



Analisis Performansi *Spray and Focus* pada *Vehicular Delay Tolerant Network* (VDTN) dengan Perubahan Kecepatan dan Kepadatan *Node*

Performance Analysis of Spray and Focus on Vehicular Delay Tolerant Network (VDTN) with Change of Speed Mechanism and Node Density

Ilman Syakir Saputra¹, Doan Perdana²

^{1,2}Fakultas Teknik Elektro, Telkom University

^{1,2}Jl. Telekomunikasi No1, Terusan Buah Batu, Bandung, 40257, Indonesia

email: ¹ilmanssaputra@gmail.com, ²doanperdana@telkomuniversity.ac.id

INFORMASI ARTIKEL

Naskah diterima 16 Mei 2018

Direvisi 9 Desember 2018

Disetujui 9 Desember 2018

Keywords:

Spray and Focus

One Simulator

Packet Delivery Ratio

Average Latency

Kata kunci :

Spray and Focus

One Simulator

Packet Delivery Ratio

Average Latency

ABSTRACT

This study aims to introduce technology that is being developed to be able to send data packages without being directly connected to the backbone network called Delay Tolerant Network (DTN). DTN is a wireless network where the time of nodes to communicate cannot be determined or in other words the relationship between nodes rare. Unlike conventional network Mobile Ad Hoc Network (MANET), the end-to-end between the sources with destination node will only be ready in a short time and are not predictable. The node on the DTN can be a source node, intermediate nodes, or destination node, consisting of mobile nodes and static nodes connected with high delay. In this research will be discussed about VDTN performance using Spray and Focus and Spray and Wait routing protocol. The design of the simulation system is divided into two clusters, cluster's speed 90 km/h and cluster's Speed 65 km/h. ONE Simulator is used to simulate the performance of the routing algorithm. The performance is evaluated using average latency and Packet Delivery Ratio (PDR). Based on observation, the result of Spray and Focus routing algorithm improves the performance of PDR than Spray and Wait routing algorithm, but Spray and Wait have better average latency than Spray and Focus routing algorithm.

ABSTRAK

Penelitian ini bertujuan memperkenalkan teknologi yang sedang dikembangkan untuk dapat mengirimkan paket data tanpa langsung terhubung dengan jaringan *backbone*, teknologi tersebut dinamakan *Delay Tolerant Network* (DTN). DTN merupakan jaringan nirkabel dengan kondisi *node* yang berkomunikasi tidak dapat ditentukan waktunya atau hubungan antara *node* jarang terjadi. Tidak seperti jaringan konvensional *Mobile Ad Hoc Network* (MANET), jalur *end-to-end* antara sumber dengan tujuan hanya akan tersedia dalam waktu yang singkat dan tidak dapat diprediksi. *Node* pada DTN dapat menjadi *source node*, *intermediate node*, maupun *node* tujuan *node*, terdiri dari *mobile node* dan *static node* yang terhubung dengan *delay* tinggi. Pada penelitian ini akan dibahas mengenai kinerja VDTN yang menggunakan protokol *routing Spray and Focus* dan *Spray and Wait*. Perancangan sistem simulasi terbagi menjadi dua *cluster* yaitu *cluster* dengan kecepatan 90 km/jam dan *cluster* dengan kecepatan 65 km/jam. Kinerja algoritma *routing* ini disimulasikan menggunakan ONE Simulator. Performansi dievaluasi dengan *average latency* dan *Packet Delivery Ratio* (PDR). Observasi yang dilakukan menunjukkan hasil bahwa *Spray and Focus* memiliki performa yang lebih baik dalam PDR dibandingkan *Spray and Wait*, tetapi *Spray and Focus* memiliki performa *average latency* yang lebih besar dibandingkan *Spray and Wait*.

1. Pendahuluan

Vehicular Delay Tolerant Networks (VDTNs) *routing protocols* telah menjadi topik *research* yang sedang banyak dibicarakan dalam *global context of Intelligent Transport Systems* (ITS) karena teknologi

ini bertujuan sebagai alternatif mentransmisikan data yang tidak dapat terpenuhi oleh TCP/IP (Hamza dkk, 2017). Contoh pengimplementasiannya di daerah urban adalah teknologi ini dapat membantu mengurangi tingkat kecelakaan di jalur bebas hambatan (tol), teknologi membuat antarkendaraan dapat berkomunikasi dengan yang lainnya sehingga ketika terjadi kecelakaan, teknologi ini dapat memberitahu kendaraan yang lain yang berada di sekitarnya tanpa menggunakan jaringan *backbone* sehingga informasi dapat dengan cepat diterima oleh kendaraan yang lain dan mengurangi dampak yang lebih besar. Contoh lainnya adalah membantu mentransmisikan data di tempat '*blank spot*' dimana transmisi radio tidak dapat menjangkaunya dikarenakan gedung-gedung dan penghalang lainnya. Untuk mencapai tujuan ini, dibutuhkan juga *protocol routing* yang baik. Ini adalah tantangan selanjutnya dalam VDTN. Oleh karena itu, jurnal ini akan membandingkan antara 2 *protocol routing* yang dikhususkan untuk teknologi VDTN sehingga tujuan dari teknologi VDTN dapat tercapai dengan baik.

Delay Tolerant Network (DTN) merupakan salah satu jenis jaringan nirkabel dimana *node* berkomunikasi dengan *node* yang lain atau hubungan antara *node* tersebut jarang terjadi atau dalam delay yang cukup besar. Tidak seperti jaringan konvensional *Mobile Ad Hoc Network* (MANET), jalur *end-to-end* antara sumber dengan tujuan hanya akan tersedia dalam waktu yang singkat dan tidak dapat diprediksi. Karakter dari DTN adalah dinamis. *Node* pada DTN dapat menjadi *source node*, *intermediate node* (*relay*), maupun *destination node*, dan terhubung dengan *delay* yang tinggi. Jaringan seperti pada kasus ini disebut *opportunistic network*.

Opportunistic network adalah pengembangan dari MANET. *Opportunistic network node* dapat terhubung dengan *node* yang lain walaupun tidak ada rute antara kedua *node* tersebut. Selanjutnya, *node* yang bekerja pada *opportunistic network* tidak mengetahui topologi pada jaringan yang sedang dijalankan *node* tersebut, hal itu sangat dibutuhkan pada MANET konvensional (Anjula Mehto dkk, 2013). Pada *opportunistic network*, topologi jaringan tidak tergambar secara jelas karena *node* selalu bergerak dengan kecepatan dan pergerakan acak. Karena koneksi *end-to-end* tidak selalu ada maka mobilitas pada *opportunistic network* dapat digunakan sebagai sarana komunikasi antara kelompok *node* yang terpisah dengan menggunakan paradigma *store-carry-and-forward* (N. Benamar dkk, 2014). Dengan metode *store*, *carry*, dan *forward* sebuah paket data saat melewati *node-node* perantara (*router*) akan disimpan terlebih dahulu sebelum diteruskan. Hal ini untuk mengantisipasi seandainya *node* berikutnya tidak dapat dijangkau (mati) atau ada kendala lain (A. Mazlan, 2014).

