

# SURVEI PENANGANAN *BROADCAST STORM PROBLEM* PADA PROTOKOL *ROUTING AODV* DI MANET

Indera Zainul Muttaqien, As'ad Arismadhani, Royyana M. Ijtihadie dan Radityo Anggoro

Institut Teknologi Sepuluh Nopember Surabaya

Jl. Teknik Kimia, Gedung Teknik Informatika, Kampus ITS Sukolilo, Surabaya, 60111

Email : [inderazainul@gmail.com](mailto:inderazainul@gmail.com)<sup>1)</sup>, [project.arisfumi@gmail.com](mailto:project.arisfumi@gmail.com)<sup>2)</sup>

## ABSTRAK

Komunikasi multi-hop pada lingkungan MANET dapat melibatkan broadcast paket dalam proses route discovery. Protokol routing pada MANET akan melakukan broadcast paket RREQ dan menjalarkannya ke node tujuan secepat dan seefisien mungkin, dimana paket RREQ dari rute yang optimal adalah paket RREQ yang terlebih dahulu sampai ke tujuan. Aktifitas broadcast yang tidak terkontrol dapat menyebabkan suatu kondisi yang disebut broadcast storm problem. Broadcast storm problem dapat berdampak pada terganggunya kinerja dari protokol routing akibat adanya packet redundancy, contention, dan collision. Broadcast storm problem dapat ditangani dengan membatasi jumlah node yang dapat melakukan broadcast. Tujuan utama dari pembuatan makalah ini adalah merangkum beberapa mekanisme terbaru yang telah dilakukan oleh para peneliti untuk menangani broadcast storm problem. Kami juga menyajikan perbandingan dari mekanisme tersebut berdasarkan karakteristik dari mekanisme ditinjau dari kesesuaian dengan beberapa skema penanganan broadcast storm problem yang diajukan oleh peneliti sebelumnya, kebutuhan informasi tertentu antar node, lingkungan uji coba dan apakah mekanisme ini dapat diterapkan pada protokol routing lainnya.

**Kata Kunci:** Broadcast storm problem, MANET, komunikasi multi-hop, route discovery

## ABSTRACT

Multi-hop communication in MANET usually involve broadcast packets in the route discovery process. In MANET routing protocols, RREQ packets need to be broadcasted to the destination node as quickly and efficiently as possible, where the RREQ packet from the optimal route is the first RREQ packet to its destination. Unmanaged or blind broadcast mechanism can lead to a condition called broadcast storm problem. The broadcast storm problem can lead to serious inefficiency to the performance of a routing protocol due to the redundancy package, contention, and collision. This problem can be addressed by limiting the number of rebroadcast node. In this paper we reviewing the main features of the latest mechanisms proposed by researchers to deal with the broadcast storm problem. We also presents characteristics comparison of the mechanisms in terms of compliance with some broadcast storm problem handling scheme proposed by previous researchers, certain information needs between nodes, and whether this mechanism can be applied to other routing protocols.

**Keywords:** Broadcast storm problem, MANET, multi-hop communication, route discovery

## I. PENDAHULUAN

MANET (Mobile Ad-hoc Network) merupakan sekumpulan *node* bergerak (*mobile host*) yang dapat berkomunikasi satu sama lain. Pada lingkungan MANET tidak tersedia *base stations* yang dapat digunakan untuk merelai paket dari *node* satu ke *node* lain. Komunikasi antar *node* dapat dilakukan melalui : (a) skenario *single-hop*, yaitu jika komunikasi dilakukan secara langsung dengan *node* yang berada dalam jangkauan transmisi *node* sumber dan (b) skenario *multi-hop*, yaitu jika komunikasi antar *node* dilakukan melalui *node-node* perantara yang merelai paket tersebut sampai *node* tujuan. Pada komunikasi *multi-hop*, setiap *node* dapat berperan sebagai entitas sumber yang menginisiasi pengiriman paket, entitas tujuan (*destination*), dan entitas *router* yang merelai paket tersebut ke *node* lainnya.

Para peneliti telah melakukan berbagai kajian tentang komunikasi *multi-hop* dan protokol *routing* untuk jaringan *ad-hoc*. Hal yang perlu menjadi perhatian dari kajian-kajian tersebut adalah adanya mekanisme *broadcast* paket ke seluruh *node* dalam jaringan (*packet flooding*). Pada MANET, transmisi *broadcast* ini digunakan setidaknya dalam 2 hal [1], yaitu : (a) dalam proses *route discovery* dan (b) untuk menyiarkan paket ke *node-node* dalam wilayah geografis tertentu. Pada proses *route discovery*, protokol *routing* pada MANET akan melakukan *broadcast* paket RREQ dan mempropagasikannya ke *node* tujuan secepat dan seefisien mungkin, dimana paket RREQ dari rute yang optimal adalah paket RREQ yang terlebih dahulu sampai ke tujuan. Pengiriman paket secara *broadcast* ini dapat menimbulkan masalah dalam MANET. Mekanisme tersebut harus dikontrol sedemikian rupa supaya efisien dengan meminimalkan adanya paket rangkap (*redundant*) dan rusaknya paket akibat terjadinya *collision*. *Packet flooding* merupakan metode dasar yang digunakan oleh hampir semua protokol *routing* pada jaringan nirkabel *ad-hoc*, maka dengan adanya perbaikan/efisiensi pada algoritma *flooding* dapat meningkatkan performa dari protokol tersebut.

Tujuan survei ini antara lain untuk membandingkan mekanisme yang digunakan dalam menangani *broadcast storm problem* pada proses *route discovery* pada protokol *routing*. Protokol *routing reactive* menjadi fokus pada

survei karena protokol tipe ini menggunakan teknik *packet flooding* untuk melakukan *broadcast* paket data pada saat proses *route discovery*. Melalui survei ini kami mendiskusikan dan membandingkan beberapa metode yang merupakan pengembangan lebih lanjut dari skema penanganan *broadcast storm problem* yang telah diajukan oleh Tseng dkk [2].

Pembahasan selanjutnya pada makalah ini dibagi menjadi beberapa bagian sebagai berikut. Bagian 2 membahas mengenai *broadcast storm problem* dan dampak yang ditimbulkan. Metode-metode baru yang telah dikembangkan oleh para peneliti untuk menangani *broadcast storm problem* disajikan dalam bagian 3. Pada bagian 4 kami melakukan perbandingan dan pembahasan metode-metode yang telah diulas pada bagian 3. Bagian 5 merupakan sesi penutup yang berisi kesimpulan dari survei ini dan usulan metode penanganan *broadcast storm problem* yang dapat dikembangkan lebih lanjut.

Kami berharap survei ini dapat memberi penyegaran informasi mengenai berbagai mekanisme yang dapat dilakukan dalam menangani *broadcast storm problem* dalam pengembangan suatu protokol *routing*. Survei ini juga diharapkan dapat memberi kontribusi dalam pengembangan protokol *routing* lainnya dalam MANET yang lebih efektif, efisien dan komprehensif.

