

KOMPARASI PERFORMASI NODE ROUTER TENGAH DAN NODE ROUTER TEPI PADA JARINGAN MANET MENGGUNAKAN PROTOKOL OLSR DAN BATMAN

Novan Ferdiansyah¹⁾, Nugroho Suharto²⁾, Rachmad Saptono³⁾

^{1),2),3)}Program Studi Jaringan Telekomunikasi Digital, Teknik Elektro, Politeknik Negeri Malang
email: ¹⁾novanxferdiansyah@gmail.com, ²⁾nugroho.suharto@polinema.ac.id

Abstrak

Perkembangan teknologi wireless saat ini telah menciptakan sebuah jaringan baru yaitu jaringan MANET. Pada jaringan MANET sebuah *node* dapat menjadi *router* maupun *end user* karena dalam jaringan MANET *node* bersifat flexibel. Pada jaringan MANET memiliki protokol OLSR dan BATMAN yang bersifat *proactiv*. Saat *node* menjadi *router* maka *node* tersebut akan dilewati oleh banyak *node* lain.

Penelitian ini dilakukan untuk mengetahui peformasi *node router* yang berada di tengah yaitu router yang dilewati oleh banyak komunikasi dan tepi yaitu router yang dilewati sedikit komunikasi dalam jaringan menggunakan parameter *CPU usage*, *RAM usage*, dan *delay processing*. Dari hasil pengujian diperoleh nilai *CPU* pada OLSR dan BATMAN untuk *node router* tengah bernilai 0.3 dan pada *router tepi* bernilai 0.2 saat berkomunikasi, saat tidak ada komunikasi nilai *CPU* pada OLSR tetap bernilai 0.3 sedangkan pada BATMAN nilai BATMAN turun menjadi 0.2. Untuk *memory* yang digunakan pada kedua protokol mendapatkan hasil 0.0 karena router hanya bersifat meneruskan paket tanpa perlu mengolah paket tersebut. Untuk *delay processing* pada kedua protokol mendapat nilai yang bervariasi karena pada pengiriman paket memiliki interval pengiriman yang tidak pasti antar paket. Pada protokol BATMAN dalam penggunaan *CPU* lebih baik dari OLSR karena jika tidak ada komunikasi data nilai *CPU* dari BATMAN turun dan pada OLSR tidak ada komunikasi nilai *CPU* tetap.

Kata Kunci : MANET, OLSR, BATMAN, *CPU usage*, *RAM usage*, *Delay processing*

1. PENDAHULUAN

1.1 Latar Belakang

Komunikasi wireless saat ini sudah sangat berkembang dan sudah memiliki banyak teknologi baru pada dunia komunikasi. Seiring dengan perkembangan teknologi di bidang telekomunikasi dan informasi, dimana hampir semua perangkat komunikasi jaringan nirkabel (*wireless*) yang mampu melibatkan banyak orang atau peralatan komunikasi tanpa ketergantungan suatu infrastruktur dengan mobilitas yang tinggi. Sehingga dibutuhkan jaringan *Mobile Ad-Hoc Network* (MANET) yang semua perangkat mobile yang dapat saling terhubung tanpa infrastruktur tertentu dan setiap *node* bersifat dinamis (bisa menjadi *end user* atau *router*). Berbeda dengan *wireless ad-hoc* pada umumnya, pada jaringan MANET memungkinkan tiap *node* untuk melakukan *routing* seperti fungsi *router* untuk menentukan dan meneruskan jalur

komunikasi antar *node*, maupun menjadi *user* (Nisac, 2017).

Dalam sebuah jaringan MANET terdapat *node* yang digunakan sebagai *router* untuk

menghubungkan antar *node* satu dengan yang lain. Semakin banyak *node* yang terhubung pada router maka data yang melewati *node* tersebut ikut bertambah. Dengan bertambahnya data yang melewati *node router* maka akan berpengaruh pada proses yang terjadi dalam *node router* tersebut.

Dalam jaringan MANET memiliki beberapa protokol, salah satu protokol tersebut adalah protokol *proactive*. Protokol *proactive* adalah dimana pembentukan tabel *routing* dibentuk dan *update* setiap waktu (secara kontinu) jika terjadi perubahan *link*. Dalam protokol *proactive* memiliki banyak algoritma. Pada penelitian ini menggunakan *Optimized Link State Routing* (OLSR) dan *Better Approach to Mobile Ad hoc Network* (BATMAN).

Pada penelitian yang lain dengan judul "Network Management of Mobile Ad hoc Networks (MANET): Architecture, Use Cases, and Applicability" oleh J. Nguyen, dkk yang mengatakan dalam jurnal bahwa protokol dalam jaringan MANET mengkonsumsi CPU dan memory yang banyak (Nguyen, 2013).

Berdasarkan penelitian di atas, penelitian ini dilakukan untuk mengetahui performansi dari *node* akan dilihat berdasarkan *CPU usage*, *RAM usage*, dan *delay process* pada *node router* tengah dan tepi dengan OLSR dan BATMAN saat *node* tersebut dalam keadaan *routing* yang padat dan tidak.

1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat dirumuskan perumusan masalah sebagai berikut:

2. Bagaimana performansi node router tengah pada protocol OLSR dan BATMAN di jaringan MANET ?
3. Bagaimana performansi node router tepi pada protocol OLSR dan BATMAN di jaringan MANET ?

