

PERANCANGAN DAN EMULASI PROTOKOL ROUTING EPIDEMIC DAN STATIC PADA JARINGAN DTN

PROTOTYPING AND ANALYSIS OF CONGESTION CONTROL ON EPIDEMIC ROUTING PROTOCOL OVER DTN NETWORK

Galih Putro Dwi Setyo^[1], Leanna Vidya Y, ST, MT.^[2], Tody Ariefianto Wibowo, ST, MT.^[3]

Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

^[1]galihputrods@gmail.com, ^[2]lvy@telkomuniversity.ac.id, ^[3]taw@telkomuniversity.ac.id

ABSTRAK

Delay Tolerant Network (DTN) merupakan konsep yang digunakan untuk komunikasi antarplanet, yang disebut juga sebagai deep space network atau interplanetary internet. Karena peruntukannya untuk komunikasi dengan jarak yang sangat jauh, DTN mampu menghadapi hambatan yang dialami komunikasi antarplanet yang tidak terbatas hanya pada delay. Oleh karena itu istilah *Disruption Tolerant Network* (dengan singkatan yang sama, DTN) dan *Challenged Network* juga dapat digunakan. Dilihat dari kemampuannya, aplikasi DTN pada masa depan akan semakin luas, sehingga penelitian di bidang DTN, baik aplikasi maupun pemodelan jaringan akan memberikan kontribusi yang cukup berarti pada pengembangan sistem DTN.

Tujuan tugas akhir ini adalah menganalisis *routing protocol epidemic* dan *static* menggunakan IBR-DTN. Pada tugas akhir ini dilakukan emulasi dimana *node-node* DTN pada NS3 akan dihubungkan dengan *Virtual Machine* sehingga setiap node berfungsi mendekati *node real* DTN. Alasan pemilihan NS3 dibandingkan dengan network simulator lain didasari pada faktor model kanal yang realistis, update teratur, dan penggunaan network stack. Tujuan utama proyek akhir ini adalah melakukan pengujian kemampuan NS3 sebagai simulator DTN dan menghubungkan lingkungan NS3 dengan jaringan eksternal. Penelitian ini menggunakan IBR-DTN sebagai perangkat lunak serta memiliki tujuan kedua membandingkan kemampuan antara *epidemic routing protocol* dan *static routing protocol* dengan membandingkan waktu pengiriman menggunakan TTL, ukuran *file*, dan ukuran *bundle* yang berbeda.

Kesimpulan yang dapat ditarik dari penelitian ini adalah ukuran *file* dan ukuran *bundle* yang berbeda mampu memberikan perbedaan hasil yang sangat signifikan dalam pengujian yang dilakukan dengan menggunakan 2 *routing protocol* yang berbeda. *Epidemic routing protocol* dan *static routing protocol* juga memiliki perbedaan yang signifikan jika dibandingkan satu sama lain, baik dari segi waktu pengiriman data serta kemampuan mengirim ukuran *file* yang berbeda

Kata kunci: DTN, NS3, Network Simulator, Epidemic, Static

ABSTRACT

Delay Tolerant Network (DTN) initially projected as a concept used for interplanetary communication, also known as deep space network or interplanetary internet. Because it's initially projected for extremely long range communication, DTN is able to overcome obstacle in interplanetary communication which are not limited to delay. Because of that, several terms such as disruption tolerant network (with same acronym, DTN) and challenged network could also be used. Based on its capability, the future holds great deals of application and development for DTN. So that research on DTN, either application or network modelling would contribute substantially on future development of DTN.

This research mainly discuss about usability of NS3 to analyze epidemic and static routing protocol with IBR-DTN. In this research, every DTN nodes in NS3 will be connected with Virtual Machine so that every nodes work in accordance to real-life DTN nodes. As for why NS3 was selected as network simulator for DTN network was based on realistic channel model, scheduled update, and usage of network stack. The goal for this project was testing the capability of NS3 as DTN simulator and connecting NS3 environment to external network node. This research utilize IBR-DTN as DTN software and as a second objective, tries to compare the capability of epidemic routing protocol and static routing protocol by comparing the performance of each routing protocol by measuring delivery time using varying TTL, bundle, and file size.

This research conclude that different file size and different bundle size gives significantly different result when applied to 2 different routing protocol. There are significant difference between epidemic and static routing protocol if we compare the delivery time and maximum file size they can send

Keyword: DTN, NS3, Network Simulator, Epidemic, Static

1. Pendahuluan

Delay Tolerant Network (DTN) pada awalnya merupakan konsep yang digunakan untuk komunikasi antarplanet, yang disebut juga sebagai deep space network atau interplanetary internet. Karena peruntukannya untuk komunikasi dengan jarak yang sangat jauh, DTN dirancang untuk mampu menghadapi hambatan yang mungkin dialami dalam komunikasi antarplanet yang tidak terbatas hanya pada delay. Oleh karena itu istilah Disruption Tolerant Network (dengan singkatan yang sama, DTN) dan Challenged Network juga dapat digunakan. Dilihat dari kemampuannya, aplikasi DTN pada masa depan akan semakin luas, sehingga penelitian di bidang DTN, baik aplikasi maupun pemodelan jaringan akan memberikan kontribusi yang cukup berarti pada pengembangan sistem DTN.

