

Simulasi Ad-Hoc Menggunakan NS2 dengan *Routing Protocol AODV*

Maulaya Radhibilla^[1], Muhammad Rafli Al-Hasyimi^[2], Taruli Jeremia Halasan Sinaga^[3], Muhammad Hilmi Izzulhaq^[4], Dara Kusumawati
Ramadani Yasir^[5]

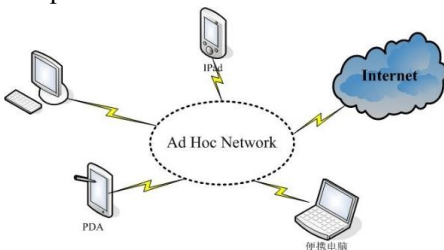
Jurusan Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung, Indonesia
radhibilla, mraflialh, muhhlmi daraaysr @student.telkomuniversity.ac.id

Abstrak— Kajian ini akan mendeskripsikan latar belakang dan fitur dasar dari *Ad-Hoc On-Demand Distance Vector* (AODV). AODV merupakan salah satu protokol reaktif pada jaringan *Ad-Hoc*. Pada AODV, rute dari *node* sumber ke *node* tujuan akan dibuat jika hanya *node* sumber ingin mengirimkan paket ke *node* tujuan. Untuk menjalankan suatu simulasi perancangan jaringan kita membutuhkan simulator alat-alat jaringan sebagai media pembelajaran atau pelatihan yaitu menggunakan NS2 (*Network Simulator versi 2*) *Network Simulator* yang dipakai pada simulasi ini adalah versi 2, di mana *software* yang dapat menampilkan secara simulasi proses komunikasi dan proses bagaimana proses komunikasi tersebut berlangsung. *Network Simulator* melayani simulasi untuk komunikasi dengan kabel dan komunikasi nirkabel.

Kata kunci — AODV, NS2, *Ad-hoc*, *Routing Protocol*

I. PENDAHULUAN

Sejalan dengan perkembangan teknologi informasi yang berkembang, tipe jaringan yang mampu melibatkan banyak orang atau perangkat komunikasi tanpa ketergantungan infrastruktur semakin diperlukan. Hal tersebut yang dinamakan dengan *Ad-Hoc Network*. Jaringan ini dapat terbentuk dari sekumpulan *node* yang menggunakan *wireless interface* untuk melakukan komunikasi antar *node*. Antara *node* yang berbeda dapat terhubung melalui transmisi *wireless*, sehingga membutuhkan *node* lain untuk meneruskan pesan.



Gambar. 1. Struktur Dasar Jaringan Ad Hoc¹

Tujuan dari paper ini yaitu untuk mencari *QOS data throughput*, memahami *script* yang digunakan untuk membangun jaringan *Ad-Hoc*, dan menganalisis hasil simulasi jaringan *Ad-Hoc* (*Throughput, packet loss, delay*,

dan lain-lain) agar bisa diimplementasikan untuk *mobile* dan *vehicular Ad-Hoc network*.

Dalam simulasi ini, kami menggunakan Ubuntu, yaitu salah satu distro dari OS Linux dan menggunakan perangkat lunak *Network Simulator-2*. Kami melakukan simulasi dengan menyiapkan *repository* dari NS2 dan melakukan instalasi pada NS2. Selain itu, kami juga melakukan instalasi *Network Animator* untuk melihat jejak jaringan dan jejak paket serta untuk melihat *layout* topologi. Pada simulasi ini, kami menggunakan protokol reaktif yaitu AODV. Setelah protokol reaktif AODV tersebut disimulasikan melalui NS2, akan menghasilkan *trace file*. *Trace file* tersebut adalah *file* yang mengandung jejak terhadap apa yang terjadi pada simulasi tersebut. Dari *trace file* ini kami dapat menentukan performa dari simulasi yang dijalankan. Dengan mengetahui performa tersebut, maka siapapun bisa melihat bagaimana pengaruh *routing* protokol terhadap jaringan *Ad-Hoc*.

Terdapat 2 jenis protokol rute konvensional, yaitu jenis *distance vector* dan jenis *link state*. Protokol rute tersebut sering digunakan pada jaringan dengan infrastruktur yang statis. Perubahan topologi yang dinamis tidak dapat menjadi acuan saat kedua jenis protokol dibuat. Keduanya tidak bisa digunakan pada protokol *routing non adaptif* untuk keperluan jaringan dimana *node* bergerak bebas.