## II. BROADCAST STORM PROBLEM

*Broadcast storm problem* terjadi jika jaringan telah ‘dibanjiri’ oleh paket-paket *broadcast* yang dikirimkan oleh *node-node* pada jaringan tersebut. Pada MANET, khususnya dalam proses *route discovery*, sebuah protokol membangun rute dari *node* sumber ke *node* tujuan dengan melakukan *broadcast* paket Route Request (RREQ). Jika *node* tujuan menerima paket ini, maka *node* tujuan akan mengirimkan paket Route Reply (RREP) ke *node* sumber dan rute pengiriman pesan terbangun. Namun jika *node* yang menerima paket ini bukan *node* tujuan, maka *node* tersebut akan melakukan re-*broadcast* paket sampai *node* tujuan ditemukan.

Cara yang paling sederhana dalam melakukan *broadcast* adalah dengan mengirimkan paket ke semua *node* (*blind broadcast*). *Node* yang menerima paket tersebut kemudian melakukan *broadcast* ulang paket tersebut ke semua *node* yang ada pada jangkauan transmisinya. Proses ini tentu saja dapat menghabiskan *bandwidth* dan sumber daya (energi) karena masing-masing *node* menggunakan media jaringan nirkabel yang sama. Paket-paket *broadcast* ini akan membebani jaringan dan dapat membuat jaringan lumpuh dan tidak dapat diakses oleh paket-paket normal.

Dampak dari *broadcast storm problem* seperti yang disampaikan oleh Y.C. Tseng dkk [2] adalah sebagai berikut :

- a. *Redundant rebroadcasts*, yaitu jika *node-node* menerima paket yang sama beberapa kali dan saat *node* tersebut akan melakukan *rebroadcast*, *node-node* tetangganya telah memiliki paket tersebut. Sehingga proses *rebroadcast* menjadi tidak efektif.
- b. *Contention*, yaitu keadaan dimana setiap *node* saling bersaing dalam melakukan transmisi ulang.
- c. *Collision*, yaitu rusaknya paket karena *node-node* mengirimkan paket secara bersamaan. Hal ini terjadi karena kurangnya mekanisme *backoff* dan tidak adanya *Collision Detection*.

## III. MEKANISME UNTUK MENANGANI BROADCAST STORM PROBLEM

Tseng, dkk [2] mengidentifikasi permasalahan *forwarding* ini dan mengajukan 5 skema untuk mengurangi redundansi paket yang menyebabkan BSP. Perbedaan dari masing-masing skema terletak pada bagaimana membatasi *node-node* yang dapat melakukan *broadcast* paket dalam melakukan penyusunan rute. Berikut adalah skema yang diajukan dalam [2]:

### a. Probabilistic-based Scheme

Skema ini menggunakan fungsi probabilitas  $P$  dalam melakukan transmisi ulang. Jika fungsi  $P$  bernilai 1 ini artinya semua *node* akan mem-*broadcast* seluruh paket yang diterima. Skema ini mengasumsikan setiap *node* memiliki kemampuan transmisi yang sama.

Secara umum, skema ini dapat dikelompokkan menjadi 2, yaitu : *fixed probability* dan *dynamic probability*. Pada skema *fixed probability*, setiap *node* akan me-*rebroadcast* pesan dengan nilai *threshold* probabilitas  $p$  yang sama yang sudah ditentukan sebelumnya (*pre-defined*). Dengan demikian, jika  $P$  bernilai 1 maka akan terjadi *blind flooding* yaitu kondisi dimana setiap *node* akan melakukan re-transmisi paket yang diterimanya ke setiap tetangganya.

Pada *dynamic probability*, setiap *node* dapat memiliki nilai probabilitas re-transmisi yang berbeda sesuai dengan kondisi yang digunakan untuk menentukan probabilitas, misalnya tingkat kepadatan *node* (jumlah *node* tetangga). Pada makalah ini, kami memilih makalah tentang *adjusted probabilistic route discovery* [3] untuk memberikan penjelasan lebih detail tentang skema *dynamic probability*.

### b. Counter-based Scheme

Pada skema ini, pembatasan jumlah transmisi ulang yang dilakukan oleh suatu *node* didasarkan oleh suatu *counter*. Jika telah mencapai nilai *counter* tertentu, sebuah *node* tidak diperkenankan untuk melakukan meneruskan paket tersebut (paket di-drop).

Saat sebuah *node* menerima paket, *node* tersebut akan menginisialisasi sebuah *counter* dan *timer*. *Counter* diatur bernilai 0 dan nilai random *RAD* (*Random Assesment Delay*) digunakan sebagai nilai awal *timer*. Nilai *RAD* ini berkisar antara 0 sampai dengan waktu maksimum *delay* yang dapat dilakukan sampai terjadinya *collision*. *Counter* ini akan di-update (ditambah 1) setiap kali *node* menerima paket yang sama (*redundant packet*). Saat nilai *timer* tercapai, dilakukan pengecekan terhadap nilai *counter*. *Node* hanya akan melakukan *rebroadcast* paket jika nilai *counter* lebih kecil dari *threshold*. Dengan demikian proses *broadcast* dapat dibatasi. Pada makalah ini, kami memilih paper dari Yassein, dkk [4] yang membahas tentang *dynamic-counter based broadcasting* untuk memberikan penjelasan lebih detil tentang skema *counter-based*.

### c. Distance-Based Scheme

Strategi ini memperhitungkan jarak antar *node* untuk melakukan transmisi ulang. Hanya *node-node* yang memenuhi jarak minimal (*threshold*) dari sumber yang dapat melakukan transmisi ulang karena semakin jauh jarak suatu *node* dari sumber, maka jangkauan keseluruhan dari transmisi tersebut akan semakin besar. Strategi ini kemudian berkembang dengan memperhitungkan kekuatan sinyal transmisi dari masing-masing *node* untuk menentukan *node* yang dapat melakukan transmisi ulang.

### d. Location-Based Scheme

Skema memanfaatkan informasi dari lokasi *node-node* untuk menentukan area yang digunakan untuk melakukan *rebroadcast*. Setiap kali sebuah *node* akan melakukan *broadcast*, *node* tersebut akan menambahkan informasi lokasi dan cakupan wilayah yang dapat digunakan untuk melakukan *rebroadcast*. Semakin luas cakupan wilayah yang dapat digunakan untuk melakukan *rebroadcast*, maka akan semakin luas jangkauan transmisi *multi-hop* yang digunakan dan secara otomatis mengurangi jumlah transmisi ulang. Jika wilayah tersebut tidak memenuhi *threshold* tertentu, maka paket akan di-drop. Akurasi lokasi dari tiap *node* dapat meningkatkan efisiensi dari skema ini, untuk itu pada skema ini, setiap *node* harus dapat menentukan lokasinya sendiri, misalnya dengan menggunakan perangkat GPS.