Terdapat sejumlah software yang dapat digunakan untuk melakukan pengujian dan simulasi terhadap jaringan DTN, tetapi yang umum digunakan adalah The One Network Simulator atau lebih dikenal sebagai OneSim. OneSim adalah sebuah perangkat lunak berbasis Java yang dikhususkan untuk melakukan simulasi jaringan DTN. Untuk saat ini, OneSim masih belum dapat dihubungkan dengan node dari jaringan eksternal, sehingga pengujian dan simulasi terhadap aplikasi/program yang memanfaatkan DTN masih belum dapat dilakukan dengan lebih mendalam.

Dikarenakan keterbatasan yang dimiliki OneSim tersebut, tugas akhir ini bertujuan untuk membuat sebuah prototype jaringan DTN berbasis NS3 untuk simulasi jaringan fisik dan IP yang dapat dihubungkan dengan node eksternal dan melakukan pengujian terhadap Software simulasi DTN berdasarkan network simulator NS3 dengan kemampuan koneksi ke node eksternal.

2. Dasar Teori

2.1 Delay Tolerant Network(DTN)

Delay Tolerant (DTN) Network merupakan sebutan untuk jaringan dengan toleransi tinggi terhadap hambatan dan delay yang mungkin terjadi selama pengiriman data antar node. Pada mulanya, DTN diciptakan untuk Deep Space Communication antar satelit yang berada di luar angkasa dan penerima di bumi, dimana jarak yang sangat jauh serta faktor luar angkasa mengakibatkan delay data dalam waktu yang tidak bisa ditentukan, dapat dikatakan hubungan antar node dalam DTN tidak selalu tersedia, sangat berbeda dengan koneksi antar node yang selalu tersedia dalam jaringan biasa[1].

2.2 NS3(Network Simulator 3)[5]

Network Simulator 3(NS3) adalah network simulator diskret yang ditujukan untuk penggunaan riset dan edukasi. Beberapa poin yang patut diingat mengenai NS3 adalah:

1. NS3 bersifat open-source dengan lingkungan pengembangan bersifat terbuka, setiap user dapat berkontribusi untuk mengembangkan fungsi dari NS3.
 2. Ekstensi yang dibuat untuk NS2 tidak dapat digunakan pada NS3. NS3 tidak mendukung API NS2
- Awalnya NS3 ditujukan sebagai pengganti NS2, dari sisi pengembangan, NS3 dibuat dari nol dan tidak menggunakan basis pemrograman NS2.

2.3 IBR-DTN

IBR-DTN adalah implementasi *bundle protocol* dan *daemon DTN*. IBR-DTN memiliki tujuan pengembangan sebagai *platform DTN* ringan yang dapat berfungsi di berbagai *platform*. IBR-DTN dikembangkan oleh *Institute of Operating Systems and Computer Networks Technische Universität Braunschweig* yang berlokasi di Jerman. Selain bersifat *multiplatform* IBR-DTN juga memiliki kelebihan lain berupa implementasi ringan yang disebabkan minimnya ketergantungan IBR-DTN akan *library* diluar paket instalasinya. Untuk saat ini IBR-DTN mendukung sejumlah sistem operasi, yaitu *Android*, *OpenWRT*, *Gentoo Linux*, *Debian/Ubuntu*, *Windows*, *MAC OSX*, dan *Debian ARM*[9].

2.4 Virtual Machine[18][19]

Virtual Machine merupakan bentuk virtualisasi dari komputer dalam bentuk fisik yang beroperasi sebagai *software/perangkat lunak*. Ada dua komponen utama dalam *virtual machine*, yaitu *host operating system* dan *guest operating system*.

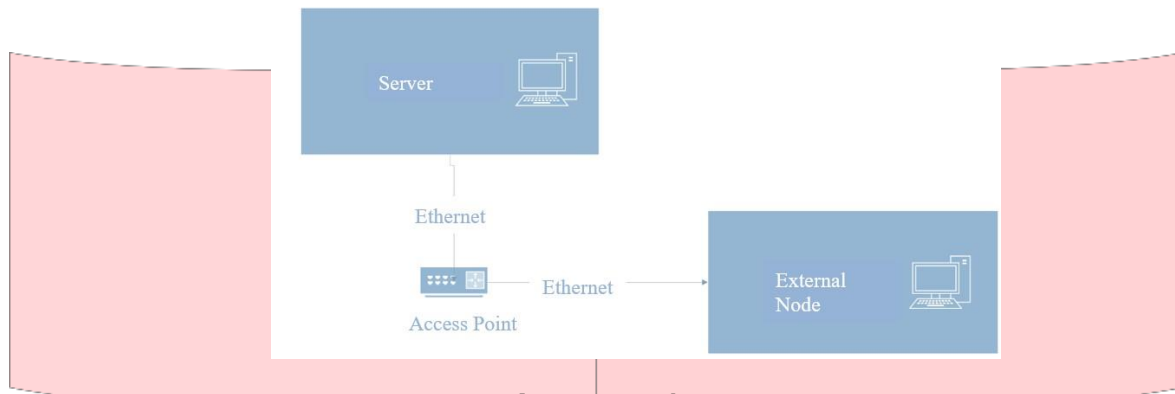
Host operating system merupakan sistem operasi induk, dimana proses virtualisasi berlangsung dan berjalan diatasnya. *Host* memiliki pengaruh yang sangat besar terhadap proses virtualisasi dan berjalannya program.