Jika *node* bergerak, maka akan menyebabkan perubahan topologi jaringan. Keadaan ini menghasilkan jaringan yang tidak memiliki infrastruktur atau biasa dikenal dengan jaringan *Ad-Hoc*. Pesan yang dikirim dalam jaringan ini akan terjadi antara dua *node* dalam cakupan transmisi masing-masing. Jika *node* tidak dapat berhubungan atau berada di luar jangkauan, secara tidak langsung dihubungkan oleh *multiple hop* melalui beberapa *node* perantara.

Protokol-protokol rute dapat diklasifikasikan menjadi tiga kategori:

- Terpusat atau tersebar
- Adaptif atau statis
- Reaktif atau proaktif atau *hybrid*

Jika kerja protokol rute dilakukan secara terpusat, semua keputusan yang dibuat ditentukan dari titik terpusat, tidak demikian dengan protokol rute terdistribusi, semua *node* melakukan kerja sama untuk memutuskan jalur

¹ Gao, Weimin, Lingzhi Zhu, and Junbin Liang. "Research on Propagation Model of Malicious Programs in Ad Hoc Wireless Network." *Sensors & Transducers* 162.1 (2014): 85.

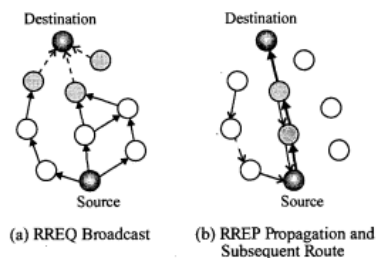
pengiriman data. Protokol rute adaptif mungkin mengubah perilaku kerjanya sesuai dengan status jaringan, mengantisipasi kongesti pada sebuah link. Sedangkan protokol rute reaktif melakukan aksi yang diperlukan seperti pencarian rute bila diperlukan, kebalikannya protokol rute proaktif menemukan rute sebelumnya untuk digunakan. Metode reaktif biasa disebut sebagai *on-demand routing protocol*. Karena aksinya dilakukan saat diperlukan, beban pengontrolan paket ditekan secara signifikan. Metode proaktif menyimpan tabel rute dan merawat tabel tersebut. Metode *hybrid* menggunakan kedua metode reaktif dan proaktif untuk membuat protokol rute semakin efisien.²

Permasalahan yang akan diangkat pada kali ini adalah pengukuran kinerja protokol routing *Ad hoc On demand Distance Vector* (AODV) dibandingkan dengan protokol routing *Destination Sequenced Distance Vector* (DSDV) dengan menggunakan Network Simulator 2 (NS2).

II. AD-HOC ON-DEMAND DISTANCE VECTOR DAN NETWORK SIMULATOR 2

Ad-Hoc On-Demand Distance Vector (AODV) merupakan salah satu protokol reaktif pada jaringan Ad-Hoc. AODV memiliki beberapa kelebihan, di antaranya AODV dapat merespons dengan cepat terhadap perubahan topologi yang diakibatkan rute aktif, AODV mendukung transmisi paket *unicast* dan *multicast*, bahkan di saat node dalam keadaan bergerak, AODV tidak menambahkan *overhead* tambahan pada paket data karena tidak menggunakan sumber routing, AODV tidak memerlukan sistem pusat administratif untuk menangani proses *routing*, dan AODV dapat menskalakan pada sejumlah besar node seluler. Pada AODV, rute dari node sumber ke node tujuan akan dibuat jika hanya node sumber ingin mengirimkan paket ke node tujuan. Proses pencarian rute pada protokol AODV terbagi menjadi dua macam, yaitu *route discovery* dan *route maintenance*.

a) *Route discovery* merupakan tahap awal di mana pencarian rute dimulai yang menggunakan dua jenis paket yaitu RREQ dan RREP. Apabila ada permintaan pengiriman paket ke suatu node, maka paket RREQ akan disebar ke semua node yang berada di sekitar node sumber sampai ke node tujuan. Setelah RREQ mencapai node tujuan, maka node tujuan akan membalas dengan mengirimkan paket RREP pada node sumber melalui rute yang sudah dilewati.



Gambar 2. *Route Discovery Circle*³

b) *Route Maintenance* merupakan proses yang dilakukan di saat ada kegagalan pengiriman paket dari node sumber ke node tujuan. Di saat node tujuan tidak bisa dicapai, maka node tujuan akan mengirimkan paket RERR pada ke semua node sampai node sumber untuk memperbarui rute.

