

## SEPARATED OPERATION STATE PADA MEDIUM ACCESS CONTROL DI JARINGAN IEEE 802.11AC

Reza Firsandaya Malik, Imam Mustofa<sup>\*)</sup>

Communication Networks and Information Security Research Lab  
Fakultas Ilmu Komputer, Universitas Sriwijaya  
Palembang, Sumatera Selatan

<sup>\*)</sup>Email: [imammustofa25.im@gmail.com](mailto:imammustofa25.im@gmail.com)

### Abstrak

Penelitian ini membahas mengenai bagaimana algoritma *Separated Operation States in Medium Access Control* (SOSMAC) dapat mengatasi *hidden node* di jaringan *Wireless Local Area Network* (WLAN) IEEE 802.11ac dan menghindari *collision*. Skenario jaringan yang dibangun adalah menempatkan 100 *node* dengan topologi grid 10 x 10 yang berkomunikasi dengan 4 *node server* dan menggunakan protokol *routing Ad-hoc on Demand Distance Vector* (AODV) dalam simulasi jaringan *Network Simulator 3* (NS-3). Parameter yang digunakan sebagai acuan analisa pada penelitian ini adalah nilai *throughput*, *packet loss*, dan *packet delivery ratio*. Hasil simulasi SOSMAC yang diperoleh dibandingkan dengan algoritma standar komunikasi yang sudah umum pada jaringan nirkabel yaitu *Medium Access Control Access* (MACA). Hasil perbandingan yang diperoleh adalah nilai rata-rata *throughput* 6,061 Kbps, *packet loss* 0,41%, dan *Packet Delivery Ratio* (PDR) 99,58% yang didapatkan dari algoritma SOSMAC sedangkan pada algoritma MACA diperoleh hasil *throughput* 5,681 Kbps, *packet loss* 7,00%, dan *Packet Delivery Ratio* (PDR) 93,00%. Oleh karena itu, perbandingan hasil tersebut menunjukkan bahwa algoritma SOSMAC lebih baik dari algoritma MACA dalam mengatasi *hidden node* dan *collision*.

**Keywords :** Algoritma SOSMAC, Algoritma MACA, AODV, NS-3

### Abstract

This research discusses how the *Separated Operation States in Medium Access Control* (SOSMAC) algorithm can overcome hidden nodes in IEEE 802.11ac *Wireless Local Area Network* (WLAN) networks and avoid collisions. The network scenario that was built was to place 100 nodes with a 10 x 10 grid topology that communicates with 4 node servers and use *Ad-hoc on Demand Distance Vector* (AODV) routing protocols in *Network Simulator 3* (NS-3) simulations. The parameters used as reference analysis in this study were the value of throughput, packet loss, and packet delivery ratio. The SOSMAC simulation results obtained were compared with the standard communication algorithms that are common on wireless networks, namely *Medium Access Control Access* (MACA). The comparison results obtained were the average throughput value of 6.061 Kbps, 0.41% packet loss, and 99.58% *Packet Delivery Ratio* (PDR) obtained from the SOSMAC algorithm while the MACA algorithm obtained throughput of 5.681 Kbps, packet loss 7, 00%, and *Packet Delivery Ratio* (PDR) 93.00%. Therefore, the comparison of these results shows that the SOSMAC algorithm is better than the MACA algorithm in overcoming hidden nodes and collisions.

**Keywords :** SOSMAC algorithm, MACA algorithm, AODV, NS-3

### 1. Pendahuluan

Bertambahnya jumlah penggunaan *Wireless Local Area Network* (WLAN) menyebabkan kebutuhan akan efisiensi terhadap *throughput* yang menjadi perhatian utama di penelitian [1]. Telah banyak penelitian yang dilakukan untuk mengatasi masalah *throughput*, hal tersebut terlihat dari beberapa perubahan amandemen standar jaringan WiFi atau IEEE 802.11, hingga ke perubahan akhir tahun 2013 yaitu IEEE 802.11ac [2]. Standar ini telah diakui pada tahun 2014 dengan memberikan *data rate* hingga 7 Gbps [3]. Berbeda dengan standar 802.11a yang selalu

menggunakan saluran 20 MHz untuk transmisi data, standar 802.11ac dapat menggunakan saluran 40, 80, atau 160 MHz (opsional) untuk transmisi data [4].

