

ANALISIS JARINGAN DENGAN PARAMETER *RANDOMIZE FILE SIZE* MENGGUNAKAN *NETWORK SIMULATOR (NS)*

Muhamad Azhari

Program Studi Ilmu Komputer FMIPA Universitas Mulawarman
Email: ktob34@yahoo.com

ABSTRAK

NS merupakan salah satu *tools* yang digunakan untuk membuat simulasi jaringan komputer. Simulasi jaringan berbasis TCP/IP dengan berbagai macam media dapat diimplementasikan dengan NS. Manajemen antrian yang dapat diterapkan dalam NS yaitu *Random Early Detection (RED)*, *Drop Tail (FIFO)*, dan *Class Based Queueing (CBQ)* dengan algoritma *routing unicast* dan *multicast*. Skenario analisis besaran paket yang digunakan dalam simulasi ini berparameter *randomize file size*. Sebanyak 200 paket per detik dikirim dalam simulasi. Ukuran paket yang diujicobakan sebesar 500 byte, 1000 byte, 1500 byte, dan 2000 byte dengan jenis antrian *Drop Tail (FIFO)*. Data keluaran NS berupa *file trace (*.tr)* dianalisa melalui proses *parsing* untuk mengetahui jumlah paket *loss*, *throughput* dan *delay time* dari simulasi.

Kata Kunci : NS, simulasi, *trace*, *parsing*

PENDAHULUAN

NS merupakan salah satu *tools* yang digunakan untuk membuat simulasi jaringan komputer. Umumnya NS bermanfaat untuk kepentingan riset. Interaksi antar protokol dalam konteks pengembangan protokol internet yang ada pada saat ini dan masa yang akan datang mampu disimulasikan dengan NS. Simulasi jaringan berbasis TCP/IP dengan berbagai macam media dapat diimplementasikan dengan NS. Beberapa protokol lapisan transport jaringan yang dapat digunakan dalam NS yaitu *Transport Control Protocol (TCP)*, *Universal Datagram Protocol (UDP)*, dan *Real Time Transport Protocol (RTP)*. Ada dua tipe dasar aplikasi yang disimulasikan pada NS, yaitu *Simulated Application* dan *Generator Traffic*. Pada saat ini baru terdapat dua jenis aplikasi yang dapat disimulasikan yaitu *File Transfer Protocol (FTP)* dan *Telnet* yang masing-masing berjalan di atas *TCP transport agent*. *Generator Traffic* digunakan untuk membangkitkan *traffic* dengan metode *on/off* yang kecepatan pembangkitan datanya dilakukan secara tetap. Terdapat empat tipe pembangkitan *traffic*, yaitu Eksponensial, Pareto, *Constant Bit Rate (CBR)*, dan *Traffic Trace*.

Pada Manajemen antrian dapat diterapkan *Random Early Detection (RED)*, *Drop Tail (FIFO)*, dan *Class Based Queueing (CBQ)*. Sedangkan algoritma *routing* yang dapat digunakan adalah *unicast* dan *multicast*. Aplikasi multimedia berupa *layered video*, *quality of service audio-video*, dan *transcoding* dapat disimulasikan dalam NS. Selain itu, dapat pula diimplementasikan beberapa MAC

(IEEE, 802.3, 802.11) di jaringan *wired* maupun *wireless* dan media satelit.

Output yang dihasilkan dari simulasi menggunakan NS berupa satu atau lebih file *text-base* yang berisi detail simulasi yang dideklarasikan pada saat membangun simulasi. File *trace* yang dihasilkan digunakan untuk menganalisa nilai numerik. Sedangkan file *namtrace* digunakan sebagai input untuk tampilan grafis simulasi (*network animator*).

Parameter unjuk kerja yang dapat dihasilkan melalui simulasi pada NS adalah *throughput*, yaitu laju rata-rata dari paket data yang berhasil dikirim melalui kanal komunikasi atau dengan kata lain *throughput* merupakan jumlah paket data yang diterima setiap detik. Rumus matematika yang digunakan untuk mencari nilai *throughput* adalah.

$$\text{Throughput } t = Pr/T \text{ detik paket/detik}; 0 \leq t \leq T$$

Pr adalah paket yang diterima (paket), T adalah waktu simulasi (detik), dan t adalah waktu saat pengambilan sampel (detik).

