

ANALISIS PERFORMANSI ROUTER MAXPROP PADA VEHICULAR AD HOC NETWORK BERBASIS DELAY TOLERANT NETWORK

PERFORMANCE ANALYSIS OF MAXPROP ROUTER ON VEHICULAR AD HOC NETWORK BASED DELAY TOLERANT NETWORK

Yasir Ahmad Abdillah¹, Tody Ariefianto Wibowo², Leanna Vidya Yovita³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹yasiroon@students.telkomuniversity.ac.id,

²ariefianto@telkomuniversity.ac.id, ³leanna@telkomuniversity.ac.id

Abstrak

Pada VANET sulit untuk membangun komunikasi *end-to-end* antara sumber dan tujuan karena cakupan dari *node* terbatas dan selalu bergerak dengan cepat. Salah satu jaringan yang dapat digunakan sebagai solusi adalah *Delay Tolerant Network* (DTN). Apabila jaringan terputus saat dalam perjalanan maka data akan disimpan pada *node* terakhir sehingga tidak diperlukan membangun hubungan dari awal lagi.

Pada Delay Tolerant Network, *bandwidth* dan *buffer* yang digunakan sangat terbatas. Akibat keterbatasan ini, kinerja jaringan DTN secara keseluruhan ditentukan oleh skema dan jenis routing yang digunakan. Protokol MaxProp menggunakan beberapa mekanisme dalam upaya meningkatkan *delivery rate* dan meminimalkan *latency* yang disampaikan oleh paket. MaxProp memberikan prioritas yang lebih tinggi untuk paket baru, dan juga upaya untuk mencegah penerimaan paket yang sama dua kali. MaxProp mempunyai performansi yang buruk dengan *buffer* yang kecil, bisa dikatakan MaxProp akan lebih baik digunakan pada *buffer* yang tinggi.

Dari hasil simulasi diperoleh bahwa besar *buffer* router Maxprop untuk lokasi simulasi yaitu Buah Batu, Bandung paling baik pada nilai *buffer* 15 MB. Berdasarkan dua skenario pengujian, *buffer* 15 MB memiliki performansi paling baik dibanding besar *buffer* 5 MB dan 10 MB. Semakin besar paket data yang dikirimkan akan menghasilkan nilai *latency* dan *packetloss* yang semakin besar. Sebaliknya akan menghasilkan nilai *packet delivery ratio* yang semakin kecil.

Kata kunci: Delay Tolerant Network, MaxProp, *latency*, *packetloss*, *packet delivery ratio*

Abstract

~~VANET can't establish end-to-end communication between the source and destination node because of limited coverage and fast movement. Delay Tolerant Network (DTN) is one of network that can be used as a solution. If the network is disconnected while on the way the data will be stored on the last node so it is not necessary to restart connection.~~

On Delay Tolerant Network, *bandwidth* and *buffer* used are very limited. As a result of these limitations, the overall DTN network performance is determined by the scheme and the type of routing that is used. MaxProp protocol uses several mechanisms in an effort to improve the delivery rate and minimize latency delivered by the package. MaxProp give higher priority to the new package, and also an attempt to prevent the receipt of the same package twice. MaxProp has poor performance with a small buffer.

The simulation results showed that a large buffer for the location Maxprop router simulation at Buah Batu Street in Bandung City is best at the value of 15 MB buffer. Based on the two test scenarios, 15 MB buffer has a good performance compared to most large buffer 5 MB and 10 MB. The greater the transmitted data packets will generate greater value of packetloss and latency. Otherwise, it will generate smaller value of packet delivery ratio.

Keywords: Delay Tolerant Network, MaxProp, *latency*, *packetloss*, *packet delivery ratio*

1. Pendahuluan

Pada VANET sulit untuk membangun komunikasi *end-to-end* antara sumber dan tujuan karena cakupan dari *node* terbatas dan selalu bergerak dengan cepat. Salah satu jaringan yang dapat digunakan sebagai solusi adalah *Delay Tolerant Network* (DTN). Apabila jaringan terputus saat dalam perjalanan maka data akan disimpan pada *node* terakhir sehingga tidak diperlukan membangun hubungan dari awal lagi.

