

OPTIMASI ROUTING PADA PROTOKOL AODV_EXT DENGAN MENGGUNAKAN LINK EXPIRATION TIME (LET)

Wiwien Windianto¹⁾, Supeno Djanali²⁾, Muchammad Husni³⁾

^{1, 2, 3)}Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Gedung Teknik Informatika, Jl. Teknik Kimia – Surabaya, Indonesia
e-mail: wien.s2its@gmail.com¹⁾, supeno@its.ac.id²⁾, husni@its-sby.edu³⁾

ABSTRAK

Proses flooding pada MANET merupakan suatu teknik pada fase route discovery yang digunakan oleh source node untuk mengirimkan route request packet (RREQ) pada node-node disekitarnya. Proses ini seringkali menimbulkan transmisi ulang yang tidak perlu dan menyebabkan congestion dan packet collision dimana fenomena ini dikenal dengan broadcast storm. Protokol AODV_EXT merupakan pengembangan dari protokol AODV yang menerapkan algoritma pengaturan RREQ forwarding pada intermediate node. Pada proses flooding tersebut maka RREQ packet akan diteruskan oleh intermediate node ke node-node selanjutnya apabila densitas node tetangga memenuhi batasan yang ditetapkan. Metode ini mengurangi terjadinya kemacetan jaringan akibat adanya transmisi ulang.

Pergerakan node memiliki peran penting pada MANET. Mobilitas node sering menyebabkan perubahan pada topologi jaringan serta putusnya jalur yang menyebabkan proses route discovery perlu dilakukan kembali. Maka pada penelitian ini diusulkan suatu optimasi pemilihan rute pada jaringan MANET dengan mempertimbangkan link lifetime yang ada akibat pergerakan node. Algoritma Link Expiration Time (LET) digunakan untuk memperkirakan lamanya dua buah node untuk dapat saling berhubungan. LET dihitung berdasarkan jangkauan transmisi serta kecepatan relatif antara dua buah node. Routing protocol yang menggunakan nilai LET akan dapat menemukan rute yang paling stabil. Berdasarkan hal itu maka algoritma LET akan diterapkan pada protokol AODV_EXT dengan tujuan dapat meningkatkan performance pada protokol tersebut.

Network Simulator 2 (NS2) digunakan untuk mengevaluasi modifikasi dari protokol AODV_EXT dengan menggunakan Link Expiration Time. Hasil penelitian menunjukkan bahwa protokol R-AODV_EXT memiliki kinerja yang lebih baik daripada AODV_EXT untuk parameter packet delivery ratio (PDR), throughput, number of packets dropped, dan average end-to-end delay.

Kata Kunci: Link Expiration Time, Rebroadcast Probability, Route Discovery.

ABSTRACT

Flooding is the simplest technique on the route discovery phase which is used by source nodes to broadcast packets to neighbouring nodes. This often leads to unnecessary retransmission, causing congestion and packet collisions in the network, a phenomenon called a broadcast storm. AODV_EXT protocol is the development of AODV protocol which implements RREQ forwarding algorithm settings performed by the intermediate nodes. During the flooding process the RREQ packet will be forwarded by the intermediate node to the neighbouring node when the density of neighboring nodes meet the restrictions set. This method reduces network congestion due to redundant transmission.

Nodes mobility play a crucial role in Mobile Ad Hoc Network (MANET). Mobility often lead to changes in the network topology as well as link breakage. This study proposed an optimization of route selection for MANET by considering the link lifetime due to the movement of nodes. Link Expiration Time (LET) algorithm is used to estimate the length of two nodes to be interconnected. Routing protocol that utilizes the LET values will be able to find the most stable path. Therefore the Link Expiration Time algorithm will be applied to AODV_EXT protocol with the aim to improve the performance of the protocol.

Network Simulator 2 (NS2) has been used to evaluate the modified AODV_EXT routing protocol using the Link Expiration Time. The results showed that R-AODV_EXT improves performance in term of packet delivery ratio (PDR), throughput, number of packets dropped, and average end-to-end delay.

Keywords: Link Expiration Time, Rebroadcast Probability, Route Discovery.

I. PENDAHULUAN

MOBILE Ad Hoc Network (MANET) merupakan sekumpulan *wireless mobile nodes* yang berhubungan satu sama lain tanpa adanya dukungan infrastruktur yang tetap dan administrasi terpusat. Jaringan tersebut dapat dibentuk setiap saat dan pada berbagai tempat apabila dibutuhkan. Teknologi ini dapat digunakan pada kondisi darurat, operasi militer, operasi penyelamatan saat terjadinya bencana alam, dan sebagainya [1].

Setiap *node* pada jaringan menggunakan baterai sebagai sumber energinya. Sehingga kelangsungan jaringan dipengaruhi oleh ketersediaan energi yang ada. Energi baterai pada *node* akan terpakai karena transmisi ataupun penerimaan data yang dilakukan.

Node-node pada MANET dapat berhubungan satu sama lain apabila mereka berada pada jangkauan transmisinya. Apabila bergerak menjauh diluar jangkauan transmisi maka *link* antara dua buah *node* dapat terputus. Dengan terputusnya suatu *link*, maka akan menyebabkan terputusnya sebuah rute yang ada. Juga dengan terputusnya *link* tertentu menyebabkan proses *route discovery* perlu dilakukan kembali. Sehingga mobilitas pada *node* merupakan masalah yang perlu untuk diperhatikan apabila ingin membentuk jaringan yang stabil.

