

## Implementasi *Congestion Avoidance* pada HWMP dengan menggunakan CCNF dan *Queue Threshold* pada Jaringan VANET

Audi Azzura Falaq<sup>1</sup>, Primantara Hari Trisnawan<sup>2</sup>, Reza Andria Siregar<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>audiazzura23@student.ub.ac.id, <sup>2</sup>prima@ub.ac.id, <sup>3</sup>reza.jalin@ub.ac.id

### Abstrak

*Vehicular Ad hoc Network* (VANET) sangat beresiko mengalami kondisi *congestion*. Kondisi *congestion* tersebut menyebabkan paket tertahan, dan gagal tiba di tujuan. Dalam penelitian ini akan dilakukan modifikasi terhadap protokol HWMP untuk mengatasi permasalahan *congestion* tersebut. Modifikasi yang dilakukan adalah menerapkan sebuah *threshold* untuk memonitor kondisi *queue* pada *node*. Apabila terjadi kondisi *congestion*, maka *node* akan mengirimkan *Congestion Control Notification Frame* (CCNF) kepada *node-node* lainnya. CCNF tersebut berfungsi untuk menotifikasi *node-node* lainnya bahwa *node* tersebut tidak dapat menerima paket untuk sementara waktu. Sehingga akan dibentuk rute alternatif yang menghindari *node* tersebut. Pengujian terhadap modifikasi tersebut dilakukan pada 3 *mobility model* yang berbeda yaitu, model Manhattan, model Freeway, dan *real map*. Dari pengujian tersebut akan diukur 4 parameter yaitu, *average end-to-end delay*, *packet delivery ratio* (PDR), *throughput*, dan *normalized routing load*. Setelah pengujian dilakukan, diketahui bahwa protokol hasil modifikasi lebih unggul dalam PDR dan *throughput*, namun terdapat penurunan performa pada *average end-to-end delay* dan *normalized routing load*.

**Kata kunci:** protokol routing, HWMP, *congestion avoidance*, VANET, NS-2

### Abstract

*Vehicular Ad hoc Network* (VANET) is prone to experience congestion problems. The congestion may cause the packet to be stuck, and fails to arrive at its destination. In this study, modifications to the HWMP protocol will be carried out to overcome the congestion problem. The modification made is to apply a threshold to monitor the queue at the node. If a node is starting to get congested, the node will send a Congestion Control Notification Frame (CCNF) to other nodes. The CCNF function is to notify other nodes that the node cannot receive packets for a while. So that an alternative route that excludes the congested node will be formed. Tests will be carried out on 3 different mobility models, namely, the Manhattan model, the Freeway model, and the real map. From those test, 4 parameters such as average end-to-end delay, packet delivery ratio (PDR), throughput, and normalized routing load will be measured. Based on the results of the tests, it is known that the modified protocol is superior in PDR and throughput, but there is a decrease in performance in average end-to-end delay and normalized routing load.

**Keywords:** routing protocol, HWMP, *congestion avoidance*, VANET, NS-2

## 1. PENDAHULUAN

Vehicular Networks, atau yang juga dikenal dengan sebutan *Vehicular Ad hoc Network* (VANET) merupakan suatu cabang teknologi di bidang *Wireless Mesh Network* (WMN). Pada dasarnya VANET menerapkan prinsip *Mobile Ad hoc Network* (MANET) pada komunikasi antar kendaraan. Dimana sekumpulan perangkat nirkabel yang bergerak (dalam kasus ini adalah

kendaraan) dapat terkoneksi tanpa adanya infrastruktur yang tetap (Zanjireh & Larijani, 2015). VANET sendiri diterapkan dengan tujuan untuk meningkatkan kenyamanan berkendara dengan mengatur lalu lintas dan juga mengurangi jumlah kecelakaan dengan mengirimkan informasi-informasi penting mengenai kendaraan lain maupun kondisi jalan yang dilalui oleh kendaraan tersebut.

Salah satu tantangan dalam penerapan

VANET sendiri adalah tingginya mobilitas setiap *node* dan pergerakan *node* yang bersifat acak. Permasalahan tersebut tentunya dapat diatasi dengan melakukan *routing*, dengan tujuan agar pesan yang ditujukan dapat sampai pada tujuannya secara cepat dan efisien (Yu, et al., 2011). Namun tentunya protokol *routing* yang digunakan harus mampu menghadapi kondisi tersebut.

Protokol *routing* dalam VANET sendiri dibagi menjadi 6 macam yaitu, *topology based routing*, *geo-cast based routing*, *position based routing*, *cluster based routing*, *infrastructure based routing*, dan *broadcast based routing*. *Topology based routing* sendiri terbagi lagi menjadi 2 yaitu *proactive*, dan *reactive* (Andrey & Hamrioui, 2015). *Hybrid Wireless Mesh Protocol* (HWMP) sendiri merupakan protokol *routing topology based*. Dalam penelitian ini, HWMP dipilih karena HWMP merupakan protokol yang digunakan pada standar IEEE 802.11s dan juga merupakan versi peningkatan dari *Ad-hoc On-demand Distance Vector Routing* (AODV), dan AODV sendiri merupakan salah satu protokol yang paling sering digunakan dalam penerapan VANET (Ghori, et al., 2018). HWMP dalam penerapannya mampu memberikan performa yang lebih baik dibandingkan dengan AODV dalam kondisi yang sama dalam seluruh skenario pengujian yang dilakukan (Nayyar, 2018).