2. TINJAUAN PUSTAKA

2.1 Jaringan Wireless

Teknologi *wireless* (nirkabel) adalah sebuah teknologi pengembangan dari jaringan komputer yang sebelumnya menggunakan kabel sebagai media penghubungnya. *wireless* memanfaatkan udara/gelombang elektromagnetik sebagai media lalu lintas pertukaran datanya. Model jaringan *wireless* terdiri dari dua jenis, salah satunya jaringan ad-hoc. Jaringan ad-hoc merupakan suatu jaringan yang terdiri dari dua atau lebih piranti *wireless* yang berkomunikasi secara langsung satu sama lain. Sinyal yang dihasilkan oleh *interface* adapter Jaringan Wifi adalah berarah Omni keluar ke rentang jangkauan yang dipengaruhi oleh faktor-faktor lingkungan, dan juga sifat dari piranti yang terlibat. (Hanafi, 2013).

2.2 Mobile Ad Hoc Network (MANET)

Mobile Ad Hoc Network (MANET) adalah kumpulan dari beberapa *wireless node* yang dapat di set-up secara dinamis dimana saja dan kapan saja tanpa menggunakan infrastruktur jaringan yang ada (Sun, Jun-Zhao. 2001). MANET juga merupakan jaringan sementara yang dibentuk oleh beberapa *mobile node* tanpa adanya pusat administrasi dan infrastruktur kabel (Suri dkk, 2010). Pada MANET, *mobile node*

yang terhubung dengan *wireless* pada jaringan dapat bergerak bebas dan juga berperan sebagai *router*. *Node-node* dalam jaringan ini bertanggung jawab untuk mencari dan menangani rute ke setiap *node* di dalam jaringan. (Mayhoney, 2008).

2.3 Better Approach To Mobile Ad-Hoc Network (BATMAN)

Better Approach To Mobile Ad-Hoc Network atau *BATMAN* merupakan sebuah *routing* protokol yang bersifat *proactive* yang dikembangkan oleh *Freifunk Community* merupakan pengembangan dari protokol *routing* OLSR. *BATMAN* dikembangkan dengan konsep membentuk sebuah protokol *routing* yang menggunakan informasi *routing* seminimum mungkin dengan hanya mengkalkulasikan *next hop*. Konsep *routing* pada *BATMAN* adalah setiap keputusan *routing* didistribusikan secara merata kepada seluruh *node*. Sehingga setiap *node* memiliki pengetahuan mengenai seluruh *node* yang tersedia beserta total *metric* untuk menuju ke tujuan dan juga *next hop* terbaik untuk mencapai tujuan. Pada *BATMAN*, informasi mengenai perubahan topologi jaringan tidak diperlukan. *BATMAN* melakukan *flooding Originator Message (OGM)* untuk menghindari informasi *routing* yang berbeda sehingga tidak terjadi *routing loop*. *BATMAN* merupakan salah satu protokol *routing* yang banyak dikembangkan dan diuji dalam banyak skenario.

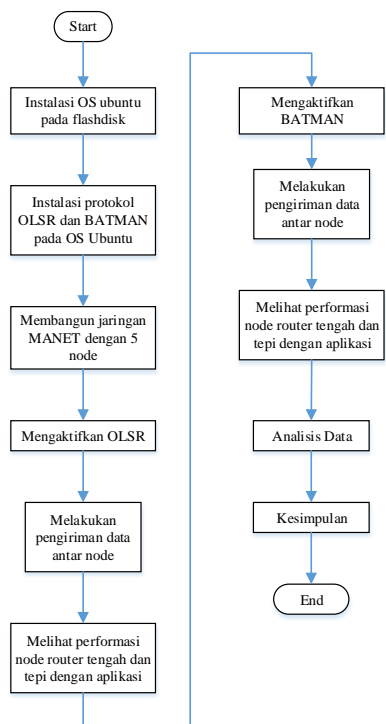
2.4 Optimized Link State Routing Protocol (OLSR)

Optimized Link State Routing Protocol (OLSR) adalah protokol *IP routing* yang dioptimalkan untuk jaringan *mobile adhoc*, yang juga dapat digunakan pada jaringan nirkabel lain. *OLSR* adalah *link-state routing protocol* proaktif, yang menggunakan *hello* dan *control* topologi (*TC*) pesan untuk menemukan dan kemudian menyebarkan informasi *link state* ke seluruh jaringan *mobile adhoc*. *Node* individu menggunakan informasi topologi ini untuk menghitung tujuan *hop* berikutnya untuk semua *node* dalam jaringan menggunakan jalur terpendek *hop forwarding*.

3. METODE PENELITIAN

3.1 Tahapan Penelitian

Rancangan penelitian yang akan dilakukan dalam pembuatan sistem ditampilkan pada Gambar 3.1 dibawah ini.



Gambar 3. 1 Flowchart utama rancangan system

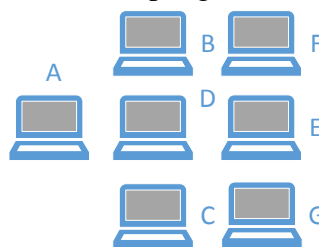
Penjelasan dari *flowchart* pada gambar 3.1 adalah:

1. Tahap Pertama adalah menginstal OS Ubuntu 16 pada *Flashdisk* dan mengandakan menjadi 5 *Flashdisk*.
2. Tahap Kedua adalah menginstal protokol OLSR dan BATMAN pada OS Ubuntu 16 menggunakan *Synaptic Package*.
3. Tahap Ketiga adalah salah satu laptop membuat jaringan ad hoc baru dan 4 laptop/node lain terhubung pada jaringan tersebut.
4. Tahap Keempat adalah mengaktifkan protokol OLSR dengan menulis perintah pada terminal di Ubuntu.
5. Tahap Kelima adalah melakukan komunikasi data dengan saling mengambil data dari laptop lain menurut skema.
6. Tahap Keenam adalah menganalisa hasil dari komunikasi data pada aplikasi wireshark.
7. Tahap ketujuh adalah mengaktifkan protokol BATMAN dengan menulis perintah pada terminal di Ubuntu.
8. Tahap Kelima adalah melakukan komunikasi data dengan saling

mengambil data dari laptop lain menurut skema.