Delay tolerant network adalah sebuah jaringan yang dirancang untuk dapat beroperasi secara lancar pada jarak yang sangat jauh. Dalam kondisi seperti itu, latensi yang sangat panjang biasanya diukur dalam

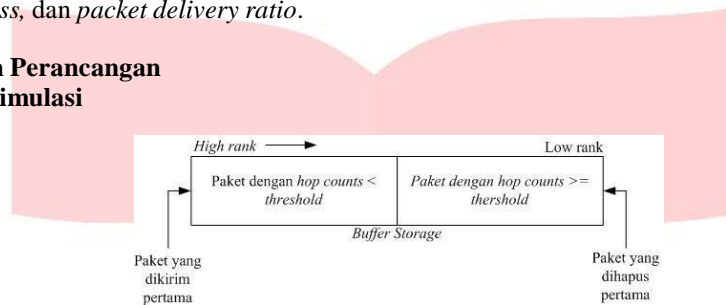
jam atau hari tidak dapat dihindari. Delay tolerant network juga merupakan model arsitektur yang meningkatkan keamanan infrastruktur jaringan dari akses yang tidak sah.^[1]

Pada Delay Tolerant Network, bandwidth dan buffer yang digunakan sangat terbatas. Akibat keterbatasan ini, kinerja jaringan DTN secara keseluruhan ditentukan oleh skema dan jenis routing yang digunakan^[2]. Protokol MaxProp menggunakan beberapa mekanisme dalam upaya untuk meningkatkan *delivery rate* dan meminimalkan *latency* yang disampaikan oleh paket. MaxProp menggunakan beberapa mekanisme untuk menentukan paket yang ditransmisikan dan dihapus. MaxProp memberikan prioritas yang lebih tinggi untuk paket baru, dan juga upaya untuk mencegah penerimaan paket yang sama dua kali. MaxProp mempunyai performansi yang buruk dengan *buffer* yang kecil, bisa dikatakan MaxProp akan lebih baik digunakan pada *buffer* yang tinggi^[3].

Tugas Akhir ini melakukan simulasi VANET berbasis Delay Tolerant Network dengan menggunakan router MaxProp dengan peta daerah Buah Batu, Bandung, Jawa Barat. Selanjutnya akan dilakukan analisis performansi router MaxProp untuk mengetahui pengaruh besar *buffer* dan besar paket untuk mendapatkan *latency*, *packetloss*, dan *packet delivery ratio*.

2. Dasar Teori dan Perancangan

2.1 Diagram Blok Simulasi



Gambar 2. 1 MaxProp Routing Strategi

MaxProp merupakan *forwarding based routing protocol*. Protokol MaxProp menggunakan beberapa mekanisme dalam upaya untuk meningkatkan *delivery rate* dan *lower latency* yang disampaikan oleh paket. MaxProp menggunakan beberapa mekanisme untuk menentukan paket yang ditransmisikan dan dihapus. MaxProp mempunyai performansi yang buruk dengan buffer yang kecil, bisa dikatakan MaxProp akan lebih baik digunakan pada buffer yang tinggi.^[3]

2.2 Perancangan

2.2.1 Perangkat Keras (Hardware)

1. Intel® Core™ i5-2410M 2.3 GHz
2. RAM 2 GB, HDD 500 GB

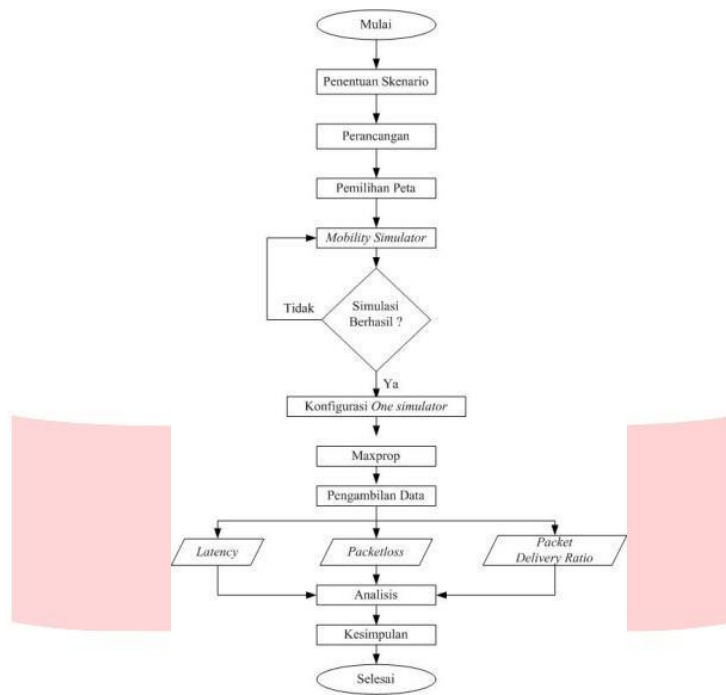
2.2.2 Perangkat Lunak (Software)

- 1 Sistem Operasi: Windows 7 Ultimate
- 2 Pengolahan Data: WKT
- 3 Java versi 1.7.0
- 4 *OpenJUMP 1.8.0 rev.4164 PLUS SUMO mobility generator* versi 0.12.3
- 5 The ONE Simulator versi 1.4.1

2.3 Perancangan Sistem

Sebelum melakukan perancangan dilakukan terlebih dahulu studi literatur. Hal pertama yang harus dilakukan dalam proses perancangan yaitu menentukan parameter. Parameter yang ditentukan adalah daerah perancangan, jumlah node DTN, kecepatan node, tipe MAC, besar data, besar buffer, mobilitas. Perancangan mobilitas node pada tugas akhir ini menggunakan bantuan *software* Open Jump. Hal ini bermaksud untuk mengatur pergerakan node agar lebih mendekati dengan keadaan lapangan. Simulasi pada tugas akhir ini menggunakan bantuan *software* ONE Simulator 1.4.1. Dengan simulasi ini didapatkan nilai *latency*, *packetloss*, *packet delivery ratio* yang selanjutnya akan dianalisis.