VDTN adalah komunikasi berbasis DTN dengan karakteristik bahwa kendaraan tersebut memiliki *router* dan berkomunikasi dengan unit kendaraan yang lain untuk menerima data dari trafik sumber sampai ke tujuan. Jurnal ini akan menganalisis performansi dari algoritma *routing Spray and Focus* pada VDTN untuk mengetahui pengaruh dari perubahan besar kepadatan *node*, *message size*, dan *time to live*. *Routing protocol* ini dinamakan *Spray and Focus* dimana paket dikirimkan ke tujuan dengan memilih *node* yang pernah berhubungan dengan *node* tujuan agar pesan dapat dengan cepat terkirim ke *node* tujuan.

2. Tinjauan Pustaka

2.1. Penelitian Terkait

Jurnal berjudul "Analisa Performansi Protokol *Routing DTN Maxprop* dan *Spray and Wait* Pada *Vehicular Ad Hoc Network* (VANET): Bandung High Way" (Putri dkk, 2016) melakukan penelitian menggunakan aplikasi ONE (*Opportunistic Network Environment*) Simulator dengan menggunakan *routing protocol Spray and Wait* dan *Maxprop*. Tempat yang diambil sebagai tempat pergerakan *node* adalah salah satu jalan bebas hambatan (tol) yang ada di Bandung.

2.2. ONE Simulator 1.4.1

ONE Simulator adalah simulator yang dirancang khusus untuk mengevaluasi *routing* pada DTN. ONE Simulator dibuat agar DTN mudah dimengerti. Tiga fungsi utama ONE Simulator adalah *node movement*,

inter-node contacts, routing and message handling. Dengan menggunakan ONE, bisa dirancang pergerakan simulasi *pe-rute-an*, visualisasi, dan laporan.

ONE Simulator memiliki beberapa pemodelan pergerakan yang dapat diimplementasikan dari yang sederhana misalnya *Random Waypoint* sampai pemodelan yang realistis *Map-Based Movement*. ONE Simulator dapat mengimpor data pergerakan dari *real-world traces*. Map yang digunakan untuk pemodelan pergerakan menggunakan *Well Known Text* (WKT) format. Dengan pemodelan *map-based movement*, *node* bergerak menggunakan jalur yang ada pada *map*. Selain itu, kelompok *node* yang berbeda dapat diatur untuk menggunakan bagian atau rute tertentu pada *map*.

2.3. VANET (Vehicular Ad hoc Network)

Vehicular Ad Hoc Network (VANET) merupakan jaringan komunikasi *wireless*, secara teori VANET juga merupakan perkembangan dari MANET, perbedaan VANET dengan MANET yaitu laju kecepatan (mobilitas) VANET sangatlah cepat, bisa melebihi 100 km/jam. Jadi dapat dikatakan bahwa VANET merupakan salah satu sistem komunikasi trafik yang cerdas. Jenis komunikasi pada VANET terbagi menjadi tiga macam, V2V (*vehicle to vehicle communication*), V2I (*vehicle to Infrastructure communication*), dan I2I (*infrastructure to infrastructure communication*).

Secara teori jaringan *wireless* dibagi dua yaitu dengan dan tanpa infrastruktur. Sedangkan topologi jaringan VANET termasuk jaringan *wireless* dengan infrastruktur, karena pada topologi jaringan VANET terdapat AP (*access point*) yang berfungsi untuk berkomunikasi antara satu *node* dan *node* lainnya. Topologi jaringan VANET juga termasuk jaringan *ad hoc* bergerak karena topologi jaringannya yang selalu berubah-ubah posisi.

2.4. DTN (Delay Tolerant Network)

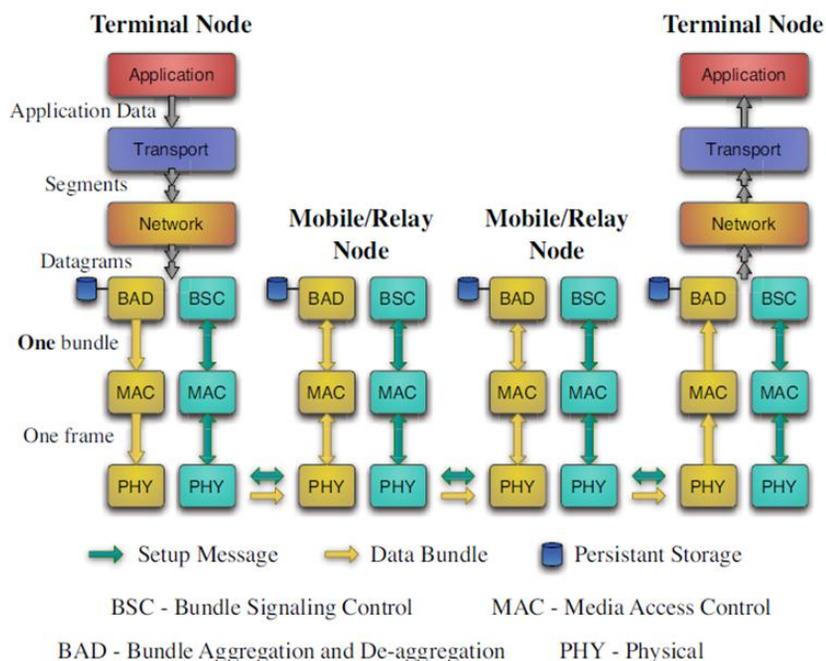
Delay Tolerant Network adalah jaringan yang awalnya dikembangkan untuk digunakan pada komunikasi antarplanet yang mensyaratkan toleransi *delay* sangat besar. Namun, DTN mungkin memiliki aplikasi lebih beragam di bumi, yang mensyaratkan toleransi *delay* sebagai kebutuhan paling besar. DTN mendukung pertukaran informasi dengan jaringan lain yang memiliki *delay* yang besar dan menerjemahkan komunikasi protokol antarjaringan tersebut (F. Warthman, 2012).

Delay Tolerant Network diperkenalkan pertama kali oleh Kevin Fall dalam makalah yang berjudul “*A Delay-Tolerant Network Architecture for Challenged Internets*”. Dalam makalah tersebut Kevin berpendapat bahwa DTN merupakan suatu arsitektur yang cocok untuk “*challenged network*”. *Challenged network* yang dimaksud adalah jaringan yang penuh masalah seperti koneksi yang sering terputus karena pergerakan dan kecepatan *node* yang *random*, *delay* yang lama, dan *error* yang tinggi (K. Fall, 2013).

2.5. VDTN (Vehicular Delay Tolerant Network)

VDTN merupakan perkembangan dari teknologi VANET, konsep VDTN sendiri secara teori adalah teknologi perpaduan antara teknologi VANET dan teknologi DTN. Permasalahan utama dalam pengiriman paket-paket data pada VANET untuk V2V, V2I, maupun I2I adalah *delay* yang sulit diprediksi. Algoritma *routing* berbasis DTN seperti *epidemic routing*, *Spray and Wait routing*, dan sebagainya merupakan algoritma *routing* yang paling efektif untuk teknologi VANET ini karena pada algoritma *routing* berbasis DTN ini menggunakan konsep probabilitas (peluang).

