

Evaluasi Unjuk Kerja Jaringan Ad Hoc Berbasis Protokol AODV

Wardi*, Intan Sari Areni*, Andani Achmad*, Irma Pratiwi Sayuti[†]

^{*†}Jurusan Teknik Elektro

Fakultas Teknik, Universitas Hasanuddin

Makassar, Indonesia

*{wardi, andani, intan}@unhas.ac.id, [†]irma.wiyaka@gmail.com

Abstract—Penelitian ini mengevaluasi unjuk kerja protokol *routing* AODV menggunakan AODV-UU pada jaringan ad hoc secara eksperimental. Skenario pengujian dibangun berdasarkan keadaan node bergerak dan tidak bergerak untuk mengevaluasi unjuk kerja protokol dalam hal mencari jalur baru ketika rute yang aktif mengalami gangguan, komunikasi secara multihop, dan kemampuan node-node untuk saling berkomunikasi dalam keadaan bergerak. Pengujian dilakukan dengan mengirimkan data teks sebesar 2,2 MB melalui protokol TCP sedangkan *streaming video* menggunakan protokol UDP sebesar 40,9 MB. Hasil evaluasi terhadap parameter-parameter *throughput*, *delay*, dan *packet loss* menunjukkan bahwa protokol *routing* AODV dapat diandalkan untuk komunikasi jaringan ad hoc.

Keywords—implementasi, AODV, bergerak, *throughput*, *delay*, *packet loss*.

I. PENDAHULUAN

Jaringan ad hoc adalah jaringan yang bersifat sementara untuk suatu tujuan khusus. Jaringan ini merupakan kumpulan dari beberapa node yang saling terhubung secara nirkabel. Node-node ad hoc dapat bergerak bebas dan berpindah secara dinamik. Oleh karena itu, topologi jaringan ad hoc dapat berubah secara cepat dan tidak dapat diprediksi. Jaringan ini tidak tergantung infrastruktur telekomunikasi yang ada sehingga setiap node harus dapat berfungsi bukan saja sebagai terminal, tetapi juga sebagai *router*.

Sejumlah protokol *routing* telah dibuat untuk jaringan ad hoc. Protokol-protokol tersebut dapat diklasifikasikan menjadi protokol proaktif, reaktif, dan hybrid [1]. Protokol *routing* proaktif secara periodik membangkitkan informasi kepada setiap node yang ada pada jaringan untuk memperbaharui tabel rutenya. OLSR dan DSDV adalah contoh protokol proaktif. Sebaliknya, protokol *routing* reaktif tidak membutuhkan informasi secara periodik ke seluruh node. Protokol ini akan membangun rute menuju terminal tujuan ketika dibutuhkan. Contoh jenis protokol ini adalah AODV dan DSR. Gabungan karakteristik kedua protokol tersebut dikategorikan sebagai protokol *routing* hybrid. Contoh protokol ini adalah TORA dan ZRP.

Evaluasi terhadap unjuk kerja protokol-protokol *routing* jaringan ad hoc banyak dilakukan secara simulasi maupun secara eksperimen. Aplikasi untuk simulasi jaringan ad hoc dapat menggunakan *Network Simulator (NS)*, *QualNet*, *NetSim* atau *OPNET*. Banyak peneliti telah melakukan

evaluasi terhadap protokol *routing* AODV secara simulasi seperti pada [2]-[4]. Namun, metode simulasi kadang-kadang dikatakan kurang presisi dalam menentukan karakteristik dan parameter-parameter pengujian karena metode simulasi memberikan asumsi-asumsi dan penyederhanaan parameter-parameter yang mungkin secara nyata tidak menggambarkan jaringan yang nyata. Sebaliknya, Pengujian secara eksperimen dapat menampilkan parameter-parameter jaringan dan kondisi-kondisi lingkungan secara nyata. [5]-[7].

Beberapa protokol untuk mengevaluasi protokol *routing* AODV secara eksperimen diantaranya: MAD-HOC, AODV-UCSB, AODV-UU, Kernel-AODV, dan AODV-UIUC [8]. Protokol-protokol tersebut dibuat secara mandiri namun protokol-protokol tersebut menampilkan pengoperasian yang sama. Penelitian ini adalah implementasikan jaringan ad hoc berbasis protokol *routing* AODV menggunakan AODV-UU. Pengujian dilakukan untuk mengetahui unjuk kerja protokol untuk parameter *throughput*, *delay*, dan *packet loss*. Bab selanjutnya dari tulisan ini diorganisasikan sebagai berikut. Bab 2 menggambarkan konsep dasar protokol AODV. Pada bab 3, dijelaskan desain dan implementasi penelitian. Bab 4 berisi hasil eksperimen protokol AODV terhadap ketiga skenario pengujian. Kesimpulan tulisan ini pada bab 5.