Proses pencarian rute *multicast* ditetapkan dengan cara yang sama, sebuah *node* yang ingin bergabung dengan sebuah grup *multicast* akan menyiarkan RREQ dengan alamat IP tujuan dan dengan *flag* 'J' (*join*) ditetapkan untuk menunjukkan bahwa *node* ingin bergabung dengan grup. Setiap *node* yang menerima RREQ merupakan anggota dari *multicast* yang memiliki nomor urut untuk grup *multicast* dapat mengirim RREP. Ketika RREP disebarluaskan kembali ke sumber, *node* akan meneruskan pesan dan mengatur *pointer* dalam *routing table multicast*. Node sumber akan *unicast* sebuah *Multicast Activation* (MACT) pesan ke *hop* berikutnya yang dipilih. esan ini bertujuan untuk mengaktifkan rute. Jika *node* tidak menerima pesan ini yang telah mendirikan sebuah rute penunjuk *multicast*, maka akan mengalami *time out* dan *pointer* akan dihapus. Jika *node* penerima belum menjadi bagian dari *multicast*, akan melacak asal dari RREP. *Node* akan terus *unicast* MACT sampai *node* yang sebelumnya anggota *multicast* tercapai. AODV mempertahankan rute aktif, hal ini termasuk memelihara *multicast*. AODV memerlukan setiap *node* untuk menjaga *tabel routing* yang berisi *field*:

- *Destination IP Address*: berisi alamat IP dari *node* tujuan yang digunakan untuk menentukan rute.
- *Destination Sequence Number* : *destination sequence number* bekerjasama untuk menentukan rute.
- *Next Hop*: Loncatan (*hop*) berikutnya, bisa berupa tujuan atau *node* tengah, *field* ini dirancang untuk meneruskan paket ke *node* tujuan.
- *Hop Count*: Jumlah *hop* dari alamat IP sumber sampai ke alamat IP tujuan.
- *Lifetime*: Waktu dalam milidetik yang digunakan untuk *node* menerima RREP. *Routing Flags*: Status sebuah rute, *up* (valid), *down* (tidak valid) atau sedang diperbaiki.⁴

Network Simulator (Versi 2), yang dikenal luas sebagai NS2, hanyalah alat simulasi *eventdriven* yang telah terbukti berguna dalam mempelajari sifat dinamis jaringan komunikasi. Simulasi jaringan kabel dan nirkabel fungsi dan protokol (misalnya, algoritma *routing*, TCP, UDP) dapat dilakukan dengan menggunakan NS2. Secara umum, NS2 menyediakan pengguna dengan cara menentukan jaringan tersebut serta protokol dan simulasi yang sesuai.

² Wismanu, Evaluasi Unjuk Kerja Protokol Rute pada Jaringan Wireless Ad Hoc Multi Hop, Tesis, JTE-FTI, 2004.

³ Royer, Elizabeth M., and Charles E. Perkins. "An implementation study of the AODV routing protocol." 2000 IEEE Wireless Communications and Networking Conference. Conference Record (Cat. No. 00TH8540). Vol. 3. IEEE, 2000.

⁴ Ian D. Chakeres dan Elizabeth M. Belding-Royer. "AODV Routing Protocol Implementation Design", University of California, Santa Barbara.

III. AwK DAN PERL

NS2 memiliki banyak dibangun di perpustakaan dan fungsi yang mendukung banyak protokol perutean, *topologi* jaringan seperti *bus*, *ring*, *hybrid*, *star topology* untuk merancang jaringan kabel dan nirkabel dengan bantuan skrip simulasi. Keuntungan utama dari NS-2 adalah bahwa kita dapat dengan mudah menghitung total *throughput*, tingkat kesalahan, penundaan ujung ke ujung, jumlah total paket yang dikirim dan diterima oleh tujuan dengan skrip khusus AWK, Perl. AWK adalah *script* yang digunakan untuk filter dan *report data*. AWK adalah singkatan dari Aho, Weinberger, Kernighan. Format dasar dari perintah (*command*) awk adalah sbb : awk 'pattern {action}' input-file > output-file, Secara mudahnya, *pattern* adalah aksi yang dilakukan terhadap data di *input-file*, dan kemudian hasilnya dimasukkan ke dalam *output-file*. Perl merupakan bahasa pemrograman tingkat tinggi.

Perl bersifat dinamis dan kerap digunakan untuk keperluan pemrograman seperti *application* dan *web development*. Perl sendiri adalah sebuah istilah yang mengacu pada "*Practical Extraction and Reporting Language*". Meskipun terlihat seperti sebuah singkatan, sejatinya tidak ada akronim resmi untuk Perl. Bahasa pemrograman ini diperkenalkan oleh Larry Wall pada tahun 1987. Ia merancang Perl secara khusus untuk keperluan *text editing*. Namun, seiring berjalannya waktu, Perl banyak digunakan untuk berbagai keperluan termasuk administrasi sistem Linux, pemrograman jaringan, hingga pengembangan server web.