Dalam mentransmisikan data yang diukur melalui *throughput*, *delay* dan *packet loss*, maka jaringan WLAN menggunakan frekuensi yang terbagi dalam beberapa channel. Pembagian *channel* mengikut standar komunikasi nirkabel dan salah satunya adalah IEEE 802.11.

Jaringan WLAN menggunakan dua metode dalam mengakses *channel* yaitu *Distributed Coordination*

*Function* (DCF) dan *Point Coordination Function* (PCF). Penelitian [5] menyatakan bahwa DCF adalah dasar dari teknik *Carrier Sense Multiple Access With Collision Avoidance* (CSMA/CA). Oleh karena semua *node* berbagi *channel* nirkabel bersama dengan *bandwidth* yang terbatas, hal ini diharapkan dapat menciptakan sistem *Medium Access Control* (MAC) yang adil dan efisien. Salah satu penggunaan dari kondisi nirkabel tersebut ialah *Mobile Ad Hoc Network* (MANET), yang merupakan satu himpunan *node* otonom yang terhubung satu sama lain tanpa administrasi pusat [6]. Penelitian [6] menyatakan bahwa ada beberapa faktor yang menyebabkan rusaknya paket data sehingga membutuhkan pengiriman ulang dari *node* asal dan mengurangi nilai *Packet Delivery Ratio* (PDR) dan meningkatkan *packet loss*. Hal ini dapat menyebabkan akan mengurangi efisiensi jaringan jika terdapat banyaknya paket data yang hilang dan harus dikirim ulang ke tujuan. Salah satu penyebab fenomena tersebut adalah terjadinya *collision* antar paket data.

*Collision* adalah salah satu faktor yang menyebabkan hilangnya paket data, *collision* dapat saja terjadi pada jaringan WLAN yaitu ketika sebuah *node* akan mengirim *frame* namun *wireless channel* dalam keadaan sibuk (*busy*) karena adanya pengiriman lain yang sedang terjadi sehingga membuat *node* harus menunggu sampai *channel* memungkinkan untuk pengiriman data atau dalam kondisi *idle*. Salah satu permasalahan muncul ketika keadaan *node* yang tersembunyi (*hidden*) pada keadaan *node* tersebut berkomunikasi dengan sebuah *Access Point* (AP) dalam *single hop* tetapi tidak diketahui *node* lainnya yang juga terhubung ke AP tersebut [7], sehingga *node* ini tersembunyi bagi *node* yang lain. Masalah klasik ini dapat diselesaikan dengan mekanisme *handshake* [7] antara pengirim dan penerima dengan melakukan skema *Request-to-Send/Clear-to-Send* (RTS/CTS) pada metode *Multiple Access with Collision Avoidance* (MACA) [8]. RTS adalah sebuah paket singkat yang dikirim untuk mendapatkan akses jalur komunikasi dan CTS adalah respon dari penerima ketika paket RTS berhasil didapatkan [5]. Mekanisme ini dapat digunakan untuk menghindari *collision* pada skenario *hidden node*, namun bermasalah terhadap *collision* antara pengiriman RTS sendiri yang terjadi dari satu *node* dengan *node* lainnya. *Channel* akan tetap dalam keadaan *clear* (kosong) hanya ketika CTS telah dikirimkan [9].