II. AD HOC ON DEMAND DISTANCE VEKTOR (AODV)

AODV [9] adalah sebuah protokol *routing* reaktif yang dibuat khusus untuk jaringan ad hoc. Setiap node pada jaringan yang menggunakan protokol AODV, hanya mengetahui alamat dari node-node tetangga 1-hop. Oleh karena itu, AODV akan membangun jalur komunikasi menuju tujuan apabila ada permintaan komunikasi data dari node sumber. Protokol tersebut menggunakan nomor urut untuk mengidentifikasi bahwa jalur yang dilalui adalah rute yang terbaru sehingga mencegah terjadinya *routing-loop*. Jika nomor urut paket lebih besar dari nomor urut yang terdapat pada tabel *routing*, maka node tersebut akan memperbaharui tabel *routing* berdasarkan nomor urut yang terbaru. Jalur yang telah terbentuk akan tetap dipelihara sepanjang masih dibutuhkan oleh terminal pengirim. Ketika terjadi kerusakan jalur yang sedang aktif, protokol AODV akan melakukan permintaan jalur baru melalui proses pencarian rute yang berakibat adanya peningkatan waktu tunda. *Hello messages* dikirim secara periodik oleh setiap node untuk memonitor jalur menuju node-node tujuan ke seluruh tetangga 1-hop.

Protokol AODV memiliki 3 *control messages* yang utama untuk proses pencarian, pembentukan, pemulihan, dan pemeliharaan rute jaringan.

A. Route Request (RREQ)

Ketika sebuah node ingin berkomunikasi dengan node lain yang tidak ada dalam tabel *routing* node tersebut, maka rute akan dibangun dengan mengirimkan route request (RREQ) kepada seluruh tetangga 1-hop. Kemudian node-node penghubung yang menerima RREQ dari node sumber akan melanjutkan RREQ kepada tetangga-tetangga 1-hop. Jika node-node tetangga belum menerima RREQ sebelumnya dan node tersebut bukan node tujuan, maka node-node tersebut meneruskan pengiriman RREQ ke node-node tetangga 1-hop. Prosedur akan terus berlanjut hingga RREQ menemukan node tujuan.

RREQ berisi informasi alamat pengirim, alamat tujuan, nomor permintaan, nomor urutan sumber, nomor urutan tujuan, dan jumlah hop untuk mengidentifikasi keunikan dari RREQ. Nomor *request ID* akan bertambah setiap node mengirim RREQ yang baru. Ketika node-node tetangga menerima RREQ, maka node-node tersebut akan memeriksa informasi-informasi yang ada pada RREQ tersebut. Jika node tersebut sudah menerima RREQ yang sama, maka RREQ yang baru tidak akan dilanjutkan ke node-node tetangganya.

B. Route Reply (RREP)

Apabila node yang menerima RREQ adalah node tujuan atau node yang memiliki jalur ke node tujuan, maka node tersebut akan membangkitkan pesan *route reply* (RREP) menuju ke node sumber secara *unicast*. Node-node *intermediate* yang menerima RREP akan memperbaharui table rutenya pada *destination sequence number* yang terbaru. Jika pengirim menerima rute lebih dari satu, akan dipilih rute dengan jumlah *hop* yang terkecil. Kemudian node pengirim akan merekam jalur ke node tujuan dan selanjutnya akan memulai mengirim data.

C. Route Error (RERR)

Sebuah node akan membangkitkan pesan *route error* (RERR) ketika node tersebut mendeteksi adanya kegagalan dalam pengiriman data pada jalur yang sedang digunakan. RERR dikirim ke node sumber melalui node-node penghubung secara *hop by hop*. Setiap node *intermediate* akan membatalkan rute untuk setiap node tujuan yang tidak dapat dihubungi. Node sumber yang menerima pesan RERR, membatalkan rutenya dan memulai mencari rute yang baru menuju node tujuan yang sama.