Gambar 1. mengilustrasikan *layer* yang bekerja pada VDTN:



Gambar 1. Transmisi *message* dan *data bundle* pada VDTN dari *source node* ke *destination node* (João N.G. Isento dkk, 2014)

2.6. Routing Protocol

Routing adalah proses mencari jalur (mencari *line* yg akan dilewati) paling optimal yang menghubungkan *nodes* sumber dan *nodes* tujuan agar paket-paket data yang dikirimkan tepat waktu. Bisa saja rute pada topologi jaringan VANET ini memiliki banyak *nodes* lain yang harus dilalui. Selain itu dari sekian banyak protokol *ad-hoc*, protokol *routing* berbasis DTN dinilai sebagai protokol *routing* yang paling efisien untuk VANET (Anjula dkk, 2013).

Beberapa *routing* protokol memanfaatkan kemampuan dari perangkat untuk mengatasi koneksi yang terputus-putus, perangkat tersebut memproses dan menyimpan data (*bundle*) sampai bertemu *intermediate node*, kemudian data (*bundle*) tersebut akan di *forward* ke *intermediate node* sesuai dengan skema pengiriman *hop-by-hop forwarding* atau skema *routing* yang digunakan. Proses ini akan diulang sampai *bundle* dapat diteruskan menuju *node* tujuan berdasarkan paradigma *store-carry-and-forward* (E. Spaho dkk, 2015).

2.7. Spray and Wait Routing Protocol

Tujuan utama dari *Spray and Wait routing protocol* adalah untuk mengurangi *copy* pesan yang terdapat di *epidemic routing protocol*. Pengurangan ini dilakukan dengan 2 tahap yaitu dengan *fase spray* dan *fase wait* (Muhammad R.P, 2016).

2.7.1. Fase Spray

Fase pertama adalah fase *spray* dimana *source node* membuat *copy* pesan untuk disebar ke *relay node*. Fase *spray* membatasi *copy* pesan untuk meminimalkan penggunaan sumber daya (*resource*) jaringan. Pada fase *spray*, proses *multi-cast* dilakukan untuk mengirim beberapa *copy* pesan dari *source* ke *internode (relay)*. Jika *destination* tidak ditemukan dalam fase *spray* maka *node* akan memasuki fase *wait*.

2.7.2. Fase Wait

Fase kedua adalah fase *wait*. Jika *destination* tidak ditemukan dalam fase *spray*, maka setiap *relay node* yang membawa *copy* pesan melakukan transmisi langsung ke tujuan yaitu meneruskan pesan hanya

untuk *node* tujuan. Pada fase *wait*, *node* dapat menyampaikan pesan ke *destination* menggunakan transmisi secara langsung ketika *time-to-live* belum berakhir.

2.8. Spray and Focus Routing Protocol

Tujuan utama dari *Spray and Focus routing protocol* adalah untuk mengurangi *delay* yang didapatkan pada *Spray and Wait routing protocol* (Maria H., 2016).

2.8.1. Fase *Spray*

Fase pertama adalah fase *spray* dimana *source node* membuat *copy* pesan untuk disebarakan ke *relay node*. Fase *spray* membatasi *copy* pesan untuk meminimalkan penggunaan sumber daya (*resource*) jaringan. Pada fase *spray*, proses *multi-cast* dilakukan untuk mengirim beberapa *copy* pesan dari *source* ke *internode* (*relay*). Jika *destination* tidak ditemukan dalam fase *spray* maka *node* akan memasuki fase *focus*.

2.8.2. Fase *Focus*

Fase kedua adalah fase *focus*. Jika *destination* tidak ditemukan dalam fase *spray*, maka setiap *relay node* yang membawa *copy* pesan melakukan transmisi langsung ke *node* yang ditemuannya dengan syarat *node* tersebut memiliki ID dari *destination node* atau *copy* pesan tersebut akan di-*drop* karena *time to live* dari pesan tersebut sudah habis.

3. Metode Penelitian

Metode penelitian yang digunakan adalah sebagai berikut :

1. Studi Literatur

Mempelajari konsep dan dasar teori yang berhubungan dengan jurnal ini, di antaranya konsep VANET, konsep DTN, dan konsep algoritma *routing Spray and Focus*.

2. Perancangan Sistem

Merupakan tahap percobaan atau mencoba membuat sistem dengan menerapkan hasil dari tahapan sebelumnya. Pada tahap ini dilakukan simulasi *Spray and Focus routing* pada DTN menggunakan *Opportunistic Network Environment* (ONE) Simulator 1.4.1 untuk mengetahui parameter performansi berdasarkan parameter yang diuji. ONE Simulator memiliki fitur untuk men-*generate* pergerakan *node* dengan menggunakan berbagai jenis *movement model*, selain itu simulator ini dapat menggunakan DTN *routing protocol*. Fitur lainnya adalah dapat memonitor secara *real time* mobilitas *node* dan perpindahan paket dengan tampilan *graphical user interface*.

3. Pengujian dan Analisis Simulasi

- a) Mengambil *file* *Roads.wkt* dan *Main Roads.wkt*. *Road.wkt* dan *Main Road.wkt* adalah tampilan jalan salah satu kota besar sehingga cukup menggambarkan situasi urban.
- b) Menyimulasikan algoritma *Spray and Focus router* dan *Spray and Wait router* pada ONE Simulator.
- c) Hasil simulasi pada ONE Simulator tersebut menghasilkan *file output* dan disimpan pada *folder Reports*.
- d) *File output* berupa data dari *Packet Delivery Ratio* (PDR) dan *average latency*. Karena 2 parameter ini dapat digunakan untuk mengetahui QoS dari suatu *routing protocol* dalam konsep DTN.
- e) Analisis dengan membandingkan hasil dari algoritma *Spray and Focus router* dan *Spray and Wait router*.

4. Hasil Penelitian dan Pembahasan

Pada bagian ini akan dijelaskan tentang performa dari algoritma *routing Spray and Focus* dan *Spray and Wait* sesuai nilai PDR dan *average latency*. Percobaan dilakukan dengan memvariasikan kepadatan

node, ukuran *message*, dan lamanya *time to live*. Masing-masing skenario dibagi menjadi dua, dengan ukuran *file* 20 kB - 30 kB dan 400 kB - 600 kB. Data didapatkan dari hasil simulasi menggunakan ONE Simulator. PDR adalah nilai yang memperlihatkan perbandingan antara jumlah paket yang dikirim dari *source node* dengan jumlah paket yang berhasil sampai di *destination node* (Dewanta dkk, 2016). *Average latency* adalah nilai rata-rata yang diambil dari *latency/end-to-end delay*. *End-to-end delay* adalah rentang waktu yang dibutuhkan suatu paket sampai paket tersebut mencapai tujuan (Tonapa dkk, 2014).

4.1. Skenario Simulasi

Skenario simulasi akan dibuat sedemikian sehingga dapat mencapai tujuan penelitian yaitu untuk mengetahui kinerja *routing* protokol *Spray and Focus*. Dengan demikian dibuat 3 skenario yaitu skenario dengan variasi ukuran paket, variasi kepadatan *node*, dan variasi *Time to Live* (TTL). Terdapat 2 *cluster* yang digunakan pada setiap skenario simulasi, yaitu *cluster* dengan kecepatan 90 km/jam dan *cluster* dengan kecepatan 65 km/jam. Secara rinci skenario simulasi dapat dilihat pada Tabel 1 dan 2.