Tujuan utama dari *routing protocol* pada MANET adalah untuk memaksimalkan *throughput* jaringan, memaksimalkan efisiensi energi, memaksimalkan *lifetime* jaringan (*network lifetime*), dan meminimalkan *delay*. *Throughput* jaringan biasanya diukur dengan *packet delivery ratio* (PDR) sedangkan konsumsi energi diukur oleh besarnya *routing overhead* yang merupakan jumlah atau ukuran dari *routing control packets* [3].

Pada penelitian oleh [5] dilakukan algoritma pembatasan terhadap RREQ *forwarding* oleh *intermediate node* pada protokol AODV yang didasarkan pada jumlah atau densitas *node* tetangga. Algoritma tersebut ternyata dapat mengurangi terjadinya *network congestion* akibat adanya transmisi ulang (*redundant transmission*) yang tidak diperlukan. Protokol tersebut kemudian diberi nama dengan AODV_EXT.

Pada MANET, *mobile node* dapat bergerak setiap saat pada arah yang acak (*random*). Dengan kata lain topologi MANET mengalami perubahan secara dinamis. Dari penelitian pada [9] didapatkan suatu algoritma untuk memperkirakan lamanya dua buah *node* untuk dapat saling berhubungan sebagai akibat adanya pergerakan *node-node* tersebut. Algoritma tersebut disebut dengan *Link Expiration Time* (LET). LET dihitung berdasarkan jangkauan transmisi radio pada *node*, informasi posisi, arah pergerakan serta kecepatan relatif antara dua buah *node*. Dengan menggunakan nilai LET maka pada *fase route discovery* yang menghasilkan beberapa alternatif rute dapat dipilih rute mana yang paling stabil untuk digunakan sebagai jalur pengiriman paket data.

Maka dari hal yang tersebut di atas diusulkan suatu algoritma *routing* yang mempertimbangkan pergerakan *node* pada protokol AODV_EXT dengan tujuan untuk mendapatkan rute yang paling stabil untuk pengiriman data. Algoritma *Link Expiration Time* digunakan untuk menghasilkan perbaikan pada protokol tersebut.

II. PROTOKOL ROUTING PADA MANET

Terdapat dua macam *routing protocol* pada MANET yaitu *proactive routing protocol* dan *reactive on-demand routing protocol*. *Proactive routing* akan membentuk dan mempertahankan keberadaan suatu rute tanpa memperhatikan apakah rute tersebut masih dibutuhkan atau tidak. Sedangkan *reactive routing* hanya membentuk suatu rute selama rute tersebut masih diperlukan. Oleh karena topologi jaringan berubah secara dinamis, maka *reactive routing* lebih disukai daripada *proactive routing*. Pada sisi yang lain *reactive routing* secara signifikan dapat mengurangi konsumsi energi, tetapi *proactive routing* memiliki kemampuan untuk memberikan *quality of service* (QoS) yang tinggi pada pengiriman data [2].

Pada bab ini akan dijelaskan protokol-protokol *routing* yang akan dipakai pada penelitian.

A. Protokol AODV

Ad hoc On Demand Distance Vector (AODV) [7] merupakan *routing protocol* yang didisain untuk *mobile ad hoc network* (MANET). AODV dapat digunakan baik untuk *unicast* maupun *multicast routing*. Protokol ini merupakan *on demand routing*, dimana algoritma ini akan membangun rute antara *node* hanya apabila diinginkan oleh *source node*. AODV memelihara rute tersebut sepanjang masih dibutuhkan oleh *source node*. AODV menggunakan *sequence number* untuk mencegah terjadinya *loop* dan memastikan bahwa rute yang dihasilkan memiliki informasi *routing* yang paling *update*.

AODV menciptakan suatu rute dengan menggunakan *route discovery process* yang terdiri dari *route request* (RREQ) dan *route reply* (RREP). Paling sedikit satu rute harus terbentuk sebelum *source node* mengirimkan data ke *destination node*. Ketika *source node* menginginkan suatu rute menuju *destination node* tetapi belum mempunyai rute yang benar ke arah *destination node*, maka *source node* akan menginisialisasi *route discovery process* untuk menemukan rute ke *destination node*. *Source node* akan *broadcast* paket RREQ menuju *node* tetangganya. RREQ paket berisi *source address*, *destination address*, *hop counter*, *source and destination sequence number*, dan *broadcast ID*. Nilai *Broadcast ID* akan bertambah satu setiap suatu *source node* mengirimkan RREQ yang baru dan digunakan sebagai identifikasi sebuah paket RREQ.

Jika *intermediate node* yang menerima RREQ memiliki informasi rute menuju *destination node*, maka *node* tersebut akan mengirim paket RREP kembali menuju *source node*. Tetapi jika tidak mengetahui maka *node* tersebut akan meneruskan transmisi RREQ ke *node* tetangganya setelah menambahkan nilai *hop counter*. *Node* yang menerima RREQ dengan nilai *source address* dan *broadcast ID* yang sama dengan RREQ yang diterima sebelumnya akan membuang RREQ tersebut.