IV. METODE PENELITIAN

Pada penelitian ini telah dibuat simulasi protokol routing *Ad-hoc On demand Distance Vector* (AODV) menggunakan NS2 sebagai simulator yang digunakan. Gambar dibawah ini merupakan flowchart dari simulasi.



Gambar 3. Flowchart simulasi pada NS2

Perencanaan penelitian dibagi menjadi 4 tahap utama, yaitu tahap perencanaan topologi, tahap coding dan *deployment*, tahap simulasi data dan tahap analisa hasil. Berikut adalah penjelasan lebih lanjut mengenai tahap-tahap implementasi:

1. Perencanaan Topologi

Perancangan topologi jaringan yang bersumber studi literatur merupakan langkah awal yang ditempuh. Topologi jaringan ini berupa node yang dibutuhkan, letak node, besarnya *bandwidth* batasan paket data, transport agent yang digunakan dan trafik yang akan membantu proses pembangunan simulasi jaringan secara keseluruhan.

2. Coding dan Deployment

Tahap ini merupakan tahapan inti dari implementasi ini, yaitu pemrograman bahasa TCL pada aplikasi NS-2.

3. Simulasi Data

Pada tahap ini simulasi dijalankan di atas aplikasi NS-2. Hasil simulasi berupa *file trace*, yang kemudian akan ditampilkan ke dalam bentuk grafik dengan menggunakan Microsoft Excel.

4. Analisa Hasil

Dalam tahap ini dilakukan analisa hasil berdasarkan data yang diperoleh. Analisis ini meliputi nilai parameter-parameter yang ingin dicari, yaitu *Packet Delivery Fraction* (PDF), *delay* dan *throughput*. Nilai parameter ini akan diperoleh dengan menggunakan *script* AWK.

V. SIMULASI DAN HASIL

Tabel 1

Hasil rangkuman dari simulasi AODV dan DSDV

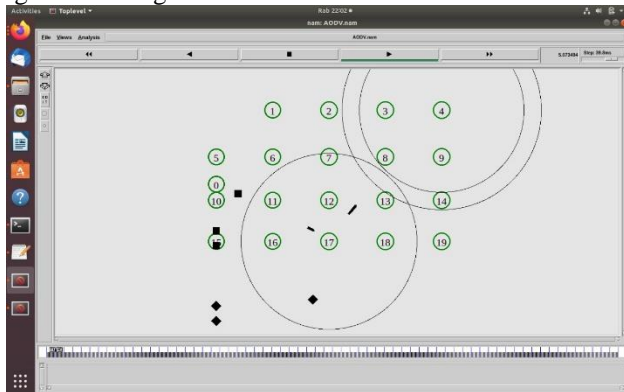
Nama	AODV	DSDV
<i>Number of packets sent</i>	188605	182856
<i>Number of packet received</i>	181085	192942
<i>Number of packet dropped</i>	22077	24668
<i>Simulation time</i>	290.997 ms	299.479 ms
<i>Initial Energy</i>	50.0	50.0
<i>Energy Consumed</i>	22.307.143	18.459.543
<i>Final Energy</i>	27.692.857	31.540.457
<i>Packet Sent</i>	188605	182856
<i>Packet Loss</i>	22077	24668
<i>Total Loss</i>	11.705.416	13.490.397
<i>Throughput</i>	407.86 Kbps	375.86 Kbps
<i>AODV delay</i>	1016.58 ms	967.093 ms

Skenario simulasi yang digunakan untuk menganalisa kinerja AODV dengan DSDV pada NS2 adalah sebagai berikut:

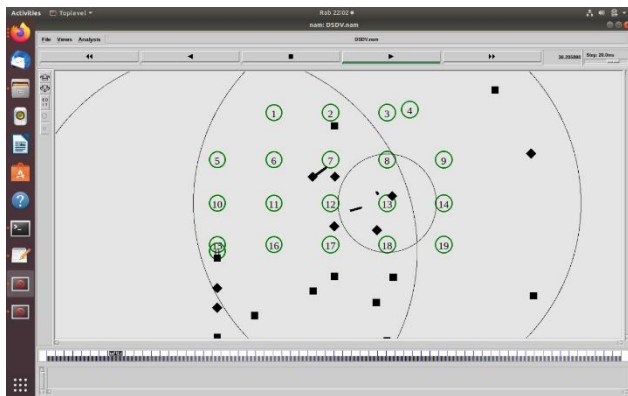
1. Topologi jaringan sesuai dengan banyaknya jumlah node.
2. Protokol routing yang digunakan adalah *Ad hoc On demand Distance Vector* (AODV) dan *Destination Sequenced Distance Vector* (DSDV).
3. Proses penempatan mobile node bersifat acak.
4. Jumlah mobile node yang ada pada yaitu 19 node.
5. Pergerakan mobile node pada saat simulasi bersifat acak.