Dalam penerapannya, VANET beresiko tinggi mengalami *congestion*. Terutama dalam jaringan dengan kondisi kepadatan *node* yang rendah (Khaliq, et al., 2014). *Congestion* tersebut menyebabkan gagalnya paket yang dikirim untuk sampai pada tujuannya. Tentunya hal ini sangat berbahaya mengingat salah satu tujuan utama dari VANET sendiri adalah untuk meningkatkan keamanan dalam berkendara. Selain itu, HWMP masih belum memiliki mekanisme *congestion control* maupun *congestion avoidance* yang tetap. Pada 802.11s disebutkan bahwa *congestion* dapat diatasi dengan menggunakan sebuah paket khusus yang disebut *Congestion Control Notification Frame* (CCNF). CCNF pada dasarnya merupakan sebuah paket yang dikirimkan untuk menotifikasi jaringan bahwa terjadi *congestion* (Hiertz, et al., 2010).

Penelitian ini sendiri dimaksudkan untuk menganalisis kinerja protokol HWMP yang telah dimodifikasi dengan menggunakan mekanisme *congestion avoidance* pada jaringan VANET. Selain itu, hasil penelitian ini diharapkan dapat

membantu dalam meningkatkan kinerja protokol *routing* HWMP dalam penerapan pada jaringan VANET.

## 2. LANDASAN KEPUSTAKAAN

Khaliq, et al. dalam penelitiannya pada tahun 2014 yang berjudul "*congestion avoidance hybrid wireless mesh protocol (CA-HWMP) for IEEE 802.11s*" juga mengemukakan suatu metode untuk mengimplementasikan mekanisme *congestion avoidance* dalam lingkungan WMN, namun diimplementasikan pada protokol HWMP. Dimana metode tersebut mengandalkan *Congestion Control Signalling* yang digunakan pada standar IEEE 802.11s. Pada penelitian tersebut dilakukan pengukuran terhadap beberapa parameter yaitu, *throughput*, *end-to-end delay*, dan *packet delivery fraction*. Khaliq, et al., menyimpulkan bahwa terjadi peningkatan kinerja pada protokol HWMP yang telah dimodifikasi dengan mekanisme *congestion avoidance* dibanding protokol HWMP biasa. Namun dibalik peningkatan tersebut, masih terdapat kekurangan dalam penerapan CA-HWMP, yaitu tidak adanya pesan yang terkirim melalui jalur yang mengalami *congestion* tersebut selama proses pencarian rute alternatif.

### 2.1. Mobile Ad-hoc Network (MANET)

*Mobile Ad-hoc Network* atau yang biasa disebut MANET merupakan sekumpulan perangkat bergerak yang saling terhubung dan bertugas sebagai *router* dan *host* dalam sebuah jaringan nirkabel yang bersifat ad-hoc. Karena setiap perangkat dalam jaringan MANET bersifat dinamis dan selalu bergerak, maka jaringan MANET bersifat *self-organizing* dan *self-healing*. Semua itu dilakukan tanpa adanya infrastruktur yang telah ditetapkan sebelumnya (Buford, et al., 2009).

### 2.2. Vehicular Ad-hoc Network (VANET)

*Vehicular Ad-hoc Network* (VANET) merupakan sekumpulan perangkat bergerak yang saling terhubung. Hal yang membedakan VANET dengan MANET adalah tingkat mobilitas *node*-nya yang jauh lebih tinggi (Badis & Rachedi, 2015). Hal ini disebabkan karena jaringan VANET terbentuk dari sekumpulan *node* yang sebagian besar terdiri dari kendaraan bermotor yang dilengkapi perangkat komunikasi nirkabel.

### 2.3. Hybrid Wireless Mesh Protocol (HWMP)

HWMP adalah sebuah routing protokol yang didasarkan pada standar 802.11s dan merupakan protokol default standar tersebut. HWMP juga merupakan salah satu protokol routing yang biasa digunakan dalam jaringan nirkabel mesh, seperti MANET dan VANET. HWMP pada dasarnya merupakan pengembangan lebih lanjut dari protokol AODV. Dimana HWMP dapat melakukan proses *routing* secara proaktif dan reaktif. Hal tersebut menjadikan HWMP sebuah protokol *routing* hybrid.

Pada HWMP, proses pencarian secara proaktif dapat dilakukan apabila terdapat *root node* dalam jaringan. *Root node* adalah *node* yang merupakan *endpoint* mayoritas dari komunikasi dalam jaringan. Pada proses pencarian rute secara proaktif, *root node* akan mem-broadcast *Route Request* (PREQ) dalam interval tertentu. Setiap *node* yang menerima PREQ tersebut akan memperbarui *routing table* nya. Dengan begitu, rute 1 arah dari *node* penerima menuju *root node* akan terbentuk. Namun apabila komunikasi 2 arah dibutuhkan, maka *node* tersebut akan mengirimkan *Route Reply* (PREP) kembali ke *root node*.

Proses pencarian rute secara reaktif pada HWMP dilakukan dengan cara yang berbeda. Dimana PREQ akan di-broadcast oleh sebuah *node* ketika *node* tersebut ingin berkomunikasi dengan *node* lain, namun tidak memiliki rute menuju *node* tersebut. Pada PREQ reaktif, *node* tujuan akan dimasukkan pada daftar tujuan PREQ tersebut. Sehingga apabila *node* yang terdapat pada daftar menerima PREQ tersebut, maka *node* tersebut akan mengirimkan PREP kembali ke *node* pengirim. Sehingga rute komunikasi 2 arah antar kedua *node* tersebut akan terbentuk.