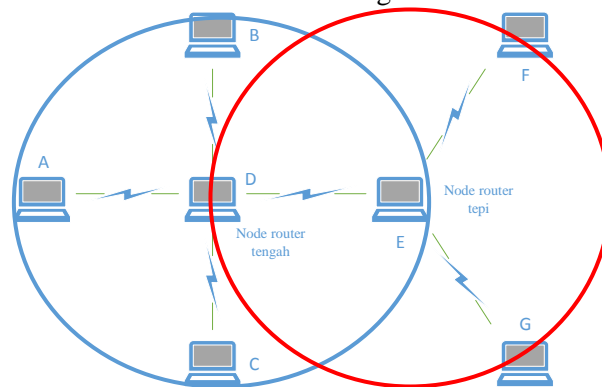
9. Tahap Keenam adalah menganalisa hasil dari komunikasi data pada aplikasi wireshark.
10. Tahap Kesepuluh adalah mengambil kesimpulan dari hasil pengujian.

Blok diagram sistem merujuk dari rumusan masalah yaitu bagaimana diagram rancangan sistem dalam menyelesaikan masalah yang diangkat dalam penelitian. Blok diagram rancangan sistem ditunjukkan pada gambar 3.2 sampai gambar 3.4.



Gambar 3. 2 Posisi awal sebelum pengujian

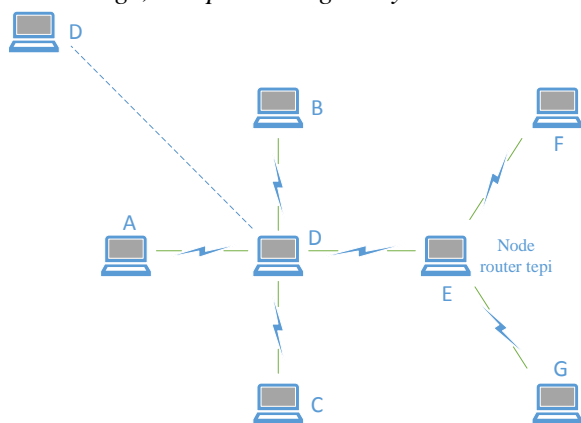
Pada gambar 3.2 adalah posisi awal seriap node sebelum melakukan pengujian. Seluruh node berkumpul terlebih dahulu untuk melakukan konfigurasi. Setelah seluruh node terhubung ke jaringan, menyalakan protokol OLSR dan BATMAN secara bergantian.



Gambar 3. 3 Blok diagram rancangan sistem skema 1

Pada blok diagram rancangan sistem diatas akan di jelaskan mengenai skema 1 dari penelitian yaitu setelah selesai melakukan konfigurasi seluruh node bergerak menjauh dari node D sesuai skema 1. Setelah seluruh node berada di tempat masing-masing dilakukan pengetestan performasi *node* dengan OLSR dan BATMAN dengan melakukan komunikasi antar node. Pengetestan dilakukan dengan cara tiap *node* mengambil data dari node lain dan data

tersebut melewati *node router* tengah dan *node router* tepi. Pengujian ini dilakukan untuk mengetahui pengaruh komunikasi yang padat pada performansi *node router* tengah (*node D*) dan komunikasi yang sedikit pada *node router* tepi (*node E*) dengan parameter *CPU usage*, *RAM usage*, dan *processing delay*.



Gambar 3.4 Blok diagram rancangan sistem skema 2

Untuk Skema 2 dilakukan pengujian menggunakan ping yang packetnya diperbesar untuk menggantikan transfer file. Saat melakukan ping antar node, node D akan bergerak keluar dari jaringan hingga terputus dari jaringan. Setelah terputus node D berhenti sejenak dan berjalan kembali memasuki jaringan ke tempat awal. Pengujian ini dilakukan untuk mengetahui jarak terputusnya node dari jaringan dan mengetahui delay transisi saat node D kembali tersambung ke jaringan.

3.2 Teknik Pengumpulan Data

Data yang didapatkan dari hasil pengujian yang telah dilakukan merupakan data primer. Data primer berisikan informasi atau data yang diperoleh dari pengujian perbandingan kedua protokol routing, yaitu OLSR dan BATMAN pada jaringan MANET yang terdiri dari 7 node. Pengujian parameter yang dilakukan meliputi *CPU usage*, *RAM usage*, dan *delay processing*.

3.3 Analisa Data

Analisis data pada penelitian dilakukan setelah membuat jaringan MANET berhasil dan pengambilan data menggunakan aplikasi *wireshark* untuk melihat *delay processing* dan perintah pada terminal untuk merekam *log CPU* dan *Memory*.

Setelah di dapatkan data *CPU usage*, *RAM usage*, dan *delay processing*, kemudian dimasukkan ke dalam tabel pengujian, dilakukan perhitungan rata – rata tiap parameter

kemudian hasilnya di analisa, kemudian ditarik kesimpulan.