Pada makalah ini, kami memilih hasil penelitian dari Gupta dan Mathur [5] yang membahas tentang *enhanced flooding mechanism based on node position* memberikan penjelasan lebih detil tentang skema *location-based*.

### e. Cluster-Based Scheme

Strategi ini membagi *graph* menjadi beberapa *cluster* yang saling berdekatan [10]. *Node-node* dari setiap *cluster* yang berdekatan membentuk sebuah sub jaringan dan setiap sub jaringan ini membentuk *dominating-set* yang saling terhubung. Proses *routing* dapat dilakukan pada sub jaringan kecil yang saling terhubung, sehingga tidak memerlukan penghitungan ulang tabel *routing* pada sub jaringan.

Beberapa mekanisme untuk menangani *broadcast storm problem* yang dibahas dalam makalah ini adalah sebagai berikut :

#### A. Adjusted probabilistic route discovery

Skema *dynamic probabilistic* yang diusulkan oleh Abdulai, dkk pada [3] menyatakan *node* pada pada lingkungan MANET dapat terdistribusi secara acak. Hal ini memiliki arti suatu *node* dapat memiliki jumlah *node* tetangga yang berbeda-beda. Jika area yang berada pada jangkauan transmisi suatu *node* disebut sebagai suatu *region*, maka pada jaringan tersebut akan terdapat *region-region* yang memiliki kepadatan berbeda (jumlah *node* yang bervariasi di tiap *region*). Berdasarkan asumsi bahwa setiap *node* memiliki kemampuan transmisi yang sama, maka untuk menjamin ketersediaan paket ke *node* tetangga, maka diperlukan pengaturan probabilitas *forwarding* dari setiap *node* secara lokal di tiap-tiap *region*, dimana pada *node* yang kepadatannya jarang memiliki tingkat probabilitas *forwarding* yang lebih tinggi dari *node* yang kepadatannya tinggi. *Node-node* pada skema ini mengirimkan paket “hello” ke tetangganya secara periodik. Dari mekanisme tersebut, tiap *node* akan memiliki daftar “1-hop neighbour” dan informasi kepadatan di *region*-nya.

Untuk dapat menentukan suatu *node* berada pada *region* dengan kepadatan yang tinggi atau rendah, digunakan nilai acuan yang diperoleh dari serangkaian percobaan terhadap jaringan yang diamati. Dengan demikian akan diperoleh nilai acuan untuk suatu luas area *A* dengan jumlah *node* sebanyak *N*, maka tiap *node* akan memiliki jumlah rata-rata *node* tetangga sebanyak  $\bar{n}$ , dengan jumlah rata-rata minimum *node* tetangga sebanyak  $\bar{n}_{min}$  dan jumlah rata-rata maksimum *node* tetangga sebanyak  $\bar{n}_{max}$ .

Abdulai, dkk mengajukan 2 macam algoritma *adjusted probabilistic*, yaitu :

##### 1) 2P-Scheme

Pada algoritma ini, setiap *region* dibagi dalam 2 kelompok, yaitu kelompok *region* dengan tingkat

kepadatan tinggi dan kelompok *region* dengan tingkat kepadatan rendah. Kelompok dengan tingkat kepadatan rendah harus memiliki probabilitas *forwarding* yang lebih tinggi daripada kelompok dengan tingkat kepadatan yang lebih tinggi.

Penjelasan algoritma dari *2P-Scheme* pada gambar 1 adalah sebagai berikut. Jika  $n_A$  adalah jumlah *node* tetangga dari suatu *node* A, *node* A dikelompokkan sebagai *node* dengan tingkat kepadatan rendah jika  $n_A \leq \tilde{n}$  dan jika  $n_B$  adalah jumlah *node* tetangga dari *node* B, *node* B dikelompokkan sebagai *node* dengan tingkat kepadatan tinggi jika  $n_B > \tilde{n}$ . Dengan demikian jika  $p_A$  adalah *threshold* probabilitas *forwarding* dari A dan  $p_B$  adalah *threshold* probabilitas *forwarding* dari B, maka  $p_A > p_B$ .

Algoritma: 2P-scheme	
1.	Saat <i>node</i> X menerima paket RREQ
2.	Hitung jumlah $n_x$ (jumlah <i>node</i> tetangga dari <i>node</i> X)
3.	Hitung jumlah rata-rata <i>node</i> tetangga $\tilde{n}$
4.	IF paket RREQ ini belum pernah diterima IF $n_x \leq \tilde{n}$ <i>Node</i> X berada pada <i>region</i> dengan tingkat kepadatan rendah $p_x = p^{high}$ ELSE <i>Node</i> X berada pada <i>region</i> dengan tingkat kepadatan tinggi $p_x = p^{low}$
5.	Hitung nilai random Rand
6.	IF Rand $\leq p_x$ Lakukan rebroadcast paket RREQ
7.	ELSE Drop paket RREQ

Gambar 1. Algoritma 2-P Scheme [3]

## 2) 3P-Scheme

Pada *3P-scheme*, *node* dibagi menjadi 3 kelompok, yaitu kelompok *node* dengan tingkat kepadatan rendah ( $n_i \leq \tilde{n}_{min}$ ), kelompok *node* tingkat kepadatan sedang ( $\tilde{n}_{min} < n_i \leq \tilde{n}_{max}$ ), dan kelompok *node* dengan tingkat kepadatan tinggi ( $n_i > \tilde{n}_{max}$ ). Sehingga jika  $n_i$  menyatakan jumlah *node* tetangga dari suatu *node* i, dan  $p_i$  adalah *threshold* probabilitas *forwarding* dari *node* i, maka jika  $n_A < n_B < n_C$ , maka  $p_A > p_B > p_C$ . Algoritma *3P-scheme* dapat dilihat pada gambar 2.

Algoritma: 3P-scheme	
1.	Saat <i>node</i> X menerima paket RREQ
2.	Hitung jumlah $n_x$ (jumlah <i>node</i> tetangga dari <i>node</i> X)
3.	Hitung jumlah rata-rata minimum <i>node</i> tetangga $\tilde{n}_{min}$ dan rata-rata maksimum <i>node</i> tetangga $\tilde{n}_{max}$
4.	IF paket RREQ ini belum pernah diterima IF $n_x \leq \tilde{n}_{min}$ <i>Node</i> X berada pada <i>region</i> dengan tingkat kepadatan rendah $p_x = p^{high}$ ELSE IF $\tilde{n}_{min} < n_x \leq \tilde{n}_{max}$ <i>Node</i> X berada pada <i>region</i> dengan tingkat kepadatan sedang $p_x = p^{sedang}$ ELSE IF $n_x > \tilde{n}_{max}$ <i>Node</i> X berada pada <i>region</i> dengan tingkat kepadatan tinggi $p_x = p^{low}$
5.	Hitung nilai random Rand
6.	IF Rand $\leq p_x$ Lakukan rebroadcast paket RREQ
7.	ELSE Drop paket RREQ

Gambar 2. Algoritma 3-P Scheme [3]

Keunggulan metode ini adalah penggunaan informasi tentang kepadatan dari tiap *region* untuk penentuan *threshold* probabilitas *forwarding* secara adaptif. Dengan demikian dapat mengurangi *redundant broadcast* pada *region* dengan kepadatan tinggi dan meningkatkan konektivitas jaringan pada *region* dengan kepadatan rendah.