Parameter lain yang dapat dihasilkan adalah *Packet Loss* (paket yang hilang) menunjukkan banyak jumlah paket yang hilang. Paket hilang terjadi ketika satu atau lebih paket data yang melewati suatu jaringan gagal mencapai tujuannya. Secara matematis *paket loss* dapat dicari dengan menggunakan rumus sebagai berikut.

$$\text{Packet loss rasio } t = Ps/(Pd+Ps) \times 100\%; 0 \leq t \leq T$$

Pd adalah paket yang mengalami *drop* (paket), Ps adalah paket yang dikirim (paket), T adalah Waktu simulasi (detik), dan t adalah waktu saat pengambilan sampel (detik).

Selain itu, dihasilkan pula waktu tunda (*delay time*) yaitu interval waktu yang dibutuhkan

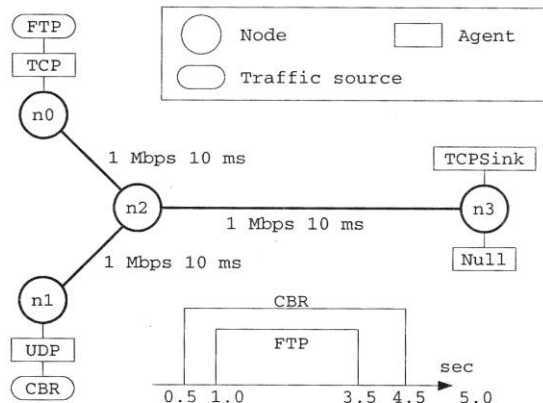
oleh suatu paket data saat data mulai dikirim dan keluar dari proses antrian dari titik sumber awal hingga mencapai titik tujuan. Secara matematis waktu tunda dapat dihasilkan melalui rumus perhitungan sebagai berikut.

$$\text{Waktu tunda } t = T_r - T_s / P_r \text{ detik; } 0 \leq t \leq T$$

T_r adalah waktu penerimaan paket (detik), T_s adalah waktu pengiriman paket (detik), P_r adalah jumlah paket yang diterima (paket), T adalah waktu simulasi (detik), dan t adalah waktu pengambilan sampel (detik).

METODOLOGI

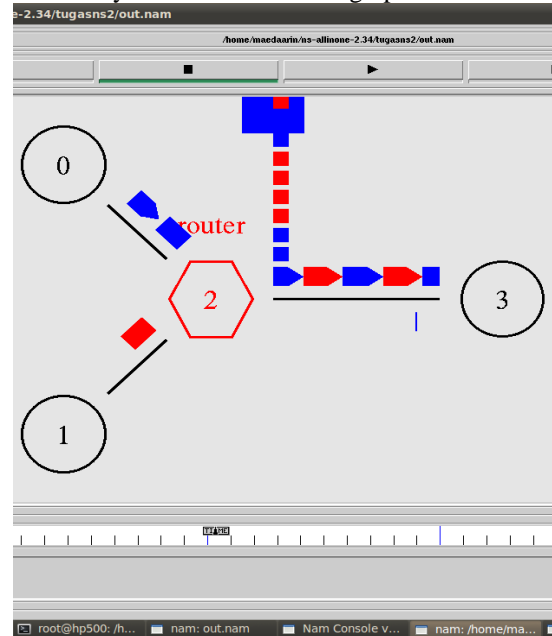
Sebuah jaringan dengan topologi *dump bell* yang terdiri dari 4 *node* akan dibuat simulasinya. Masing-masing *node* tersebut yaitu *node 0*, *node 1*, *node 2*, dan *node 3*. Pada *node 0* menggunakan aplikasi FTP yang dijalankan di atas TCP sebagai transport *agent*. *Node 1* menggunakan aplikasi CBR yang dijalankan di atas UDP sebagai transport *agent*. *Node 2* difungsikan sebagai *forward router* yang mengirimkan data dari *node 0* dan *node 1* ke *node 3*. Hubungan yang terjadi antar *node* tersebut dibuat *duplex* dimana antara *node 0* ke *node 2* dan *node 1* ke *node 2* diberikan *bandwidth* sebesar 2 Mbps dengan *delay* 10ms, sedangkan *bandwidth* yang diberikan antara *node 2* ke *node 3* adalah sebesar 1,7 Mbps dengan *delay* 20ms. *Node 0* mengirimkan *traffic* FTP dengan tujuan *node 3* mulai dari 0,5s simulasi dan diakhiri pada 4,0s simulasi. Sedangkan *node 1* mengirimkan *traffic* CBR dengan tujuan *node 3* mulai dari 1,0s simulasi dan diakhiri pada 4,5s simulasi. Skenario analisis besaran paket yang digunakan adalah *randomize file size* dengan nilai parameter yang digunakan dalam simulasi ini adalah mengirimkan paket sebanyak 200 paket per detik. Dengan ukuran paket yang diujicobakan adalah sebesar 500 byte, 1000 byte, 1500 byte, dan 2000 byte. Jenis antrian yang digunakan adalah *Drop Tail* (FIFO).