Skema untuk meningkatkan performa *throughput*, *delay*, *packet loss* dan PDR serta menghindari *collision* pada IEEE 802.11 ac adalah rumusan masalah penelitian ini. Pada umumnya *Separated Operation States in Medium Access Control* (SOSMAC) digunakan untuk jaringan WLAN 802.11 lainnya sebelum IEEE 802.11 ac. Tujuan tersebut akan ditunjang dengan metode SOSMAC untuk mengatasi kondisi jaringan yang padat *channel* di IEEE 802.11ac. Metode MACA digunakan untuk perbandingan hasil pengujian. Penerapannya akan dibuat dalam bentuk simulasi menggunakan *network simulation* atau lebih

dikenal dengan sebutan NS-3. NS-3 adalah simulator *open source* yang bertujuan untuk mensimulasikan semua skenario jaringan nirkabel ataupun tidak [10]. NS-3 menyediakan modul yang terdapat implementasi dari standar 802.11 dan mendukung penggunaan dari DCF di dalam infrastruktur dan topologi *Ad-Hoc* [11].

Protokol *routing* pada penelitian [10] yaitu *Ad-hoc on Demand Distance Vector* (AODV) sebagai metode *routing* yang digunakan dalam penransmisi data di layer 3. Penelitian ini menggunakan protokol AODV sebagai protokol *routing* karena sifatnya yang sederhana, efisien, dan efektif [10]. Penelitian ini diperuntukkan mengetahui kemampuan SOSMAC dalam jaringan WLAN IEEE 802.11ac dalam menghadapi kondisi saluran data yang padat dengan mengubah skema RTS/CTS dan diharapkan akan mempengaruhi beberapa faktor dalam parameter *Quality of Service* (QoS) yaitu *throughput*, *delay*, *packet loss* dan PDR.

Seperti halnya pada penelitian [12] yang membahas mengenai *throughput* dari beberapa protokol *routing* dan *packet loss* yang menjadi dampak dari kehadiran *hidden node* pada penelitian [13] akan dibahas dalam penelitian ini. Parameter *packet delivery ratio* menjadi pelengkap dari parameter analisa oleh karena menyangkut banyaknya paket data yang diterima oleh *server* dan dikatakan bahwa nilai PDR akan berkurang dengan meningkatnya jumlah paket yang datang pada *server* [14]. Dengan hasil yang baik dan relevan dari beberapa parameter tersebut akan menimbulkan kepuasan pada pengguna jaringan dalam pengiriman data meskipun mengalami peningkatan *node* dalam jumlah yang besar.

## 2. MEDIUM ACCESS CONTROL

Lapisan *Medium Access Control* (MAC) menyediakan pengalamatan dan pengaturan *channel access* yang digunakan banyak STA di dalam sistem komunikasi. Sehubungan dengan hal tersebut, IEEE 802.11 memperlihatkan operasi yang terlihat sama dengan IEEE 802.3 (*Ethernet*). Bagaimanapun juga, pengalamatan dan pengaturan *channel access* pada IEEE 802.11 berbeda dari operasi yang ada pada IEEE 802.3, mengingat adanya variasi *channel* pada jaringan *wireless*. Sehingga, jumlah dari fitur baru telah diperkenalkan untuk mengatur mobilitas, *hidden nodes*, dan kondisi *channel*.

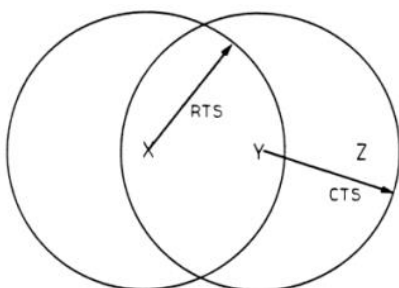
Lapisan MAC melibatkan fungsi dan prosedur untuk mentransfer data antara dua atau lebih *node* dalam jaringan [15]. Dua metode akses dasar yang didefinisikan pada protokol MAC yaitu PCF dan DCF, metode PCF menjadi opsional bagi STA dan merupakan metode *centralized* di atas DCF. DCF adalah algoritma terdistribusi yang membuat setiap *node* akan merasakan *channel* selama suatu periode, sebelum dilakukannya transmisi data. DCF menjadi dasar dari CSMA/CA, sebagaimana ditentukan dalam standar, DCF harus didukung oleh semua *node*