3.4 Bahan dan Alat

Table 3. 1 Bahan Penelitian

NO	Software	Versi
1	OS Linux Ubuntu Desktop	16.04
2	OLSRD	0.6.6.2-1ubuntu1
3	BATMAN - adv	0.3.2-17
4	Wireshark	Version 2.6.1

Table 3. 2 Alat Penelitian

NO	Hardware	Spesifikasi	Jumlah
1	Flashdisk	8/16 GB	5 buah
2	Laptop (node)	Prosesor 64 bit	5 buah

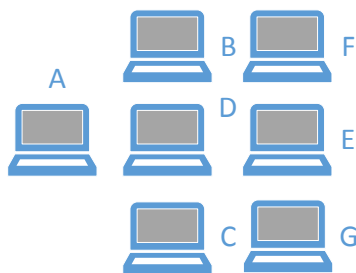
3.5 Penggunaan Sarana

Penggunaan sarana yang digunakan antara lain halaman sekitar Gedung AI dan AH yang cukup luas untuk membangun jaringan MANET, menggunakan laptop untuk node, flashdisk untuk menaruh OS Ubuntu dan protokol OLSR dan BATMAN.

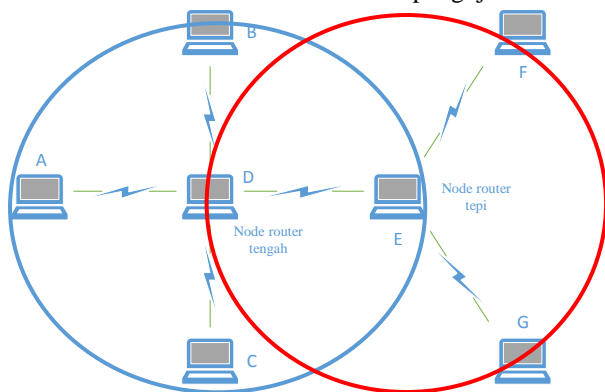
4. Pengujian dan Analisa

4.1 Hasil Implementasi Perancangan Sistem

Implementasi perancangan sistem merupakan penerapan sistem yang telah di buat. Tahap awal implementasi diawali dengan membangun jaringan MANET menggunakan protokol OLSR dan BATMAN. Kemudian dilanjutkan dengan pengujian koneksi menggunakan skema yang telah dibuat untuk mengambil data menggunakan parameter *CPU usage*, *RAM usage*, dan *processing delay* untuk mengetahui performansi *node router* menggunakan protokol OLSR dan BATMAN pada jaringan MANET.



Gambar 4.1 Posisi awal sebelum pengujian



Pada gambar 4.1 ditunjukkan desain *hardware* penelitian, jaringan MANET yang dibangun terdiri dari 7 *node* dimana tiap *node* dalam jaringan saling terhubung melalui koneksi *wireless*. Setiap *node*, menggunakan *flashdisk* yang sudah berisi OS ubuntu 16 yang sudah terinstal protokol OLSR dan BATMAN untuk melakukan *boot system*. Setelah semua laptop sudah menyala dengan OS ubuntu 16, laptop yang menjadi *router* tengah membuat jaringan ad hoc. Setelah semua *node* tersambung pada jaringan ad hoc, semua *node* mengatur IP secara manual mulai dari “192.168.1.1 – 192.168.1.7.”. Setelah semua *node* sudah mengubah IP masing-masing, semua *node* menghidupkan OLSR dengan cara menulis perintah pada terminal “olsrd -i wlo1 -d 2”.

Setelah semua olsr menyala, seluruh *node* bergerak seperti gambar 4.2. Setelah semua *node* berada pada tempat masing-masing, semua *node* menghidupkan wireshark dengan cara menulis pada terminal “sudo wireshark”. Setelah wireshark menyala pilih *interface* wifi(wlo1), secara otomatis wireshark akan mencapture komunikasi yang melewati *node* tersebut. Untuk *capture cpu* dan *memory usage* adalah dengan menulis di terminal perintah “while true; do (echo “%CPU %MEM ARGS \$(date)” && ps -e -o pcpu,pmem,args --sort=pcpu | cut -d" " -f1-5 | tail) >> ps.log; sleep 1; done” secara otomatis pada folder *home* di akan tercipta nama file “ps.log” yang berisikan rekaman dari

penggunaan *cpu* dan *memory* pada *node* tersebut. Pada skema *router* tengah, *node* IP 192.168.1.1 membuka *folder sharring* pada IP 192.168.1.5 dan mengkopi file yang berada di dalam folder tersebut. Di tengah proses mengkopi file, *node* dengan IP 192.168.1.4(*router*) mengkopi file yang berada pada *folder sharring* pada IP 192.168.1.1. Setelah proses mengkopi IP 192.168.1.1 selesai, IP 192.168.1.2 mengkopi file dari *folder sharring* IP 192.168.1.3. Setelah salah satu proses mengkopi selesai *node* IP 192.168.1.3 mengkopi file dari *folder file sharring* pada IP 192.168.1.2. Setelah semua proses pengkopian selesai, hasil *capture* pada wireshark dan ps.log di simpan. Setelah skema *router* tengah untuk OLSR selesai, OLSR di matikan dengan cara terminal yang menjalankan OLSR di *cancel(ctrl+c)*. Setelah itu mengkonfigurasi BATMAN, dengan cara menulis pada terminal “modprobe batman-adv” lalu *enter*, berikutnya “echo bat0 > /sys/class/net/wlo1/batman_adv/mesh_interface” lalu *enter*. Untuk menyalakan BATMAN menulis perintah “batmand -d 1 wlo1”

Setelah protokol BATMAN menyala, seluruh *node* merestart wireshark dan perintah ps.log. Setelah itu melakukan transfer file sama seperti dengan OLSR yaitu IP 192.168.1.1 mengkopi dari IP 192.168.1.5, IP 192.168.1.4 mengkopi dari 192.168.1.1, dan IP 192.168.1.2 dan 192.168.1.3 saling mengkopi. Setelah proses mengkopi selesai simpan hasil *capture* wireshark dan ps.log.