### 2.4. Congestion Avoidance

*Congestion Avoidance* merupakan salah satu mekanisme yang diterapkan dalam suatu jaringan untuk mencegah agar jaringan tersebut tidak mencapai kondisi macet (*congested*). Mekanisme *congestion avoidance* sendiri bertujuan untuk menjaga agar jaringan tetap beroperasi dengan *delay* yang relatif rendah dan *throughput* yang relatif tinggi (Jain & Ramakrishnan, 1988). *Congestion avoidance* berbeda dengan *congestion control*, dimana *congestion avoidance* berfungsi untuk mencegah kondisi *congestion* terjadi, sedangkan

*congestion control* berfungsi untuk menangani situasi apabila *congestion* telah terjadi.

Pada standar 802.11s sendiri disebutkan bahwa mekanisme *congestion control* dan *congestion avoidance* dapat dilakukan dengan memanfaatkan sebuah *management frame* yang disebut *Congestion Control Notification Frame* (CCNF). Pada dasarnya CCNF merupakan sebuah paket yang dikirimkan oleh sebuah *node* yang menandakan bahwa *node* tersebut mengalami *congestion*. Namun pada 802.11s, tidak didefinisikan mekanisme *congestion control* atau *congestion avoidance* apa yang digunakan. Dan sampai saat ini HWMP masih belum menggunakan CCNF untuk manajemen *congestion* (Khaliq, et al., 2014).

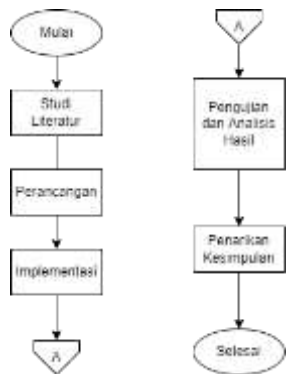
### 2.5. Simulation of Urban Mobility (SUMO)

Simulation of Urban Mobility (SUMO) merupakan simulator lalu lintas yang bersifat *open source*, dan mikroskopik. SUMO pada dasarnya dirancang untuk menyimulasikan jaringan lalu lintas berskala besar. Maksud dari simulator yang bersifat mikroskopik adalah setiap kendaraan pada jalan raya dimodelkan secara eksplisit, mempunyai rutenya masing-masing, dan bergerak secara individu dalam lalu lintas (Harri, et al., 2009). Hal ini menjadikan simulasi mikroskopik jauh lebih akurat dibanding simulasi makroskopik walaupun diikuti dengan kebutuhan daya komputasi yang lebih tinggi.

### 2.6. OpenStreetMap (OSM)

OpenStreetMap (OSM) merupakan sebuah proyek *open source* berbasis *web* yang menyediakan peta yang bersifat bebas dan terbuka dari peta seluruh dunia. Data peta yang disediakan OSM sendiri merupakan *geodata* diperoleh melalui survei manual, perangkat *Global Positioning System* (GPS), fotografi aerial, serta sumber-sumber publik lainnya (OpenStreetMap Foundation, 2021). *Geodata* tersebut kemudian dapat digunakan untuk kebutuhan-kebutuhan lainnya, seperti peta pada perangkat GPS, maupun penelitian berupa simulasi dengan bantuan simulator seperti SUMO.

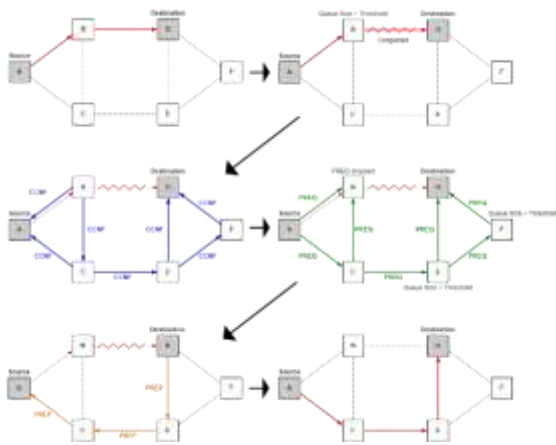
## 3. METODOLOGI



Gambar 1. Metodologi Penelitian

Seperti yang dapat dilihat pada Gambar 1, proses penelitian dimulai dari tahapan studi literatur yang berfungsi untuk mencari referensi sekaligus mempelajari teori-teori yang berkaitan dengan penelitian ini. Tahapan selanjutnya adalah melakukan perancangan untuk merumuskan rancangan dari implementasi yang akan dilakukan. Selanjutnya, akan dilakukan implementasi berdasarkan rancangan yang telah dibuat sebelumnya. Hasil dari implementasi kemudian akan diujikan dan dilakukan analisis terhadap hasil pengujian yang didapat. Penelitian diakhiri dengan menarik kesimpulan dan memberikan saran terhadap penelitian yang telah dilakukan.

### 3.1. Perancangan



Gambar 2. Rancangan Mekanisme Congestion Avoidance

Mekanisme *congestion avoidance* yang akan diimplementasikan pada penelitian ini bekerja dengan menggunakan CCNF dan *queue threshold*. Cara kerja mekanisme *congestion avoidance* tersebut dapat dilihat pada Gambar 2. Ketika terjadi penumpukan paket pada sebuah *node* yang ditandai dengan *queue* lebih besar dari *threshold*, maka *node* tersebut akan mem-

*broadcast* CCNF kepada seluruh *node* dalam jaringan. Ketika *node* sumber menerima CCNF tersebut, maka akan dilakukan pencarian rute ulang dengan mem-*broadcast* PREQ baru menuju *node* tujuan. Namun untuk kali ini, *node* yang mengalami *congestion* tidak akan merespon PREQ tersebut. Dengan begitu, rute baru yang menghindari *node* yang mengalami *congestion* akan terbentuk.