Source sequence number digunakan oleh suatu *node* untuk memelihara informasi yang valid mengenai *reverse path* (jalur balik) menuju ke *source node*. Pada saat RREQ mengalir menuju *destination node* yang diinginkan, dia akan menciptakan *reverse path* menuju ke *source node*. Apabila *node* yang memiliki informasi rute menuju *destination node* menerima RREQ, maka *node* tersebut akan membandingkan nilai *destination sequence number*

yang dia miliki dengan nilai *destination sequence number* yang ada di RREQ. Jika nilai *destination sequence number* yang ada di RREQ lebih besar dari nilai yang dimiliki oleh *node* maka paket RREQ tersebut akan *broadcast* kembali ke *node* tetangganya. Sebaliknya jika nilai *destination sequence number* yang ada di RREQ lebih kecil atau sama dengan nilai yang ada di *node* maka *node* tersebut akan mengirim *route reply* (RREP) menuju *source node* dengan menggunakan *reverse path* yang telah dibentuk oleh RREQ. *Intermediate node* yang menerima RREP akan meng-*update* informasi *timeout* (masa aktif rute) jalur yang telah diciptakan. Informasi rute *source* ke *destination* akan dihapus apabila waktu *timeout* habis.

Di dalam AODV setiap *node* bertanggung jawab untuk memelihara informasi rute yang telah disimpan di dalam *routing table*-nya. Pada saat pengiriman data apabila terjadi perubahan topologi yang mengakibatkan suatu *node* tidak dapat dituju dengan menggunakan informasi rute yang ada di *routing table*, maka suatu *node* akan mengirimkan *route error packet* (RRER) ke *node* tetangganya (*predecessor node*) dan *node* tetangganya akan mengirim kembali RRER demikian seterusnya hingga menuju *source node*. Setiap *node* yang memperoleh RRER ini akan menghapus informasi yang mengalami error di dalam *routing table*-nya. Kemudian *source node* akan melakukan *route discovery process* kembali apabila rute ke *destination node* masih diperlukan.

B. Protokol AODV_EXT

Pada AODV, saat *route discovery process* maka *source node* mengirimkan (*broadcast*) RREQ paket ke semua *node* tetangganya. Proses ini disebut dengan *flooding* dan seringkali menimbulkan transmisi ulang yang tidak perlu dan menyebabkan *congestion* dan *packet collision* dimana hal ini dikenal dengan *broadcast storm*.

Pada penelitian yang dilakukan oleh [5] yang diterapkan pada protokol AODV, saat RREQ dikirimkan maka berdasarkan kondisi tertentu hanya beberapa *intermediate node* yang akan meneruskan paket. *Intermediate node* yang telah menerima RREQ *message* akan memeriksa apakah densitas atau jumlah *node* tetangga memenuhi persyaratan. Apabila memenuhi, *intermediate node* tersebut akan meneruskan paket. Apabila tidak maka paket dibuang (*drop*). Jumlah minimum *node* tetangga yang menjadi persyaratan tersebut dapat bervariasi tergantung dari aplikasi dan *quality of service* yang diinginkan. Algoritma RREQ *message forwarding* yang digunakan dapat dilihat pada Gambar 1 di bawah. Metode tersebut ternyata dapat mengurangi kemacetan jaringan (*network congestion*) yang disebabkan oleh transmisi ulang (*redundant transmission*). Protokol tersebut diberi nama AODV_EXT.

```

//Any node  $F_i, i = 1, 2, \dots, n$  receiving the RREQ message
IF  $F_i \neq S$  and  $F_i \neq D$  (i.e  $F_i$  is intermediate node)
    Node  $F_i$  resolves its neighbourhood density  $\beta_i$ 
    IF  $\beta_i \leq d$  THEN
        Forward the RREQ message
    ELSE
        //Calculate message forwarding probability  $P_i$  at node  $F_i$ 

$$P_i = \frac{100}{\beta_i} * (d * C_f) \quad \text{for } 0 < C_f \leq 1$$

        IF  $R < P_i$  THEN
            Forward the RREQ message
        ELSE
            Ignore and Drop the RREQ message
        END IF
    END IF
END IF
END IF

```

Gambar 1. Algoritma RREQ *message forwarding* oleh *intermediate node* pada AODV_EXT [5]

dimana:

- n = Jumlah *node* dalam jaringan
- F_i = *Node* yang menerima RREQ
- P_i = Probabilitas RREQ *Packet Forwarding*
- β_i = Jumlah atau densitas *node* tetangga
- d = *Minimum number of neighbouring nodes*

C_f = Control factor yang berfungsi untuk mengatur nilai probabilitas P_i
 R = Nilai acak (random number) diantara 0 dan 100

III. ALGORITMA LINK EXPIRATION TIME

Link Expiration Time (LET) merupakan algoritma untuk memperkirakan lama waktu koneksi atau waktu kadaluwarsa antara dua buah node pada MANET. Jika parameter-parameter yang ada diantara dua buah node diketahui seperti kecepatan, arah pergerakan dan jangkauan propagasi radio, maka dapat ditentukan lamanya waktu antara dua buah node dapat saling berhubungan.