Setelah itu melakukan pengujian *node router* tepi. Pada *node router* tepi hanya menggunakan 4 laptop untuk komunikasi yaitu IP 192.168.1.4 – 192.168.1.7. Pengujian yang dilakukan sama dengan *node router* tengah yaitu OLSR terlebih dahulu setelah itu BATMAN. Untuk komunikasi yang dilakukan yaitu IP 192.168.1.6 mengkopi file dari folder *sharring* di IP 192.167.1.7 dan *node* IP 192.167.1.5(*router* tepi) mengkopi file dari IP 192.168.1.4. Setelah proses pengkopian selesai simpan *capture* wireshark dan ps.log.

Pada skema 2, pertama menyalakan OLSR dan wireshark, setelah itu seluruh *node* bergerak menuju posisi masing-masing. Setelah OLSR dinyalakan dilakukan ping yang datanya sudah ditambah dengan cara menulis “ping -s 1472(jumlah data) (no IP tujuan)” pada terminal. *Node* yang melakukan ping adalah *node* disekitar *node router* tengah yaitu *node* IP 192.168.1.2 ping ke *node* IP 192.168.1.3, *node* IP 192.168.1.1 ping ke *node* IP 192.168.1.5, dan *node* IP 192.168.1.4(*router*) melakukan ping ke

node IP 192.168.1.1. Saat proses ping node router tengah akan bergerak keluar dari jaringan perlahan. Saat perjalanan keluar dari jaringan seluruh node disekitar node router tengah memantau proses ping yang akan terputus pada jarak posisi node router tengah berada di posisi tertentu dan mencatat jarak saat terjadi ping yang putus. Setelah node tengah terputus dari jaringan, node router tengah kembali menuju tempat awal secara perlahan. Saat node router menuju awal, pada wireshark di lihat paket ARP untuk melihat proses transisi node router tengah masuk kedalam jaringan. Setelah node router tengah kembali ke posisi simpan seluruh wireshark. Untuk pengujian BATMAN sama seperti pengujian OLSR hanya mengganti protokol yang digunakan.

4.2 Metode Pengukuran

Metode pengukuran dilakukan untuk mendapat nilai-nilai dari parameter yaitu *CPU usage*, *RAM usage*, dan *processing delay*, dengan menggunakan wireshark dan perintah untuk *capture cpu* dan *memory*. Pengambilan data dilakukan disisi *node router*. Berikut cara mendapat nilai setiap parameter yang diuji :

- a. CPU Usage dan RAM usage

Untuk mendapatkan nilai CPU usage dan RAM usage dapat kita lihat di dalam ps.log yang sudah di capture saat melakukan transfer file sebagai berikut :

```

root@kali:~/j-r-gdell# cat ps.log
#CPU MEMINFO psutil Sat Jul 19 17:34:50 WIB 2018
0.1 0.0 [worker:/s1:0]
0.1 0.0 [worker:/s1:1]
0.1 0.0 [sshd]
0.2 0.0 [rsyncd]
0.2 0.0 [sdd]
0.3 0.0 [lsid]
0.9 0.5 /usr/lib/gnome-terminal/gnome-terminal-server
1.4 0.9 /usr/lib/ncarg/ncarg
2.1 2.4 compiz
#CPU MEMINFO psutil Sat Jul 19 17:35:00 WIB 2018
0.1 0.0 [worker:/s1:0]
0.1 0.0 [worker:/s1:1]
0.1 0.0 [sshd]
0.2 0.0 [rsyncd]
0.2 0.0 [sdd]
0.3 0.0 [lsid]
0.9 0.5 /usr/lib/gnome-terminal/gnome-terminal-server
1.4 0.9 /usr/lib/ncarg/ncarg
2.1 2.4 compiz
#CPU MEMINFO psutil Sat Jul 19 17:35:01 WIB 2018
0.1 0.0 [worker:/s1:0]
0.1 0.0 [worker:/s1:1]
0.1 0.0 [sshd]
0.2 0.0 [rsyncd]
0.2 0.0 [sdd]
0.3 0.0 [lsid]
0.9 0.5 /usr/lib/gnome-terminal/gnome-terminal-server
1.4 0.9 /usr/lib/ncarg/ncarg
2.1 2.4 compiz
#CPU MEMINFO psutil Sat Jul 19 17:35:02 WIB 2018
0.1 0.0 [worker:/s1:0]
0.1 0.0 [worker:/s1:1]
0.1 0.0 [sshd]
0.2 0.0 [rsyncd]
0.2 0.0 [sdd]
0.3 0.0 [lsid]
0.9 0.5 /usr/lib/gnome-terminal/gnome-terminal-server
1.4 0.9 /usr/lib/ncarg/ncarg
2.1 2.4 compiz
    
```

Gambar 4. 1 Tampilan ps.log pada ubuntu 16

- b. Processing delay

Untuk mendapatkan parameter *processing delay* pada aplikasi wireshark dapat dijelaskan sebagai berikut :