III. DESAIN DAN IMPLEMENTASI

Uji coba unjuk kerja jaringan dilakukan di lantai 3 gedung jurusan teknik elektro fakultas teknik Universitas Hasanuddin. Lokasi ini digunakan untuk pengujian dalam ruangan baik dengan penghalang maupun tanpa penghalang dan di luar ruangan tanpa penghalang. Node-node menggunakan 5 buah laptop yang mempunyai spesifikasi perangkat keras yang dapat dilihat pada Tabel 1. Setiap laptop menggunakan sistem operasi Linux Ubuntu 10.04.4 LTS [10]. Masing-masing

laptop juga dilengkapi dengan perangkat lunak Wireshark 1.4.6-1 [11] yang digunakan untuk menganalisis kinerja jaringan, Giver untuk *sharing* file, dan VLC Media Player 2.01 [12] untuk streaming video.

Pengujian jaringan dilakukan dengan menggunakan video streaming dan pengiriman teks. Streaming video sebesar 40.2 MB dikirim melalui protokol UDP dengan *frame rate* 30 fps. Sedangkan data teks sebesar 2,2 MB dikirim melalui protokol TCP.

TABLE I. SPESIFIKASI LAPTOP

| Node | Brand | Processor | RAM |
|------|--------------------------|-------------------|------|
| A | Acer Aspire 4810T | Intel Core2 Solo | 4 GB |
| B | Toshiba Satellite Turion | AMD Turion II Neo | 4 GB |
| C | Toshiba Satellite | Intel Atom N2600 | 2 GB |
| D | Acer Aspire One | Intel Core i3 | 2 GB |
| E | Acer Aspire 4741 | Intel Core i5 | 2 GB |

A. Konfigurasi Protokol AODV

Semua node diinstalasi dengan protokol AODV-UU 0.9.6 yang dikembangkan oleh E. Nordstrom di Uppsala University [13]. AODV-UU adalah implementasi dari protokol AODV yang distandarasi oleh IETF RFC 3561 [9].

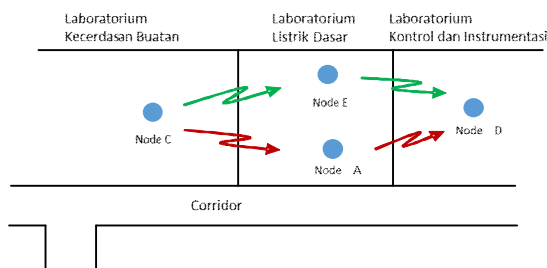
B. Skenario Pengujian

Skenario pengujian dibuat untuk keadaan node bergerak dan node tidak bergerak. Node-node dievaluasi untuk melihat kemampuan node-node untuk mencari jalur baru ketika jalur yang dilalui rusak, komunikasi secara multihop, dan kemampuan node untuk saling berkomunikasi dalam keadaan bergerak.

1) Skenario self-configuration

Skenario ini untuk mengetahui kemampuan protokol AODV dalam mencari jalur baru ataupun memperbaiki hubungan bila terjadi kerusakan pada jalur yang sedang dilalui. Pada skenario pertama ini, laptop di tempatkan pada posisinya masing-masing, yaitu pada 3 laboratorium jurusan Teknik Elektro yang dibatasi oleh tembok penghalang seperti terlihat pada Gambar 1.

Langkah-langkah pengujian sebagai berikut: Pertama-tama Node sumber (node C) melakukan streaming video ke Node tujuan (node D). Selanjutnya Mengamati jalur yang dipilih oleh node C sebagai *intermediate node* (node A atau node E) dan merekam komunikasi data yang terjadi. Setelah itu



Gambar 1. Topologi Pengujian Skenario 1

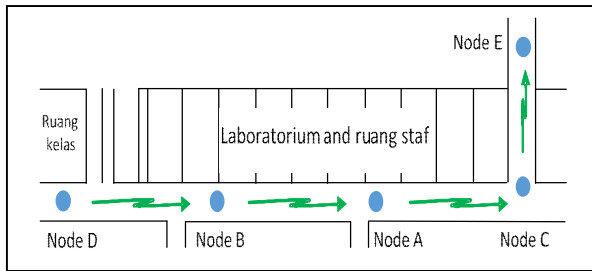
mengulangi pengiriman data lalu menon-aktifkan *intermediate node*. Kemudian mengamati komunikasi data yang terjadi.