Gambar 1. Topologi yang digunakan untuk simulasi.

HASIL SIMULASI

Untuk menganalisa hasil yang didapat melalui simulasi dipergunakan *software* pendukung yang telah disertakan dalam paket instalasi NS, diantaranya adalah AWK dan *xgraph*.



Gambar 2. Capture animasi file simulasi pada NS

Data keluaran NS berupa *file trace* (*.tr) dengan format data numerik dapat dianalisa secara matematis maupun ditampilkan dalam bentuk grafis. *File trace* ini merupakan pencatatan seluruh *event* yang dialami oleh suatu simulasi paket pada simulasi yang dibangun.

```
File: /media/DATA/kuliah_Q/tugas2/tugasns2/outnew1.txt
+ 0.1 1 2 cbr 1000 ----- 2 1.0 3.1 0 0
- 0.1 1 2 cbr 1000 ----- 2 1.0 3.1 0 0
+ 0.1 1 2 cbr 1000 ----- 2 1.0 3.1 1 1
- 0.104 1 2 cbr 1000 ----- 2 1.0 3.1 1 1
r 0.114 1 2 cbr 1000 ----- 2 1.0 3.1 0 0
+ 0.114 2 3 cbr 1000 ----- 2 1.0 3.1 0 0
- 0.114 2 3 cbr 1000 ----- 2 1.0 3.1 0 0
+ 0.116 1 2 cbr 1000 ----- 2 1.0 3.1 2 2
- 0.116 1 2 cbr 1000 ----- 2 1.0 3.1 2 2
+ 0.116 1 2 cbr 1000 ----- 2 1.0 3.1 3 3
r 0.118 1 2 cbr 1000 ----- 2 1.0 3.1 1 1
+ 0.118 2 3 cbr 1000 ----- 2 1.0 3.1 1 1
- 0.118706 2 3 cbr 1000 ----- 2 1.0 3.1 1 1
- 0.12 1 2 cbr 1000 ----- 2 1.0 3.1 3 3
r 0.13 1 2 cbr 1000 ----- 2 1.0 3.1 2 2
+ 0.13 2 3 cbr 1000 ----- 2 1.0 3.1 2 2
- 0.13 2 3 cbr 1000 ----- 2 1.0 3.1 2 2
+ 0.132 1 2 cbr 1000 ----- 2 1.0 3.1 4 4
- 0.132 1 2 cbr 1000 ----- 2 1.0 3.1 4 4
+ 0.132 1 2 cbr 1000 ----- 2 1.0 3.1 5 5
r 0.134 1 2 cbr 1000 ----- 2 1.0 3.1 3 3
+ 0.134 2 3 cbr 1000 ----- 2 1.0 3.1 3 3
```

Gambar 3. Contoh *file trace* yang dihasilkan dari eksekusi *script tcl*

Hasil yang didapat melalui pengambilan data *trace* simulasi kemudian *diparsing*. *Parsing* adalah suatu teknik untuk mendapatkan informasi yang dibutuhkan. Untuk itu perlu dibuat *file awk* yang berfungsi untuk memfilter *trace file* yang dibutuhkan. *Parsing* data yang pertama difokuskan

terhadap paket *loss* yang diperoleh dengan perlakuan menggunakan *packet randomize* dengan nilai 500 byte, 1000 byte, 1500 byte, dan 2000 byte. Hasil yang diperoleh dari data *parsing* tersebut dapat ditunjukkan dalam tabel 1 berikut.