Menurut kami, kelemahan dari metode ini adalah dalam menentukan nilai *threshold* probabilitas. Penentuan

nilai *threshold* harus dilakukan secara periodik dengan mempertimbangkan jumlah *node* pada area yang dicakup (*network space*) yang tentunya dapat menjadi *overhead*. Selain itu, pada kelompok *node* dengan tingkat kepadatan rendah, *end-to-end delay* cenderung menurun akibat terbatasnya konektifitas jaringan.

**B. Dynamic counter-based broadcasting scheme**

Metode yang diajukan oleh Yassein, dkk [4] adalah skema *counter-based* yang adaptif dengan menggunakan nilai *threshold counter* berdasarkan kepadatan *node* (jumlah *node* tetangga dari suatu *node*). Nilai *threshold* yang digunakan ada 3 macam, yaitu *threshold* untuk *node* yang berada pada kelompok *node* dengan tingkat kepadatan rendah, *threshold* untuk *node* yang berada pada kelompok *node* dengan tingkat kepadatan sedang, dan *threshold* untuk *node* yang berada pada kelompok *node* dengan tingkat kepadatan tinggi.

Pada proses awal, dilakukan penghitungan nilai dari 3 *threshold* yang digunakan. Algoritma ini memanfaatkan paket “hello” yang dikirimkan antar *node* secara periodik untuk mengetahui jumlah *node* tetangga tiap *node*. Dari jumlah *node* tetangga tersebut ditentukan *threshold*  $C_{min}$  (dihitung dari jumlah *node* tetangga terbanyak suatu *node*), *threshold*  $C_{mid}$  (dihitung dari jumlah rata-rata *node* tetangga dari tiap *node*), dan  $C_{max}$  (dihitung dari jumlah *node* tetangga paling sedikit dari suatu *node*).

Algoritma *dynamic counter-base broadcasting* sebagaimana ditampilkan pada gambar 3, dapat dijelaskan sebagai berikut. Kika  $n_i$  adalah jumlah *node* tetangga dari suatu *node* i dan  $\bar{n}$  adalah jumlah rata-rata *node* tetangga dari tiap *node*, maka jika  $n_A < \bar{n}$ , maka A adalah *node* yang berada pada kelompok dengan tingkat kepadatan rendah, sehingga nilai *counter threshold* untuk A adalah  $C_A = C_{max}$ . Demikian juga jika  $n_B > \bar{n}$ , maka B adalah *node* yang berada pada kelompok *node* dengan tingkat kepadatan tinggi, sehingga nilai *counter threshold* untuk B adalah  $C_B = C_{min}$ . Sebuah *node* C akan memiliki nilai *counter threshold*  $C_C = C_{mid}$  jika  $n_C = \bar{n}$

Nilai awal timer (RAD) ditentukan dengan membangkitkan nilai acak antara 0 dan  $T_{max}$  yang memenuhi persamaan  $RAD = \frac{X}{RF}$ , dengan  $T_{max}$  adalah waktu maksimum *delay* yang dapat dilakukan sebelum terjadinya *collision*, X adalah nilai acak antara 0 dan 1, dan RF adalah *Random Factor* (*pre-defined*).

Algoritma: dynamic counter-based broadcasting	
1.	Saat <i>node</i> X menerima paket RREQ
2.	Catat Broadcast ID, hitung $n_1$ (jumlah <i>node</i> tetangga paling sedikit), $n_2$ (jumlah <i>node</i> tetangga terbanyak), hitung nilai <i>threshold</i> $C_{min}$ , $C_{med}$ , $C_{max}$
3.	Hitung jumlah $n_x$ (jumlah <i>node</i> tetangga dari <i>node</i> X)
4.	IF paket RREQ ini belum pernah diterima IF $n_x < n_1$ <i>Node</i> X berada pada region dengan tingkat kepadatan rendah $C_x = C_{min}$ ELSE IF $n_1 \leq n_x \leq n_2$ <i>Node</i> X berada pada region dengan tingkat kepadatan sedang $C_x = C_{med}$ ELSE IF $n_x > n_2$ <i>Node</i> X berada pada region dengan tingkat kepadatan tinggi $C_x = C_{max}$ Timer = 1 Counter = 1 ELSE Counter = Counter + 1
5.	WHILE (timer < RAD) Wait sampai timer berakhir Timer berakhir
6.	IF (Counter < $C_x$ ) Lakukan rebroadcast paket RREQ
7.	ELSE Drop paket RREQ

Gambar 3. Algoritma dynamic counter-based broadcasting.

Keunggulan metode ini adalah penggunaan informasi tentang kepadatan dari tiap *region* untuk penentuan *threshold counter* secara dinamis. Dengan demikian dapat mengurangi *redundant broadcast* pada *region* dengan kepadatan tinggi dan meningkatkan konektifitas jaringan pada *region* dengan kepadatan rendah.

Menurut kami, kelemahan dari metode ini adalah dalam menentukan nilai *threshold* probabilitas. Penentuan nilai *threshold* harus dilakukan secara periodik dengan mempertimbangkan jumlah *node* pada area yang dicakup (*network space*) yang tentunya dapat menjadi *overhead*.

### C. Irresponsible Forwarding (IF)

*Irresponsible Forwarding* adalah strategi *forwarding* paket yang berdasarkan pendekatan probabilitas (*probabilistic approach*). Pendekatan probabilitas menyatakan bahwa sebuah *node* memiliki kemungkinan untuk meneruskan paket dengan probabilitas  $p$  dan kemungkinan tidak meneruskan paket dengan probabilitas  $1-p$ . Pada IF [6], probabilitas ini dihitung oleh setiap *node* dengan mempertimbangkan kepadatan *node* dalam suatu area, jangkauan transmisi, dan jarak antara *node* tersebut dengan *node* pengirim transmisi. Informasi ini dapat diperoleh dengan lebih akurat dengan menggunakan GPS pada tiap-tiap *node*.

Pada mekanisme ini, *node* yang menerima paket akan melakukan penghitungan kemampuan dirinya dan *node-node* tetangganya dalam melakukan *rebroadcasting* paket. Sebuah *node* didefinisikan dapat melakukan *rebroadcast* paket dengan baik adalah jika *node* tersebut ( $N_i$ ) memiliki jarak terjauh dengan pengirim transmisi ( $N_{i-1}$ ) dan berada dalam *range* transmisi  $N_{i-1}$ . Jika sebuah *node* mendeteksi ada *node* tetangga yang memiliki nilai probabilitas *forwarding* paket yang lebih tinggi, maka *node* tersebut tidak akan melakukan *rebroadcast* paket.

