

ANALISIS KINERJA PROTOKOL ROUTING IoT PADA TOPOLOGI STATIS DENGAN NS2

Arya Sony^{*)}, and Selo Sulisty^{*)}

Departemen Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada Yogyakarta
Jl. Bulak Sumur kampus UGM, Sleman, Yogyakarta

^{*)}E-mail: aryasny@gmail.com, selo@mail.ugm.ac.id

Abstrak

salah satu area penelitian didunia yang berkembang pesat saat ini adalah IoT, terlebih penelitian pada area berbasis pergerakan atau biasa disebut MANET (Mobile Adhoc Network), hal tersebut tidak terhindarkan lagi karena teknologi ini menjelma menjadi kebutuhan manusia moderen, oleh karena itu tren penelitian selalu mengarah dan mengasumsikan parameter penelitian dan simulasinya pada topologi adhoc. Indonesia dilain pihak belum membutuhkan teknologi semacam ini karena sebagian besar penggunaan perangkat IoT masih berbasis pada topologi infrastruktur tetap/pasti. Paper ini menggunakan Network Simulator 2 (NS2) sebagai perangkat lunak simulasi untuk membandingkan kinerja dari 3 protokol routing terpopuler MANET diantaranya AODV, DSDV dan DSR. Hasil analisa menunjukkan AODV menjadi protokol pencarian jalur yang paling dapat diandalkan namun paling boros dalam penggunaan energi.

Kata kunci: Topologi Statis, MANET, Analisis Kinerja, NS2

Abstract

one of the most buzzing area of research around the world this days is IoT technology, especially research in the area of movement device sensor network using MANET (Mobile Adhoc Network) that seems emerged to be our needs in modern society, that is why the tren of research and simulation is always set the parameter with adhoc topology. Indonesia in the other side doesn't really need this kind of technology yet, because most of our IoT devicees are still based on the fixed infrastructure technology. This paper uses NS2 Network Simulator software to compare and analysis 3 of the most used MANET routing protocol AODV, DSDV and DSR. The analysis show that AODV is the most reliable routing protocol but most wastfull in energy used.

Keywords: Fixed Topology, MANET, Performance Analysis, NS2

1. Pendahuluan

Perkembangan teknologi IoT didunia merupakan salah satu yang tercepat untuk bidang IT, terlebih untuk mengakomodasi permintaan dalam pemantauan wilayah berbasis pergerakan menggunakan MANET (Mobile Adhoc Network). Berbagai jenis penelitian untuk mencari mekanisme dan kinerja selalu mengasumsikan penelitiannya pada topologi *infrastructureless* [1][2]. Di lain pihak indonesia sebagai negara berkembang belum membutuhkan teknologi semacam ini, sebagian besar penggunaan perangkat IoT kita masih berbasis pada topologi *fixed infrastructure*.

Penelitian pada paper ini adalah lanjutan dari penelitian yang telah dilakukan sebelumnya sejak tahun 2008–2014, penelitian [3] membandingkan protokol DSR dan AODV dengan topologi AdHoc yang berubah-ubah dalam

studi kasus kapal perangnya. Penelitian [4] membandingkan protokol DSR dan GRP dan lebih mendalami dampak geografis. Peneliti [5] menganalisis secara rinci protokol AODV secara simulasi dan membandingkan dengan tesbed dengan PDA.

Tujuan dilakukannya penelitian untuk mengetahui mana diantara ketiga protokol uji yang paling cocok digunakan pada topologi statis/tidak berubah. protokol terpopuler AODV, DSDV dan DSR di simulasikan dengan variabel banyaknya node untuk mencari hasil yang biasa digunakan dalam evaluasi kinerja suatu protokol. parameter hasil yang diambil diantaranya *throughput*, reliabilitas data transaksi, waktu pencarian jalur.

Struktur paper diawali pendahuluan dan penelitian terdahulu pada bab 1, dilanjutkan dengan alat bahan dan metode penelitian pada bab 2, bab 3 berisi tampilan hasil

simulasi dan analisis, kemudian kesimpulan dan referensi sebagai penutup.