### 3.2. Implementasi

Setelah didapatkan hasil perancangan, maka dapat dilanjutkan dengan mengimplementasikan rancangan tersebut pada protokol HWMP. Implementasi dilakukan langsung pada file kode sumber HWMP pada NS-2. Proses implementasi yang dilakukan dapat dibagi menjadi 2 bagian, yaitu implementasi mekanisme *congestion detection* dan *notification*, dan implementasi mekanisme *update route*. Algoritme dari kedua mekanisme tersebut dapat dilihat pada Algoritme 1 dan 2 dibawah ini secara berurutan.

```

Algoritme 1: Mekanisme Congestion
Detection dan Notification
1  If node receive Packet
2  If (currentQueue >= threshold)
3
4  If (congestionFlag == 0)
5  Set congestionFlag = 1
6  Broadcast CCNF towards Packet
   source
7
8  If (congestionFlag == 1 &&
   Packet == PREQ)
9
10 Drop Packet
    
```

```

Algoritme 2: Mekanisme Update Route
1  If node receive CCNF
2  If (CCNF seqno >= current seqno)
3
4  Update routing table
5  If (CCNF destination == this
   node)
6
7  Drop packet
8  Else
9  Forward CCNF
10 Else
11 Drop packet
    
```

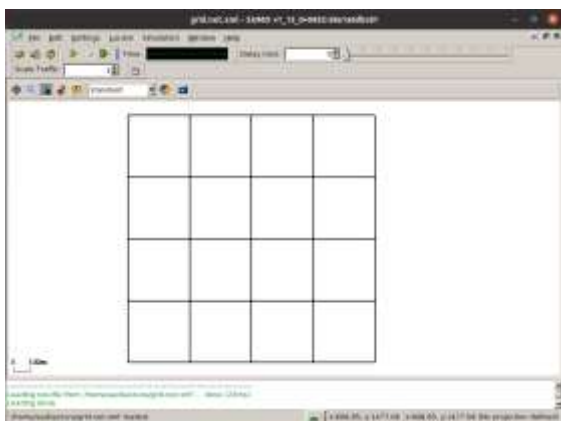
### 3.3. Pengujian

Pengujian dilakukan pada 2 skenario berbeda, yaitu skenario *model-based* dan skenario *real map*. Pada skenario *model-based* digunakan 2 buah *mobility model* yaitu model



Manhattan (*grid*) dan model Freeway (*highway*). Sedangkan pada skenario *real map*, peta jalan yang digunakan merupakan hasil konversi dari peta sungguhan dengan menggunakan bantuan OSM dan SUMO.

Pada skenario Manhattan (*grid*), terdapat 5 persimpangan untuk setiap jalan. Dimana jarak antar persimpangan adalah 350 m dan setiap jalan mempunyai 2 lajur yang berlawanan arah. Selain itu, kecepatan kendaraan dibatasi pada 13,89 m/detik atau sekitar 50 km/jam. Peta jalan yang digunakan untuk skenario Manhattan (*grid*) dapat dilihat pada Gambar 3.



Gambar 3. Peta Skenario Manhattan (*Grid*)

Pada skenario Freeway (*highway*), hanya terdapat 2 buah jalan yang saling berlawanan arah. Dimana masing-masing jalan mempunyai panjang 1400 m dan masing-masing jalan mempunyai 3 lajur. Pada skenario ini, kecepatan kendaraan dibatasi pada 27,78 m/detik atau sekitar 100 km/jam. Peta jalan yang digunakan untuk skenario Freeway (*highway*) dapat dilihat pada Gambar 4.



Gambar 4. Peta Skenario Freeway (*Highway*)

Pada skenario *real map*, peta yang digunakan merupakan peta di sekitar Kecamatan Blimbing, Kota Malang, Jawa Timur. Peta area simulasi dapat dilihat pada Gambar 5. Total area peta yang akan digunakan untuk simulasi adalah

sekitar 1500m x 1500m.



Gambar 5. Area Simulasi Skenario *Real Map*

Setelah peta area simulasi didapatkan dengan menggunakan bantuan OSM, maka kemudian akan dilakukan konversi dengan bantuan modul-modul yang ada pada SUMO. Hasil konversi tersebut dapat dilihat pada Gambar 6.



Gambar 6. Peta Hasil Konversi Skenario *Real Map*

Skenario tersebut kemudian disimulasikan dengan menggunakan simulator dengan parameter seperti yang dapat dilihat pada Tabel 1. *Network Simulator* yang digunakan adalah NS-2.35 yang telah di *patch* agar dapat menunjang simulasi jaringan dengan protokol *routing* HWMP. *Traffic data* yang digunakan adalah *Constant Bit-Rate* (CBR) dengan ukuran paket 512 *byte* (0.5 *kilobyte*) sebanyak 20 kali per detik, dan dikirim menggunakan UDP sebagai *transport layer protocol*-nya. Ukuran panjang *queue* maksimal yang diterapkan pada tiap *node* adalah sebanyak 64 paket. Dalam proses pengujian, topologi jaringan yang digunakan adalah model Manhattan, Freeway, dan peta asli hasil konversi. Proses simulasi tersebut berjalan selama 5 menit (300 detik).