Tabel 1. Variasi kepadatan *node*

Parameter Uji	Nilai
Ukuran Paket	20kB – 30kB, 400kB – 600kB
Ukuran Buffer	5 MB
Kecepatan Node	90 km/jam (<i>Cluster 1</i>), 65 km/jam (<i>Cluster 2</i>)
Kepadatan Node	60 – 150 <i>node</i>
<i>Time to Live</i>	60 menit

Tabel 2. Variasi TTL

Parameter Uji	Nilai
Ukuran Paket	20kB – 30kB, 400kB – 600kB
Ukuran Buffer	5 MB
Kecepatan Node	90 km/jam (<i>Cluster 1</i>), 65 km/jam (<i>Cluster 2</i>)
Kepadatan Node	150 <i>node</i>
<i>Time to Live</i>	5 - 50 menit

4.2. Parameter Simulasi

Simulasi dilakukan dengan menggunakan perangkat lunak ONE Simulator versi 1.4.1. Parameter simulasi ditentukan berdasarkan standar-standar yang digunakan pada jaringan VDTN. Pada Tabel 2 diperlihatkan parameter-parameter yang digunakan dalam simulasi ini.

Tabel 3. Parameter simulasi

Sistem Operasi	Linux Ubuntu 12.04 64 bit (dalam vmware windows 10 64 bit)
Simulator Jaringan	ONE Simulator (version 1.4.1)
Algoritma Routing	<i>Spray and Focus</i> dan <i>Spray and Wait</i>
Model Mobilitas	<i>Shortest Path Map Based Movement</i>
Kepadatan Node	60 – 150 <i>node</i>
Ukuran Buffer	5MB
Waktu Simulasi	18000 detik/ 5 jam
Kecepatan Pengiriman Data	2Mbps
Message Size	20 kB – 30 kB, 400 kB – 600 kB
Time to Live	5 – 50 menit, 60 menit
Kecepatan Node Cluster 1	90 km/jam
Kecepatan Node Cluster 2	65 km/jam

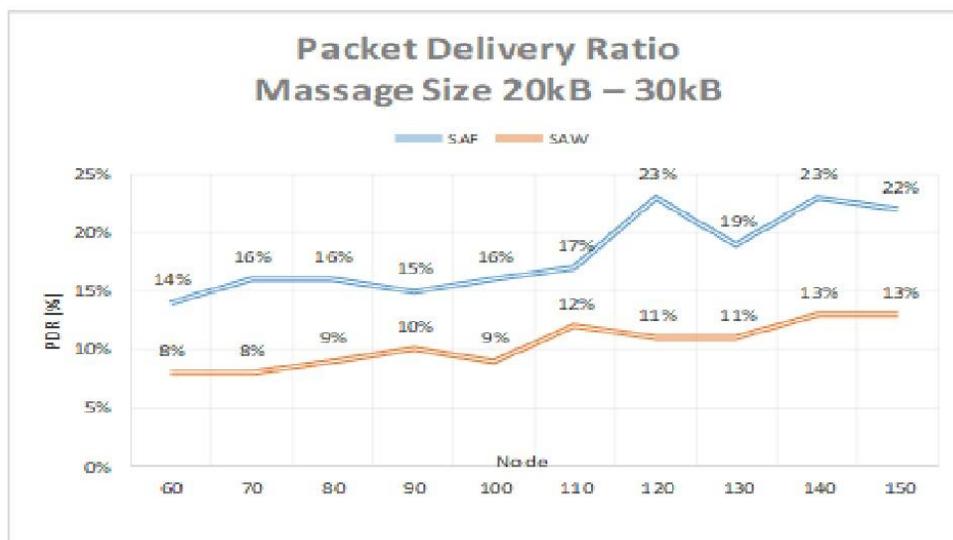
4.3. Analisis Pengujian dengan Penambahan Kepadatan *Node* (20 kB – 30 kB)

Pada bagian ini akan dijelaskan performansi dengan scenario memvariasikan kepadatan *node*. Nilai kepadatan *node* yang digunakan adalah 60 – 150 *node* dengan penambahan 10 *node* setiap *step* dengan ukuran *message* 20 kB – 30 kB.

Tabel 4. Hasil pengujian dengan variasi kepadatan *node*

Kepadatan <i>Node</i>	Jumlah <i>Message</i> yang Dibuat		Jumlah <i>Message</i> yang Sampai	
	SaF	SaW	SaF	SaW
60	1344	979	87	52
70	1836	1231	100	51
80	1894	1407	96	54
90	2119	1579	90	63
100	2219	1607	94	58
110	2410	1821	104	76
120	2950	2269	144	70
130	2928	2203	116	69
140	3128	2359	140	80
150	3382	2493	132	83

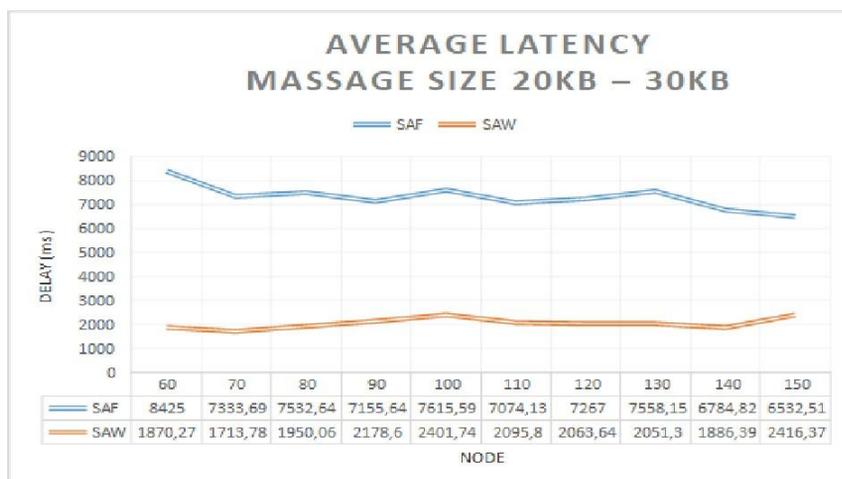
Dan berikut grafik yang dihasilkan dari skenario ini:



Gambar 2. Hasil PDR dengan skenario penambahan *node*

Pengukuran pertama menganalisis PDR dengan menggunakan *message size* 20kB – 30kB. Hasilnya menunjukkan bahwa kepadatan *node* dari 50 *node* sampai 150 *node* berpengaruh terhadap PDR yang didapatkan, semakin bertambah kepadatan *node*-nya maka persentase PDR nya semakin besar. Selain itu dari kedua algoritma *routing* yang digunakan *Spray and Focus* menunjukkan hasil yang lebih baik dibandingkan *Spray and Wait*. Pada saat kepadatan 150 *node* persentase PDR sebesar 22% sedangkan *Spray and Wait* mendapatkan persentase PDR 14%.