2. Metode

2.1. Alat dan Bahan

Penelitian dikerjakan dengan bantuan perangkat lunak simulasi NS2, NS2 adalah *software* simulasi diskrit yang mendasarkan kinerjanya berdasarkan kejadian [6]. *Software* simulasi ini berjalan dengan baik di lingkungan UNIX yang merupakan *default environment* nya, Java JDK disertakan dalam paper ini karena berperan cukup banyak dalam membantu tercapainya hasil, begitu juga dengan Wine yang digunakan oleh NS Trace Analyser untuk menganalisa *file trace* .tr dari NS2. Berikut alat dan bahan yang digunakan dalam penelitian:

1. Notebook
2. Linux 15.10 64 bit
3. Ns-allinone-2.35
4. Java JDK 7
5. NSG 2.jar
6. Wine 1.6
7. Network Trace Analyser.exe

2.2. Asumsi dan Parameter Simulasi

Untuk menjaga konsistensi terdapat beberapa asumsi yang digunakan saat penelitian, node yang terletak pada ujung kiri topologi menjadi node sumber data, dan node di ujung kanan menjadi node tujuan data.

Table 1. Luas wilayah untuk setiap variabel jumlah node

Jumlah Node	Ukuran Topologi (m)
25	600 x 600
100	1200 x 1200
200	1500 x 1200
500	3000 x 3000

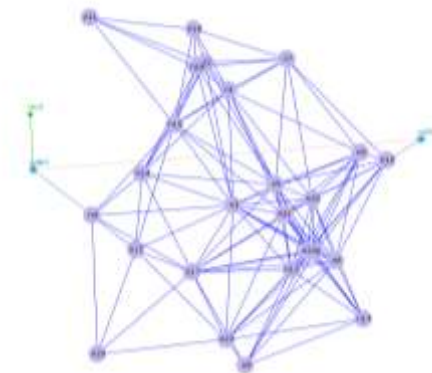
Tabel 1 memuat data luas wilayah yang digunakan dalam *software* NSG2.jar, untuk memperoleh topologi yang sesuai dengan yang di inginkan masukan data *TopologiSize* dan pilih jenis peletakan node untuk topologi secara acak. Banyaknya variasi jumlah node diperlukan untuk mensimulasikan besar kecilnya medan pada keadaan sebenarnya.

Dalam mengatur simulasi, parameter diperlukan sebagai kontrol dari keadaan awal dapat dilihat pada tabel 2, tiga parameter terakhir adalah variabel yang diubah-ubah dan dibandingkan dalam penelitian. Yang perlu diperhatikan adalah untuk tipe antrian data pada saat protokol routing menggunakan DSR maka tipe antrian harus diganti dengan *CMUPriQueue*, hal ini untuk menghindari pesan kesalahan *errorFragmantation(coreDump)*, ini adalah permasalahan teknis terkait penggunaan format dalam *trace file*.

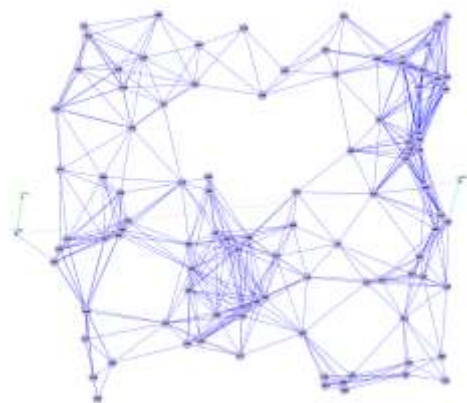
Table 2. Parameter Simulasi NS2

Parameter	Nilai
Tipe Kanal	Channel/WirelessChannel
Model Propagasi	Propagation/TwoRayGround
Model Physical Layer	Phy/WirelessPhy
Tipe Protokol MAC	Mac/802_11
Tipe Antrian Data	Queue/DropTail/PriQueue
Tipe Link Layer	LL
Tipe Antena	Antenna/OmniAntenna
Maksimum Paket dalam Antrian	50
Waktu Simulasi Total	200
Waktu Paket Mulai Dikirimkan	0.01-190
Jenis Sumber Data	CBR, 0,001 interval, 1000b/paket
Protokol Routing	DSDV DSR AODV
Jumlah Node	25 100 200 500
Luas Topologi (KM2)	0.36 1.44 1.8 9