6. Data dikirimkan dari mobile node berjenis UDP

Pada gambar 4 di bawah dapat dilihat ada 19 node yang berada dalam satu wilayah, ketika topologi jaringan topologi belum berubah. Jika node-node ini bergerak maka 5 akan mempengaruhi topologi jaringan sehingga routing-nya juga berubah. Semua node yang disimulasikan bergerak secara acak, maka algoritma protokol routing akan berubah menyesuaikan dan mengupdate tabel routing-nya disesuaikan dengan jarak perpindahan node. Penelitian ini diuji berdasarkan parameter-parameter QoS, yaitu *delay*, *throughput*, dan *packet loss*. Berdasarkan data hasil simulasi menggunakan NS2 didapat data untuk parameter *delay*, *throughput* dan *packet loss*. sebagai perbandingan kinerja protokol routing digunakan DSDV. Skenario posisi dan pergerakan sebagai berikut:



Gambar 4. Skenario AODV



Gambar 5. Skenario DSDV

1. Delay

Hasil dari pengukuran *delay* pada simulasi ditunjukkan pada table 1. Dari hasil simulasi pada Tabel 1 didapat nilai *delay protocol routing* AODV lebih besar dibandingkan dengan *protocol routing* DSDV. Di sini dapat dilihat *protocol routing* DSDV memiliki kinerja yang baik dibandingkan dengan *protocol routing* AODV pada parameter *delay* di simulasi kali ini. Jarak pada node pada simulasi kali ini mengakibatkan trafik jaringan akan semakin padat sehingga mengakibatkan *delay* yang besar.

2. Throughput

Dari ketiga gambar di atas performansi parameter *throughput protocol routing* AODV lebih besar dari *protocol routing* DSDV. Semakin padatnya jaringan maka paket data yang dikirim mengalami antrian dan saat node terputus koneksinya, *node source* akan membroadcast paket data untuk melakukan *routing* ke node tujuan dan ini menyebabkan paket yang dapat dilewatkan mengalami penurunan.

3. Packet loss

Berdasarkan hasil pengukuran *packet loss* diatas dapat diamati bahwa performansi dari *routing* AODV lebih baik dibandingkan dengan protokol *routing* DSDV. *Packet loss* adalah parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang. Paket yang hilang ini dapat terjadi karena *collision* dan *congestion* pada jaringan. Dilihat dari table 1 *protocol routing* DSDV lebih banyak paket yang hilang dibanding dengan protokol routing AODV.

Dari hasil simulasi pada tabel 1, diketahui bahwa performansi *protocol routing* AODV untuk jaringan *mobile ad hoc* lebih baik dibandingkan dengan *protocol* DSDV

VI. KESIMPULAN

Pada simulasi kali ini kita bisa mencari, *packet received*, *packet dropped*, *Simulation time*, *Initial Energi*, *Energy Consumed*, *Final Energi*, *Packet Sent*, *Packet Loss*, *Total Loss*, *Throughput*, *AODV delay*, sehingga hal ini berguna untuk simulasi jaringan Ad-Hoc ke depannya. Sehingga dalam sistem jaringan yang dapat mempermudah siapa saja yang ingin membangun sebuah jaringan telekomunikasi *wireless*. Kami menggunakan *Network Simulator 2* untuk melakukan simulasi jaringan komputer.

Berdasarkan hasil pengujian simulasi yang telah dibuat pada penelitian ini dapat diambil kesimpulan :

1. Tingkat kerapatan node akan mempengaruhi kesuksesan pengiriman data antar node
2. Protokol routing AODV memiliki *delay* yang besar, *throughput* yang lebih besar, *packet loss* yang lebih kecil dibanding DSDV.

Daftar Pustaka

- [1] Perkins, Charles E. "Ad hoc networking: an introduction." Ad hoc networking 40 (2001): 20-22.)
- [2] Issariyakul, Teerawat, and Ekram Hossain. "Introduction to network simulator 2 (NS2)." Introduction to network simulator NS2. Springer, Boston, MA, 2009. 1-18.
- [3] Perkins, Charles, Elizabeth Belding-Royer, and Samir Das. "RFC3561: Ad hoc on-demand distance vector (AODV) routing." (2003).
- [4] Wheeb, Ali Hussein. "Performance comparison of transport layer protocols." International Journal 5.12 (2015)..