Pada pengukuran pertama, hasil analisis *average latency* dengan menggunakan *message size* 20kB – 30kB menunjukkan bahwa kepadatan *node* berpengaruh terhadap *average latency* nya. Semakin besar kepadatan *node* nya semakin bagus pula *average latency* yang didapatkan. Tetapi melalui Gambar 3 dapat dilihat bahwa algoritma *routing Spray and Focus* mendapatkan *average latency* yang lebih besar dibandingkan dengan algoritma *routing Spray and Wait*. Hal ini dijelaskan melalui Tabel 4 bahwa paket yang terkirim oleh algoritma *routing Spray and Focus* selalu lebih besar atau sama dengan *Spray and Wait*.



Gambar 3. Hasil average latency dengan skenario penambahan node

Sebagai contoh pada kepadatan 70 node, *Spray and Focus* dapat mengirimkan 100 pesan sampai kepada node tujuan, sedangkan pada *Spray and Wait* hanya mampu mengirimkan 51 pesan yang berhasil sampai ke node tujuan. Hasil yang didapatkan dari *result file* adalah dengan menggunakan *average latency* sehingga nilai yang didapatkan oleh *Spray and Focus* akan menjadi besar dikarenakan banyak paket yang berhasil terkirim ke node tujuan. Sebagai contoh jika pesan yang berhasil dikirim oleh *Spray and Focus* ke node tujuan pada menit–menit terakhir sebelum simulasi berakhir, maka akan sangat memengaruhi hasil dari *average latency* yang didapatkan.

4.4. Analisis Pengujian dengan Penambahan Kepadatan Node (400 kB – 600 kB)

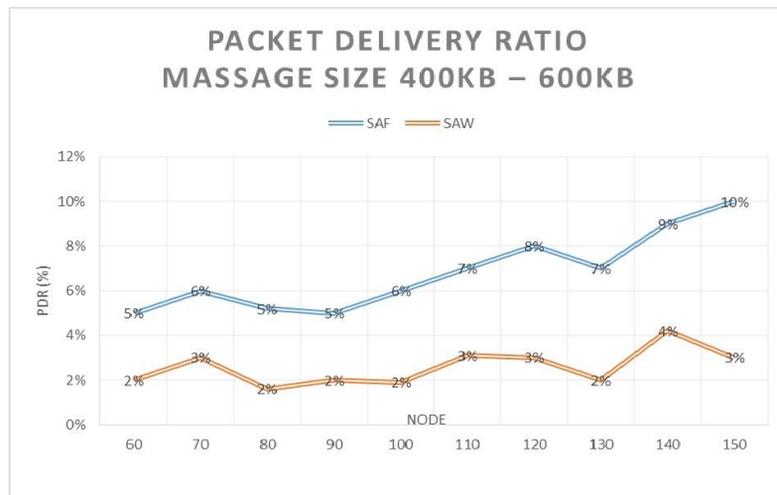
Pada bagian ini akan dijelaskan performansi dengan skenario memvariasikan *Time to Live*. Nilai *Time to Live* yang digunakan adalah 5–50 menit dengan penambahan 5 menit setiap *step* dengan ukuran *message* 20 kB – 30 kB.

Pada bagian ini akan dijelaskan performansi dengan scenario memvariasikan kepadatan node. Nilai kepadatan node yang digunakan adalah 60–150 node dengan penambahan 10 node setiap *step* dengan ukuran *message* 400 kB – 600 kB.

Tabel 5. Hasil pengujian dengan variasi kepadatan Node

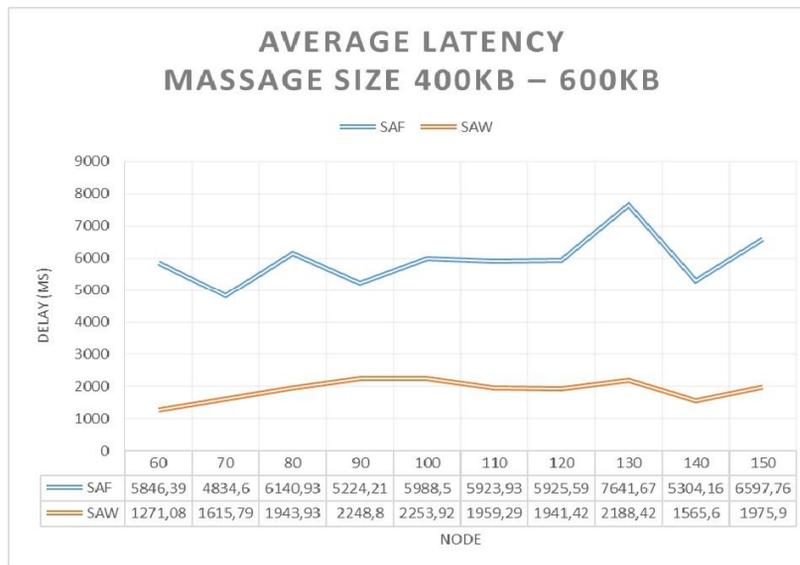
Kepadatan Node	Jumlah Message yang Dibuat		Jumlah Message yang Sampai	
	SaF	SaW	SaF	SaW
60	747	818	31	12
70	1094	1094	35	19
80	1161	1226	30	15
90	1349	1396	33	15
100	1411	1444	36	12
110	1691	1706	41	17
120	2092	2205	49	19
130	2081	2189	43	12
140	2366	2422	56	25
150	2511	2633	59	21

Dan berikut grafik yang dihasilkan dari skenario ini:



Gambar 4. Hasil PDR dengan skenario penambahan *node*

Pengukuran kedua menganalisis PDR dengan menggunakan *message size* 400 kB – 600 kB. Hasil menunjukkan bahwa kepadatan *node* dari 50 *node* sampai 150 *node* berpengaruh terhadap PDR yang didapatkan, semakin bertambah kepadatan *node*-nya maka persentase PDR nya semakin besar. Walaupun tidak setiap kenaikan kepadatan *node* meningkatkan persentase PDR nya, tetapi secara umum kepadatan *node* dari 50 sampai 150 meningkatkan PDR yang didapatkan. Jika dibandingkan dengan Gambar 2 dapat dilihat bahwa Gambar 2 lebih baik dibandingkan Gambar 4, dikarenakan algoritma *routing* mendapatkan beban lebih dengan mengirim *message size* yang besarnya 20 kali dibandingkan pada Gambar 2, oleh karena itu terdapat penurunan kualitas PDR. Selain itu dari kedua algoritma *routing* yang digunakan *Spray and Focus* menunjukkan hasil yang lebih baik dibandingkan *Spray and Wait*. Pada saat kepadatan 140 *node*, persentase PDR *Spray and Focus* sebesar 9% sedangkan *Spray and Wait* mendapatkan persentase PDR 4%.