Tabel 2 berisi parameter yang perlu diinputkan dalam konfigurasi awal simulasi pada NS2. Simulasi akan mulai meggenerate paket pada detik ke 0.01-190 dan program keseluruhan akan dihentikan pada detik ke 200.



Gambar 1. Topologi pada simulasi dengan jumlah node 25 menggunakan skema acak NSG2.jar



Gambar 2. Topologi pada simulasi dengan jumlah node 100 menggunakan skema acak NSG2.jar

Gambar 1 dengan topologi sebesar 600m x 600m, masih terlihat dengan jelas detail data per node karena ukuran topologinya yang masih kecil. Pemilihan luas wilayah topologi pada tabel 1 didasari atas pertimbangan konektivitas antar node, jika wilayah yang diberikan pada software NSG2.jar terlalu luas maka node akan menyebar sesuai dengan luas topologi yang diinginkan, sebagai tambahan informasi tiap node memiliki cakupan area sebesar 250m untuk *transmission range* dan 550 untuk *interference range* [7], jika *setting* luas topologi terlalu besar maka akan banyak node yang tidak saling berhubungan, penempatan node diatur sedemikian rupa agar masih dapat terhubung satu dengan lainnya [8]. Gambar topologi dengan node 200 dan 500 tidak disertakan karena terlalu kecil, hal yang perlu diperhatikan adalah node yang paling pada topologi dijadikan node sumber dan node paling kanan dari topologi menjadi node tujuan. Tidak ada metode khusus untuk menempatkan node, yang ada hanya aturan dasar tata peletakan [8], yang diperhatikan pada paper ini adalah node-node tidak bergerak dalam simulasinya, tidak seperti kebanyakan analisis pada paper lain yang menggerakkan nodenya [3-5].

3. Hasil dan Analisa

Hasil dari penelitian dengan simulasi ini banyak dibantu oleh perangkat lunak Network Trace Analyser, perangkat lunak ini memiliki lingkungan di sistem operasi Windows, dengan bantuan emulator Wine maka simulasi dapat di jalankan di lingkungan Linux. Unjuk kerja routing *Wireless Sensor Network* (WSN) dapat dinilai secara umum dari nilai *Througput*, *Packet Drop*, *Discovery Time* dan *Energy Consumption*

3.1. Throughput

Througput adalah parameter unjuk kerja yang menunjukkan nilai aliran bit data dari paket yang ditransmisi per total satuan waktu. Semakin besar nilai *throughput* maka protokol routing yang diamati semakin baik, biasa juga disebut dengan kecepatan transfer data.

$$\text{Throughput} = \text{Total Data terkirim} / \text{Waktu Simulasi}$$

Table 3. Hasil Simulasi Throughput Sistem

Jumlah Node	Throughput (bit/s)		
	AODV	DSDV	DSR
25	25109	32936	33317
100	9664	11327	9741
200	8684	9051	7525
500	6900	1000	-

Data dalam tabel 3 menunjukkan bahwa AODV dalam hal *throughput* kalah dari DSDV dan DSR, sedangkan *throughput* terbesar tidak dapat dipastikan karena juga tergantung dari banyaknya node yang divariabelkan.

Satu hal yang menarik adalah data pada jumlah node 500 untuk protokol DSR tidak menunjukkan nilai (-), hal ini disebabkan juga karena banyaknya node sehingga mekanisme pencarian jalur (*route discovery*) tidak berjalan dengan baik, selanjutnya analisis *route discovery* akan dibahas pada bab 3.2

3.2. Waktu Pencarian Jalur

Adalah waktu yang diperlukan oleh node untuk mulai mengirimkan paket data, nilai dari *route discovery* berbeda dengan *request latency* yang merupakan nilai rata-rata waktu tunggu perpindahan rute dalam pengiriman data.