Guest operating system merupakan sistem operasi yang berjalan diatas program virtualisasi. Sistem operasi yang digunakan jumlahnya beragam, selama perangkat lunak yang digunakan untuk proses virtualisasi mendukung.

3. Perancangan dan Implementasi Sistem

3.1 Arsitektur Jaringan

Secara garis besar, simulasi yang akan dilaksanakan memerlukan rancangan topologi jaringan yang akan digunakan dalam tugas akhir ini tampak seperti pada gambar 3.1.



Gambar 3.1 Arsitektur Jaringan

NS3 berjalan sebagai network simulator yang berjalan diatas Host OS. NS3 memungkinkan adanya node eksternal yang mampu berjalan dalam jaringan yang sudah disimulasikan oleh NS3. NS3 dalam sistem ini berfungsi untuk melakukan manajemen dan simulasi jaringan. Penggunaan NS3 bertujuan untuk mengemulasikan layer 3 jaringan dan mensimulasikan pergerakan node dalam jaringan yang akan divisualisasikan menggunakan PyViz.

Dalam diagram tersebut, terdapat tiga perangkat keras yang berperan dalam pengujian yang terdiri dari:

a. Server

Dalam pengujian ini, server berperan sebagai *Host OS* dan lingkungan operasi NS3. dalam scenario yang akan diuji, Server bertugas menyediakan sejumlah *Virtual Machine* yang akan difungsikan sebagai *DTN node* dalam NS3.

b. Access Point

Access Point bertugas menghubungkan Server dengan External Node melalui antarmuka Ethernet.

c. External Node

External node berfungsi sebagai *DTN Node* dalam NS3 yang dikendalikan oleh sistem di luar *Virtual Machine* yang berjalan dalam server.

3.2 Implementasi Jaringan

Instalasi NS3 dilakukan dengan cara men-download source code NS3 dari NSNAM kemudian meng-compile source code tersebut. Setelah itu NS3, scenario *tap-bridge*, yaitu *tap-csma-virtual-machine* dan *tap-wifi-virtual-machine* ditambahkan beberapa baris kode. Tujuan dari perubahan kode ini adalah:

1. *tap-csma-virtual-machine* diubah untuk pengujian konektivitas. Tujuannya membuktikan koneksi *tun/tap bridge* yang digunakan dalam NS3 sudah berfungsi sebagaimana seharusnya.
2. *tap-wifi-virtual-machine* diubah untuk model mobilitas yang nantinya akan dipergunakan dalam DTN. Model pergerakan yang digunakan dalam skenario ini adalah *random waypoint mobility model*.

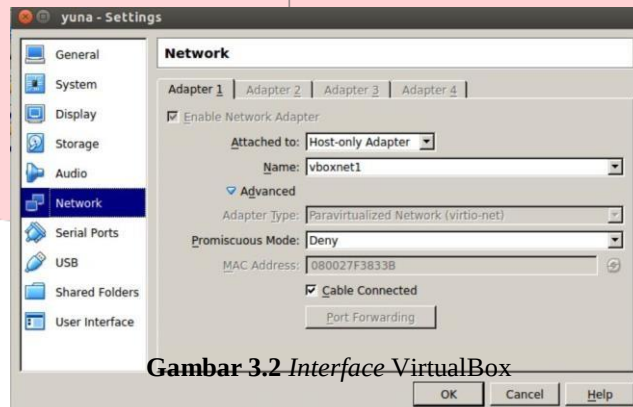


Gambar 3.2 Implementasi Jaringan

pengujian ini menggunakan total 5 *node*. Dimana 4 *node* akan dijalankan dalam komputer yang sama dengan NS3 yang disebut sebagai *node internal*, sedangkan satu node dihubungkan ke yang ada di luar sebagai *node eksternal*.

berikutnya adalah instalasi Virtualbox di Komputer yang menjalankan NS3 dan Komputer yang akan menjadi node eksternal. Hal ini dilakukan karena IBR-DTN secara *native* berjalan diatas system operasi *Linux*.

Untuk *node* internal, pada tab *settings* pilih *host only adapter* dan pilih *network interface* yang nantinya akan di *bridge* dengan *tun/tap interface*. Secara default, *interface* ini akan bernama *vboxnet* yang nantinya akan diikuti nomor.