No.	Time	Source	Destination	Protocol	Length	Info	File
1	0.000000	192.168.1.1	192.168.1.2	TCP	60	64720 → 64720 [RST] Seq=1921681154 Win=0 Len=0	/usr/share/doc/ncurses5-bin/examples/CPU-MEMINFO.psutil
2	0.000000	192.168.1.2	192.168.1.1	TCP	60	64720 → 64720 [RST] Seq=1921681154 Win=0 Len=0	/usr/share/doc/ncurses5-bin/examples/CPU-MEMINFO.psutil
3	0.000000	192.168.1.1	192.168.1.2	TCP	60	64720 → 64720 [RST] Seq=1921681154 Win=0 Len=0	/usr/share/doc/ncurses5-bin/examples/CPU-MEMINFO.psutil
4	0.000000	192.168.1.2	192.168.1.1	TCP	60	64720 → 64720 [RST] Seq=1921681154 Win=0 Len=0	/usr/share/doc/ncurses5-bin/examples/CPU-MEMINFO.psutil
5	0.000000	192.168.1.1	192.168.1.2	TCP	60	64720 → 64720 [RST] Seq=1921681154 Win=0 Len=0	/usr/share/doc/ncurses5-bin/examples/CPU-MEMINFO.psutil
6	0.000000	192.168.1.2	192.168.1.1	TCP	60	64720 → 64720 [RST] Seq=1921681154 Win=0 Len=0	/usr/share/doc/ncurses5-bin/examples/CPU-MEMINFO.psutil
7	0.000000	192.168.1.1	192.168.1.2	TCP	60	64720 → 64720 [RST] Seq=1921681154 Win=0 Len=0	/usr/share/doc/ncurses5-bin/examples/CPU-MEMINFO.psutil
8	0.000000	192.168.1.2	192.168.1.1	TCP	60	64720 → 64720 [RST] Seq=1921681154 Win=0 Len=0	/usr/share/doc/ncurses5-bin/examples/CPU-MEMINFO.psutil

Gambar 4. 2 Tampilan untuk file yang di melewati node router

Pada gambar 4.5 merupakan tampilan dari hasil yang sudah di filter TCP pada bagian *search* di wireshark dengan menulis “tcp” lalu *enter*. Setelah data di filter sesuai protokol TCP

akan terlihat data yang melewati *router* tersebut dengan melihat pada bagian info. Pada bagian info tertulis “[TCP Retransmission]...” yang berarti data yang di terima oleh *node router* di kirim ulang ke *node* tujuan. Untuk mengetahui dari mana data dan akan dikirim kemana dapat di lihat pada bagian seperti di gambar 4.6 dan 4.7.

Gambar 4. 3 Capture data dari ip 192.168.1.

Gambar 4. 4 Capture data retransmission dari ip 192.168.1.5

Pada gambar 4.7 tercatat data dari *mac address ...dc:d5*(ip 192.168.1.5) melewati *mac address ...c4:d7*(ip 192.168.1.4/node router). Pada gambar 4.7 tercatat data *retransmission* dari *mac address ...c4:d7*(ip 192.168.1.4/node router) dikirim ke *mac address ...be:9d* (ip 192.168.1.1). Setelah itu dilakukan perhitungan *delay* dengan persamaan berikut :

$$\text{Processing Delay} = \text{waktu paket keluar} - \text{waktu paket masuk}$$

Dimana pada *packet* dari ip 192.168.1.5 merupakan *packet* yang datang dan pada *packet(TCP retransmission)* dari ip 192.168.1.4 merupakan *packet* keluar.

4.3 Hasil Pengujian

- a. CPU dan Memory Usage

CPU dan *Memory usage* adalah penggunaan *CPU* dan *memory* pada saat menggunakan sebuah program pada saat itu. Tujuan dari pengujian *CPU* dan *memory usage* adalah untuk melihat pengaruh dari protokol pada saat melakukan transfer data yang padat dan tidak pada *node router*. Dalam penelitian ini mengambil 30 *sample* data pada saat terjadi transfer file antar *node*.

Tabel 4.1 Data sample 30 paket skema router tengah

No.	Skema Router Tengah			
	OLSR		BATMAN	
	CPU	Memory	CPU	Memory
1	0.3	0.0	0.3	0.0
2	0.3	0.0	0.3	0.0
3	0.3	0.0	0.3	0.0
4	0.3	0.0	0.3	0.0
5	0.3	0.0	0.3	0.0
6	0.3	0.0	0.3	0.0
7	0.3	0.0	0.3	0.0
8	0.3	0.0	0.3	0.0

9	0.3	0.0	0.3	0.0
10	0.3	0.0	0.3	0.0
11	0.3	0.0	0.3	0.0
12	0.3	0.0	0.3	0.0
13	0.3	0.0	0.3	0.0
14	0.3	0.0	0.3	0.0
15	0.3	0.0	0.3	0.0
16	0.3	0.0	0.3	0.0
17	0.3	0.0	0.3	0.0
18	0.3	0.0	0.3	0.0
19	0.3	0.0	0.3	0.0
20	0.3	0.0	0.3	0.0
21	0.3	0.0	0.3	0.0
Skema Router Tengah				
No.	OLSR		BATMAN	
	CPU	Memory	CPU	Memory
22	0.3	0.0	0.3	0.0
23	0.3	0.0	0.3	0.0
24	0.3	0.0	0.3	0.0
25	0.3	0.0	0.3	0.0
26	0.3	0.0	0.3	0.0
27	0.3	0.0	0.3	0.0
28	0.3	0.0	0.3	0.0
29	0.3	0.0	0.3	0.0
30	0.3	0.0	0.3	0.0
Mean	0.3	0.0	0.3	0.0

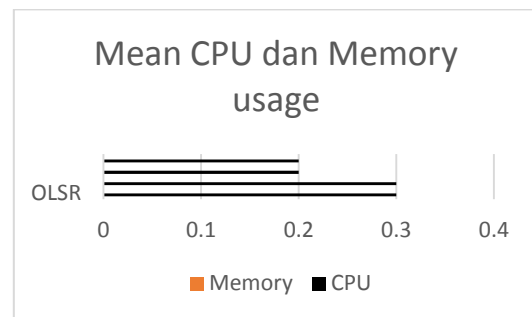
Tabel 4.2 Data sample 30 paket skema router tepi