dalam *Basic Service Set* (BSS) [16]. Fitur lain dari lapisan MAC adalah fragmentasi, yaitu mencegah *node* untuk menempati channel dalam waktu lama ketika beroperasi pada tingkat PHY dengan rate yang rendah [9]. Metode pengaksesan lapisan MAC pada standar 802.11ac dapat dibedakan berdasarkan dua bagian utama yaitu *Directional Multi Gigabit Station* (DMG STA) dan non-DMG STA. Salah satu teknik penyebaran spektrum radio pada lapisan PHY dapat digunakan oleh non-DMG STA, sedangkan DMG STA hanya dapat menggunakan DMG PHY sebagai operasi pengaksesan lapisan medium. Untuk metode aksesnya sendiri, non-DMG STA menggunakan metode akses DCF dan protokol yang digunakan adalah CSMA/CD.

A. Multiple Access with Collision Avoidance (MACA)

Metode yang paling terkenal dalam mengatasi masalah *hidden node* adalah *Multiple Access with Collision Avoidance* (MACA) dengan mekanisme *four-way handshake* yang diinisiasi oleh pengirim, yang mana pengiriman paket data dan ACK (*acknowledgement*) didahului oleh paket RTS dan CTS. MACA adalah metode yang terinspirasi dari CSMA/CA (digunakan oleh *Apple localtalk network* untuk sesuatu yang berbeda) dan skema “*prioritized ACK*” yang diusulkan oleh Eric Gustafson [18]. Mekanisme RTS/CTS pada metode MACA dibakukan dalam penelitian [17] dengan secara tegas menamakan MACA yang disematkan dalam judulnya. Berbeda dengan CSMA/CA yang bekerja pada sebagian besar tipe jaringan *Ethernet* yang didasarkan pada *packet contention* [18], MACA diaplikasikan dalam jaringan saluran khusus atau *point-to-point link* dan juga *ad-hoc*.

Mekanisme dari metode MACA yaitu RTS/CTS/Data/ACK akan menjadi metode pembanding dengan metode yang diusung dari penelitian [5] yang menjadi objek utama dari penelitian ini. Oleh karena kedua metode tersebut bekerja pada lapisan yang sama sehingga akan mendapatkan perbandingan yang adil dalam mengatasi masalah yang dihadapi.



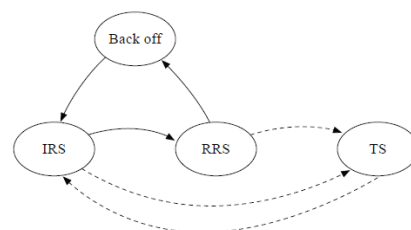
Gambar 1. MACA pada *hidden terminal* [17].

Gambar 1 menerangkan kondisi *hidden terminal* dan terjadi proses *broadcast* paket RTS dan CTS, *node Y* membuat *node Z* menahan transmisinya karena *Y*

mengirimkan paket CTS sebagai respon paket RTS yang didapatkan dari *node X*. *Z* akan merasakan saluran data dalam keadaan sibuk dan akan melakukan perhitungan *backoff* secara acak sampai saluran kembali *idle* sehingga *Z* dapat mengirimkan paket data.

B. Separate Operation State Medium Access Control (SOSMAC)

Skema SOSMAC adalah sebuah siklus yang berisikan dari tiga buah *state* atau interval. Dua bagian untuk *contention* dan satu untuk pengiriman *frame*. Gambar di bawah memperlihatkan siklus dari SOSMAC.



Gambar 2. Diagram Siklus SOSMAC [5].