Gambar 3.2 Interface VirtualBox

3.3 Skenario Pengujian

a. Uji Konektivitas

Pengujian konektivitas meliputi 2 jenis pengujian, yaitu uji *ping* ke *node* internal dan uji *ping* ke *node* eksternal

b. Pengujian Dengan Variasi Ukuran *Bundle*

Pengujian variasi ukuran *bundle* dilaksanakan dengan parameter sesuai tabel 3.1 dan untuk ukuran *bundle* yang diuji sesuai dengan Tabel 3.2

Tabel 3.1 Parameter Skenario Variasi Ukuran *Bundle*

Paramater	Besaran
Ukuran File	32.2MB
TTL	7200 Seconds
Jumlah Node	5

Tabel 3.2 Variasi Ukuran *Bundle*

Routing Protocol	Bundle 1	Bundle 2	Bundle 3
Epidemic	100 KB	500 KB	1500 KB
Static	100 KB	500 KB	1500 KB

c. Pengujian Dengan Variasi Ukuran *File*

Pengujian variasi ukuran *file* dilaksanakan dengan parameter sesuai tabel 3.3 dan untuk ukuran *file* yang diuji sesuai dengan tabel 3.4

Tabel 3.3 Parameter Skenario Variasi Ukuran *File*

Paramater	Besaran
Ukuran <i>Bundle</i>	500KB
TTL	7200 Seconds
Jumlah Node	5

Tabel 3.4 Variasi Ukuran *File*

Routing Protocol	File 1	File 2	File 3
Epidemic	10.8 MB	32.2 MB	40 MB
Static	10.8 MB	32.2 MB	40 MB

d. Pengujian Dengan Variasi Ukuran TTL

Pengujian variasi ukuran TTL dilaksanakan dengan parameter sesuai tabel 3.5 dan untuk ukuran *bundle* yang diuji sesuai dengan Tabel 3.6

Tabel 3.5 Parameter Skenario Ukuran TTL

Paramater	Besaran
Ukuran File	32.2MB
Ukuran Bundle	500KB
Jumlah Node	5

Tabel 3.6 Parameter Ukuran TTL

Routing Protocol	TTL 1	TTL 2	TTL 3
Epidemic	3600 Seconds	7200 Seconds	14400 Seconds
Static	3600 Seconds	7200 Seconds	14400 Seconds

e. Perbandingan Node Diam dan Node Bergerak

Pengujian ini menunjukkan seberapa jauh perbedaan waktu untuk pengiriman data antara node bergerak dengan node diam. Parameter yang digunakan dibahas pada tabel 3.7

Tabel 3.7 Parameter Perbandingan Node Diam dan Node Bergerak

Parameter	Node Diam	Node Bergerak
Ukuran File	32.2 MB, 10.8 MB	32.2 MB, 10.8 MB
Ukuran Bundle	500 KB	500 KB
Ukuran TTL	3600 Seconds	3600 Seconds
Jumlah Node	5	5

4. Pengujian dan Analisa Sistem

a. Uji Konektivitas

Pengujian pertama dalam tahap ini adalah *dtntping* dari *node 1* ke *node 2*, *node 3*, dan *node 4*. Setelah skenario pada NS3 dijalankan dengan perintah `./waf --run tap-wifi-virtual-machine`.

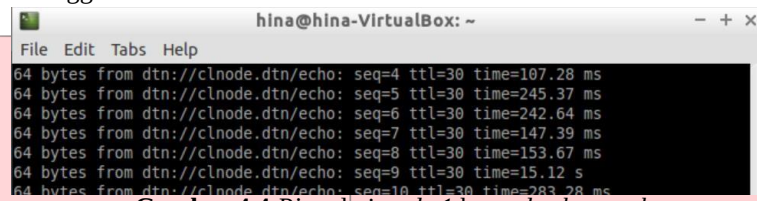
```
hina@hina-VirtualBox:~$ dtntping dtn://node2.dtn/echo
ECHO dtn://node2.dtn/echo 64 bytes of data.
64 bytes from dtn://node2.dtn/echo: seq=1 ttl=30 time=10.86 s
64 bytes from dtn://node2.dtn/echo: seq=2 ttl=30 time=135.41 ms
64 bytes from dtn://node2.dtn/echo: seq=3 ttl=30 time=113.86 ms
64 bytes from dtn://node2.dtn/echo: seq=4 ttl=30 time=227.90 ms
64 bytes from dtn://node2.dtn/echo: seq=5 ttl=30 time=69.24 ms
64 bytes from dtn://node2.dtn/echo: seq=6 ttl=30 time=136.36 ms
64 bytes from dtn://node2.dtn/echo: seq=7 ttl=30 time=204.35 ms
```

```
64 bytes from dtn://node3.dtn/echo: seq=18 ttl=30 time=96.90 ms
64 bytes from dtn://node3.dtn/echo: seq=19 ttl=30 time=97.72 ms
64 bytes from dtn://node3.dtn/echo: seq=20 ttl=30 time=191.82 ms
64 bytes from dtn://node3.dtn/echo: seq=21 ttl=30 time=145.21 ms
64 bytes from dtn://node3.dtn/echo: seq=22 ttl=30 time=153.48 ms
64 bytes from dtn://node3.dtn/echo: seq=23 ttl=30 time=244.73 ms
64 bytes from dtn://node3.dtn/echo: seq=24 ttl=30 time=155.59 ms
64 bytes from dtn://node3.dtn/echo: seq=25 ttl=30 time=251.74 ms
```

```
ECHO dtn://node4.dtn/echo 64 bytes of data.
64 bytes from dtn://node4.dtn/echo: seq=1 ttl=30 time=616.31 ms
64 bytes from dtn://node4.dtn/echo: seq=2 ttl=30 time=981.61 ms
64 bytes from dtn://node4.dtn/echo: seq=3 ttl=30 time=1.04 s
64 bytes from dtn://node4.dtn/echo: seq=4 ttl=30 time=349.30 ms
64 bytes from dtn://node4.dtn/echo: seq=5 ttl=30 time=248.87 ms
64 bytes from dtn://node4.dtn/echo: seq=6 ttl=30 time=341.32 ms
64 bytes from dtn://node4.dtn/echo: seq=7 ttl=30 time=10.86 s
```