Dari tiga tahap utama perancangan terdapat sub-tahap pendukung yang berguna untuk mendapatkan hasil terbaik. Tahap-tahap tersebut tersusun dalam diagram alir perancangan sebagai berikut :



Gambar 2. 2 Flowchart Simulasi Router Maxprop

2.4 Mekanisme MaxProp^[4]

Protokol MaxProp menggunakan beberapa mekanisme dalam meningkatkan delivery rate dan lower latency dari paket yang dikirimkan.

MaxProp menggunakan beberapa mekanisme untuk menentukan urutan paket yang akan ditransmisikan dan dihapus. Gambar 2.12 menggambarkan mekanisme ini. Inti dari protokol MaxProp adalah daftar paket yang disimpan node berdasarkan cost untuk setiap tujuan. Cost adalah perkiraan kemungkinan pengiriman. Selain itu, MaxProp menggunakan acknowledgment yang dikirim ke semua node untuk memberitahukan tentang pengiriman paket. MaxProp memprioritaskan paket baru, dan juga berupaya untuk mencegah penerimaan paket yang sama dua kali. Sisa dari bagian ini mempresentasikan rincian estimasi cost tujuan, mekanisme lainnya, dan manajemen buffer ini.

1. *Estimating Delivery Likelihood*: Variasi dari algoritma Dijkstra dapat menentukan jalur terpendek, jika tersedia.
2. *Complementary Mechanisms*: Tidak seperti protokol lain, MaxProp melibatkan beberapa mekanisme lain yang meningkatkan *delivery rate* dan mengurangi *latency*.
3. *Managing buffers* Perbedaan antara mengelola penyimpanan terbatas dan transmisi terbatas adalah pada paket yang dikirim dalam satu kesempatan transfer dapat dikirim lagi dalam peluang berikutnya. Sebaliknya, jika paket di-drop dari buffer, pesan tidak pernah dikirimkan. Paket yang belum dikirim diberikan prioritas lebih tinggi. Oleh karena itu, MaxProp mencoba untuk memberikan paket baru sebuah *head start* pada jaringan dengan prioritas yang lebih tinggi.

2.5 Penentuan Parameter

2.5.1 Daerah Perancangan

Daerah Buah Batu, Bandung, Jawa Barat merupakan daerah yang tergolong padat. Pada daerah Buah Batu banyak terdapat perumahan, rumah makan, dan sekolah. Dengan melihat kondisi lapangan seperti itu, maka dipilih Buah Batu sebagai peta yang digunakan untuk perancangan dan simulasi Delay Tolerant Network.

2.5.2 Jumlah Node

Dilihat dari banyaknya jumlah armada angkot yang berada di Bandung, maka memungkinkan apabila terdapat banyak angkot pada setiap ruas jalan Bandung. Terutama pada daerah dekat pusat perbelanjaan, cafe, tempat hiburan dan sekolah. Dilihat juga dari gaya hidup masyarakat Bandung yang lebih suka menggunakan kendaraan umum daripada pribadi. Maka diputuskan angkot sebagai node Delay Tolerant Network, dikarenakan jumlah angkot di Bandung tergolong banyak terutama pada daerah Buah Batu

2.5.3 Mobilitas

Untuk model yang lebih baik sesuai dengan mobilitas pada keadaan aslinya , mobilitas berdasarkan map based movement dengan membatasi gerakan node pada jalur yang telah ditetapkan dimana rute ditentukan berasal dari peta. Map based movement bergerak sesuai dengan jalur yang sudah ditentukan. Digunakan map based movement dikarenakan node yang digunakan adalah angkot, sehingga sesuai dengan jalur yang ada.