Gambar 5. Hasil *average latency* dengan skenario penambahan *node*

Pengukuran kedua untuk menganalisis *average latency* dengan menggunakan *message size* 400 kB– 600 kB. Hasilnya menunjukkan bahwa kepadatan *node* berpengaruh terhadap *average latency* nya. Semakin besar kepadatan *node* nya semakin besar pula *average latency* yang didapatkan. Tetapi dari Gambar 5 dapat dilihat bahwa algoritma *routing Spray and Focus* mendapatkan *average latency* yang lebih besar dibandingkan dengan algoritma *routing Spray and Wait*. Hal ini dapat dijelaskan melalui Tabel 5 bahwa paket yang terkirim oleh algoritma *routing Spray and Focus* selalu lebih besar atau sama dengan *Spray and Wait*. Sebagai contoh pada kepadatan 70 *node*, *Spray and Focus* dapat mengirimkan 35 pesan

sampai kepada *node* tujuan, sedangkan pada *Spray and Wait* hanya mampu mengirimkan 19 pesan yang berhasil sampai ke *node* tujuan, padahal *message* yang dibuat oleh kedua algoritma sama-sama 1094 *message*. Selain itu, hasil yang didapatkan dari *result file* adalah dengan menggunakan *average latency* sehingga membuat nilai yang didapatkan oleh *Spray and Focus* akan menjadi besar dikarenakan banyak paket yang berhasil terkirim ke *node* tujuan. Sebagai contoh jika pesan yang berhasil dikirim oleh *Spray and Focus* ke *node* tujuan pada menit-menit terakhir sebelum simulasi berakhir, maka akan sangat memengaruhi hasil dari *average latency* yang didapatkan.

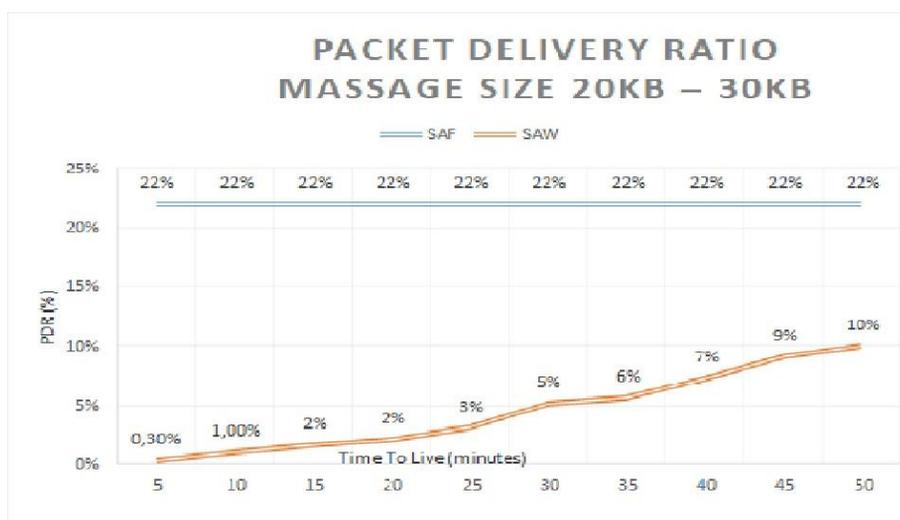
4.5. Analisis Pengujian dengan Penambahan *Time to Live* (20 kB – 30 kB)

Pada bagian ini akan dijelaskan performansi dengan skenario memvariasikan *Time to Live*. Nilai *Time to Live* yang digunakan adalah 5–50 menit dengan penambahan 5 menit setiap *step* dengan ukuran *message* 20 kB – 30 kB.

Tabel 6. Hasil Pengujian dengan Variasi *Time to Live*

<i>Time to Live</i>	Jumlah <i>Message</i> yang Dibuat		Jumlah <i>Message</i> yang Sampai	
	SaF	SaW	SaF	SaW
5	3382	307	132	2
10	3382	705	132	6
15	3382	1047	132	9
20	3382	1365	132	12
25	3382	1601	132	21
30	3382	1807	132	30
35	3382	1967	132	37
40	3382	2108	132	44
45	3382	2245	132	55
50	3382	2340	132	63

Dan berikut grafik yang dihasilkan dari skenario ini:

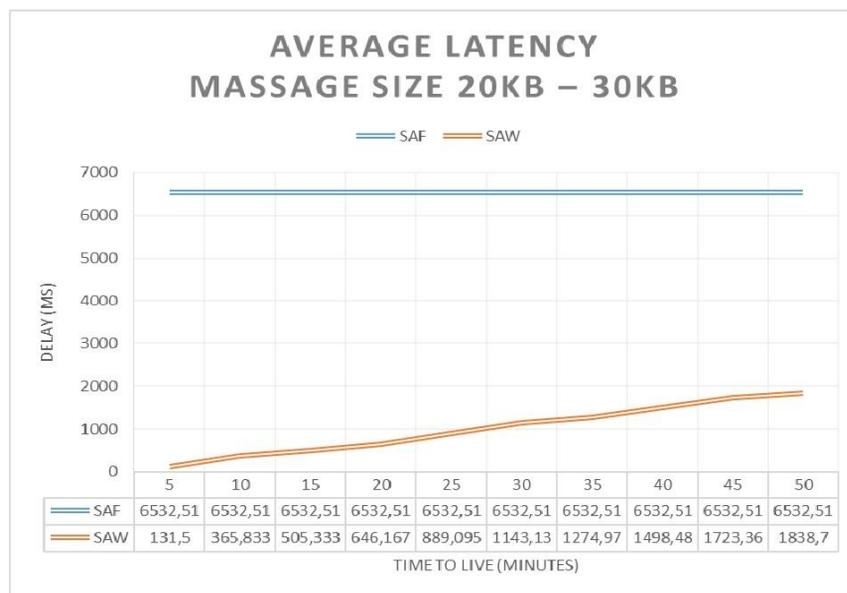


Gambar 6. Hasil PDR dengan skenario penambahan *Time to Live*

Pengukuran ketiga untuk menganalisis PDR dengan menggunakan *message size* 20kB–30kB. Hasilnya menunjukkan bahwa *Time to Live* pada *node* berpengaruh terhadap PDR yang didapatkan. Semakin besar *Time to Live* yang digunakan *node* maka persentasi PDR nya semakin besar. Tetapi dapat dilihat dari Gambar 6 bahwa *Spray and Focus* memberikan hasil pengukuran yang tidak berubah walaupun TTL ditingkatkan. Pada bagian *message* Tabel 6 terlihat bahwa *Spray and Focus* memiliki 3382 *copies* dan

Spray and Wait hanya memiliki 2340 pada TTL 50 menit. Oleh karena itu, *Spray and Focus* memberikan hasil PDR yang stagnan walaupun dengan TTL yang sangat kecil. Selain itu dari kedua algoritma *routing* yang digunakan, *Spray and Focus* menunjukkan hasil yang lebih baik dibandingkan *Spray and Wait*. Pada saat TTL *node* 50 menit persentase PDR sebesar 22% sedangkan *Spray and Wait* mendapatkan persentase PDR 10%.