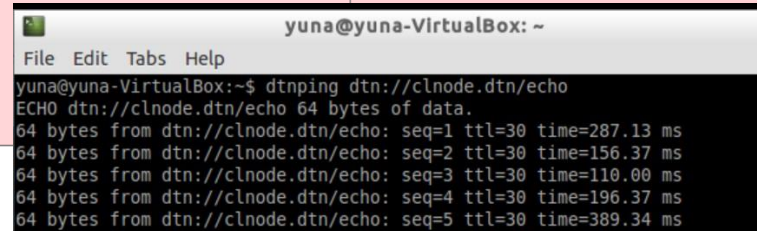
Dari hasil pengujian, kesimpulan yang dapat diambil adalah *node* internal dapat berfungsi dengan baik. Hal ini terlihat dari komunikasi antara kedua *node* tersebut. Baik menggunakan protokol *TCP/IP* dan protokol *DTN*. *Tun/tap interface* yang digunakan dalam NS3 berfungsi dengan baik dan NS3 dapat menjalankan fungsi virtualisasi jaringan dan sudah sesuai untuk pemodelan sistem.

Setelah sebelumnya menguji komunikasi antar *node* internal, pengujian berikutnya adalah konektivitas *node* eksternal. Pengujian ini menggunakan 5 *node*.



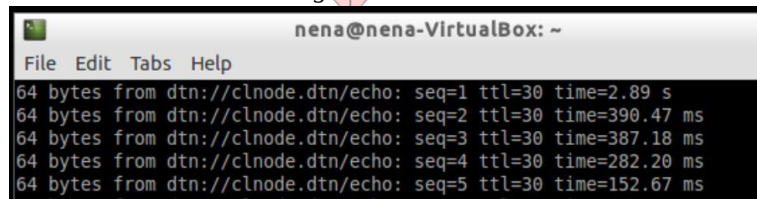
```
hina@hina-VirtualBox: ~
File Edit Tabs Help
64 bytes from dtn://clnode.dtn/echo: seq=4 ttl=30 time=107.28 ms
64 bytes from dtn://clnode.dtn/echo: seq=5 ttl=30 time=245.37 ms
64 bytes from dtn://clnode.dtn/echo: seq=6 ttl=30 time=242.64 ms
64 bytes from dtn://clnode.dtn/echo: seq=7 ttl=30 time=147.39 ms
64 bytes from dtn://clnode.dtn/echo: seq=8 ttl=30 time=153.67 ms
64 bytes from dtn://clnode.dtn/echo: seq=9 ttl=30 time=15.12 s
64 bytes from dtn://clnode.dtn/echo: seq=10 ttl=30 time=283.28 ms
```

Gambar 4.4 Ping dari node 1 ke node eksternal



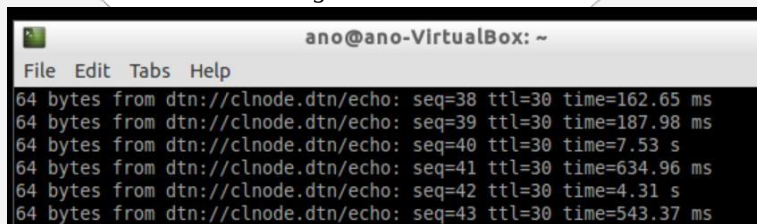
```
yuna@yuna-VirtualBox: ~
File Edit Tabs Help
yuna@yuna-VirtualBox:~$ dtnping dtn://clnode.dtn/echo
ECHO dtn://clnode.dtn/echo 64 bytes of data.
64 bytes from dtn://clnode.dtn/echo: seq=1 ttl=30 time=287.13 ms
64 bytes from dtn://clnode.dtn/echo: seq=2 ttl=30 time=156.37 ms
64 bytes from dtn://clnode.dtn/echo: seq=3 ttl=30 time=110.00 ms
64 bytes from dtn://clnode.dtn/echo: seq=4 ttl=30 time=196.37 ms
64 bytes from dtn://clnode.dtn/echo: seq=5 ttl=30 time=389.34 ms
```

Gambar 4.5 Ping dari node 2 ke node eksternal



```
nena@nena-VirtualBox: ~
File Edit Tabs Help
64 bytes from dtn://clnode.dtn/echo: seq=1 ttl=30 time=2.89 s
64 bytes from dtn://clnode.dtn/echo: seq=2 ttl=30 time=390.47 ms
64 bytes from dtn://clnode.dtn/echo: seq=3 ttl=30 time=387.18 ms
64 bytes from dtn://clnode.dtn/echo: seq=4 ttl=30 time=282.20 ms
64 bytes from dtn://clnode.dtn/echo: seq=5 ttl=30 time=152.67 ms
```