A. Gorrieri dan G. Ferrari [7] melakukan modifikasi untuk menangani BSP pada AODV dengan cara mengganti mekanisme flooding pada proses *route discovery* AODV dengan menggunakan *Irresponsible Forwarding*.

Saat *node* sumber S akan mentransmisikan pesan ke *node* tujuan D, maka S akan melakukan *initial broadcast* (*transmission hop* ke-0). Selanjutnya *node-node* tetangga yang menerima *initial broadcast* ini bertanggung jawab untuk meneruskan transmisi tersebut (*transmission hop* ke-1). *Node-node* ini disebut sebagai domain transmisi ke-1. Aturan transmisinya dilakukan berdasarkan konsep *Irresponsible Forwarding* seperti pada [5]. Saat sebuah *node* pertama kali menerima request RREQ, maka *node* tersebut menyimpan nilai *broadcast\_id*. Jika kemudian *node* tersebut menerima RREQ, maka *node* tersebut akan membandingkan nilai *broadcast\_id* tersebut dengan nilai yang sudah disimpan sebelumnya, jika hasilnya sama maka *request* ini dikategorikan sebagai paket redundan dan harus di-*drop*. Namun jika hasilnya berbeda dan *node* tidak memiliki informasi *routing* ke tujuan, maka dilakukan *rebroadcast* dengan strategi *Irresponsible Forwarding*. Dengan aturan yang sama, *node-node* yang menerima transmisi dari domain transmisi ke-1 akan melakukan *transmission hop* k-2 ke *node-node* lainnya. Proses ini dilakukan oleh tiap-tiap domain transmisi sampai pesan sampai ke D.

*Irresponsible Forwarding* bermanfaat untuk membatasi jumlah transmisi ulang karena proses transmisi ulang hanya dilakukan oleh *node-node* yang memiliki probabilitas *rebroadcast* yang tinggi.

### D. Enhanced Distance Based [11]

Mekanisme EDB [11] (*Enhanced Distance Based*) adalah pengembangan dari metode DB (*distance-based*) yang difokuskan dalam penghematan energi pada masing-masing *node* dengan tetap mempertahankan performa jaringan. Berbeda dengan DB yang menggunakan jarak antar *node* sebagai *threshold* (hanya *node* yang memiliki jarak dengan source lebih besar dari *threshold* yang dapat melakukan meneruskan paket), pada EDB menggunakan parameter kekuatan sinyal sebagai *threshold*.

Tiap *node* pada EDB mendeteksi jarak dengan *node* tetangganya berdasarkan kekuatan sinyal yang diterima, sehingga EDB tidak mempersyaratkan adanya GPS pada tiap *node*. Semakin kuat sinyal yang diterima dari suatu *node* tetangga, maka *node* penerima dapat menentukan bahwa jarak *node* tersebut semakin dekat dengan dirinya. Konsekuensi lainnya adalah semakin banyak penghalang antar *node*, maka kekuatan sinyal transmisi yang diterima oleh *node* tetangga tersebut akan semakin berkurang, maka pada EDB jarak antar *node* tersebut semakin jauh. Untuk itu, *threshold* yang digunakan dalam EDB adalah dalam satuan dBm bukan meter. *Threshold* yang digunakan berada dalam interval kekuatan sinyal minimum sampai maksimum yang memungkinkan sebuah pesan dapat dikirimkan ke sebuah *node* yang memiliki jarak cukup jauh dari *node* sumber, namun tetap dapat melakukan pemrosesan pesan tersebut dengan baik. Dalam EDB digunakan nilai *threshold* [-95, -90], yang diperoleh dari hasil uji coba [11], dimana -90 dBm adalah *border\_threshold*. Hanya *node-node* yang menyatakan yang dapat menerima pesan ini dengan kekuatan sinyal tidak lebih besar dari -90 dBm yang dapat melakukan *forwarding* paket. Nilai -95 dBm adalah *end\_threshold* yang menyatakan batas minimum kekuatan sinyal yang diterima oleh suatu *node* untuk dapat memproses suatu pesan dengan baik. Jika kekuatan sinyal yang diterima kurang dari nilai *end\_threshold*, maka *node* akan mengasumsikan paket ini rusak / tidak dapat diproses.

Seperti halnya pada DB, setiap *node* pada mekanisme EDB mengirimkan 'hello' message secara periodik dengan kekuatan pengiriman sinyal yang sama. Selain digunakan untuk mengetahui *node-node* tetangga di sekitar suatu *node*, 'hello' message juga digunakan untuk menentukan jarak *node-node* tetangga ditinjau dari kekuatan sinyal yang dapat diterima dari *node-node* tersebut. Kemudian *node* dapat menentukan kekuatan sinyal minimum yang dapat digunakan untuk mengirimkan pesan dengan baik ke *node* tetangganya.

Pada awal proses *route discovery*, *node* tujuan S akan mem-broadcast pesan RREQ ke seluruh tetangga dengan kekuatan sinyal minimum (yang diperoleh dari pemrosesan 'hello' message). *Node* tetangga yang menerima pesan ini kemudian menghitung kekuatan sinyal pesan RREQ yang diterimanya. Jika kekuatan sinyal tersebut

lebih besar dari *threshold*, maka *node* tidak melakukan *forwarding* paket, karena *node* ini menganggap ada *node* lain yang memiliki jarak lebih jauh yang dapat meneruskan pesan RREQ. Jika kekuatan sinyal tersebut masih dalam range *threshold*, maka *node* tersebut akan meneruskan pesan RREQ dengan kekuatan sinyal minimum ke tetangga terjauhnya sesuai dengan data yang diperoleh saat pemrosesan 'hello' message. Proses *route discovery* ini akan berlangsung dengan mekanisme yang sama di tiap *node* sampai *node* tujuan D ditemukan.

EDB mengurangi jumlah *rebroadcast* dengan melakukan membatasi *node-node* yang dapat melakukan *forwarding paket* berdasarkan kekuatan sinyal. Dengan demikian EDB memiliki keuntungan lain yaitu dalam hal penghematan sumber daya dan meminimalkan *colision*. Penghematan sumber daya diperoleh dengan adanya pembatasan kekuatan sinyal untuk melakukan transmisi ulang pesan ke *node* lain yang dipilih untuk melakukan *rebroadcast* sampai pada batas kekuatan sinyal yang dianggap cukup sehingga *node* tersebut dapat melakukan pemrosesan pesan dengan baik. Keuntungan dari hal ini adalah berkurangnya interferensi sinyal yang tidak perlu yang mengakibatkan *collision*. Tingkat *collision* perlu dijaga seminimal mungkin supaya *node* tidak perlu melakukan transmisi ulang sinyal akibat rusaknya pesan akibat *collision*.

