 15.002 "puts \"NS EXITING..\" ;$ns_ halt"
```

12. Definisikan prosedur "stop", yaitu prosedur mengakhiri simulasi

```
# Define a finish procedure
proc stop {} {
    global ns_ tracefd nam1
    $ns_ flush-trace
    close $tracefd
    close $nam1
    exec nam statik.nam &
    exit 0
}
puts "Starting Simulation....."
$ns_ run
```

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

```
# ns statik.tcl
```

14. Jika tidak ada kesalahan dalam penulisan, maka akan ada 3 jenis output yang bisa dilihat saat simulasi dijalankan, yaitu output pada editor CLI yang dikeluarkan dengan perintah “puts”, output pada statik.tr yang berisi data-data hasil simulasi dan output pada statik.nam yang merupakan animasi simulasi.

### **3.4.2. Membangun simulasi jaringan Nirkabel Mobile**

Skenario kedua adalah membangun jaringan nirkabel dengan node-node mobile, dengan aturan sebagai berikut:

Sebuah jaringan wireless terdiri dari 2 node mobile (n0, n1). Ke dua node tersebut diletakkan pada koordinat sbb: n0 =(5,2,0) dan n1=(390,385,0). Node-node tersebut disebar pada luasan 500x500 m<sup>2</sup>. Node n0 mengirim paket FTP melalui jalur TCP ke node n1. Jaringan akan disimulasikan sbb:

Pada detik ke 2 node (0) akan menuju koordinat (100,40,0) dengan kecepatan 25 m/s.

Pada detik ke 4 node (1) akan menuju koordinat (300,18,0) dengan kecepatan 40 m/s.

Pada detik ke 10 node(1) menuju koordinat (490,480,0) dengan kecepatan 20 m/s.

Pengiriman paket FTP n0 – n2 mulai pada detik ke 4 dan selesai pada detik ke 10.

Simulasi berakhir pada detik ke 20.0001.

#### Langkah-langkah Pemrograman Tcl

1. Buka file tcl baru dengan perintah gedit. Misalkan nama file adalah fileku.tcl, maka **# gedit nirkabel.tcl**
2. Ulangi langkah 2 sampai 7 pada percobaan sebelumnya. Berikan output file nir1.tr untuk output text dan nir1.nam untuk output visualisasi gambar.
3. Create node baru, disable-kan random motion.

```
# Create node0 dan node1
for {set i 0} {$i < $val(n)} {incr i} {
    set node_($i) [$ns_ node]
}
$node_(0) random-motion 0
$node_(1) random-motion 0
```

4. Atur posisi awal node.

```
# Node initial position topology
$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0
$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
```

5. Atur ukuran node untuk display nam.

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

6. Pada detik tertentu, gerakkan kedua node tersebut kearah koordinat tertentu dan kecepatan pergerakan tertentu.

```
$ns_ at 2.0 "$node_(1) setdest 100.0 40.0 25.0"
$ns_ at 3.0 "$node_(0) setdest 300.0 400.0 40.0"

$ns_ at 10.0 "$node_(1) setdest 490.0 480.0 40.0"
```

7. Buat aliran traffic data antar 2 node. Set koneksi TCP, alirkan trafik FTP di atasnya.

```
# Create a TCP flow from n0 to n1
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(0) $tcp
$ns_ attach-agent $node_(1) $sink
$ns_ connect $tcp $sink

set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 4.0 "$ftp start"
```

8. Atur akhir simulasi

```
for {set i 0} {$i < $val(n)} {incr i} {
    $ns_ at 20.0 "$node_(i) reset"
}
$ns_ at 20.001 "stop"
$ns_ at 20.002 "puts \"NS EXITING...\" ;$ns_ halt"
```



9. Definisikan prosedur “stop”, yaitu prosedur mengakhiri simulasi

```
# Define a finish procedure
proc stop {} {
    global ns_ tracefd nam1
    $ns_ flush-trace
    close $tracefd
    close $nam1
    exec nam nirl1.nam &
    exit 0
}
puts "Starting Simulation....."
$ns_ run
```

10. Simpan konfigurasi di atas. Jalankan simulasi dengan perintah:

```
# ns nirkabel.tcl
```

### 3.4.3. Menganalisa Hasil Trace file pada simulasi Nirkabel dengan NS-2

Trace file adalah output text dari simulasi NS-2. File ini berisi beberapa parameter yang terukur saat simulasi dijalankan. Untuk Jaringan nirkabel, ada perbedaan dalam pembacaan nilai parameternya dibandingkan jaringan berbasis kabel. Format dari trace file jaringan nirkabel seperti pada gambar 3.2.

```
ACTION: [s|r|D|f]: s -- sent, r -- received, D -- dropped, f -- forwarded
WHEN: the time when the action happened
WHERE: the node where the action happened
LAYER: AGT -- application,
        RTR -- routing,
        LL -- link layer (ARP is done here)
        IFQ -- outgoing packet queue (between link and mac layer)
        MAC -- mac,
        PHY -- physical
flags:
SEQNO: the sequence number of the packet
TYPE: the packet type
      cbr -- CBR data stream packet
      DSR -- DSR routing packet (control packet generated by routing)berikut:

      RTS -- RTS packet generated by MAC 802.11
      ARP -- link layer ARP packet
SIZE: the size of packet at current layer, when packet goes down,
      size increases, goes up size decreases
[a b c d]: a -- the packet duration in mac layer header
           b -- the mac address of destination
           c -- the mac address of source
           d -- the mac type of the packet body
flags:
[.....]: [
          source node ip : port_number
          destination node ip (-1 means broadcast) : port_number
```

Gambar 3.2. Format parameter yang ter-record pada trace file

Contoh 1:

Jika didapatkan sebuah urutan data sebagai berikut:

**r 0.010176954 \_0\_ RTR - - - - 1 gpsr 29 [0 ffffffff 8 800] - - - - [8:255 -1:255 32 0]**

Artinya:

Routing Agent pada node 0, saat  $t=0,010176954$  menerima broadcast GPSR MAC address 0xff, ip address -1 dan tipe MAC ETHERTYPE\_IP 800), merutekan paket yang ID nya 1 dengan ukuran 29 byte, dari node 8.

Contoh 2:

Jika didapatkan sebuah urutan data sebagai berikut:

**r 40.639943289 \_1\_ AGT - - - - 1569 tcp 1032 [a2 1 2 800] - - - - [0:0 1:0 32 1] [35 0] 2 0**

Artinya:

Saat  $t=40,639943289$  node 1 menerima paket transport (AGT) berupa TCP pada sequence ke 1569 dengan ukuran paket 1032 byte. broadcast GPSR MAC address 0xff, ip address -1 dan tipe MAC ETHERTYPE\_IP 800), merutekan paket yang ID nya 1 dengan ukuran 29 byte, dari node 8.

### 3.5. Tugas dan Pertanyaan

1. Konfigurasilah sebuah jaringan dengan 3 node statis yang akan disimulasikan pada NS-2 dengan skenario sebagai berikut:  
Tiga node berturut-turut di posisi (25,50,0), (100,100,0) dan (300,350,0).  
Luas area 400x400 m<sup>2</sup>.  
Node(1) merupakan agent TCP, sedangkan node(0) dan node(2) adalah sink-nya. Paket yang dikirim adalah paket FTP.  
Simulasikan saat paket dikirim ke node(0) pada  $t=2$  detik, dan ke node(2) pada  $t=6$  detik.  
Simulasi berakhir pada  $t=15$  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 m<sup>2</sup>.

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() 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: 7 December 2015 by Prima Kristalina*