Apabila ada dua buah node i dan j dengan jangkauan transmisi r , kecepatan v_i dan v_j , koordinat (x_i, y_i) dan (x_j, y_j) , serta besar sudut arah pergerakan θ_i dan θ_j [9], maka nilai LET dapat diperkirakan dengan menggunakan persamaan 1.

$$LET_{ij} = \frac{-(ab + cd) + \sqrt{(a^2 + c^2)r^2 - (ad - bc)^2}}{a^2 + c^2} \tag{1}$$

dimana:

$$\begin{aligned} a &= v_i \cos \theta_i - v_j \cos \theta_j \\ b &= x_i - x_j \\ c &= v_i \sin \theta_i - v_j \sin \theta_j \\ d &= y_i - y_j \end{aligned}$$

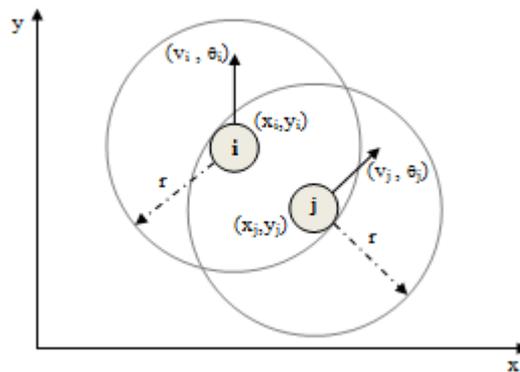
Arah pergerakan node θ dapat dihitung dengan menggunakan persamaan 2 dan 3.

$$\tan \theta = \frac{y' - y}{x' - x} \tag{2}$$

$$\theta = \arctan \frac{y' - y}{x' - x} \tag{3}$$

Parameter-parameter tersebut (kecepatan, arah pergerakan, dan posisi node) diperoleh dari GPS yang ada pada node. Semakin besar perbedaan sudut pergerakan diantara dua buah node maka semakin kecil nilai Link Expiration Time, sebaliknya apabila perbedaan sudut semakin kecil maka nilai LET semakin besar. Dari parameter di atas dapat diketahui apabila kecepatan relatif antara dua buah node adalah nol atau dengan kata lain $v_i = v_j$ dan $\theta_i = \theta_j$, maka koneksi antara dua buah node dapat dilakukan terus-menerus atau harga LET menjadi ∞ .

Ilustrasi dari persamaan 1 sampai 3 dapat dilihat pada Gambar 2 di bawah.



Gambar 2. Posisi, kecepatan dan arah pergerakan dari dua buah mobile node [4]

Selain LET maka hal yang perlu dilakukan adalah menghitung Path Expiration Time (PET). Merupakan jangka waktu maksimum (maximum lifetime) yang ada pada suatu rute. Nilainya ditentukan dengan menggunakan persamaan 4. Nilai PET tersebut didapat dari nilai minimum dari beberapa LET yang ada pada suatu rute yang terbentuk.

$$PET = \underset{1 \leq n \leq k}{\text{Min}} (LET_n) \quad (4)$$

dimana:

n = nomor urut *link*

k = jumlah *link* pada suatu rute

Apabila beberapa rute yang terbentuk selama *route discovery process* telah mencapai *node* tujuan, maka *node* tujuan akan memilih rute yang akan dipakai berdasarkan harga terbesar (*maximum*) dari beberapa nilai PET yang ada.

Proses pemilihan tersebut dinyatakan pada persamaan 5.

$$\text{Most Reliable Path} = \underset{1 \leq h \leq m}{\text{Max}} (PET_h) \quad (5)$$

dimana:

h = nomor urut *path* (rute)

m = jumlah *path* yang terbentuk

Pada pemakaian algoritma ini apabila ada beberapa rute yang sampai pada *destination node* selama proses *route request*, maka *destination node* akan memilih rute yang memiliki *Path Expiration Time* paling besar walaupun ada rute lain yang memiliki jumlah hop lebih kecil.

IV. PENELITIAN TERKAIT

Ad hoc On Demand Multipath Distance Vector (AOMDV) adalah pengembangan dari protokol *routing* AODV untuk menghitung jalur *multiple loop-free* dan *link disjoint*. Protokol tersebut merupakan *multi-path routing protocol* dimana dalam cara kerjanya protokol akan mengalihkan pengiriman data pada *path* yang lain apabila jalur yang digunakan mengalami kegagalan. Peneliti pada [6] mengusulkan penambahan algoritma LET pada AOMDV yang kemudian diberi nama S-AOMDV. Penerapan algoritma LET untuk mencari jalur yang paling stabil pada S-AOMDV dilakukan saat pengiriman beberapa RREP dari *destination node* ke *source node*. Saat *source node* menerima RREP, *source node* akan menghitung stabilitas rute dan mulai mengirimkan paket data sambil menerima RREP berikutnya. *Source node* akan membandingkan stabilitas rute untuk setiap RREP dan memindahkan rute ke rute lain apabila memiliki nilai stabilitas lebih tinggi. Penggunaan rute cadangan tidak menunggu hingga rute yang sedang digunakan rusak. Hasil penelitian menunjukkan bahwa S-AOMDV memiliki nilai *Packet Delivery Ratio* dan rata-rata *Throughput* yang lebih baik serta *Packet Drop* yang lebih rendah dari AOMDV. Tapi peningkatan kinerja yang terjadi masih kecil atau kurang signifikan.