Route Discovery adalah waktu paling awal paket data dikirim dan sampai ke node tujuan dan sangat mempengaruhi pada nilai unjuk kerja yang lain, jika protokol gagal untuk mencari jalur maka data tidak dapat ditransmisikan sehingga semua paket akan *drop* dan menyebabkan nilai *throuput* yang juga kosong (-).

Table 4. Hasil Simulasi Waktu Pencarian Jalur

Jumlah Node	Route Discovery (s)		
	AODV	DSDV	DSR
25	0.1338	35.598	0.218
100	0.2893	77.298	0.277
200	2.0072	119.228	0.446
500	0.6599	194.725	-

Terdapat anomali data pada tabel 4 diatas, nilai *route discovery* pada jumlah node 200 untuk protokol AODV tiba-tiba bernilai tinggi yaitu 2 detik, sangat jauh jika dibandingkan pada node dengan jumlah lebih sedikit maupun lebih banyak. Hal ini dikarenakan mekanisme *route discovery* pada protokol AODV pada variasi jumlah node 200 tidak dapat menjangkau node tujuan saat pertama kali disebarkan, sehingga node sumber harus beberapa kali *broadcast* paket untuk mencari jalur ke node tujuan [9][10], besarnya waktu yang diperlukan untuk mencari node tujuan bergantung pada besarnya topologi dan banyaknya *hop*, sedangkan data saat jumlah node 500 menunjukan hal sebaliknya karena topologi yang dibuat sepenuhnya acak, sehingga jumlah node tidak linier dengan banyaknya *hop* yang diperlukan transfer data dari node sumber ke tujuan.

Mekanisme yang berbeda ditemui pada data jumlah node 500, dalam hal ini data kosong (-) karena untuk protokol DSR *broadcast* paket *route request* hanya dilakukan

sekali saja, tidak seperti AODV yang berkali-kali dengan tiap *route request* yang baru bertambah TTL nya[11].

3.3. Paket Drop

Paket *drop* dapat dianalogikan dengan reliabilitas dari protokol, walaupun pernyataan diatas tidak selalu benar karena ada hal-hal lain selain reliabilitas yang dapat dijadikan acuan baik buruknya dalam mengevaluasi protokol routing. Kembali lagi kepada kebutuhan dari masing-masing kasus akan berbeda parameter yang ditonjolkan. Berikut hasil simulasi dari paket *drop*.

Table 5. Hasil Simulasi Paket Drop

Jumlah Node	Paket Drop (packet)		
	AODV	DSDV	DSR
25	18928	18578	17352
100	21903	22401	21790
200	22099	23043	22244
500	22431	23748	-

Hasil unjuk kerja paket *drop* dari data tabel 5 tidak menunjukan hasil yang signifikan, semua protokol bergantian nilainya untuk mendapat predikat protokol yang paling sering *drop* paketnya.

3.4. Konsumsi Energi

Pada jaringan WSN, nilai dari konsumsi energi memiliki daya tarik tersendiri karena masih terbatasnya permasalahan baterai yang digunakan oleh perangkat. Data aktual dari konsumsi energi tidak dapat didapatkan secara spesifik karena energi yang digunakan baik dalam mengirim atau menerima semua tergantung dari vendor perangkat yang digunakan, pada tabel 6 ditampilkan total paket yang diproduksi dari tiap protokol routing. Untuk mengetahui total energi yang terpakai sesuai perangkat dapat dilihat pada [12], pada referensi tersebut tertera energi yang dibutuhkan node saat mengirim paket atau menerima paket. Tabel 6 adalah data hasil total paket yang dihasilkan tiap protokol routing pada jumlah node yang bervariasi