2) Skenario Komunikasi Multihop

Skenario ini untuk melihat kemampuan protocol dalam komunikasi multihop. Pada skenario kedua ini, *node-node* di letakkan di sepanjang koridor jurusan teknik elektro tanpa penghalang yang diilustrasikan pada Gambar 2.

Pengujian dilakukan berdasarkan langkah-langkah pengujian sebagai berikut:

- 1-hop : Node D mengirim streaming video dan data teks ke node B
- 2-hop : Node D mengirimkan streaming dan data teks ke node A melalui intermediate node B
- 3-hop : Node D mengirimkan streaming dan data teks ke node C melalui intermediate node B dan node A
- 4-hop : Node D mengirimkan streaming dan data teks ke node E melalui intermediate node B, node A, dan node C.



Gambar 2. Topologi Pengujian Skenario 2

3) Skenario Node Bergerak

Skenario ini memperlihatkan kemampuan protocol dalam melakukan komunikasi secara bergerak. Konfigurasi node-node pengujian dapat dilihat pada Gambar 3. Pada Gambar 3, terdapat 4 node yang dapat saling berkomunikasi tanpa penghalang. Setiap node berpindah sesuai dengan jalur yang sudah ditentukan dengan kecepatan rata-rata 0,467 m/s (kecepatan rata-rata melangkah). Langkah-langkah pengujian adalah sebagai berikut:

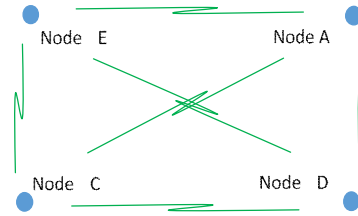
- Node A mengirimkan streaming video ke hanya node C.
- Node A mengirimkan steaming video ke node C dan node D.
- Node A mengirimkan streaming video ke node C, node D, dan node E.

IV. HASIL DAN PEMBAHASAN

Selama proses pengujian ketiga scenario, komunikasi data antar node-node dalam jaringan ad hoc yang menggunakan protokol AODV direkam oleh Wireshark [11].

A. Skenario Self-configuration

Pengamatan dilakukan ketika node sumber (node C) mengirimkan data ke node tujuan (node D). Pengamatan pertama dilakukan ketika node C mengirim data ke node D melalui node penghubung (node A) yang dipilih secara random oleh jaringan. Kemudian node A di matikan untuk



Gambar 3. Topologi Pengujian Skenario 3

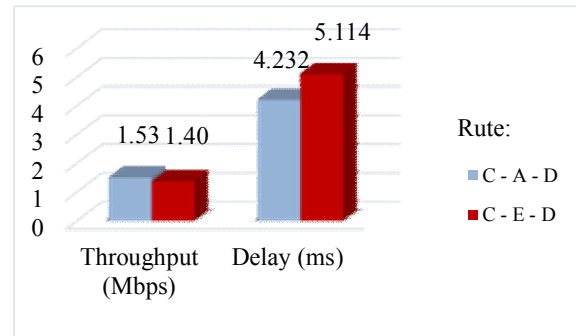
melihat kemampuan dari protokol AODV untuk mencari jalur lain ketika jalur yang sedang dilalui rusak/hilang. Dari hasil pengamatan terlihat bahwa terjadi penurunan throughput sebesar 8,49%. Hal ini terjadi akibat adanya paket yang hilang ketika proses informasi jalur yang gagal (RERR) dan ketika proses permintaan jalur baru oleh node C (RREQ dan RREP). Perubahan jalur tersebut juga berdampak pada delay time yang meningkat dari 4,232 ms menjadi 5,114 ms. Data hasil pengamatan dapat dilihat pada Gambar 4.

B. Skenario Komunikasi Multihop

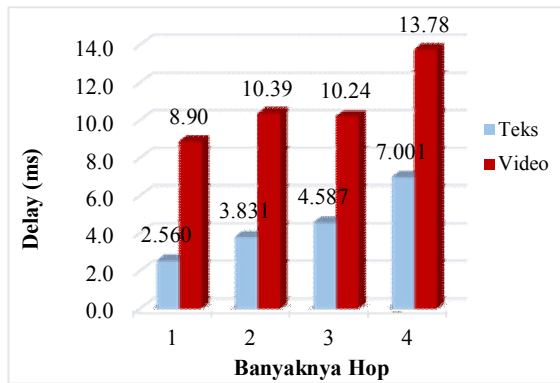
Pengujian terhadap komunikasi multihop pada jaringan ad hoc dilakukan dengan menguji mulai dari 1-hop sampai 4-hop. Hasil pengujian dapat dilihat pada Gambar 5, Gambar 6, dan Gambar 7.