E. Enhanced flooding mechanism based on node position [5]

Algoritma ini diajukan oleh Gupta dkk [5] untuk memperbaiki proses *route discovery* pada AODV dengan cara membatasi area yang digunakan untuk melakukan *route discovery*, dengan demikian akan mengurangi proses transmisi ulang paket RREQ. Algoritma ini mengasumsikan seluruh *node* menggunakan GPS untuk memperoleh informasi lokasinya yang kemudian dipertukarkan dengan *node* di sekitarnya dengan mekanisme paket "hello".

Beberapa informasi yang digunakan dalam proses *route discovery* pada algoritma ini adalah sebagai berikut :

1) Informasi Lokasi

Dari perangkat GPS yang ada pada tiap *node* akan diperoleh lokasi fisik, kecepatan, dan informasi lainnya.

2) Expected Zone

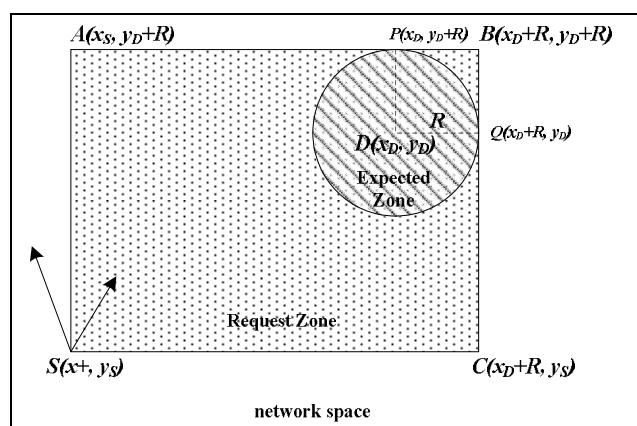
*Expected zone* adalah zona yang diperkirakan memuat lokasi dari *node* tujuan D. Dari sudut pandang *node* sumber S, *expected zone* adalah sebuah area lingkaran pada *network space* dengan yang memiliki radius R. Untuk dapat mengetahui *expected zone*, S harus memiliki informasi lokasi dari D pada waktu tertentu  $t_0$ , dan kecepatan  $V_D$ . Dengan demikian, pada waktu  $t_1$ , *expected zone* dari D adalah sebuah lingkaran dengan titik tengah  $D(x_D, y_D)$  dan radius R, dimana  $R = V_D (t_1 - t_0)$ .

Namun jika S tidak memiliki informasi tentang lokasi D, maka *expected zone* adalah *network space* itu sendiri. Konsep tentang *expected zone* akan menjadi lebih jelas pada pembahasan *request zone* di bawah ini.

3) Request Zone

*Request zone* adalah zona yang digunakan untuk melakukan *forwarding* paket ke *node* tetangga. Hanya *node-node* yang berada dalam *request zone* yang dapat melakukan *forwarding* paket.

*Request zone* berbentuk persegi panjang terkecil yang dapat memuat lokasi dari S dan *expected zone*. Ilustrasi dari *expected zone* dan *request zone* pada algoritma Gupta dkk [5] dapat dilihat pada gambar 4.



Gambar 4. Ilustrasi *expected zone* dan *request zone* pada *network space* (diadaptasi dari [5])

Saat *node* S akan melakukan proses *route discovery*, S akan memilih 4 *node* tetangganya yang masing-masing berada pada 4 zona yang berada pada jangkauan transmisi S untuk dipilih sebagai NNRR (*Nominated Neighbour to Rebroadcast RREQ*). Tabel I memuat syarat/aturan pembagian zona transmisi.

TABEL I  
PEMBAGIAN ZONA TRANSMISI NODE PADA METODE [5]



No	Zona	Syarat
1	Zona 1	Node A berada pada zona 1 jika : $x_S \leq x_A$ dan $y_S \leq y_A$ , dimana $(x_S, y_S)$ adalah lokasi node S dan $(x_A, y_A)$ adalah lokasi node A
2	Zona 2	Node B berada pada zona 2 jika : $x_S \leq x_B$ dan $y_S \leq y_B$ , dimana $(x_S, y_S)$ adalah lokasi node S dan $(x_B, y_B)$ adalah lokasi node B
3	Zona 3	Node C berada pada zona 3 jika : $x_S \leq x_C$ dan $y_S \leq y_C$ , dimana $(x_S, y_S)$ adalah lokasi node S dan $(x_C, y_C)$ adalah lokasi node C
4	Zona 4	Node D berada pada zona 4 jika : $x_S \leq x_D$ dan $y_S \leq y_D$ , dimana $(x_S, y_S)$ adalah lokasi node S dan $(x_D, y_D)$ adalah lokasi node D

Pada masing-masing zona, akan dipilih *node NNRR* yaitu *node* yang memiliki jarak terjauh dari S dengan menggunakan rumus jarak Euclidean. Proses selanjutnya adalah menyertakan informasi *NNRR* dan *request zone* akan ke dalam paket RREQ dan kemudian proses *route discovery* dimulai.

Saat *node* tetangga menerima RREQ, *node* tersebut akan memeriksa apakah dirinya merupakan *NNRR* dan berada dalam *request zone*. Jika ya, maka *node* tersebut akan menghitung ulang nilai *request zone* dan *NNRR* yang akan digunakan untuk meneruskan paket RREQ ini. Informasi tersebut akan dimuat dalam paket RREQ dan *node* tersebut akan *re-broadcast* paket RREQ untuk melanjutkan proses *route discovery* sampai ditemukannya *node tujuan D*.

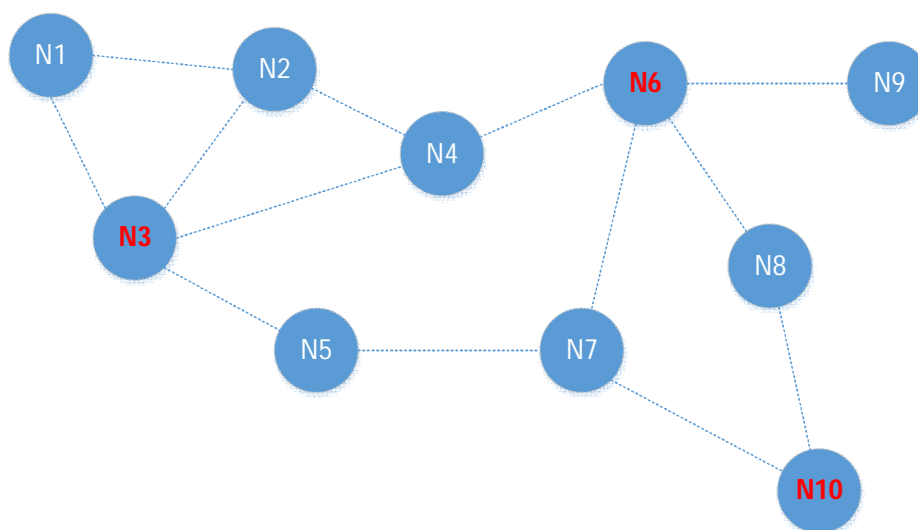
Keunggulan metode ini adalah pada pemilihan *node NNRR*. *Node* yang dipilih sebagai *NNRR* adalah *node* yang memiliki jarak terjauh untuk memperluas jangkauan transmisi. Selain itu hanya *node NNRR* yang berada pada *request zone* saja yang dapat melakukan *rebroadcast*.