Tabel 2.1. Parameter Perancangan 1

Parameter	Nilai
Jangkauan <i>Node</i>	250 m
Waktu Simulasi	3600 s
Mac Type	802.11p
Data rates	600 Mbps
Jumlah <i>Node</i> DTN	5,10,15,20,25,30
<i>Node</i> Sumber	<i>Node</i> 0
<i>Node</i> Tujuan	<i>Node</i> 4
Model Mobilitas	Map Based Movement
Ukuran Data	500 KB
<i>Buffer</i>	5 MB, 10 MB, 15 MB
Kecepatan Minimum	0 km/jam
Kecepatan Maksimum	113 km/jam
Router	Maxprop

Tabel 2.2. Parameter Perancangan 2

Parameter	Nilai
Jangkauan <i>Node</i>	250 m
Waktu Simulasi	3600 s
Mac Type	802.11p
Data rates	600 Mbps
Jumlah <i>Node</i> DTN	5,10,15,20,25,30
<i>Node</i> Sumber	<i>Node</i> 0
<i>Node</i> Tujuan	<i>Node</i> 4
Model Mobilitas	Map Based Movement
Ukuran Data	1 MB, 2 MB, 3 MB, 4 MB
<i>Buffer</i>	15 MB
Kecepatan Minimum	0 km/jam
Kecepatan Maksimum	113 km/jam
Router	Maxprop

2.6 Parameter Pengukuran

1. Latency

Waktu tunda merupakan selang waktu yang dibutuhkan oleh suatu paket data saat data mulai dikirim dan keluar dari proses antrian sampai mencapai titik tujuan. Rata-rata waktu tunda merupakan waktu yang diperlukan oleh suatu paket data untuk melakukan pengiriman dari sumber sampai ke tujuan.

2. Packet Loss

Packet Loss adalah banyaknya paket yang hilang selama proses transmisi ke tujuan. Misal dalam perjalanan paket tersebut mengalami tabrakan atau antrian penuh

3. Packet Delivery Ratio

$$PacketLoss = \frac{Packet\ Transmitted - Packet\ received}{Packet\ Transmitted}$$

Packet delivery ratio adalah rasio yang digunakan untuk menghitung jumlah paket data yang dikirimkan oleh node sumber terhadap data yang hilang. Hal ini digunakan untuk menghitung tingkat kehilangan paket data sementara selama transmisi data dalam jaringan. Mengevaluasi tingkat kerugian dan mengukur keberhasilan dan efisiensi routing protokol ad-hoc.

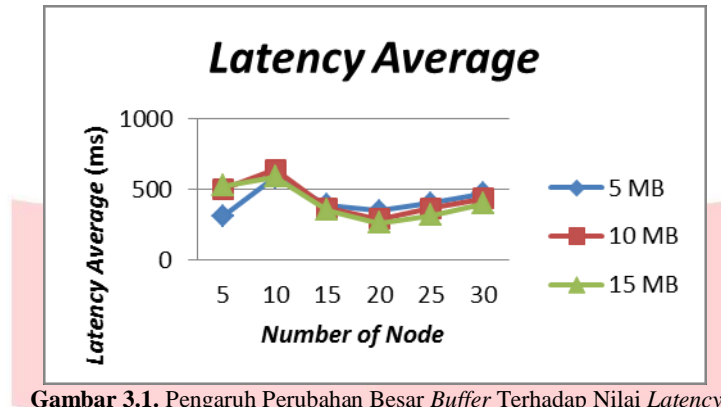
$$Packet\ Delivery\ Ratio = \frac{Jumlah\ paket\ yang\ diterima}{Jumlah\ paket\ yang\ dikirim}$$

3. Hasil

3.1 Pengujian Pengaruh Ukuran Buffer

Skenario kali ini digunakan untuk mengetahui pengaruh *buffer* dan penambahan node pada nilai *latency*, *packetloss*, dan *packet delivery ratio*.

3.1.1 Latency



Gambar 3.1. Pengaruh Perubahan Besar Buffer Terhadap Nilai Latency

Berdasarkan data hasil simulasi pada gambar 3.1, terlihat bahwa *router* Maxprop memiliki nilai *latency* yang cenderung turun ketika ukuran *buffer* semakin besar. *Latency* dengan nilai terkecil didapat pada ukuran *buffer* 15 M. Pada jumlah *node* 5 nilai *latency* cenderung naik dan pada jumlah *node* 10 nilai *latency* naik saat ukuran *buffer* 10 MB dan turun saat nilai *buffer* 15 MB.