*Request state* berisikan *Initial Request State* (IRS) dan *Retry Request State* (RRS) untuk membantu mengkoordinasikan transmisi paket dari perangkat, sebelum transmisi paket yang sebenarnya pada bagian *Transmission State* (TS). Pada fase *contention*, ketika sebuah STA dapat meminta akses saluran data untuk melakukan transmisi, permintaan akses tersebut terbagi dalam IRS dan RRS. STA dapat memulai percobaan akses (*contending*) kapanpun dari slot waktu yang ada di IRS, STA yang tidak berhasil di IRS akan melakukan percobaan akses kembali dengan memilih slot waktu secara acak di RRS, *state* ini hanya digunakan untuk bagian percobaan ulang terhadap akses reservasi. Sebagai hasilnya, jumlah keseluruhan dari perangkat yang mengambil bagian di dalam TS mungkin meningkat, hal ini akan menyebabkan peningkatan *throughput*. STA yang gagal pada RRS akan melalui waktu *back-off* seperti pada IEEE 802.11, STA akan memilih masa tunggu sebagaimana jumlah eksponensial dari slot waktu dan akan mencoba akses ke *channel* kembali pada IRS. STA dapat melakukan percobaan akses jika memiliki setidaknya sebuah *frame* untuk dikirimkan, oleh karena itu pada metode ini akan diasumsikan bahwa setiap *frame* yang akan dikirim tersimpan dalam antrian (*queue*). Untuk itu, STA mungkin bersaing untuk slot waktu yang dipilih secara acak di IRS atau RRS setelah menghasilkan sebuah *frame*.

Untuk lebih memahami bagaimana cara skema SOSMAC dalam pengiriman paket data yaitu dengan membagi bagian-bagiannya dengan berbasiskan pada RTS/CTS. Dimulai dengan lingkup IRS, yaitu sebuah STA dengan  $Q > 0$  akan memilih secara acak satu slot waktu  $m$  untuk memulai percobaan (*contend*) di dalam jalur data (*channel*). Dimulai dengan pengiriman RTS untuk

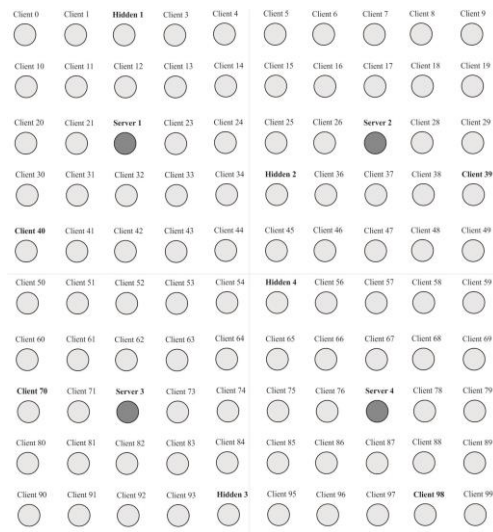
pembukaan slot waktu di dalam TS. Jika permintaan yang dilakukan berhasil, *Base Station* (BS) atau *node* penerima akan merespon dengan sebuah CTS, sebagaimana terjadi di 802.11. Lalu STA pengirim akan melalui TS dan menunggu penetapan slot waktu untuk pengiriman *frame*. Ketidakterhasilan STA di dalam slot waktu pada bidang ini tidak akan melakukan percobaan IRS lagi melainkan melalui satu slot waktu di dalam RRS.

Pada bagian RRS, sebuah STA akan memilih secara acak slot waktu untuk memulai percobaan pengiriman RTS. Sebagaimana di IRS, RTS akan mendapatkan balasan CTS yang dikirimkan oleh BS, termasuk di dalamnya sebuah *id* slot waktu di dalam TS. Kemudian proses akan berpindah ke bidang TS. Jika tidak berhasil proses di dalam RRS, maka waktu tunggu *back-off* akan diperlakukan sampai mendapatkan slot waktu di IRS pada siklus selanjutnya. Pada masa *back-off*, STA dengan RTS yang tidak berhasil di dalam RRS akan menunggu selama waktu slot  $2p$ , nilai  $p$  adalah sama dengan jumlah dari *collision* yang dialami sejak berhasilnya transmisi sebelumnya. Akhir dari masa tunggu dari waktu *back-off*, STA akan berpindah ke proses di dalam IRS lagi. Kemudian bagian terakhir dari siklus yaitu bagian TS, STA yang memiliki jaminan slot waktu menunggu untuk pembukaan (reservasi slot waktu) dan segera akan mengirimkan *frame* ke BS. Akhir dari sebuah transmisi adalah sebuah bidang STA akan menetapkan kembali ke dalam IRS.