Skema Router Tepi				
No.	OLSR		BATMAN	
	CPU	Memory	CPU	Memory
1	0.2	0.0	0.2	0.0
2	0.2	0.0	0.2	0.0
3	0.2	0.0	0.2	0.0
4	0.2	0.0	0.2	0.0
5	0.2	0.0	0.2	0.0
6	0.2	0.0	0.2	0.0
7	0.2	0.0	0.2	0.0
8	0.2	0.0	0.2	0.0
9	0.2	0.0	0.2	0.0
10	0.2	0.0	0.2	0.0
11	0.2	0.0	0.2	0.0
12	0.2	0.0	0.2	0.0
13	0.2	0.0	0.2	0.0
14	0.2	0.0	0.2	0.0
15	0.2	0.0	0.2	0.0
16	0.2	0.0	0.2	0.0
17	0.2	0.0	0.2	0.0
18	0.2	0.0	0.2	0.0
19	0.2	0.0	0.2	0.0
20	0.2	0.0	0.2	0.0
21	0.2	0.0	0.2	0.0
22	0.2	0.0	0.2	0.0
23	0.2	0.0	0.2	0.0

24	0.2	0.0	0.2	0.0
25	0.2	0.0	0.2	0.0
26	0.2	0.0	0.2	0.0
27	0.2	0.0	0.2	0.0
28	0.2	0.0	0.2	0.0
29	0.2	0.0	0.2	0.0
30	0.2	0.0	0.2	0.0
Mean	0.2	0.0	0.2	0.0

Berdasarkan dari tabel 4.1 dan 4.2 dapat dilihat nilai *CPU usage* pada skema 1 lebih besar dikarenakan jumlah komunikasi yang melewati *node router* lebih banyak. Nilai *CPU* dan *memory usage* yang kecil dikarenakan *node router* hanya bekerja meneruskan data dengan tanpa harus melihat isi data tersebut. Jadi saat data dikirim, *node router* hanya melihat *header* pada paket untuk melihat kemana paket tersebut akan dikirim. Lalu *node router* meneruskan paket tersebut.

Dari hasil tabel 4.1 dan tabel 4.2 dilakukan perhitungan rata-rata *CPU usage* dan dapat ditunjukkan pada grafik di gambar 4.8 :



Gambar 4. 5 Grafik rata-rata CPU dan Memory usage

Dari hasil grafik diatas OLSR maupun BATMAN memiliki *CPU usage* yang sama yaitu bernilai 0,3 pada skema 1 dan 0,2 pada skema 2. Untuk nilai *memory* juga sama yaitu bernilai 0,0 untuk OLSR dan BATMAN skema 1 maupun skema 2. Pada awal data *CPU* terdapat nilai *CPU* BATMAN turun menjadi 0.2 karena tidak adanya komunikasi.

4.4 Processing Delay

Processing delay adalah lama waktu dari node memproses paket yang diterima oleh node tersebut. Tujuan pengujian processing delay adalah untuk melihat berapa lama paket di proses oleh node router saat terjadi komunikasi yang padat dan tidak. Dalam penelitian ini mengambil rata-rata dari 20 sample data pada proses transfer file.

Tabel 4.3 Tabel rata-rata node router tengah

Node Router Tengah		
No.	OLSR(ms)	BATMAN(ms)
1	0.000014662	0.000024780
2	0.000107739	0.000077909
3	0.000016431	0.000032589
4	0.000014796	0.000027263

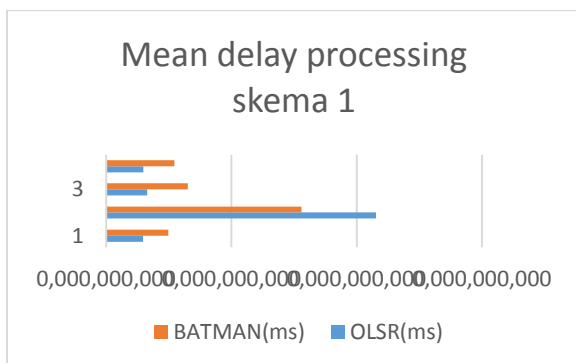
Tabel 4.4 Tabel rata-rata node router tepi

Node Router Tepi		
No.	OLSR(ms)	BATMAN(ms)
1	0.018125308	0.000036075
2	0.001336813	0.127453889

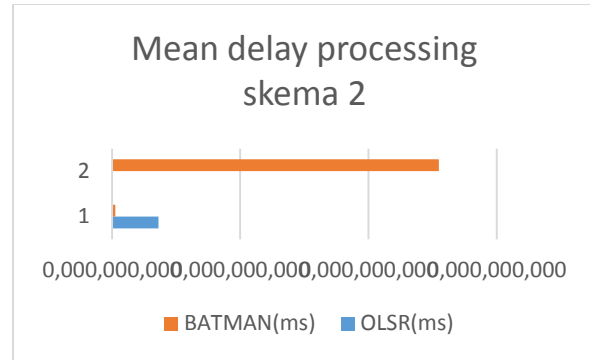
Berdasarkan dari tabel 4.3 dapat dilihat nilai *processing delay* pada skema 1 paling kecil berada pada no 2 yaitu komunikasi antara *node* 192.168.1.1 dan *node* 192.168.1.4 dikarenakan komunikasi ini dilakukan tanpa melewati router/komunikasi direct jadi dalam komunikasi ini tidak terjadi pengiriman ulang yang melewati *node router*.

Berdasarkan dari tabel 4.4 dapat dilihat nilai *processing delay* cukup tinggi pada protokol BATMAN komunikasi yang ke 2 (antara 192.168.1.4 dan 192.168.1.5) dikarenakan saat pengujian komunikasi yang dilakukan cukup bersamaan dan hasil dari pengiriman paket saling bergantian dan menimbulkan *delay* antara paket dan *ack(acknowledge)* cukup jauh.