V. METODE PENELITIAN

Pada penelitian ini digunakan *Network Simulator 2.35* (NS-2.35) untuk simulasi. Guna mendukung hal ini maka dilakukan pemilihan baik dari segi *hardware* maupun *software*. Oleh karena *Network Simulator 2* bekerja di lingkungan sistem operasi Linux, maka dipilih salah satu *distro* yaitu Ubuntu 14.04 LTS. Adapun *hardware* dan *software* yang digunakan ditunjukkan sebagai berikut:

- Processor Intel® Core™ 2 Duo CPU
- Memory 2048 MB RAM
- Sistem Operasi Ubuntu 14.04 LTS
- Simulasi menggunakan *Network Simulator 2.35* (NS-2.35)
- AWK *software*
- Microsoft Excel 2007

TABEL I
PARAMETER SIMULASI

Parameter	Keterangan
Protokol	AODV, AODV_EXT, R-AODV_EXT
Area	800 m × 800 m
Waktu Simulasi	80 s
Jumlah Node	10, 20, 30, 40, 50, 100, 150, dan 200
Kecepatan Node	1 m/s sampai 40 m/s
Pause Time	0 s
Pergerakan Node	Random Waypoint
Maksimum Koneksi	10
Channel	Wireless
Propagation	TwoRayGround
Jangkauan Transmisi	250 m
MAC Layer	IEEE 802.11
Antena	OmniAntenna
Pola Trafik	Constant Bit Rate (CBR)
Transport Type	UDP
Ukuran Paket	512 Bytes = 4.096 bits
Kecepatan Paket	1 Paket per Detik = 4.096 bps
Transmission Interval	1 s
Seed	1,0
PROTOKOL AODV_EXT DAN R-AODV_EXT	
Minimum number of neighbouring nodes (d)	$d = (N - 1) 0,8 \frac{\pi r^2}{A}$

Simulasi jaringan MANET yang digunakan untuk analisa kinerja dari *routing protocol* AODV, AODV_EXT maupun R-AODV_EXT menggunakan *area* 800 m × 800 m dengan jumlah *node* yang divariasikan dari 10 hingga 50 *node*, kemudian dilanjutkan untuk total *node* yang relatif besar yaitu 100, 150, dan 200. Setiap *node* mempunyai jangkauan transmisi sebesar 250 m. Model pergerakan yang digunakan adalah *random waypoint*. Pada model pergerakan tersebut setiap *node* pada awal simulasi akan diam selama *pause time* tertentu, kemudian mulai bergerak menuju titik tujuan yang dipilih secara acak dengan kecepatan tertentu dari 1 m/s hingga kecepatan maksimal (1, *max_speed*). Setelah *node* mencapai titik tujuan, akan kembali diam selama *pause time* yang ditentukan untuk kemudian bergerak lagi menuju titik tujuan yang lain. Proses ini akan terus berulang hingga simulasi berakhir. Kecepatan minimal 1 m/s dan maksimal sebesar 40 m/s serta *pause time* 0 detik digunakan pada penelitian ini. Jangka waktu simulasi ditetapkan sebesar 80 s. Sedangkan nilai *d* ditentukan berdasarkan algoritma tertentu yang ditunjukkan pada persamaan 6. Nilai parameter simulasi dapat dilihat pada Tabel I.

Untuk variabel *minimum number of neighbouring nodes* (d), maka dari penelitian yang dilakukan oleh [10] nilai tersebut dapat ditentukan yang didasarkan pada jumlah total *node* pada jaringan MANET, jangkauan transmisi serta luas area yang digunakan. Besarnya nilai *d* tersebut diperoleh dari persamaan 6 dibawah.

$$d = (N - 1) 0,8 \frac{\pi r^2}{A} \tag{6}$$

dimana:

- N = Jumlah *node* dalam jaringan
- r = Jangkauan transmisi
- A = Luas area jaringan *ad hoc*

Dari persamaan 6 tersebut dapat dilihat bahwa nilai *minimum number of neighbouring nodes* (d) akan berbeda-beda pada tiap-tiap total *node* pada jaringan. Harganya semakin besar pada jumlah total *node* serta jangkauan transmisi yang semakin tinggi, dan nilainya semakin kecil pada *area* yang lebih luas.

Agar algoritma *Link Expiration Time* dapat bekerja pada protokol routing, maka perlu pula modifikasi terhadap RREQ *message*. Saat suatu *source node* mengirimkan RREQ, maka RREQ *packet* tersebut harus dapat menyimpan informasi lokasi, kecepatan serta arah pergerakan *node* [8]. Sehingga *node* berikutnya (*next hop*) dapat memperkirakan lamanya waktu koneksi atau LET dari *node* tersebut terhadap *source node*. Jika *node* B merupakan *next hop* dari *node* A, maka *node* A akan menambahkan informasi lokasi, kecepatan serta arah pergerakannya pada paket RREQ sehingga *node* B dapat menghitung nilai LET antara *node* A dan *node* B.