### 3. Metodologi

Pemodelan dari sebuah sistem jaringan nirkabel tidak terlepas dengan tolak ukur yang menjadikan simulasi menyerupai kenyataan yang ada. Tolak ukur tersebut antara lain adalah ketersediaan perangkat pengguna jaringan seperti komputer dan *handphone*. Dibutuhkan pula perancangan topologi yang dapat menyusun perangkat-perangkat jaringan untuk saling berkomunikasi dan berbagi data. Topologi yang akan digunakan dalam penelitian ini adalah topologi *ad-hoc grid* 2 dimensi dengan ukuran *grid* 10 x 10 dan jumlah *node* adalah 100.

Bentuk topologi dari penelitian ini dapat dilihat pada gambar 3, pada gambar tersebut masing-masing *node* diposisikan secara diam dan saling berkomunikasi melalui transmisi jaringan *ad-hoc* tanpa adanya *Access Point* (AP). Masing-masing *node* dapat menjadi pengirim paket ataupun sebagai penerima paket, hal ini ditentukan melalui program NS-3 untuk menjadikan satu *node* sebagai *server* dan beberapa *node* lainnya sebagai *client*.



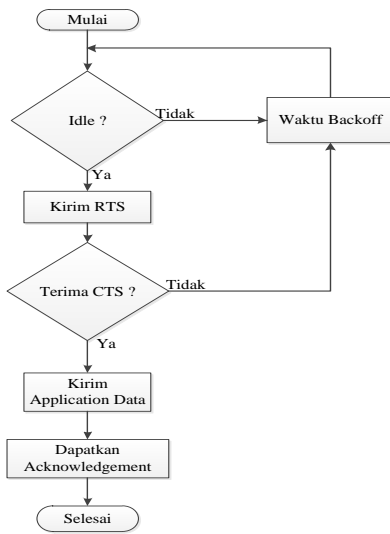
Gambar 3. Topologi Grid 10 x 10.

Tabel 1. Parameter Pengujian.

No	Parameter	Spesifikasi
1	Topologi	Ad-hoc Grid
2	Routing Protocol	AODV
3	Spatial Stream	1
4	Control Mode, Data Mode	MCS 0, MCS 9
5	Jumlah Paket	10 Paket/node client
6	Ukuran Paket	1024 Byte
7	Server Node	Node 22, 27, 72, dan 77
8	Waktu Simulasi	3900 detik

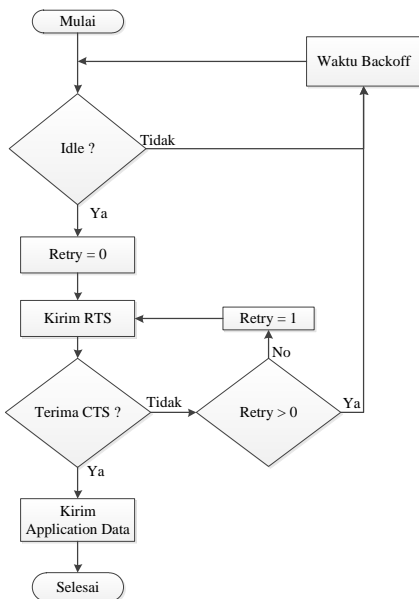
Diketahui dari gambar di atas bahwa terdapat 4 *node server* dengan 4 *hidden node* di tiap *node server*. *Hidden 1* akan mengirimkan paket bersamaan dengan *client 40* ke server 1 sebagai objek masalah *hidden node*. Begitu juga kondisi pada server lainnya agar diperoleh hasil yang relevan terhadap metode yang digunakan.