Pengukuran ketiga untuk menganalisis *average latency* dengan menggunakan *Message size* 20kB–30kB. Hasilnya menunjukkan bahwa *Time to Live* berpengaruh terhadap *average latency*. Semakin besar *Time to Live node* nya semakin besar pula *average latency* yang didapatkan. Melalui Gambar 7 dapat dilihat bahwa algoritma *routing Spray and Focus* mendapatkan *average latency* yang lebih besar dibandingkan dengan algoritma *routing Spray and Wait*. Hal ini dijelaskan melalui Tabel 6 bahwa paket yang terkirim oleh algoritma *routing Spray and Focus* selalu lebih besar atau sama dengan *Spray and Wait*. Sebagai contoh pada TTL 50 menit, *Spray and Focus* telah mengirimkan 132 pesan sampai kepada *node* tujuan, sedangkan pada *Spray and Wait* hanya mampu mengirimkan 63 pesan yang berhasil sampai ke *node* tujuan. Oleh karena itu, *Spray and Focus* memberikan hasil yang stagnan walaupun dengan TTL yang sangat kecil.



Gambar 7. Hasil *average latency* dengan skenario penambahan *node*

4.6. Analisis Pengujian dengan Penambahan *Time to Live* (400 kB – 600 kB)

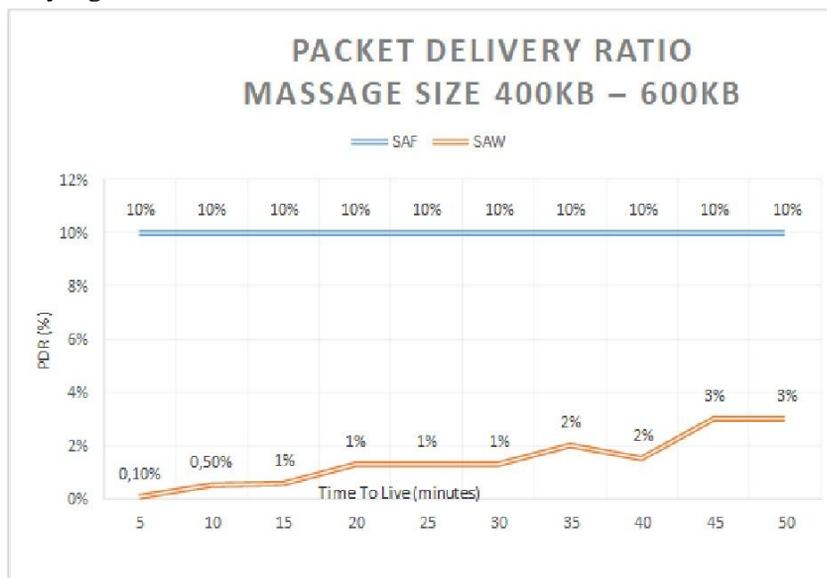
Pada bagian ini akan dijelaskan performansi dengan skenario memvariasikan *Time to Live*. Nilai *Time to Live* yang digunakan adalah 5–50 menit dengan penambahan 5 menit setiap *step* dengan ukuran *message* 400 kB – 600 kB.

Tabel 7. Hasil pengujian dengan variasi *Time to Live*

<i>Time to Live</i>	Jumlah <i>Message</i> yang Dibuat		Jumlah <i>Message</i> yang Sampai	
	SaF	SaW	SaF	SaW
5	2511	283	59	1
10	2511	621	59	3
15	2511	929	59	4
20	2511	1206	59	5
25	2511	1429	59	8
30	2511	1645	59	9

<i>Time to Live</i>	Jumlah Message yang Dibuat		Jumlah Message yang Sampai	
	SaF	SaW	SaF	SaW
35	2511	1864	59	11
40	2511	2054	59	13
45	2511	2225	59	16
50	2511	2381	59	20

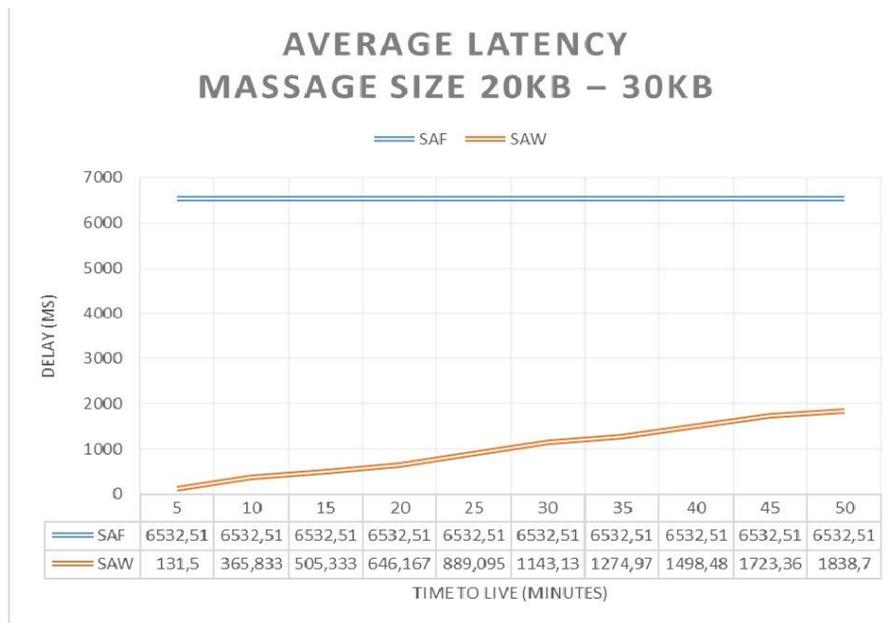
Dan berikut grafik yang dihasilkan dari skenario ini:



Gambar 8. Hasil PDR dengan skenario penambahan *Time to Live*

Pengukuran keempat untuk menganalisis PDR menggunakan *message size* 400kB – 600kB. Hasilnya menunjukkan bahwa *Time to Live* pada *node* berpengaruh terhadap PDR yang didapatkan. Semakin besar *Time to Live* yang digunakan *node* maka persentase PDR nya semakin besar. Melalui Gambar 8 dapat dilihat bahwa *Spray and Focus* memberikan hasil pengukuran yang tidak berubah walaupun TTL ditingkatkan. Pada bagian *message* Tabel 7 terlihat bahwa *Spray and Focus* memiliki 2511 *copies* dan *Spray and Wait* hanya memiliki 2381 pada TTL 50 menit. Oleh karena itu, *Spray and Focus* memberikan hasil PDR yang stagnan walaupun dengan TTL yang sangat kecil. Selain itu dari kedua algoritma *routing* yang digunakan, *Spray and Focus* menunjukkan hasil yang lebih baik dibandingkan *Spray and Wait*. Pada saat TTL *node* 50 menit, persentase PDR sebesar 10% sedangkan *Spray and Wait* mendapatkan persentase PDR 3%.

Pengukuran keempat untuk menganalisis *average latency* dengan menggunakan *message size* 20kB–30kB. Hasilnya menunjukkan bahwa *Time to Live* berpengaruh terhadap *average latency* nya. Semakin besar *Time to Live* *node* nya, semakin besar pula *average latency* yang didapatkan. Melalui Gambar 9 dapat dilihat bahwa algoritma *routing* *Spray and Focus* mendapatkan *average latency* yang lebih besar dibandingkan dengan algoritma *routing* *Spray and Wait*. Hal ini dijelaskan melalui Tabel 7 bahwa paket yang terkirim oleh algoritma *routing* *Spray and Focus* selalu lebih besar atau sama dengan *Spray and Wait*. Sebagai contoh pada TTL 50 menit, *Spray and Focus* telah mengirimkan 59 pesan sampai kepada *node* tujuan, sedangkan pada *Spray and Wait* hanya mampu mengirimkan 20 pesan yang berhasil sampai ke *node* tujuan. Oleh karena itu, *Spray and Focus* memberikan hasil yang stagnan walaupun dengan TTL yang sangat kecil.