Hop Count	Broadcast ID	Destination IP Address	Destination Sequence Number	Source IP Address	Source Sequence Number	X Position	Y Position	Speed	Moving Direction	Path Expiration Time
-----------	--------------	------------------------	-----------------------------	-------------------	------------------------	------------	------------	-------	------------------	----------------------

Gambar 3. Modifikasi pada AODV Route Request (RREQ) Message

Pada saat RREQ telah mencapai *destination node*, maka paket tersebut juga diharapkan dapat membawa nilai PET dari rute yang telah dilaluinya. Sehingga untuk kemudian *destination node* dapat memilih harga PET terbesar dari beberapa RREQ yang sampai kepadanya. Untuk itu perlu ditambahkan informasi nilai PET pada RREQ message.

Penghitungan nilai PET dapat dijelaskan dengan metode sebagai berikut: saat suatu *source node* S mengirimkan RREQ ke *node* berikutnya A, maka setelah sampai di *node* A nilai PET disamakan dengan nilai LET_{SA} dan nilai PET tersebut disimpan pada RREQ message. Apabila berlanjut sampai ke *node* B maka nilai PET yang dibawa oleh RREQ akan dibandingkan dengan harga LET_{AB} . Apabila LET_{AB} lebih kecil dari PET, maka nilai PET akan diganti dengan nilai LET_{AB} . Apabila tidak, maka nilai PET tetap. Proses perbandingan nilai ini akan terus berlanjut dari *intermediate node* ke *intermediate node* berikutnya hingga RREQ mencapai *destination node* dengan membawa nilai PET atau minimum LET dari rute yang dilaluinya. Metode ini sama dengan algoritma untuk mencari nilai minimum pada suatu deretan angka.

Modifikasi terhadap RREQ message dapat dilihat pada Gambar 3. Informasi yang perlu ditambahkan ditunjukkan oleh kotak dengan *background* warna gelap.

Untuk mengetahui kinerja dari ketiga protokol routing tersebut, maka ditentukan beberapa parameter yang digunakan untuk evaluasi pada penelitian ini yaitu:

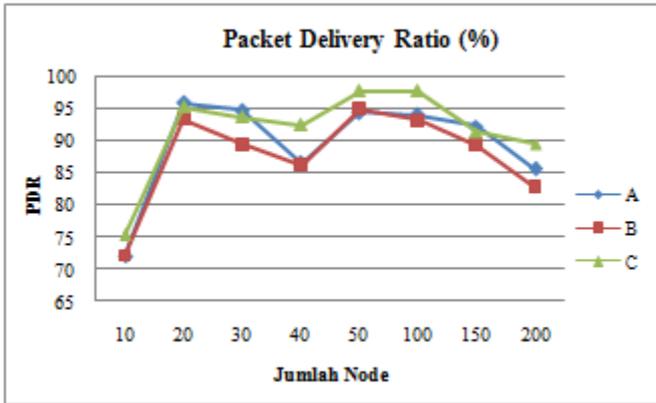
- a. *Packet Delivery Ratio (PDR)*
Packet Delivery Ratio merupakan perbandingan antara jumlah paket data yang diterima oleh *destination node* dengan jumlah paket data yang dikirim oleh *source node*. PDR dinyatakan dalam persen.
- b. *Throughput*
 Menyatakan ukuran seberapa cepat suatu jaringan dalam mengirimkan atau menerima data. *Throughput* adalah jumlah paket data yang berhasil diterima oleh *destination node* setiap detiknya. Dinyatakan dalam satuan *kilobits per second (kbps)*.
- c. *Number of Packets Dropped*
 Merupakan jumlah paket data yang tidak berhasil dikirimkan ke tujuan selama transmisi.
- d. *Average End-to-End Delay*
 Merupakan waktu rata-rata yang dibutuhkan oleh paket data saat dikirimkan oleh *source node* dengan waktu penerimaan paket data oleh *destination node*. Dinyatakan dalam *millisecond (ms)*.
- e. *Routing Overhead*.
 Merupakan perbandingan antara total jumlah paket *routing* yang dikirim dengan total jumlah paket data yang diterima. *Routing Overhead* dinyatakan dalam persen.

VI. HASIL DAN PEMBAHASAN

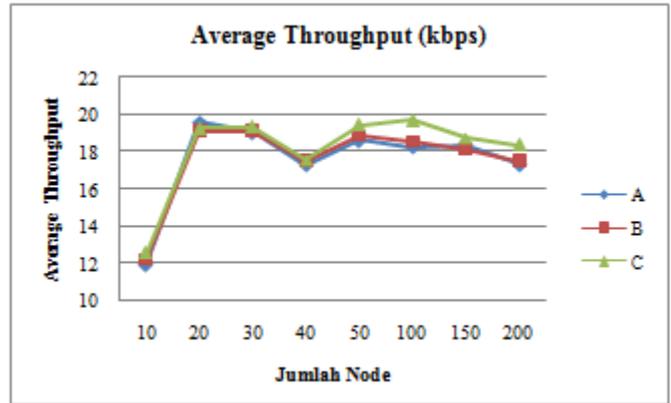
Pada bab ini dipaparkan hasil uji coba dari sistem yang telah dirancang serta dijelaskan pada bab sebelumnya. Hasil pengukuran terhadap *trace file* hasil *output* NS2 dilakukan dengan AWK untuk kemudian data ditabulasikan dan disajikan dalam bentuk grafik.