Table 6. Hasil Simulasi Total Konsumsi Energy

No de	AODV			DSDV			DSR		
	Tx	Rx	F	Tx	Rx	F	Tx	Rx	F
25	125544	120888	14469	107528	114705	10373	125279	127254	12833
100	206445	503773	23075	116078	210044	11977	176284	171364	25468
200	262306	1123522	26876	112251	628972	7849	295388	447418	51507
500	566094	2371600	57253	156689	1634710	15	32125	187331	8292

Tx: Paket Terkirim (s)

Rx: Paket Diterima (r)

F : Paket Diteruskan (f)

(setiap jenis paket (s/r/f) memiliki data konsumsi energi/paket dapat dilihat pada [12])

Hasil yang didapat dengan menjumlahkan semua paket data, AODV sangat boros konsumsinya dimana jumlah paket mencapai 5,4 juta, DSDV berada ditengah-tengah dengan jumlah paket 3,1 juta dan DSR adalah protokol paling irit sumber daya dengan hanya menghasilkan total paket 1,6 juta.

4. Kesimpulan

Dari analisa yang telah dilakukan dan simulasikan menunjukkan bahwa pada topologi statis AODV adalah routing protokol yang paling dapat diandalkan dalam penyampaian data dan juga paling cepat dalam menemukan rute pencarian jalur namun paling boros dalam penggunaan sumber daya. DSR adalah protokol yang paling cepat dalam transfer data dengan penggunaan energi paling minimal dengan pengecualian untuk topologi yang sangat besar karena memiliki permasalahan *route discovery* yang berimbas terlalu lama node untuk mencari rute pengiriman data. DSDV berada ditengah tengah untuk hampir semua unjuk kinerja simulasi yang sudah dilakukan.

Referensi

- [1]. Kamaljit Kaur, Amanpreet Chela. Simulation and Review of SAODV NS2 Simulation. *International Journal of Engineering science & Research Technology*. 2014;
- [2]. Monzur Morshed, Et al. Performance Comparison of TCP variants over AODV, DSDV, DSR, OLSR in NS-2. *Informatic, Electronic and Vision (ICIEV)*. 2012; page 1069-1074.
- [3]. Hendratoro Gamantyo, Dhamayanti Yulia. Analisis Perbandingan Kinerja Protokol DSR dan AODV pada MANET untuk Sistem Komunikasi Taktis Kapal Perang. *Jurnal Ilmiah Elite Elektro*, Vol.4 no.1. 2013; 5-10.
- [4]. Amilia Fitri, dkk. Analisis Perbandingan Kinerja Protokol DSR dan GRP pada MANET. *Jurnal Sains, Teknologi dan Industri*, Vol 12 no.1. 2014; 9-15.
- [5]. Sari Riri, dkk. Analisis Kinerja Routing AODV pada jaringan Ad Hoc Hybri: Perbandingan Hasil Simulasi dengan NS2 dan Implementasi pada Testbed dengan PDA. *Makara, Teknologi*, Vol.12 no.1. 2008; 7-18.
- [6]. Teerawat Issariyakul, Ekram Hossain. Introduction to Network Simulator Tool. Second Edition. New York: Springer.2012: hal 12.
- [7]. IEEE Standard for Information Technologi. *802.11 Local and Metropolitan Area Network Standard*. USA. IEEE. 2012.
- [8]. J. Biagoni , E.S Sasaki. Wireless Sensor Placement For Reliable and Efficient Data Collection. *International Journal of Advancement Research in Computer Science and Software Engineering*. 2003
- [9]. Ibrahim Ahmad, Et al. Modified Exponential Backoff Algorithm to Minimize Mobile Communication Time. *International Journal Information Technology and Computer Science*. 2014; page 20-29.

- [10]. Mubbasi Rehmani. *A Tutorial on the Implementation of Ad-hoc On Demand Distance Vector (AODV) Protocol in Network Simulator (NS-2)*. Version 1. National School of Computer Science of Tunisia. 2009.
- [11]. IETF Internet Engineering Task Force. RFC 4728. The Dynamic Source Routing Protocol (DSR) for Mobile Adhoc Network for Ipv4. 2007.
- [12]. http://en.wikipedia.org/wiki/List_of_wireless_sensor_networks