Gambar 4.6 Ping dari node 3 ke node eksternal



```
ano@ano-VirtualBox: ~
File Edit Tabs Help
64 bytes from dtn://clnode.dtn/echo: seq=38 ttl=30 time=162.65 ms
64 bytes from dtn://clnode.dtn/echo: seq=39 ttl=30 time=187.98 ms
64 bytes from dtn://clnode.dtn/echo: seq=40 ttl=30 time=7.53 s
64 bytes from dtn://clnode.dtn/echo: seq=41 ttl=30 time=634.96 ms
64 bytes from dtn://clnode.dtn/echo: seq=42 ttl=30 time=4.31 s
64 bytes from dtn://clnode.dtn/echo: seq=43 ttl=30 time=543.37 ms
```

Gambar 4.7 Ping dari node 4 ke node eksternal

Komunikasi antara seluruh *node* internal dengan *node* eksternal berfungsi dengan baik. Ke-4 *node* yang berjalan dalam simulasi NS3 dapat berkomunikasi dan berhubungan dengan *node* eksternal. *Node* eksternal dapat berkomunikasi dengan *node* eksternal melalui interface *eth0/Ethernet* yang di-*bridge* dengan *tun/tap interface*. Komunikasi dengan protokol *TCP/IP* dan *DTN* berfungsi dengan baik dan NS3 berfungsi sebagaimana seharusnya dalam pemodelan sistem

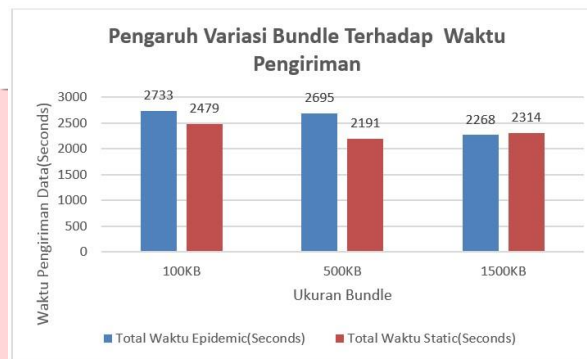
b. Pengujian Dengan Variasi Ukuran *Bundle*

Pengujian ini dibagi menjadi 2 bagian. Bagian pertama adalah pengujian pengaruh ukuran *bundle* terhadap waktu pengiriman data dengan algoritma *routing epidemic*. Bagian Kedua adalah pengujian pengaruh ukuran *bundle* terhadap waktu pengiriman data dengan algoritma *routing static* dengan *packet forwarding* diaktifkan.

Berikut ini adalah hasil pengujian dengan menggunakan algoritma *epidemic*.

Tabel 4.1 Pengaruh Variasi Bundle Terhadap Waktu Pengiriman Data pada Algoritma Routing Epidemic dan Static

Ukuran Bundle(KB)	100KB	500KB	1500KB
Total Waktu Epidemic(Seconds)	2733	2695	2268
Total Waktu Static(Seconds)	2479	2191	2314



Gambar 4.1 Pengaruh Variasi Bundle Terhadap Waktu Pengiriman Data pada Algoritma Routing Epidemic dan Static

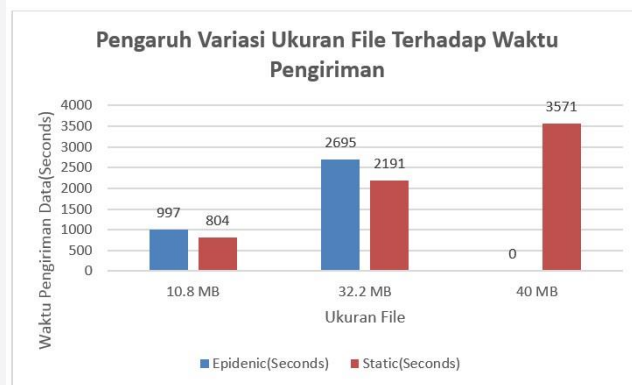
Pada algoritma *epidemic* dan *static* perubahan ukuran bundle memiliki dampak yang signifikan terhadap waktu pengiriman *file*. Secara umum dapat disimpulkan bahwa ukuran bundle yang lebih besar berakibat pada waktu pengiriman data yang lebih cepat.

c. Pengujian Dengan Variasi Ukuran File

Pengujian ini bertujuan untuk mengetahui pengaruh ukuran *file* terhadap waktu pengiriman data menggunakan *IBR-DTN*.