Metode *Link Expiration Time* yang diterapkan bertujuan untuk mendapatkan rute yang stabil atau handal (*reliable*) pada AODV_EXT, sehingga protokol yang diusulkan diberi nama *Reliable AODV_EXT* atau R-AODV_EXT. Untuk mempersingkat nama protokol yang digunakan dalam penyajian grafik hasil uji coba maka nama protokol routing diganti dengan urutan (A=AODV, B=AODV_EXT, C= R-AODV_EXT).

Untuk parameter *Packet Delivery Ratio (PDR)* maka R-AODV_EXT mempunyai nilai yang lebih baik dari pada AODV_EXT pada nilai *control factor (Cf)* yang berbeda-beda untuk tiap-tiap total *node*. Begitu pula untuk parameter *Throughput* maupun *Number of Packets Drop*. Hal ini ialah karena nilai Cf dapat diubah-ubah antara 0 dan 1 dan berfungsi untuk memperkecil nilai probabilitas dari paket RREQ apakah diteruskan (*forward*) atau di buang (*drop*). Apabila nilainya kurang dari 1 maka nilai probabilitas P_i akan diperkecil yang mana berarti memperkecil tingkat kemungkinan suatu paket RREQ untuk diteruskan (*forward*).



Gambar 4. Grafik jumlah node terhadap PDR

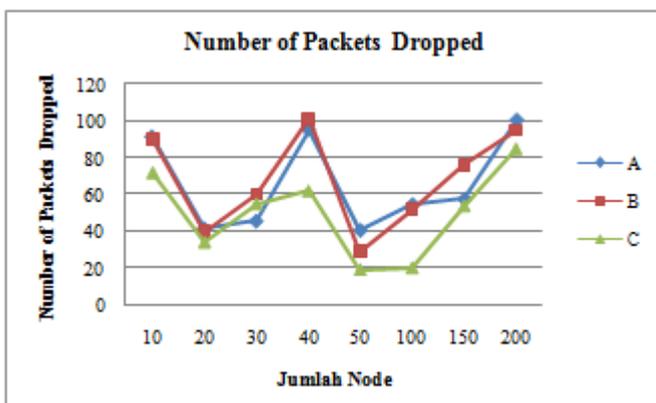


Gambar 5. Grafik jumlah node terhadap Throughput

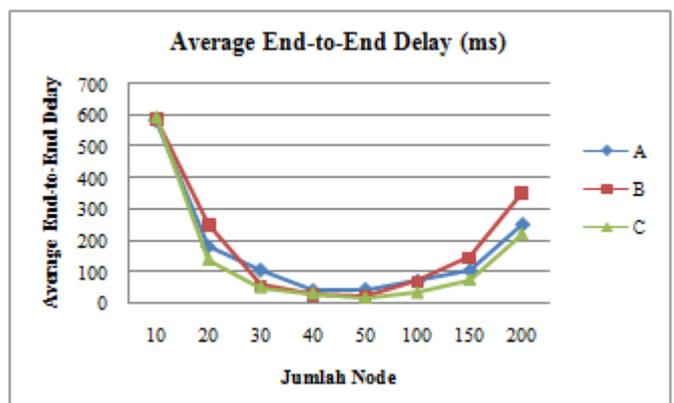
Hasil simulasi untuk parameter PDR, *Throughput* serta *Number of Packets Drop* dapat dilihat pada Gambar 4 sampai 6. Peningkatan yang terjadi pada R-AODV_EXT ialah karena protokol tersebut dapat mencari rute yang paling stabil atau handal (*reliable*) dari beberapa rute yang terbentuk, sehingga kejadian terputusnya *link* akan berkurang yang berakibat juga berkurangnya paket yang terbuang (*Number of Packets Drop*). Dengan berkurangnya paket yang terbuang maka terjadi peningkatan pada PDR serta *Throughput*.

End-to-end delay adalah jumlah waktu yang digunakan oleh sebuah paket ketika dikirim oleh sebuah *node* dan diterima di *node* tujuan. *End-to-end delay* merupakan jumlah dari waktu pengiriman, propagasi, proses dan antrian dari suatu paket pada setiap *node* di jaringan. Faktor pertama yang mempengaruhi *End-to-end delay* adalah waktu untuk menemukan rute. Hal ini berguna sebelum pesan yang akan dikirim, *source node* harus mengetahui terlebih dahulu jalur atau rute untuk mencapai *destination node*. Faktor lain yang mempengaruhi *End-to-end delay* adalah *delay* proses. Ketika *intermediate node* menerima sebuah pesan, *node* tersebut akan menganalisa header untuk mengetahui untuk siapa paket ditujukan, dan kemudian mengecek *node* untuk menentukan kemana harus meneruskan paket tersebut. Dapat dilihat pada gambar 7 bahwa nilai *Average End-to-End Delay* pada R-AODV_EXT cenderung lebih rendah dibandingkan AODV_EXT, hal ini disebabkan karena pada saat pengiriman data pada protokol R-AODV_EXT tidak banyak terjadi *route discovery* ulang yang menjadikan paket-paket tidak banyak menunggu pada antrian *buffer* sehingga waktu pengiriman bisa lebih cepat.