Tabel 1. Parameter Simulasi

Parameter	Spesifikasi
<i>Network Simulator</i>	NS-2.35
<i>Network Standard</i>	IEEE 802.11

Protokol <i>Routing</i>	HWMP, modifikasi	HWMP
Protokol <i>Transport Layer</i>	UDP	
Tipe Kanal	<i>Wireless Channel</i>	
Propagasi Sinyal	<i>Two-ray Ground</i>	
Range	350m	
Aliran <i>Traffic Data</i>	<i>Constant Bit Rate (CBR)</i>	
Ukuran Paket	512 Bytes	
<i>Packet Rate</i>	20 <i>packets/s</i>	
<i>Max Queue Length</i>	64 <i>packet</i>	
<i>Mobility Model</i>	Manhattan (Grid), Freeway (Highway), Real Map	
Area Simulasi	1500m x 1500m	
Banyak <i>Node</i>	100, 200, 300	
Waktu Simulasi	300s	
<i>Threshold</i> (pada kode sumber)	25%, 50%, 75%	

Hasil simulasi akan dianalisis berdasarkan 4 parameter *Quality of Service (QoS)* yang meliputi:

*Average E2E Delay* merupakan waktu yang dibutuhkan oleh suatu paket yang dikirimkan oleh *node* sumber untuk mencapai *node* tujuan. *Average E2E delay* dapat dihitung dengan menggunakan Persamaan 1, Persamaan 2, dan Persamaan 3 secara berurutan.

$$E2Edelay = ReceiveTime - SentTime \quad (1)$$

$$Total_{E2E} = \sum_{i=1}^{recvNum} E2Edelay \quad (2)$$

$$Average_{E2E} = \frac{Total_{E2E}}{recvNum} \quad (3)$$

*Packet Delivery Ratio (PDR)* merupakan rasio perbandingan jumlah paket yang diterima oleh *node* tujuan dengan jumlah paket yang dikirimkan oleh *node* pengirim. PDR dapat dihitung dengan menggunakan Persamaan 4.

$$PDR = \frac{packetReceived}{packetSent} \quad (4)$$

*Throughput* adalah total besarnya paket yang berhasil terkirim oleh pengirim menuju penerima dalam satuan waktu *Throughput* dapat dihitung dengan menggunakan Persamaan 5.

$$Throughput(kbps) = \frac{(N_{recvBytes} * 8)}{simulationTime} \quad (5)$$

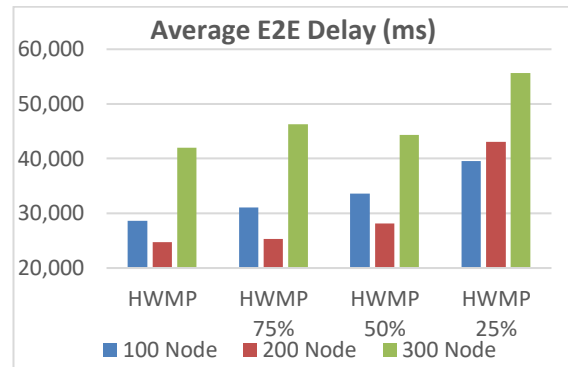
*Normalized Routing Load* merupakan persentase jumlah paket *routing* yang dikirimkan oleh setiap *node* dalam jaringan terhadap jumlah

total paket yang diterima *normalized routing load* dapat dihitung dengan menggunakan Persamaan 6.

$$NormalizedRoutingLoad(\%) = \frac{N_{routingPacket}}{N_{totalPacket}} \times 100\% \quad (6)$$

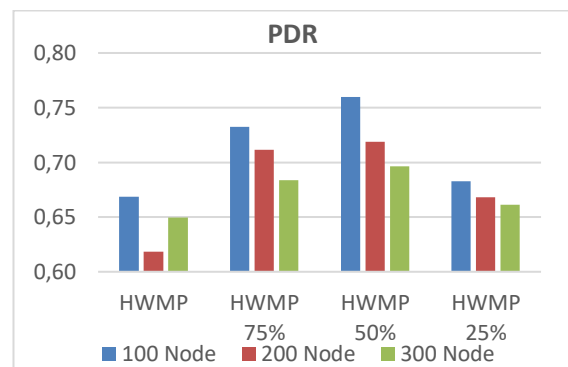
#### 4. HASIL DAN PEMBAHASAN

##### 4.1. Skenario Manhattan (Grid)



Gambar 7. Diagram *Average E2E Delay* pada Skenario Manhattan (*Grid*)

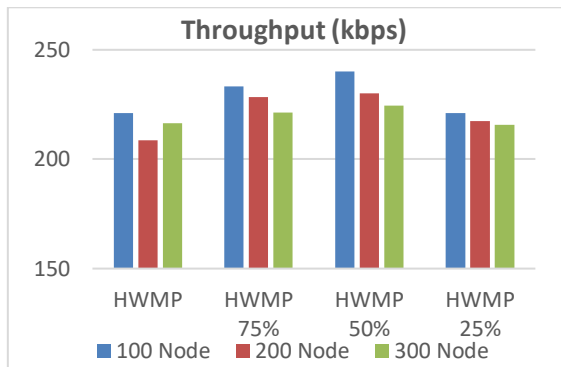
Berdasarkan Gambar 7, dapat dilihat bahwa pada skenario Manhattan (*grid*), protokol HWMP hasil modifikasi cenderung mengalami penurunan performa pada *average E2E delay* nya. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 2,42ms atau 7% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 3,56ms atau sekitar 12,2% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan *delay* sebesar 14,32ms atau 48,3% dari HWMP asli.