Tabel 1. Nilai paket *loss* yang diperoleh dengan beberapa perlakuan jumlah paket.

<i>Packet Randomize</i> (byte)	<i>Number of</i> <i>Packet Sent</i>	<i>Lost</i>
500	876	21
1000	438	20
1500	584	18
2000	438	24

Pada saat paket yang diberikan terhadap jaringan kecil maka jumlah paket yang dikirim persatuan waktu menjadi sangat banyak sehingga menyebabkan paket *loss* nya lebih besar. Apabila paket diperbesar sampai dengan batas atas *bandwidth* yang dimiliki maka paket *loss* yang diakibatkannya semakin sedikit. Sedangkan apabila paket yang dikirimkan sudah terlalu besar dibandingkan *bandwidth* yang dimiliki maka akan terjadi *bottleneck* yang menyebabkan paket *loss* akan kembali besar tetapi dengan *number of packet* yang tetap kecil.

Adapun *throughput* yang diperoleh dari simulasi dengan *packet randomize* dapat dilihat pada tabel 2 berikut.

Tabel 2. *Throughput* rata-rata yang diperoleh.

<i>Packet Randomize</i> (byte)	<i>Throughput</i> rata-rata
500	260,4680093
1000	365,6373796
1500	327,3043068
2000	366,2727073

Rata-rata *delay time* yang diperoleh dari simulasi dengan perlakuan *packet randomize* dapat dilihat pada tabel 3 berikut.

Tabel 3. *Delay time* rata-rata yang diperoleh.

<i>Packet</i> <i>Randomize</i> (byte)	<i>Delay time</i> rata-rata
500	0,043900961
1000	0,05867617
1500	0,051184219
2000	0,058844447

KESIMPULAN

Besarnya nilai paket *loss* pada data yang diamati disebabkan karena banyaknya pengiriman paket bernilai kecil. Sehingga semakin banyak paket yang dikirim maka akan semakin besar

terjadinya paket *loss*. Pengiriman data paket yang cukup besar juga menyebabkan terjadinya paket *loss* akibat adanya *bottleneck*.

Peningkatan nilai *throughput* dicapai seiring penambahan jumlah paket yang dikirim namun penambahan paket menyebabkan semakin besarnya *delay time* pada pengiriman paket tersebut.

DAFTAR PUSTAKA

[1]Altman & Jimenez. *NS for Beginners*. <http://www.sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf>

[2]Anonymous.2004. *NS-2 Tutorial (1)*. Multimedia Networking Group, The Department of Computer Science, UVA. Jianping Wang

[3]Anonymous. 2007. *Tutorial NS2 Simulasi Jaringan Nirkabel Pada NS2*. www.infolinux.web.id, INFOLINUX edisi 06/2007

[4]Dewo ES. 2003. *Bandwidth dan Throughput*. Artikel Populer IlmuKomputer.Com

[5]Fall K & Varadhan K. 2008. *The ns Manual*. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC

[6]Greis M. *Tutorial for The Network Simulator*, <http://www.isi.edu/nsnam/ns/tutorial/>

[7]Nofianti D & Rochim AF. *Simulasi Kinerja WPAN 802.15.4 (Zigbee) dengan Algoritma Routing AODV dan DSR*. http://eprints.undip.ac.id/32020/1/Dwi_Nofianti.pdf

[8]Susanto B. *Modul Jaringan Komputer*. robby.c.staff.gunadarma.ac.id/Downloads/files/5659/tcpudp.pdf