Tabel 4.2 Pengaruh Variasi Ukuran File Terhadap Waktu Pengiriman pada algoritma Routing Epidemic dan Static

Ukuran File (MB)	10.8 MB	32.2 MB	40 MB
Total Waktu Pengiriman Epidemic(Seconds)	997	2695	0
Total Waktu Pengiriman Static(Seconds)	804	2191	3571



Grafik 4.2 Pengaruh Variasi Ukuran File Terhadap Waktu Pengiriman pada Algoritma Routing Epidemic dan Static

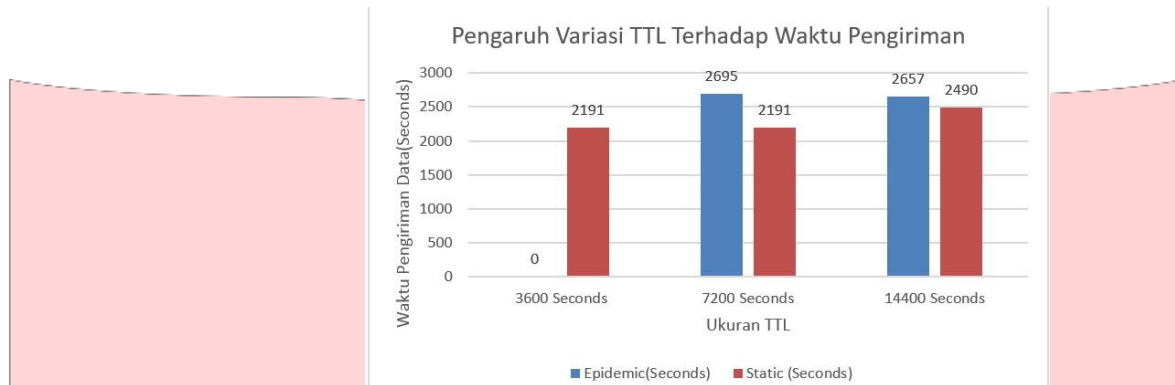
Semakin Besar Ukuran *file*, semakin lama waktu yang dibutuhkan untuk menyelesaikan pengiriman *file* tersebut dan semakin besar pula kemungkinan kegagalan pengiriman. Dalam pengujian tersebut, ukuran file 40MB diblok merah karena file gagal terkirim. Perlu diketahui bahwa berdasarkan pengujian[7] yang telah dilakukan sebelumnya bahwa *IBR-DTN* memiliki ukuran *file* maksimum yang dapat dikirim berukuran 130 MB.

d. Pengujian Dengan Variasi Ukuran TTL

Pengujian ini memiliki tujuan untuk mengetahui pengaruh perubahan TTL terhadap waktu pengiriman *file*.

Tabel 4.3 Pengaruh Variasi TTL Terhadap Waktu Pengiriman pada Algoritma Routing Epidemic dan Static

Ukuran TTL(Seconds)	3600	7200	14400
Total Waktu Epidemic(Seconds)	0	2695	2657
Total Waktu Static (Seconds)	2191	2191	2490



Grafik 4.3 Pengaruh Variasi TTL Terhadap Waktu Pengiriman pada Algoritma Routing Epidemic dan Static

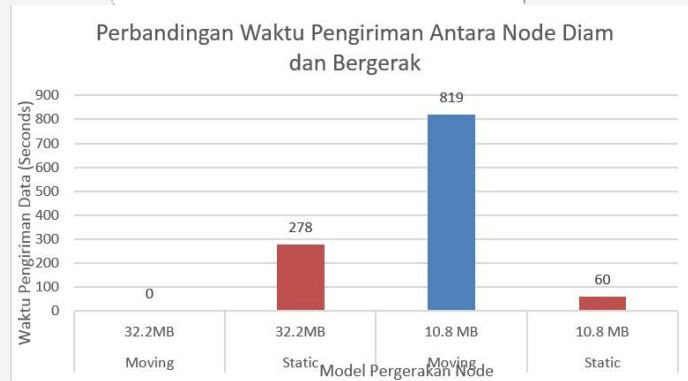
Dari grafik tersebut, dapat dilihat bahwa perubahan TTL relatif tidak signifikan dalam waktu pengiriman, namun sangat berpengaruh terhadap keberhasilan pengiriman informasi. Dari grafik tersebut tampak jelas bahwa pengiriman dengan TTL 3600 second gagal dalam pengiriman dengan menggunakan *epidemic routing protocol* dimana *static routing protocol* berhasil mengirimkan data tersebut

e. Perbandingan Node Diam dan Bergerak

Pengujian ini menunjukkan seberapa jauh perbedaan waktu untuk pengiriman data antara node bergerak dengan node diam.

Tabel 4.4 Perbandingan Waktu Pengiriman Data Pada Node Bergerak dan Diam

Kondisi Node	Bergerak	Statis	Bergerak	Statis
Ukuran File	32.2MB	32.2MB	10.8 MB	10.8 MB
Waktu Pengiriman	0	278	819	60



Tabel 4.4 Perbandingan Waktu Pengiriman Data Pada Node Bergerak dan Diam

Hasil pengujian menunjukkan perbedaan yang signifikan antara pengiriman data dalam kondisi *node* diam dan saat *node* bergerak. Bahkan kegagalan pengiriman data saat mengirim file berukuran 32.2MB dengan TTL 3600S. Variasi waktu pengiriman dan kegagalan ini disebabkan oleh perbedaan kondisi pergerakan node.