Gambar 8. Diagram PDR pada Skenario Manhattan (*Grid*)

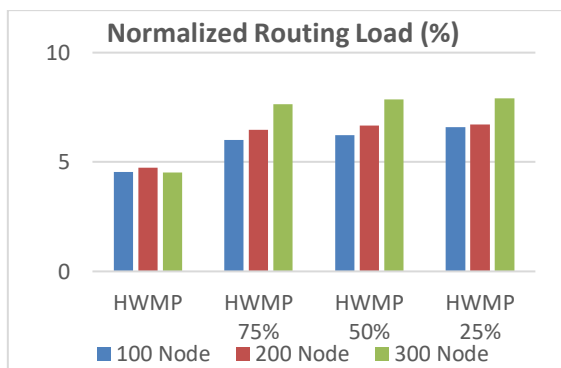
Berdasarkan Gambar 8, dapat dilihat bahwa pada skenario Manhattan (*grid*), PDR protokol HWMP hasil modifikasi lebih baik daripada HWMP asli. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 0,064 atau 9,98% dari HWMP asli.

Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 0,08 atau sekitar 12,37% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 0,025 atau 3,98% dari HWMP asli.



Gambar 9. Diagram *Throughput* pada Skenario Manhattan (*Grid*)

Berdasarkan Gambar 9, dapat dilihat bahwa pada skenario Manhattan (*grid*), *throughput* protokol HWMP hasil modifikasi lebih baik daripada HWMP asli. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 12,25 kbps atau 5,7% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 16,13 kbps atau sekitar 7,5% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 2,69 kbps atau 1,3% dari HWMP asli.

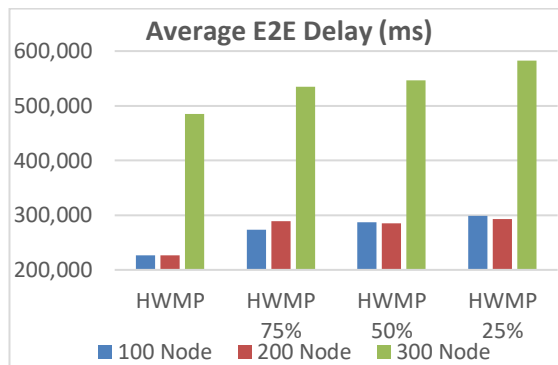


Gambar 10. Diagram *Normalized Routing Load* pada Skenario Manhattan (*Grid*)

Berdasarkan Gambar 10, dapat dilihat bahwa pada skenario Manhattan (*grid*), terjadi penurunan performa *normalized routing load* pada protokol HWMP hasil modifikasi. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 2,1 atau 9,98% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar

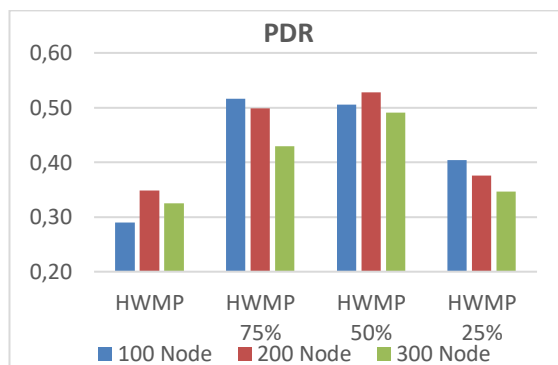
2,32 atau sekitar 50,7% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 2,47 atau 74,9% dari HWMP asli.

#### 4.2. Skenario Freeway (*Highway*)



Gambar 11. Diagram *Average E2E Delay* pada Skenario Freeway (*Highway*)

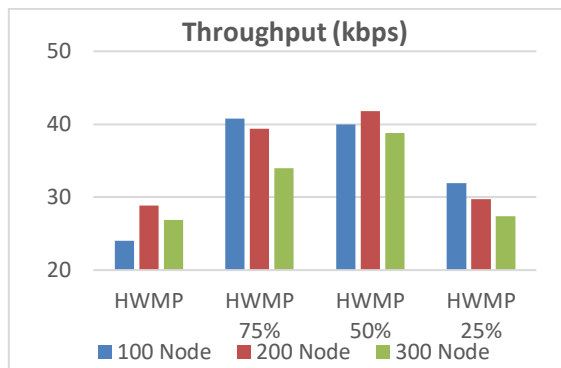
Berdasarkan Gambar 11, dapat dilihat bahwa pada skenario Freeway (*highway*), protokol HWMP hasil modifikasi cenderung mengalami penurunan performa pada *average E2E delay* nya. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 53,09ms atau 19,5% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 60,07ms atau sekitar 21,7% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan *delay* sebesar 53,09ms atau 19,5% dari HWMP asli.



Gambar 12. Diagram PDR pada Skenario Freeway (*Highway*)

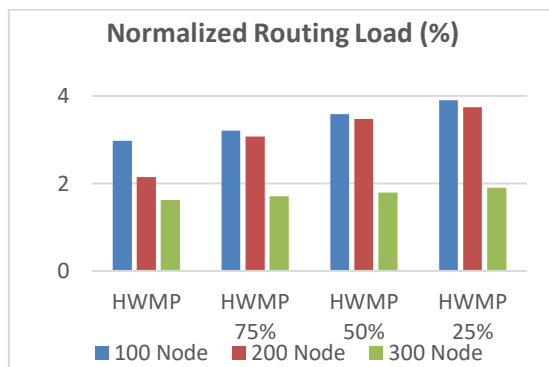
Berdasarkan Gambar 12, dapat dilihat bahwa pada skenario Freeway (*highway*), PDR protokol HWMP hasil modifikasi lebih baik daripada HWMP asli. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 0,16 atau 51% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 0,19 atau sekitar 59%

dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 0,05 ms atau 17,9% dari HWMP asli.