Dari hasil tabel 4.3 dan tabel 4.4 dilakukan perhitungan rata-rata *CPU usage* dan dapat ditunjukkan pada grafik di gambar 4.9 :



Gambar 4. 6 Grafik Mean dealy processing skema 1



Gambar 4. 7 Grafik Mean delay processing skema 2

Dari hasil grafik diatas *delay processing* sangat berpengaruh pada jarak antara paket yang dikirim dengan *ack(acknowledge)* ataupun dengan paket *retransmisi*.

4.5 Proses Transisi

Proses transisi adalah proses dimana node luar dikenal oleh sebuah jaringan. Tujuan pengujian ini adalah untuk mengetahui proses transisi ketika node router tengah masuk ke dalam jaringan MANET. Dari penelitian ini di dapatkan hasil sebagai berikut :

Tabel 4.5 Tabel proses transisi 1

OLSR berangkat		
jumlah komunikasi	jarak (m)	delay(s)
4	0	0
3	45	71.592546225
2	68	81.663835411
1	94	169.412575863
0	136	348.374958501

Tabel 4.6 Tabel proses transisi 2

OLSR kembali		
jumlah komunikasi	jarak (m)	delay(s)
0	144	548.341774802
1	133	555.024348384
2	80	602.160336742
3	40	773.896372854
4	10	821.433692913

Tabel 4.7 Tabel proses transisi 3

BATMAN berangkat		
------------------	--	--

jumlah komunikasi	jarak (m)	delay(s)
4	0	0
3	50	68.022012152
2	71	86.104453316
1	95	192.622866345
0	140	206.766909159

Tabel 4.8 Tabel proses transisi 4

OLSR kembali		
jumlah komunikasi	jarak (m)	delay(s)
0	144	403.715061265
1	140	436.789119499
2	80	498.159227652
3	40	529.263213762
4	10	548.367242695

Dari data yang proses transisi didapatkan hasil jarak maksimal komunikasi di jarak rata-rata 135 meter meter dari posisi tengah untuk OLSR dan 140 meter untuk BATMAN. Perbedaan jarak di setiap komunikasi dipengaruhi oleh keadaan disekitar tempat pengujian. Untuk hasil delay transisi OLSR bernilai 0.000030851 detik dan BATMAN bernilai 0.000038403 detik saat node router tengah berada di luar jaringan dan berjalan memasuki jaringan.

5. KESIMPULAN

5.1. Kesimpulan

Dari hasil pengujian dan pembahasan yang telah dilakukan maka dapat disimpulkan sebagai berikut :

1. Berdasarkan pada pengujian yang telah dilakukan untuk mengamati performansi

node router tengah didapatkan hasil CPU untuk OLSR 0.3% dan BATMAN 0.3%. Untuk *memory* OLSR dan BATMAN didapatkan hasil 0.0%. Pada pengujian ini didapat kan hasil *delay processing* bervariasi karena pada proses pengiriman paket memiliki jarak interval yang tidak pasti.

2. Berdasarkan pada pengujian yang telah dilakukan untuk mengamati perfarmasi *node router* tepi didapatkan hasil CPU untuk OLSR 0.2% dan BATMAN 0.2%. Pada pengujian ini didapatkan hasil *delay processing* pada BATMAN mengalami *delay processing* yang cukup tinggi karena disaat pengujian pengiriman data bergantian dengan komunikasi yang lain.
3. Pada pengujian didapatkan hasil CPU dan RAM usage yang kecil di karena kan node router hanya bekerja meneruskan paket tanpa harus mengolah paket yang diterima.
4. Pada nilai CPU di BATMAN di dapatkan hasil 0.2% ketika tidak ada komunikasi dan naik menjadi 0.3% ketika terjadi komunikasi. Untuk OLSR nilai CPU nilai tetap berada pada nilai 0.3% dari awal dinyalakan hingga dimatikan.
5. Dari hasil perhitungan *delay processing* hasil yang didapat *delay processing* pada protokol OLSR dan BATMAN sangat bervariasi. Dan pada saat skema 2 protokol BATMAN mendapat *delay processing* yang cukup tinggi yaitu 0.127453889 s karena paket yang dikirim dengan balasan ack memiliki jarak yang cukup jauh, di antara paket dan ack terdapat proses pengiriman paket *node* lain.
6. Dari hasil yang didapat dari pengujian skema 2 saat node router tengah menjauh dari jaringan dan kembali mendapat hasil *delay* transisi OLSR sebesar 0.000030851 detik dan BATMAN sebesar 0.000038403 detik dari tengah/posisi awal.

5.2 Saran

Beberapa poin yang dapat dijadikan saran dalam skripsi ini adalah

1. Melakukan pengujian performansi *node* dengan protokol lain.
2. Menambah jumlah *node* yang lebih banyak.

6. REFERENSI

- arifttegar. (2013, januari 19). pengertian dan fungsi memori cache. Diambil kembali dari attegar.wordpress.com:
<https://attegar.wordpress.com/2013/01/19/pengertian-dan-fungsi-memori-cache/>
- Nisac, H. C. (2017). PERBANDINGAN PERFORMANSI PROTOKOL BATMAN DAN AODV. Malang.
- Amiva Ratna Vitrotulia, Erica Petricia .2013. Komparasi Protokol B.A.T.M.A.N dan OLSR Pada Routing MANET untuk Pengukuran Performasi Jaringan Di Area Politeknik Negeri Malang, Malang.
- Nguyen, J. (2013). Network Management of Mobile Ad hoc Networks (MANET): Architecture, Use Cases, and Applicability. *Internet-Draft*, 16.