Gambar 5 menunjukkan rata-rata waktu tunda jaringan ad hoc untuk 1-hop hingga 4-hop. Lamanya waktu yang dibutuhkan ketika mengirim data berupa teks untuk 1-hop sebesar 2,560 ms, untuk 2-hop sebesar 3,831 ms. Ketika data dikirim sampai 3-hop, waktu tunda terjadi sebesar 4,587 ms, sedangkan untuk pengiriman data teks hingga 4-hop selama 7,001 ms. Dari data terlihat bahwa terjadi peningkatan delay time dari 1-hop sampai 4-hop. Peningkatan delay time juga terjadi ketika mengirimkan data berupa streaming video dari 8,900 ms untuk 1-hop dan 13,78 ms untuk 4-hop.

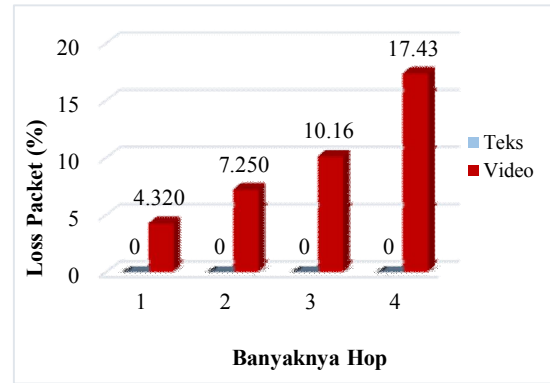
Pengamatan hasil pengujian terhadap throughput dapat dilihat pada gambar 6. Gambar tersebut memperlihatkan terjadi penurunan jumlah rata-rata throughput dari komunikasi 1-hop hingga komunikasi 4-hop. Semakin banyak jumlah hop dalam berkomunikasi maka akan semakin kecil jumlah throughput yang terjadi. Rata-rata throughput untuk pengiriman data teks mulai dari 3,230 Mbps, 2,450 Mbps,



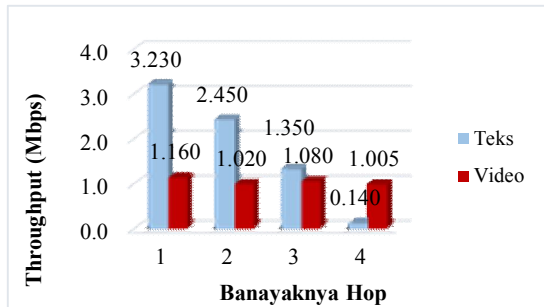
Gambar 4. Hasil Pengujian Skenario 1



Gambar 5. Delay untuk Komunikasi Multihop



Gambar 7. Packet Loss untuk Komunikasi Multihop



Gambar 6. Throughput untuk Komunikasi Multihop

1,350 Mbps, dan 0,140 Mbps untuk masing-masing jumlah hop berturut-turut 1-hop, 2-hop, 3-hop, dan 4-hop. Penurunan throughput juga terjadi ketika data berupa *streaming video* dikirim mulai dari 1,160 Mbps untuk 1-hop hingga 1,005 Mbps untuk 4-hop.

Gambar 7 memperlihatkan banyaknya paket yang hilang terhadap jumlah komunikasi hop dari node sumber ke node tujuan. Gambar tersebut menunjukkan tidak adanya paket yang hilang ketika mengirim data teks dari terminal sumber (node D) ke terminal tujuan (node E). Sejumlah paket hilang selama proses komunikasi pada pengiriman *streaming video* sebesar 4,32 % untuk 1-hop. *Packet loss* sebesar 7,20%, 10,16%, dan 17,43% terjadi untuk *streaming video* berturut-turut pada komunikasi 2-hop, 3-hop, dan 4-hop.