Gambar 13. Diagram Throughput pada Skenario Freeway (*Highway*)

Berdasarkan Gambar 13, dapat dilihat bahwa pada skenario Freeway (*highway*), *throughput* protokol HWMP hasil modifikasi lebih baik daripada HWMP asli. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 11,47 kbps atau 44,3% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 13,61 kbps atau sekitar 51,9% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 3,11 kbps atau 12,7% dari HWMP asli.

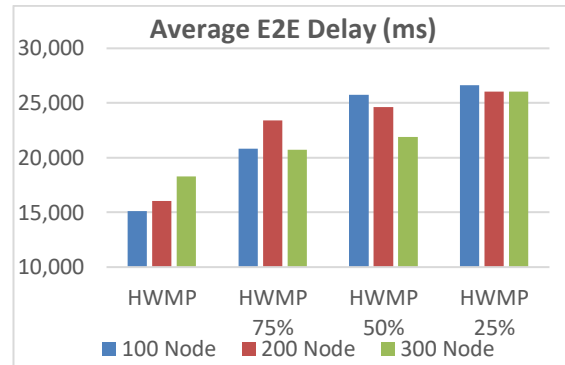


Gambar 14. Diagram *Normalized Routing Load* pada Skenario Freeway (*Highway*)

Berdasarkan Gambar 14, dapat dilihat bahwa pada skenario Freeway (*highway*), terjadi penurunan performa *normalized routing load* pada protokol HWMP hasil modifikasi. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 0,41 atau 18,5% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 0,70 atau sekitar 30,8% dari HWMP asli.

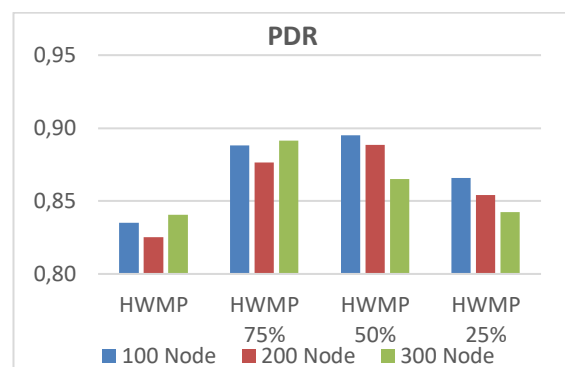
Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 0,93 kbps atau 40,7% dari HWMP asli.

### 4.3. Skenario Real Map



Gambar 15. Diagram *Average E2E Delay* pada Skenario *Real Map*

Berdasarkan Gambar 15, dapat dilihat bahwa pada skenario *real map*, protokol HWMP hasil modifikasi cenderung mengalami penurunan performa pada *average E2E delay* nya. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 5,17 ms atau 13,35% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 7,59 ms atau sekitar 7,59 dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan *delay* sebesar 9,74 ms atau 60,18% dari HWMP asli.

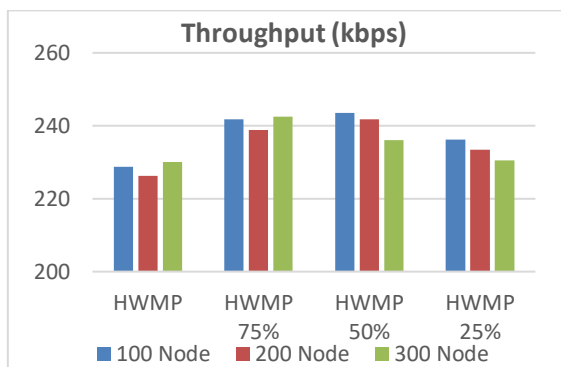


Gambar 16. Diagram PDR pada Skenario *Real Map*

Berdasarkan Gambar 16, dapat dilihat bahwa pada skenario *real map*, PDR protokol HWMP hasil modifikasi lebih baik daripada HWMP asli. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 0.051 atau 6.18% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 0.049 atau sekitar 5.91% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 0.0204

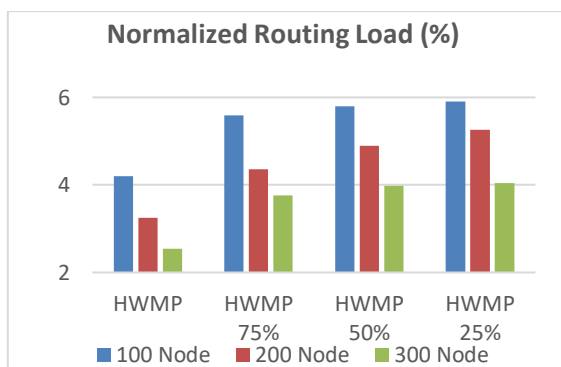


atau 2.46% dari HWMP asli.



Gambar 17. Diagram *Throughput* pada Skenario *Real Map*

Berdasarkan Gambar 17, dapat dilihat bahwa pada skenario *real map*, *throughput* protokol HWMP hasil modifikasi lebih baik daripada HWMP asli. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 12,62 kbps atau 5,5% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 12,05 kbps atau sekitar 5,3% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 5,01 kbps atau 2,2% dari HWMP asli.



Gambar 18. Diagram *Normalized Routing Load* pada Skenario *Real Map*

Berdasarkan Gambar 18, dapat dilihat bahwa pada skenario *real map*, terjadi penurunan performa *normalized routing load* pada protokol HWMP hasil modifikasi. HWMP hasil modifikasi dengan 75% *threshold* mempunyai rata-rata kenaikan sebesar 1,23 atau 38,2% dari HWMP asli. Ketika *threshold* sebesar 50% digunakan, terjadi kenaikan sebesar 1,56 atau sekitar 48,3% dari HWMP asli. Sedangkan ketika *threshold* sebesar 25% digunakan, terjadi kenaikan sebesar 1,73 atau

53,7% dari HWMP asli.