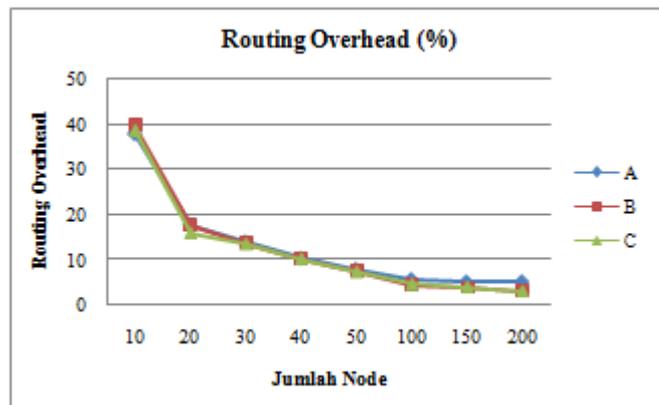
Pada gambar 8 ditunjukkan bahwa protokol R-AODV_EXT mempunyai nilai *Routing Overhead* yang lebih kecil dari AODV_EXT tapi selisihnya kurang signifikan. Untuk jumlah *node* yang sama R-AODV_EXT memiliki *Routing Overhead* yang relatif lebih rendah daripada AODV_EXT karena R-AODV_EXT memiliki kemampuan untuk memilih rute yang paling stabil sehingga melakukan prosedur *route discovery* lebih sedikit dibandingkan AODV_EXT.



Gambar 6. Grafik jumlah node terhadap Number of Packets Drop



Gambar 7. Grafik jumlah node terhadap Average End-to-End Delay



Gambar 8. Grafik jumlah node terhadap Routing Overhead

VII. KESIMPULAN

Berdasarkan hasil uji coba dengan menggunakan *Network Simulator 2* maka penerapan algoritma *Link Expiration Time* pada AODV_EXT dapat meningkatkan kinerja protokol routing dalam pengiriman data. Protokol R-AODV_EXT menunjukkan kinerja terbaik dibandingkan AODV_EXT untuk parameter PDR dan *Throughput* pada kondisi jumlah *node* 100. Sedangkan kinerja terbaik untuk parameter *Number of Packet Drop* berada pada kondisi jumlah *node* 50. Pada semua kondisi maka nilai *Routing Overhead* pada R-AODV_EXT adalah lebih rendah daripada AODV_EXT tapi selisih nilainya kurang signifikan. Khusus pada jumlah *node* 10 nilai *Routing Overhead* untuk R-AODV_EXT adalah lebih tinggi daripada AODV. Nilai *Average End-to-End Delay* pada R-AODV_EXT cenderung lebih rendah dibandingkan AODV_EXT walaupun pada beberapa total *node* harganya lebih tinggi.

Dari hasil tersebut di atas maka dapat diambil kesimpulan bahwa protokol R-AODV_EXT memiliki kinerja yang lebih baik dibandingkan dengan AODV_EXT untuk parameter *Packet Delivery Ratio (PDR)*, *Throughput*, *Number of Packet Dropped*, dan *Average End-to-End Delay*. Sedangkan untuk *Routing Overhead* memiliki nilai yang lebih kecil daripada AODV_EXT tapi selisihnya kurang signifikan.

DAFTAR PUSTAKA

- [1] Badarneh, O., & Kadoch, M. (2009), "Multicast Routing Protocol in Mobile Ad Hoc Networks: A Comparative Survey and Taxonomy", *EURASIP Journal on Wireless Communications and Networking*.
- [2] Chang, B.-J., Liang, Y.-H., & Lin, Y.-M. (2009), "Distributed Route Repair for Increasing Reliability and Reducing Control Overhead for Multicasting in Wireless MANET", *Information Sciences Journal*, Volume 179, Pages 1705-1723, Elsevier Science Inc., New York, NY, USA.
- [3] Fareena, N., Priya Mala, A., & Ramar, Dr.k. (2012), "Mobility Based Energy Efficient Multicast Protocol for MANET", *International Conference on Modelling Optimization and Computing*.
- [4] Jatmika, A.H. (2011), "Optimasi Routing pada Jaringan MANET Menggunakan MEDSR dan LET", Magister, Institut Teknologi Sepuluh Nopember, Surabaya.
- [5] Kanakaris, V., Ndzi, D., & Ovaliadis, K. (2011), "Improving AODV Performance using Dynamic Density Driven Route Request Forwarding", *International Journal of Wireless & Mobile Networks*, Vol. 3, Is. 3.
- [6] Nurfiiana (2012), "Perbaikan Protokol Routing Ad Hoc On-demand Multipath Distance Vector (AOMDV) untuk Mendapatkan Rute yang Stabil Menggunakan Link Expiration Time (LET)", Magister, Institut Teknologi Sepuluh Nopember, Surabaya.
- [7] Perkins, C., & Royer, E. (2002), "Ad-hoc On-Demand Distance Vector Routing", RFC 3561, Addison Wesley.
- [8] Santhi, G., Nachiappan, Dr. A. (2010), "Adaptive QoS Multicast Routing with Mobility Prediction in MANETs", *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)* Vol.1, No.3, September 2010.
- [9] Su, W., Lee, S.-J., & Gerla, M. (2000), "Mobility Prediction in Wireless Networks", *21st Century Military Communications Conference Proceedings*, Los Angeles, CA.
- [10] Yassein, M.B., Khaoua, M.O., Mackenzie, L.M., & Papanastasiou, S. (2005), "Improving the Performance of Probabilistic Flooding in MANETs", *Proc. of International Workshop on Wireless Ad-hoc Networks*, London.