C. Skenario Node Bergerak

Pengujian untuk node-node bergerak menggunakan 4 buah laptop. Masing-masing terminal dapat saling berkomunikasi dan bergerak sepanjang 120m pada jalur yang ditentukan dengan kecepatan rata-rata 0,467 m/s. Skenario komunikasi antara terminal-terminal dilakukan dengan mengamati komunikasi data ketika pengiriman data oleh node sumber (node A) untuk 1 node tujuan (node C), 2 node tujuan (node C, node D), dan 3 node tujuan (node C, node D, node E). Hasil pengamatan oleh Wireshark dapat dilihat pada Gambar 8, Gambar 9, dan Gambar 10.

Gambar 8 memperlihatkan banyaknya paket yang hilang ketika mengirim data kepada 1, 2, dan 3 terminal tujuan.

Jaringan ini mengalami maksimum paket yang hilang ketika mengirimkan data kepada 3 node tujuan secara bersamaan dibandingkan kepada 1 node tujuan atau 2 node tujuan. Setiap terminal tujuan terlihat memiliki jumlah *packet loss* yang berbeda-beda karena setiap terminal memiliki spesifikasi yang berbeda.

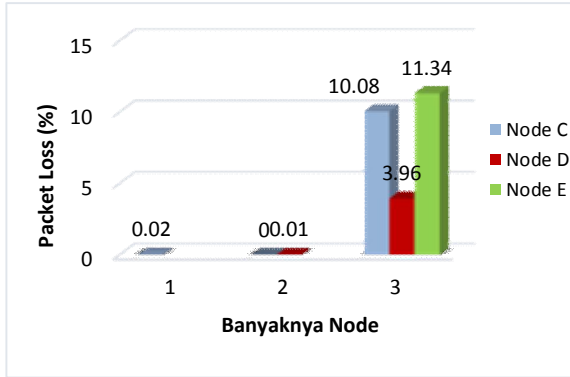
Gambar 9 menggambarkan suatu peningkatan lamanya waktu tunda sejalan dengan bertambahnya jumlah node tujuan. Hal ini diakibatkan keterbatasan kapasitas dari jaringan yang digunakan. Waktu tunda yang tertinggi terlihat ketika node A mengirimkan *streaming video* secara bersamaan kepada node C, node D, dan node E. Node D membutuhkan waktu tunda terlama dibandingkan node C dan node E sebesar 9,877ms.

Banyaknya throughput terhadap jumlah terminal tujuan diperlihatkan oleh Gambar 10. Gambar tersebut menunjukkan bahwa semakin banyak node sebagai tujuan pengiriman data maka besarnya rata-rata throughput akan menurun. Hasil pengamatan pada node C, ketika node A mengirim data hanya pada node C, rata-rata throughput pada node C sebesar 1,401 Mbps. Rata-rata throughput pada node C akan menurun sebesar 1,363 Mbps ketika node A menyebarkan data pada node C dan node D. Begitu pula ketika node A mengirim data kepada tiga node (node C, node D, dan node E) secara bersamaan, maka rata-rata throughput pada node C sebesar 1,113 Mbps.

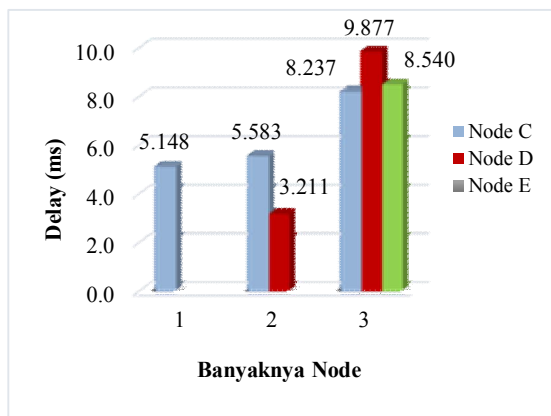
V. KESIMPULAN

Penelitian ini memperlihatkan kinerja protokol AODV pada jaringan ad hoc secara nyata.