Kelemahan metode ini adalah saat *node S* tidak memiliki informasi lokasi dari *node D*. Performa dari algoritma ini ditentukan oleh luas dari *request zone*, semakin luas *request zone* maka jumlah *node* yang berpotensi dipilih untuk melakukan *rebroadcast* akan semakin banyak. Selain itu, penggunaan GPS untuk menentukan lokasi secara periodik dapat berakibat pada penggunaan sumber daya / baterai.

#### F. Domination-set Based Routing (DBR)

Preetha K. G. dan A. Unnikrishnan [8] mengusulkan sebuah metode untuk mengurangi *delay* dan *routing overhead* dengan menggunakan *dominating nodes* pada jaringan. *Dominating-set* merupakan kumpulan beberapa *node* pada jaringan yang terhubung dengan *dominating node* secara langsung maupun terhubung melalui perantara *node* lain. Pada metode ini rute terbentuk dengan melewati *dominating node*, yaitu merupakan *node* yang terhubung dengan *node-node* lain pada jaringan dengan jumlah *node* yang terhubung lebih banyak dari *node* yang ada di sekitarnya.

Untuk menentukan *dominating-node* pada metode DBR adalah dengan menghitung kedekatan antar *node* dengan menggunakan matriks *adjacency* atau matriks kedekatan. Proses ini melibatkan paket HELLO yang dikirimkan ke *node* terdekat untuk memberitahukan *node* mana saja yang merupakan *node* tetangga. Setelah menentukan mana saja yang merupakan *node* tetangga, daftar *node* tetangga dikirimkan ke setiap *node* yang berdekatan dan setiap *node* mempersiapkan matriks *adjacency*. Hasil dari matriks ini digunakan untuk mencari *dominating node* dan *dominating set*.



Gambar 5. MANET dengan domination-set [8]



Pada gambar 5, *node* N3, N6 dan N10 merupakan *dominating node*. Misalnya, jika N1 merupakan *source node* dan N10 adalah *target node*. N10 dapat dicapai dengan menggunakan rute melalui *node* N3 dan N6 (N1-N3-N4-N6-N8-N10).

Keunggulan dari metode DBR adalah deteksi pada rute yang rusak / gagal dapat dilakukan dengan cepat. Hal ini dapat dilakukan karena *dominating node* terdekat melakukan identifikasi masalah dan memperbaiki secara lokal (dalam *dominating set*), atau paket data dapat sampai ke *node* tujuan dengan menggunakan rute lain yang memungkinkan. Cara yang lain adalah *dominating node* pada *dominating-set* yang terkait dengan rute yang rusak / gagal mengirimkan pesan *route failure report* ke *dominating node* yang lain agar paket data diarahkan menggunakan rute yang lain.

Sedangkan kelemahan dari metode DBR adalah komunikasi dari satu *dominating node* ke *dominating node* yang lain. Jadi meskipun ada jalur dengan rute yang lebih pendek, selama jalur tersebut tidak menghubungkan antar *dominating node* maka jalur tersebut tidak akan digunakan.

#### IV. PEMBAHASAN

Pada makalah ini kami menyajikan pembahasan berupa perbandingan mekanisme penanganan broadcast storm problem dengan karakteristik seperti yang ditampilkan pada tabel 2.

Kolom pertama berisikan nama dari metode yang diajukan untuk menangani *broadcast storm problem* yang dibahas dalam makalah ini. Kolom kedua memberikan catatan pengelompokan metode tersebut dalam kategori sesuai skema [2]. Kolom ketiga menunjukkan apakah mekanisme tersebut mempersyaratkan masing-masing *node* yang terlibat untuk mengetahui keberadaan dari *node-node* lainnya. Kolom keempat berisikan lingkungan uji coba yang digunakan oleh masing-masing metode. Kolom terakhir menyajikan protokol *routing* yang digunakan untuk uji coba dari masing-masing metode.

Dari survei yang kami lakukan, kami menemukan beragam metode yang sudah diajukan untuk menangani BSP. Metode-metode tersebut merujuk kepada pengkategorian penanganan BSP menurut Tseng, dkk. Metode yang diajukan juga berkembang tidak hanya berfokus pada penerapan 1 skema saja, tetapi juga menerapkan beberapa skema seperti pada *irresponsible forwarding* untuk i-AODV di [6] yang menerapkan skema *probabilistic-based* dan *distance-based*. Seperti yang terlihat pada kolom ketiga, dalam mengoptimalkan proses broadcasting, masing-masing metode memerlukan informasi dari *node-node* di sekitarnya. Semakin banyak jumlah *node* yang terlibat dalam pertukaran informasi, berdampak pada meningkatnya *packet overhead*. Dengan demikian dapat kita simpulkan bahwa overhead pada [3] dan [4] tentunya lebih tinggi daripada [5], [6], [8], [11].

Hampir semua metode yang dibahas dalam makalah ini melakukan uji coba proses *route discovery* ini pada protokol *routing* AODV (kolom 5). Menurut kami hal ini dikarenakan protokol *routing* ini banyak dikaji sehingga akan lebih mudah untuk melakukan perbandingan antar metode. Selain itu AODV juga dikenal sebagai protokol *routing on-demand* yang sebenarnya [9] dimana proses *route discovery* dan *route maintain* dilakukan hanya jika diperlukan. Khusus untuk EDB, karena mekanisme ini berorientasi pada pengoptimalan penggunaan energi pada broadcast, maka pada metode Ruiz dan Bouvry [11] tidak ditemukan penjelasan secara spesifik tentang protokol *routing* yang digunakan untuk uji coba, sehingga menurut kami mekanisme ini juga layak digunakan untuk mengoptimalkan proses *route discovery* pada protokol AODV.

Dari metode yang kami bahas, proses pengujian terbatas dilakukan di simulator dengan kasus uji yang berbeda, sehingga tidak dapat dibandingkan hasil uji dari masing-masing metode. Pengujian dengan simulator dapat dipahami karena keterbatasan perangkat uji yang dapat digunakan untuk melakukan pengujian di dunia nyata.