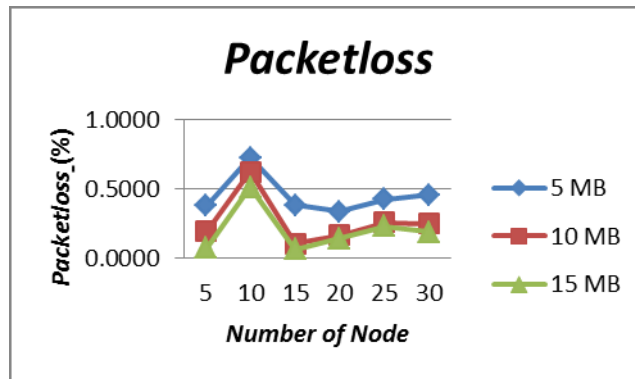
Pada tahap pengiriman, *threshold* bekerja untuk menentukan paket yang akan dikirim terlebih dahulu. Saat kedua *node* bertemu, paket dengan nilai *hopcount* < *threshold* akan dikirimkan terlebih dahulu dengan melihat nilai *hopcount* terkecil. Sedangkan paket dengan nilai *hopcount* ≥ *threshold* akan disimpan. Dengan ukuran *buffer* yang besar maka paket dengan nilai *hopcount* ≥ *threshold* kemungkinan paket dibuang lebih kecil. Ketika paket dengan nilai *hopcount* < *threshold* sudah dikirimkan semua, maka paket dengan *hopcount* ≥ *threshold* menjadi prioritas pengiriman dengan melihat nilai *hopcount* terkecil. Sehingga kemungkinan paket terkirim dengan tujuan yang jauh bertambah. Dengan ukuran *buffer* yang besar, paket dengan tujuan yang jauh akan tetap terkirim karena kemungkinan paket dibuang lebih kecil. Nilai *latency*-nya pun menjadi lebih kecil karena tidak perlu mengirim ulang paket ketika paket dibuang dari *buffer*.

Dilihat pada gambar 3.1, nilai *latency* pada jumlah *node* 5 cenderung naik karena probabilitas bertemunya dengan *node* kecil. Ketika *node* mempunyai kesempatan untuk mengirim paket, paket akan dikirimkan sesuai dengan prioritasnya. Dengan jaranganya *node* bertemu dengan *node* lain, saat ada kesempatan transfer maka paket akan dikirimkan karena sedikitnya paket dalam daftar antrian dan nilai *hopcount* setiap paket kecil. Nilai *latency* mengalami kenaikan karena dengan semakin besar ukuran *buffer* maka paket yang ditampung akan semakin banyak. Sehingga daftar antrian pada *buffer* menjadi semakin banyak. Sedangkan dengan jumlah *node* yang sedikit pada jaringan, terjadinya kesempatan pengiriman sedikit dan paket yang dikirimkan dilihat dari prioritas nilai *hopcount* terkecil.

Semakin banyak *node* yang digunakan dalam simulasi menghasilkan nilai *latency* yang naik turun. Ini diakibatkan pergerakan *node* yang acak sehingga nilai *latency* tidak bisa diprediksi berdasarkan perbedaan jumlah *node*. Pada Maxprop pengiriman paket dipengaruhi oleh *hopcount*, tetapi dipengaruhi juga oleh pergerakan *node*. Walaupun jumlah *hopcount*-nya sudah bervariasi dengan banyaknya *node*, nilai *latency* tetap dipengaruhi oleh pergerakan dari setiap *node* dalam jaringan. Nilai *latency* kecil apabila *node* sumber dengan *node* tujuan sering bertemu secara langsung. Akan tetapi nilai *latency* tetap cenderung turun ketika nilai *buffer* dinaikkan.

3.1.2 Packetloss

Packetloss merupakan suatu parameter yang menggambarkan suatu kondisi dimana jumlah total paket yang hilang terhadap paket yang dikirim. Dapat terjadi karena paket yang di *drop*, aborsi, atau dihapus pada saat transmisi sehingga tidak sampai ke tujuan. Berdasarkan data hasil simulasi pada gambar 3.2, terlihat bahwa *router* Maxprop memiliki nilai *packetloss* yang cenderung turun ketika ukuran *buffer* semakin besar. *Packetloss* dengan nilai terkecil didapat pada ukuran *buffer* 15 M.



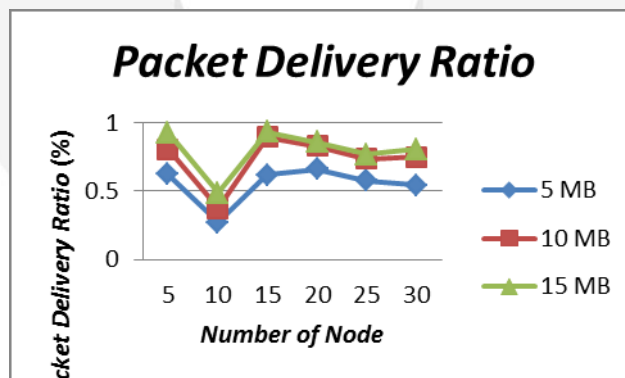
Gambar 3.2. Pengaruh Perubahan Besar Buffer Terhadap Nilai Packetloss