Gambar 9. Hasil *average latency* dengan skenario penambahan *node*

5. Simpulan dan Saran

Kesimpulan yang dapat diambil dari seluruh proses perancangan, simulasi, dan analisis adalah sebagai berikut :

1. Algoritma *routing Spray and Focus* merupakan algoritma *routing* terbaik untuk PDR dibanding *Spray and Wait routing*. Dengan capaian persentase PDR tertinggi 22% pada saat kepadatan *node* = 150 *node*, *message size* 20 kB–30 kB, 10% pada *message size* 400 kB–600 kB, dan dengan kepadatan 150 *node*. Karena pada saat *Focus Phase*, setiap *node* dapat menduplikasi *message*-nya dan mengirimkannya ke *node relay* yang mempunyai ID *destination node* atau langsung ke *destination node*.

2. Algoritma *routing Spray and Focus* mempunyai hasil *average latency* yang lebih besar dibandingkan *Spray and Wait routing* untuk *message size* 20 kB–30 kB dan *message size* 400 kB–600 kB. Hal ini dikarenakan jumlah *message* pada *Spray and Focus* yang sampai membuat *average latency* menjadi besar.

3. Algoritma *routing Spray and Focus* memiliki hasil yang stagnan pada skenario penambahan TTL, sedangkan *Spray and Wait* memberikan peningkatan performa pada skenario penambahan TTL. Tetapi *Spray and Focus* masih memiliki performa yang lebih baik dibandingkan *Spray and Wait*.

4. Pada penambahan kepadatan *node* kedua *routing*, secara keseluruhan mengalami peningkatan performa, baik pada *Spray and Focus* dan *Spray and Wait* karena penambahan *node* sampai 150 *node* membuat *message* yang dikirim menjadi lebih cepat menemukan *destination node*.

5. Pada penambahan ukuran *message* kedua *routing*, secara keseluruhan mengalami penurunan performa, baik pada *Spray and Focus* dan *Spray and Wait* karena dengan ukuran *message* yang besar dan dengan kecepatan pengiriman yang terbatas, hanya 2 Mbps, membuat *message* lebih lama terkirim dan besarnya *message* membuat *layer bundle* menjadi cepat penuh, sehingga membuat banyak *message* di-drop.

6. Algoritma *routing Spray and Focus* memiliki performa yang lebih baik dibandingkan algoritma *routing Spray and Wait* pada VDTN. Selain itu penelitian ini membuktikan bahwa dengan penambahan ukuran paket yang dikirim dapat mengurangi performa dari masing–masing *routing protocol*.

7. Semakin padatnya *traffic* paket data dan tidak pada setiap tempat dapat terjangkau transmisi radio membutuhkan teknologi yang tidak perlu terhubung dengan jaringan *backbone* tetapi tetap dapat mengirimkan paket data. Oleh karena itu, dirancanglah teknologi DTN dan VDTN untuk menjawab tantangan tersebut. Pada sistem VDTN, dibutuhkan algoritma *routing protocol* yang memiliki performa

yang bagus agar diharapkan teknologi ini dapat menjadi jawaban dari permasalahan yang sudah dijabarkan sebelumnya. Dari beberapa algoritma *routing protocol* yang dapat digunakan, diambil 2 *routing protocol* dan kemudian dibandingkan sehingga diharapkan dapat mengetahui *routing protocol* yang lebih baik atau lebih cocok digunakan dalam teknologi VDTN. Untuk mencapai hasil yang akurat maka digunakanlah aplikasi ONE Simulator karena fitur yang cocok untuk mensimulasikan teknologi VDTN ini. Untuk mendekati dengan keadaan sebenarnya maka digunakan 2 jenis Cluster, Cluster 1 dengan kecepatan 90km/h dan Cluster 2 dengan kecepatan 65 km/h. Cluster adalah pengelompokan kendaraan yang dikonfigurasi untuk lebih menggambarkan keadaan yang lebih nyata.

Daftar Pustaka

- E. Spaho, L. Barolli, V. Kolicic dan A. Lala. (2015). "Performance Evaluation of Different *Routing Protocols* in a Vehicular *Delay Tolerant Network*," International Conference on Broadband and Wireless Computing, Communications and Applications, pp. 157- 162.
- Mehto, A, Chawla, M. (2013). Comparing *Delay Tolerant Network Routing Protocols* for Optimizing L-Copies in *Spray and wait Routing* for Minimum Delay. *Advances in Communication and Control Systems*, 239- 244
- F. Warthman. (2012). *Delay- and Disruption- Tolerant Networks (DTNs)*.
- Tonapa, O., Rahmiati, P., & Komba, D. (2014). Analisis Performansi Konektifitas Pada Jaringan Wireless Broadband di Bandung. *Jurnal Elkomika Institut Teknologi Nasional*, 168.
- K. Fall. (2003). A Delay-Tolerant *Network Architecture* for *Challenged Internets*, *SIGCOMM*, 27-34.
- João N.G. Isento, Joel J.P.C. Rodrigues, Sandra Sendra, and Guangjie Han. (2014). Performance Analysis of Aggregation Algorithms for Vehicular Delay-Tolerant *Networks*.
- Febryan, Perdana, D., & Dewanta, F. (2016). *ANALISA PERFORMANSI PENGIMBANGAN BEBAN TRAFIK KENDARAAN MENGGUNAKAN VANET*. Bandung, Indonesia: School of Electrical Engineering, Telkom University.
- Dr. Mazlan Abbas. (2014). Trends and Challenges in *Delay Tolerant Network (DTN) or Mobile Opportunistic Network (OppNet)*, UTHM 2
- N. Benamar, K. D. Singh, M. Benamar, D. El Ouadghiri, and J.-M. Bonnin. (2014). "Routing Protocols in Vehicular *Delay Tolerant Networks*: A Comprehensive Survey", *Computer Communications*, vol. 48.
- Muhammad, R. P. (2016). Analisis Performansi dan Energy Consumption pada Vehicular *Delay Tolerant Network (VDTN)* dengan Memodifikasi Algoritma *Routing Spray and Wait*. Universitas Telkom.
- Maria, H. (2016). Analisis Unjuk Kerja Protokol *Spray and focus* di Jaringan Opportunistic. Universitas Sanata Dharma.
- Hamza, A.C, Khaled, B., Gladys, D., Didi, F.. (2017). Improving the performances of geographic VDTN routing protocols.
- Putri, S.N, Yovita, L.V, Perdana. D. (2016). Analisa Performansi Protokol Routing DTN Maxprop dan Spray and Wait Pada Vehicular Ad Hoc Network (VANET) : Bandung High W. SENIATI, B. 362-B. 370