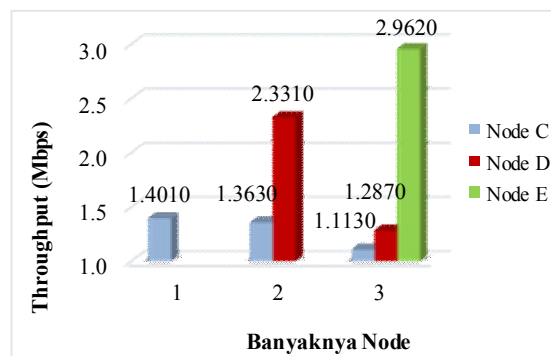
- Protokol AODV mencari jalur baru ketika jalur yang sedang dilalui mengalami kegagalan dengan penurunan throughput dari 1,53 Mbps menjadi 1,40 Mbps dan waktu tunda dari 4,232 ms menjadi 5,114 ms.
- Terjadi penurunan unjuk kerja jaringan ad hoc ketika jumlah hop terminal pengirim ke terminal tujuan bertambah. Pada komunikasi 4-hop, terlihat bahwa:
 - o Waktu tunda sebesar 7,001 ms untuk pengiriman data teks dan 13,78 ms untuk *streaming video*.



Gambar 8. *Packet Loss* untuk Node Bergerak



Gambar 9. *Delay* untuk Node Bergerak



Gambar 10. *Throughput* untuk Node Bergerak

- Kehilangan paket sebesar 17,43.% untuk streaming video dan tidak ada paket yang hilang ketika mengirim data berupa teks.
- Rata-rata throughput sebesar 0,140 Mbps untuk data teks dan 1,005 Mbps untuk streaming video.
- Pengujian node-node bergerak, rata-rata waktu tunda, throughput dan packet loss ketika terminal sumber mengirimkan data kepada 3 terminal tujuan secara bersamaan berturut-turut sebesar 8,885 ms, 1,787 Mbps, dan 8,46 %.

Dari hasil pengujian terhadap semua parameter (throughput, waktu tunda, *packet loss*) memperlihatkan unjuk kerja yang baik untuk node-node tidak bergerak maupun dalam keadaan bergerak. Oleh karena itu, protokol AODV dapat diandalkan untuk jaringan ad hoc.

DAFTAR PUSTAKA

- [1] M. Albolhasan, T. Wysocki, and E. Dutkiewicz, "A Review of Routing Protocols for Mobile Ad Hoc Networks," *Ad Hoc Network*, vol. 2, no. 1, pp. 1-22, January 2004.
- [2] A. Moravejsharieh, H. Modares, R. Salleh, and E. Mostajeran, "Performance Analysis of AODV, AOMDV, DSR, DSDV Routing Protocols in Vehicular Ad Hoc Network", *Research Journal of Recent Sciences*, vol. 2, no. 7, pp. 66-73, July 2013.
- [3] M. S. Chaudhary and V. Singh "Simulation and Study of AODV Routing Protocol under CBR and TCP Traffic Source", *International Journal of Future Computer and Communication*, vol. 3, no. 2, pp. 84-88, April 2014.
- [4] M. Muamer.; S. Norrozila "Performance Analysis of DSR, AODV On-Demand Routing Protocols in Mobile Ad Hoc Networks", *Advanced Science Letter*, vol. 20, no. 2, pp.359-363, Februari 2014.
- [5] X.B Timothy, D. Sheeatalkumar, J. Sushant, H. Daniel, and G. T. Roshan " A full scale wireless ad hoc network test bed", in *Proc. The ISART*, 2005, pp 51-60.
- [6] Y Lin, A. H. M Rad, V.W.S Wong, and J.H Song, "Experimental comparisons between SAODV and AODV routing protocols", the 1st ACM workshop on Wireless multimedia networking and performance modeling, New York, 2005, pp. 113-122.
- [7] J. Yun, B. Lee, Y. Baek, and J. Kim, "An Implementation of AODV Testbed with Multi-metrics", *Eighth International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, April 2011, pp. 1072-1073.
- [8] I.D. Chakeres and E. M. Royer, "AODV routing protocol implementation design", *24th International Conference on Distributed Computing Systems Workshops*, March 2004. Pp. 698-703
- [9] C.Perkin, E. M. Royer, "Ad hoc on Demand Distance Vector Routing", RFC 3561, July 2003, <http://www.ietf.org/rfc/rfc3561.txt>.
- [10] "Linux ubuntu 10.04 LTS" [Online]. Available: <http://releases.ubuntu.com/10.04/>
- [11] Wireshark software." [Online]. Available: <http://www.wireshark.org/>
- [12] "Vlc player." [Online]. Available: <http://www.videolan.org/vlc/download-ubuntu.htm>.
- [13] E. Nordstrom, "SourceForge AODV-UU download," <http://sourceforge.net/projects/aodvuu/>