Umumnya semakin besar ukuran *buffer* maka nilai *packetloss*-nya semakin kecil. Karena dengan besarnya ukuran *buffer* maka paket yang dapat ditampung akan semakin banyak sehingga kemungkinan paket dibuang kecil. Pada tahap pengiriman, *threshold* bekerja untuk menentukan paket yang akan dikirim terlebih dahulu. Saat kedua *node* bertemu, paket dengan nilai *hopcount* < *threshold* akan dikirimkan terlebih dahulu dengan melihat nilai *hopcount* terkecil. Sedangkan paket dengan nilai *hopcount* ≥ *threshold* akan disimpan. Dengan ukuran *buffer* yang besar maka paket dengan nilai *hopcount* ≥ *threshold* kemungkinan paket dibuang lebih kecil. Ketika paket dengan nilai *hopcount* < *threshold* sudah dikirimkan semua, maka paket dengan *hopcount* ≥ *threshold* menjadi prioritas pengiriman dengan melihat nilai *hopcount* terkecil. Sehingga dengan ukuran *buffer* yang besar nilai *packetloss*-nya pun menjadi lebih kecil. Karena paket dengan tujuan yang jauh akan tetap terkirim karena kemungkinan paket dibuang lebih kecil.

Semakin banyak *node* yang digunakan dalam simulasi menghasilkan nilai *packetloss* yang naik turun. Ini diakibatkan pergerakan *node* yang acak sehingga nilai *packetloss* tidak bisa diprediksi berdasarkan perbedaan jumlah *node*. Pada Maxprop pengiriman paket dipengaruhi oleh *hopcount*, tetapi dipengaruhi juga oleh pergerakan *node*. Walaupun jumlah *hopcount*-nya sudah bervariasi dengan banyaknya *node*, nilai *packetloss* tetap dipengaruhi oleh pergerakan dari setiap *node* dalam jaringan. Nilai *packetloss* kecil apabila *node* sumber dengan *node* tujuan sering bertemu secara langsung. Akan tetapi nilai *packetloss* tetap cenderung turun ketika nilai *buffer* dinaikkan.

3.1.3 Packet Delivery Ratio

Packet delivery ratio merupakan suatu parameter yang menggambarkan tingkat keberhasilan paket yang dikirimkan. Berdasarkan data hasil simulasi pada gambar 3.3, terlihat bahwa router Maxprop memiliki nilai *packet delivery ratio* yang cenderung naik ketika ukuran *buffer* semakin besar. *Packet delivery ratio* dengan nilai terbesar didapat pada ukuran *buffer* 15 M.



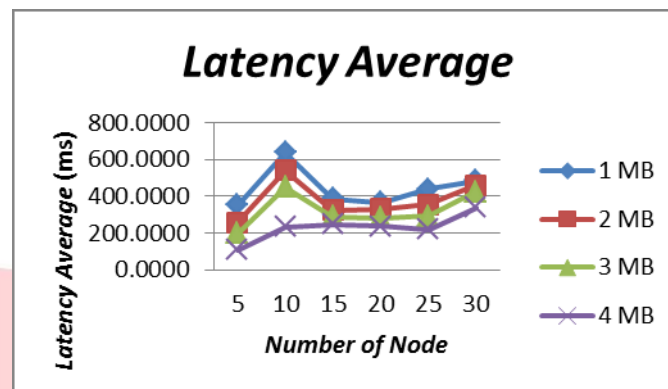
Gambar 3.3. Pengaruh Perubahan Besar Buffer Terhadap Nilai Packet Delivery Ratio

Umumnya semakin besar ukuran *buffer* maka nilai *packet delivery ratio*-nya semakin kecil. Karena dengan besarnya ukuran *buffer* maka paket yang dapat ditampung akan semakin banyak sehingga kemungkinan paket dibuang kecil. Pada tahap pengiriman, *threshold* bekerja untuk menentukan paket yang akan dikirim terlebih dahulu. Saat kedua *node* bertemu, paket dengan nilai *hopcount* < *threshold* akan dikirimkan terlebih dahulu dengan melihat nilai *hopcount* terkecil. Sedangkan paket dengan nilai *hopcount* ≥ *threshold* akan disimpan. Dengan ukuran *buffer* yang besar maka paket dengan nilai *hopcount* ≥ *threshold* kemungkinan paket dibuang lebih kecil. Ketika paket dengan nilai *hopcount* < *threshold* sudah dikirimkan semua, maka paket dengan *hopcount* ≥ *threshold* menjadi prioritas pengiriman dengan melihat nilai *hopcount* terkecil. Sehingga dengan ukuran *buffer* yang besar nilai *packet delivery ratio*-nya pun menjadi lebih besar. Karena paket dengan tujuan yang jauh akan tetap terkirim karena kemungkinan paket dibuang lebih kecil.

3.2 Pengujian Pengaruh Ukuran Paket

Skenario kali ini digunakan untuk mengetahui pengaruh besar paket dan penambahan node pada nilai *latency*, *packetloss*, dan *packet delivery ratio*.