## 5. PENUTUP

### 5.1. Kesimpulan

1. Implementasi mekanisme congestion avoidance pada HWMP terbagi dalam 2 tahapan. Tahap pertama yaitu dengan mengimplementasi mekanisme *congestion detection* dan *notification*. Mekanisme ini berfungsi untuk mendeteksi potensi terjadi *congestion* dengan cara memonitor panjang *queue* pada *node*. Kemudian apabila terdeteksi terjadi *congestion* pada sebuah *node*, maka *node* tersebut akan mengirimkan pesan notifikasi berupa CCNF secara broadcast ke arah *node* sumber. Tahap kedua yaitu dengan melakukan pembaruan terhadap routing table dan melakukan pencarian rute baru. Pada tahapan ini, rute yang melalui *node* yang mengalami *congestion* akan diberi *flag* yang menandakan bahwa rute tersebut sudah kadaluarsa (RTF OLD). Dengan begitu, *node* sumber akan melakukan pencarian rute baru. Pada proses pencarian rute baru tersebut, *node* yang mengalami *congestion* akan mengabaikan PREQ dari *node* sumber. Sehingga rute baru tidak akan melalui *node* yang mengalami *congestion* tersebut.
2. Setelah dilakukan pengujian, dapat diketahui bahwa performa HWMP hasil modifikasi unggul pada PDR dan *throughput*, namun cenderung lemah pada *average E2E delay* dan *normalized routing load*. Hal ini dapat dilihat dari penurunan performa *average E2E delay* sebesar 7% hingga 60.18%, kenaikan sebesar 2.46% hingga 59% untuk PDR, kenaikan sebesar 1.3% hingga 51.9% pada *throughput*, serta penurunan performa sebesar 18.5% hingga 74.9% untuk parameter *normalized routing load*. Selain itu, ukuran *threshold* yang digunakan juga mempengaruhi performa dari HWMP hasil modifikasi tersebut. HWMP hasil modifikasi yang menggunakan *threshold* sebesar 50% dan 75% cenderung memiliki performa yang lebih baik dibanding *threshold* sebesar 25%. Hal tersebut terjadi akibat *threshold* sebesar 25% terlalu ketat, sehingga kondisi *congestion* lebih sering terjadi dan *node* akan lebih banyak menghabiskan waktu mencari rute baru.

## 5.2. Saran

1. Implementasi protokol HWMP pada NS2 masih belum sempurna, sehingga sedikit menyulitkan proses penelitian. Selain itu, hal tersebut dapat mempengaruhi hasil yang didapat selama penelitian. Untuk pengembangan lebih lanjut, ada baiknya menggunakan NS3 yang mempunyai implementasi HWMP yang lebih baik.
2. Pada penelitian ini, setiap percobaan hanya dilakukan sebanyak 5 kali untuk menghemat waktu penelitian. Untuk kedepannya, ada baiknya percobaan dilakukan lebih dari 5 kali. Hal tersebut bertujuan agar hasil uji yang didapat bisa lebih akurat.
3. Melakukan pengujian terhadap parameter QoS yang lain dan jumlah *node* yang lebih beragam. Dengan begitu, hasil yang didapat dapat lebih detail dan performa protokol hasil modifikasi ini dapat diketahui untuk parameter-parameter lainnya

## 6. DAFTAR PUSTAKA

- Andrey, O. V. & Hamrioui, E.-h., 2015. Comparative Study of Routing Protocols in Vehicular Ad-Hoc Networks (VANETs). *International Journal of Advanced Studies*, Volume 4, pp. 9-16.
- Badis, H. & Rachedi, A., 2015. Modeling tools to evaluate the performance of wireless multi-hop networks. In: M. S. Obaidat, P. Nicopolitidis & F. Zarai, eds. *Modeling and Simulation of Computer Networks and Systems*. s.l.:Morgan Kaufmann, pp. 653-682.
- Buford, J. F., Yu, H. & Lua, E. K., 2009,. Mobility and Heterogeneity. In: J. F. Buford, H. Yu & E. K. Lua, eds. *P2P Networking and Applications*. s.l.:Morgan Kaufmann, pp. 299-317.
- Ghori, M. R., Sadiq, A. S. & Ghani, A., 2018. VANET Routing Protocols: Review, Implementation and Analysis. *Journal of Physics: Conference Series*, Volume 1049.
- Hiertz, G. R. et al., 2010. IEEE 802.11s: The WLAN Mesh Standard. *IEEE Wireless Communications*.
- Jain, R. & Ramakrishnan, K. K., 1988. Congestion avoidance in computer networks with a connectionless network layer: concepts, goals and methodology. Washington, DC, IEEE.
- Khaliq, K. A., Qayyum, A., Elahi, E. & Akbar, M. S., 2014. Congestion avoidance hybrid wireless mesh protocol (CA-HWMP) for IEEE 802.11s. *Procedia Computer Science*, Volume 32, p. 229–236.
- Nayyar, A., 2018. Flying Adhoc Network (FANETs): Simulation Based Performance Comparison of Routing Protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP. 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), pp. 1-9.
- OpenStreetMap Foundation, 2021. OpenStreetMap. [Online] Available at: <https://www.openstreetmap.org/about> [Accessed 13 September 2021].
- Yu, X., Guo, H. & Wong, W.-C., 2011. A reliable routing protocol for VANET communications. 2011 7th International Wireless Communications and Mobile Computing Conference, pp. 1748-1753.
- Zanjireh, M. M. & Larijani, H., 2015. A Survey on Centralised and Distributed Clustering. Glasgow, s.n.