5. Kesimpulan

1. NS3 berhasil dihubungkan dengan jaringan eksternal dengan memanfaatkan koneksi Ethernet yang di-bridge dengan tun/tap interface. Hasil *ping* antara *node* internal dengan *node* internal dan antara *node* internal dan *node* eksternal membuktikan bahwa seluruh *node* dapat saling mendeteksi dan bertukar informasi dan data.
2. Makin besar ukuran *bundle* waktu yang dibutuhkan untuk pengiriman data menjadi lebih singkat. Peningkatan performa ini lebih terlihat pada *routing protocol epidemic* dibandingkan dengan *routing protocol static*.
3. Makin besar ukuran *file*, maka akan terjadi peningkatan waktu pengiriman antar *node*. Kenaikan waktu pengiriman ini memiliki efek yang lebih signifikan pada *epidemic routing protocol* jika dibandingkan dengan *static routing protocol*.
4. Perubahan TTL sangat berpengaruh terhadap keberhasilan pengiriman *file*. Secara signifikan, peningkatan ukuran TTL terlihat pada *epidemic routing protocol* dibandingkan dengan *static routing protocol*.

5. Batas maksimum ukuran *file* yang dapat dikirim oleh IBR-DTN, dalam pengujian ini dengan *epidemic routing protocol* dengan menggunakan *node* bergerak adalah 32.2MB.
6. *Static routing protocol* secara konsisten memberikan performa yang baik dan kemampuan memanfaatkan jaringan secara baik jika dibandingkan dengan *epidemic routing protocol* pada pengujian yang dilakukan dalam tugas akhir ini. Dapat dilihat dari 3 pengujian yang telah dilakukan, *epidemic routing protocol* gagal mengirimkan file berukuran 32.2 MB pada pengujian variasi ukuran *file* dan gagal mengirimkan *file* 10.8MB dengan TTL 3600 seconds pada pengujian variasi TTL dimana *static routing protocol* berhasil pada kedua pengujian tersebut. Perlu diingat bahwa jaringan yang dibangun dalam pengujian tugas akhir ini tergolong jaringan kecil dikarenakan keterbatasan alat pengujian.
7. Dibandingkan saat *node* bergerak, *node* diam mampu mengirim *file* lebih cepat dan dalam tempo waktu yang lebih singkat.

Daftar Pustaka

- [1] Warthman, Forrest. 2012. *Delay-and-Disruption-Tolerant Networks*. Warthman Associates
- [2] Wikipedia. 2015. *Routing in Delay Tolerant Networking*. [Online] Available at: https://id.wikipedia.org/wiki/Routing_in_delay_tolerant_networking [Accessed 11 Desember 2015]
- [3] Muri, Paul & Mcnair, Janise. 2012. *Simulating Delay Tolerant Networking for CubeSats*. Wireless and Mobile Systems lab, Electrical and Computer Engineering Department. University of Florida. Florida, USA
- [4] Fall, Kevin. 2003. *Delay-Tolerant Networking: Architecture and Application*. Intel Research, Berkeley
- [5] Ns-3 dev. 2013. Ns-3 Tutorial. Ns-3 project
- [6] Carneiro, Gustavo J.A.M. 2010. NS-3 Network Simulator 3. INESC porto, Faculdade de Engenharia, Universidade de Porto.
- [7] Georgescu, Marius, Nara Institute of Science and Technology .2014. Performance Analysis of File Transmission in DTN2 and IBR-DTN
- [8] Pöttner, Wolf-Bastian & Morgenroth , Johannes & Schildt , Sebastian & Wolf, Lars. IBR, Technische Universität Braunschweig. An Empirical Performance Comparison of DTN Bundle Protocol Implementations
- [9] Pöttner, Wolf-Bastian & Morgenroth , Johannes & Schildt , Sebastian & Wolf, Lars . Workshops der wissenschaftlichen Konferenz Kommunikation in verteilten Systemen . 2014. IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation
- [10] A. Broder, M. Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics* 1(4):485–509, 2004
- [11] H. Kruse, S. Ostermann. UDP Convergence Layers for the DTN Bundle and LTP Protocols. IETF Draft, 2008. <http://tools.ietf.org/pdf/draft-irtf-dtnrg-udp-clayer-00.pdf>
- [12] A. Lindgren, E. Davies, A. Doria. Epidemic Routing Module for Generic Opportunistic Routing Framework, IETF Draft, 2014
- [13] J. Dowdell, N. Benamar. Static Routing for DTN, IETF Draft, 2015. <http://tools.ietf.org/pdf/draft-dowdell-dtnwg-static-00.pdf>
- [14] Ali Marandi, Mahdi Faghi Imani, Kav'e Salamatian. Practical Bloom filter based epidemic forwarding and congestion control in DTNs: A comparative analysis. *Computer Communications*, Elsevier, 2014, 48, pp.98-110.
- [15] A. Broder, M. Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics* 1(4):485–509, 2004.
- [16] B. Sipos, M. Demmer, J. Ott, S. Perreaul. Delay-Tolerant Networking TCP Convergence Layer Protocol Version 4, 2016. <http://tools.ietf.org/pdf/draft-ietf-dtn-tcpclv4-01.pdf>
- [17] Jain, Sushanti. Fall, Kevin. Patra. *Routing In A Delay Tolerant Network*
- [18] *Virtualization in Education*. IBM. October 2007
- [19] VirtualBox.org. 2017. Virtual Machines. [Online] Available at: <https://www.virtualbox.org/wiki/Virtualization> [Accessed 17 January 2017]