3.2.1 Latency

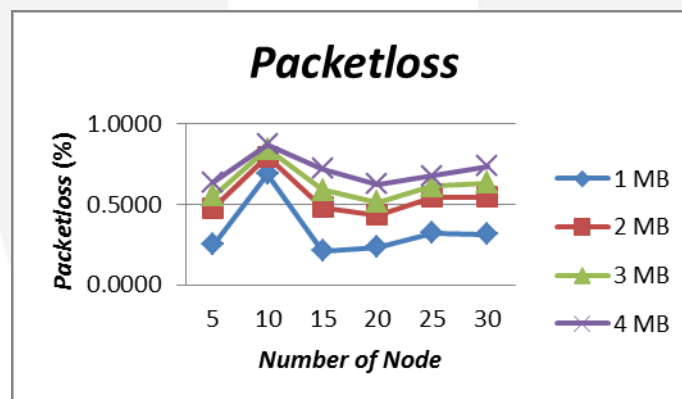


Gambar 3.4. Pengaruh Perubahan Besar Paket Terhadap Nilai *Latency*

Dilihat dari gambar 3.4, semakin besar paket yang dikirimkan nilai *latency* cenderung semakin kecil. Karena dengan ukuran paket yang kecil maka pada *buffer* akan terjadi penumpukan antrian. Sehingga paket akan dikirimkan bergantung pada daftar antriannya berdasarkan $hopcount < threshold$. Sedangkan semakin besar ukuran paket maka *buffer* hanya akan menampung sedikit jumlah paket. Saat bertemu dengan *node* lain kemungkinan paket dikirimkan terlebih dahulu lebih besar karena hanya ada sedikit paket dalam daftar antrian.

Semakin banyak *node* yang digunakan dalam simulasi menghasilkan nilai *latency* yang naik turun. Ini diakibatkan pergerakan *node* yang acak sehingga nilai *latency* tidak bisa diprediksi berdasarkan perbedaan jumlah *node*. Pada Maxprop pengiriman paket dipengaruhi oleh *hopcount*, tetapi dipengaruhi juga oleh pergerakan *node*. Walaupun jumlah *hopcount*-nya sudah bervariasi dengan banyaknya *node*, nilai *latency* tetap dipengaruhi oleh pergerakan dari setiap *node* dalam jaringan. Nilai *latency* kecil apabila *node* sumber dengan *node* tujuan sering bertemu secara langsung. Akan tetapi nilai *latency* tetap cenderung naik ketika ukuran paket data yang dikirimkan semakin besar.

3.2.2 Packetloss



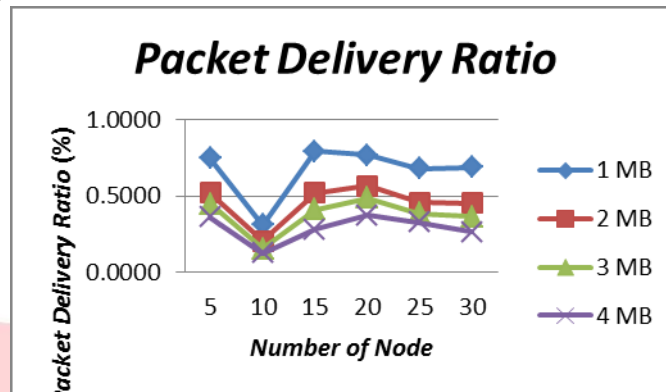
Gambar 3.5. Pengaruh Perubahan Besar Paket Terhadap Nilai *Packetloss*

Pada gambar 3.5 didapat bahwa nilai *packetloss* cenderung naik ketika ukuran paket semakin besar. Dalam *buffer*, diprioritaskan paket dengan $hopcount < threshold$ yang akan dikirim terlebih dahulu. Dengan ukuran paket yang besar, maka daftar antrian pada *buffer* menjadi sedikit. Paket akan mempunyai prioritas dikirim lebih besar. Akan tetapi saat *node* mempunyai kesempatan mengirim, *buffer* penerima hanya bisa menampung sedikit dari jumlah paket. Dengan sedikitnya paket yang dapat ditampung oleh *buffer*, maka *buffer* akan sering penuh. Oleh karena itu banyak paket yang tidak diterima (*aborted*) oleh penerima karena sudah tidak ada kapasitas yang dapat menampung paket tersebut. Dengan begitu paket harus dikirim ulang melalui *node* lain yang masih mempunyai kapasitas lebih. Dengan begitu semakin besar ukuran paket maka nilai *packetloss* akan semakin besar.

Semakin banyak *node* yang digunakan dalam simulasi menghasilkan nilai *packetloss* yang naik turun. Ini diakibatkan pergerakan *node* yang acak sehingga nilai *packetloss* tidak bisa diprediksi berdasarkan perbedaan jumlah *node*. Pada Maxprop pengiriman paket dipengaruhi oleh *hopcount*, tetapi dipengaruhi juga oleh pergerakan *node*. Walaupun jumlah *hopcount*-nya sudah bervariasi dengan banyaknya *node*, nilai *packetloss* tetap dipengaruhi oleh pergerakan dari setiap *node* dalam jaringan. Nilai *packetloss* kecil apabila *node* sumber

dengan *node* tujuan sering bertemu secara langsung. Akan tetapi nilai *packetloss* tetap cenderung naik ketika ukuran paket data yang dikirimkan semakin besar.