TABEL II  
PERBANDINGAN MEKANISME PENANGANAN BSP PADA MANET

Metode	Kategori sesuai Skema [2]	Memerlukan Informasi Keberadaan <i>Node</i> Lain	Lingkungan Uji Coba	Protokol <i>Routing</i> untuk Uji Coba
Adjusted probabilistic route discovery [3]	Probabilistic-based scheme	Ya, seluruh <i>node</i>	Simulasi	AODV, (DSR)
Dynamic counter-based broadcasting [4]	Counter-based scheme	Ya, seluruh <i>node</i>	Simulasi	AODV
Irresponsible forwarding [6]	Probabilistic-based scheme Distance-based scheme	Ya, <i>node</i> tetangga	Simulasi	AODV
Enhanced distance based [11]	Distance-based scheme	Ya, <i>node</i> tetangga	Simulasi	Tidak spesifik
Enhanced flooding scheme based on <i>node</i> position [5]	Location-based scheme	Ya, sebagian <i>node</i> tetangga	Simulasi	AODV
Domination-set based routing [8]	Cluster-based scheme	Ya, <i>node</i> tetangga	Simulasi	AODV

#### V. KESIMPULAN DAN PENGEMBANGAN LEBIH LANJUT

Beberapa mekanisme yang dapat digunakan untuk menangani *broadcast storm problem* pada proses *route discovery* pada protokol *routing* di MANET telah disajikan dalam makalah ini. Penanganan *broadcast storm*

*problem* secara tepat diperlukan untuk menjamin performa dari protokol *routing*. Mekanisme tersebut telah dibandingkan dengan kriteria yang sudah ditunjukkan pada tabel 2. Pengujian terhadap metode yang dibahas di makalah ini, semuanya dilakukan di simulator dengan demikian kita masih belum dapat mengetahui hasil pengujian pada lingkungan *testbed*. Melalui survei ini penulis mengharapkan dapat memberi kontribusi dalam pengembangan protokol *routing* lainnya dalam MANET yang lebih efektif, efisien dan komprehensif.

Penanganan *broadcast storm problem* tidak harus terpaku kepada satu skema saja. Pada beberapa penelitian sebelumnya, telah diajukan penggunaan metode dengan menggunakan lebih dari satu skema, seperti *irresponsible forwarding* yang sudah dibahas pada paper ini. Untuk penelitian lebih lanjut, dapat menggunakan metode dengan menerapkan skema baru yaitu menggabungkan keunggulan-keunggulan dari skema yang sudah ada. Sebagai contoh penggunaan skema *cluster-based* yang digabungkan dengan *distance-based*.

Pada skema *cluster-based* (metode DBR) diketahui memiliki kelemahan tidak dapat mengarahkan atau membentuk rute dengan jarak yang lebih pendek. Hal ini dikarenakan komunikasi pada metode DBR dilakukan antar *dominating node* pada masing-masing *cluster* yang berada pada rute yang sudah ditentukan. Sehingga meskipun terdapat jalur dengan rute yang lebih pendek, selama jalur tersebut tidak menghubungkan antar *dominating node* maka jalur tersebut tidak akan digunakan. Untuk mengatasi permasalahan ini, digunakanlah skema *distance-based* untuk menutupi kelemahan dari skema *cluster-based*. Metode *distance-based* digunakan untuk membelokkan rute paket data supaya tidak melalui *dominating-node*, sehingga paket data dapat menggunakan rute terpendek yang tersedia sesuai skema *distance-based*.

Tetapi tidak selamanya hanya sisi positif yang didapat dari metode penggabungan ini. Dengan menggunakan metode penggabungan skema *cluster-based* dengan metode *distance-based* memiliki kelemahan, antara lain penentuan rute yang dipengaruhi oleh topologi jaringan itu sendiri. Ketika topologi jaringan dalam keadaan stabil atau tidak terjadi banyak perubahan, maka penentuan rute bisa dilakukan dengan cepat karena posisi tiap *node* cenderung tidak mengalami perubahan. Tetapi ketika topologi jaringan mengalami perubahan dikarenakan oleh adanya pengurangan atau penambahan *node* maka penentuan rute terpendek dapat membutuhkan waktu yang lebih lama. Hal ini dikarenakan adanya proses untuk melakukan pemindaian ulang terhadap posisi terakhir *node*.

#### DAFTAR PUSTAKA

- [1] O. K. Tonguz, N. Wisitpongphan, J. S. Parikh, F. Bai, P. Mudalige, V. K. Sadekar, "On the Broadcast Storm Problem in Ad hoc Wireless Networks", dalam *Proceedings of 3<sup>rd</sup> International Conference on Broadband Communications, Networks, and Systems*, 2006, hal. 1-11.
- [2] Y. C. Tseng, S.Y. Ni, Y.S. Chen, J.P. Sheu, "The broadcast storm problem in mobile ad hoc network", *Wireless Networks*, vol. 8, no. 2-3, hal. 153-167, 2002.
- [3] J. D. Abdulai, M. Ould-Khaoua, L. M. Mackenzie, "Adjusted probabilistic route discovery in mobile ad-hoc networks", *Computers and Electrical Engineering*, vol. 35, no. 1, hal. 168-182, Jan. 2009.
- [4] M. B. Yassein, S. F. Nimer, A. Y. Al-Dubai, "A new dynamic counter-based broadcasting scheme for Mobile Ad hoc Networks", *Simulation Modelling Practice and Theory*, vol. 19, no. 1, hal. 553-563, Jan. 2011.
- [5] S. Gupta, A. Mathur, "Enhanced Flooding Scheme for AODV Routing Protocol in Mobile Ad hoc Networks", dalam *Proceedings of 2014 International Conference on Electronic Systems, Signal Processing, and Computing Technologies (ICESC)*, Nagpur, India, 2014, hal. 316-321.
- [6] S. Panichpapiboon, G. Ferrari, "Irresponsible forwarding", dalam *Proceedings of 8th International Conference on Intelligent Transport System Telecommunication (ITST)*, Phuket, Thailand, 2008, hal. 311-316.
- [7] A. Gorrieri, G. Ferrari, "Irresponsible AODV routing", *Vehicular Communications*, vol. 2, no. 1, hal. 47-57, Jan. 2015.
- [8] Preetha K. G., A. Unnikrishnan, "Improving the Routing Performance of Mobile Ad hoc Networks Using Domination Set", dalam *International Conference on Information and Communication Technologies (ICICT 2014)*, Kochi, India, 2015, hal. 1209-1215.
- [9] Y. Yen, H. Chang, R. Chang, H. Chao, "Routing with adaptive path and limited flooding for mobile ad-hoc networks", *Computers and Electrical Engineering*, vol. 36, no. 2, hal. 280-290, Mar. 2010.
- [10] Jie Wu, Hailan Li. A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks, *Telecommunication Systems*, vol. 18, hal. 13-36, September 2001.
- [11] P. Ruiz, P. Bouvry, "Enhanced distance based broadcasting protocol with reduced energy consumption", dalam *Proceedings of 2010 International Conference on High Performance Computing and Simulation (HPCS)*, Caen, Perancis, 2010, hal. 249-258.