3.2.3 Packet Delivery Ratio



Gambar 3.6. Pengaruh Perubahan Besar Paket Terhadap Nilai *Packet Delivery Ratio*

Pada gambar 3.6 didapat bahwa nilai *packet delivery ratio* cenderung turun ketika ukuran paket semakin besar. Dengan ukuran paket yang besar, maka daftar antrian pada *buffer* menjadi sedikit. Paket akan mempunyai prioritas dikirim lebih besar. Akan tetapi saat *node* mempunyai kesempatan mengirim, *buffer* penerima hanya bisa menampung sedikit dari jumlah paket. Dengan sedikitnya paket yang dapat ditampung oleh *buffer*, maka *buffer* akan sering penuh. Oleh karena itu banyak paket yang tidak diterima (*aborted*) oleh penerima karena sudah tidak ada kapasitas yang dapat menampung paket tersebut. Sehingga paket harus dikirim ulang melalui *node* lain yang masih mempunyai kapasitas lebih. Sedikit dari jumlah paket yang dapat sampai ke tujuan, dengan begitu semakin besar ukuran paket maka nilai *packet delivery ratio* akan semakin kecil.

Semakin banyak *node* yang digunakan dalam simulasi menghasilkan nilai *packet delivery ratio* yang naik turun. Ini diakibatkan pergerakan *node* yang acak sehingga nilai *packet delivery ratio* tidak bisa diprediksi berdasarkan perbedaan jumlah *node*. Pada Maxprop pengiriman paket dipengaruhi oleh *hopcount*, tetapi dipengaruhi juga oleh pergerakan *node*. Walaupun jumlah *hopcount*-nya sudah bervariasi dengan banyaknya *node*, nilai *packet delivery ratio* tetap dipengaruhi oleh pergerakan dari setiap *node* dalam jaringan. Nilai *packet delivery ratio* besar apabila *node* sumber dengan *node* tujuan sering bertemu secara langsung. Akan tetapi nilai *packet delivery ratio* tetap cenderung turun ketika ukuran paket data yang dikirimkan semakin besar.

4. Kesimpulan

Setelah dilakukan analisa terhadap hasil simulasi, maka kesimpulan yang dapat diambil adalah sebagai berikut.

1. Router Maxprop memiliki nilai *latency* dan *packetloss* semakin kecil ketika ukuran *buffer* diperbesar. Karena semakin besar *buffer* maka dapat menampung banyak jumlah paket yang memungkinkan sedikit paket yang dibuang.
2. Semakin besar ukuran *buffer* maka nilai *packet delivery ratio* semakin besar. Kemungkinan paket dibuang kecil karena paket masih disimpan pada *buffer*, sehingga meningkatkan probabilitas paket sampai ke tujuan.
3. Semakin besar ukuran paket, nilai *latency* dan *packetloss* semakin besar. Karena semakin besar paket maka paket yang dapat ditampung oleh *buffer* semakin sedikit. Sehingga probabilitas diaborsinya paket lebih besar karena *buffer node* tetangga yang sudah penuh.
4. Semakin besar ukuran paket, nilai *packet delivery ratio* semakin kecil. Karena semakin besar paket maka paket yang dapat ditampung oleh *buffer* semakin sedikit. Probabilitas diaborsinya paket lebih besar karena *buffer node* tetangga yang sudah penuh. Sehingga menurunkan probabilitas paket sampai ke tujuan.
5. Semakin banyak *node* menghasilkan nilai *latency*, *packetloss*, dan *packet delivery ratio* yang naik turun. Ini diakibatkan pergerakan *node* yang acak sehingga nilai *latency*, *packetloss*, dan *packet delivery ratio* tidak bisa diprediksi berdasarkan perbedaan jumlah *node*.

Daftar pustaka

- [1.] Warthman, Forest et al (2003), Delay-Tolerant Networks (DTNs) A Tutorial, DTN Research Group Internet Draft.
- [2.] C. E. Palazzi, A. Bujari, S. Bonetta, G. Marfia, M. Rocchetti, A. Amoroso. 2011.MDTN: Mobile Delay/Disruption Tolerant Network. 2011. IEEE
- [3.] Ari Keranen, Jorg Ott, Teemu Karkkainen. 2009. The ONE Simulator for DTN Protocol Evaluation.
- [4.] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, Max-prop: Routing for vehicle-based disruption tolerant networks, in Proc. 25th IEEE Int. Conf. on Computer Communications, (April 2006), pp. 1–11.