Masing-masing *node* yang akan mengirimkan data memiliki interval waktu awal yang berbeda. Hal ini dilakukan untuk menghindari *collision* pada program saat awal pengujian. Berdasarkan program yang dibuat penulis, maka ada 10 paket data yang dikirimkan untuk masing-masing *node*, paket data tersebut akan mengisi *traffic network* dan akan membangkitkan data uji dari masing-masing *node*. Jika masing-masing *node* mengirimkan 10 paket data kecuali 4 *node* yang menjadi penerima sehingga akan ada 960 paket data untuk dikirimkan. Setelah keseluruhan data dikirimkan, maka percobaan simulasi selesai. Metode yang digunakan dalam penelitian ini adalah MACA dan SOSMAC. Skema pertukaran pada mekanisme RTS/CTS sebelum pengiriman paket data menjadi sebuah landasan dasar dari algoritma MACA dapat dilihat pada gambar berikut.



Gambar 4. Flowchart algoritma MACA.

Sedangkan gambar dari skema algoritma SOSMAC yang tidak berbeda jauh dari gambaran yang terlihat pada algoritma MACA dapat dilihat pada gambar dibawah ini.



Gambar 5. Flowchart algoritma SOSMAC.

Perulangan yang terjadi pada algoritma SOSMAC diharapkan dapat meningkatkan *throughput* dan mengurangi terjadinya kehilangan data sehingga dapat mempertahankan kualitas jaringan yang baik.

#### 4. Hasil dan Pembahasan

Setelah diperoleh data hasil pengukuran, maka data tersebut dijumlahkan untuk mendapatkan hasil rata-rata perhitungan sebagai hasil akhir dan untuk memudahkan

dalam mengambil kesimpulan dari masing-masing metode yang digunakan.

```

FLOW ID = 23 Src Address 192.168.1.3 Dst Address 192.168.1.23
Tx Packets = 10
Rx Packets = 10
Tx Bytes = 10520
Rx Bytes = 10520
Time Last Receive Packet = +583500164998.0ns
Time First Receive Packet = +570000000000.0ns
Packet Forwarded = 0
Packet Loss = 0
Sum Delay = 0.009573380Detik
Throughput = 6.08789 Kbps
-----
FLOW ID = 24 Src Address 192.168.1.41 Dst Address 192.168.1.23
Tx Packets = 10
Rx Packets = 0
Tx Bytes = 10520
Rx Bytes = 0
Time Last Receive Packet = +0.0ns
Time First Receive Packet = +571000000000.0ns
Packet Forwarded = 0
Packet Loss = 10
Sum Delay = 0Detik
Throughput = -0 Kbps
    
```

Gambar 6. Data *hidden node* dan *client node* algoritma MACA

Gambar di atas memperlihatkan bagaimana hasil dari *id 23 (hidden node 1)* kehilangan 10 paket dan *id 24 (client 40)* berhasil mengirimkan seluruh paket data ke server meskipun pengiriman dilakukan secara bersamaan dan pada jarak yang berjauhan.

Meskipun di tiap *node server* dihubungkan dengan pengiriman paket data dengan jumlah yang sama yaitu dari tiap 24 *node client*, ringkasan hasil data yang diperoleh akan berbeda oleh karena interferensi dari masing-masing *node* dan jumlah *collision* yang berbeda. Berikut adalah hasil data rata-rata di tiap *node server*.

Tabel 2. Nilai rata-rata pengukuran *node server 22* Algoritma MACA.

No	Parameter	Nilai Rata-rata
1	Throughput	5,681 Kbps
2	Packet Loss	7,00%
3	Packet Delivery Ratio	93,00%

Nilai *throughput* di atas menggambarkan nilai rata-rata yang berdasarkan pengiriman 10 paket data dari tiap *node client* dan merupakan besaran kbps di masing-masing pengiriman yang dilakukan. Dengan nilai sebesar 7,00% pada parameter kedua yaitu *packet loss* yang merepresentasikan tingkat persentase kegagalan paket data, maka dapat dikatakan bahwa adanya paket data yang gagal dikirimkan. Dengan adanya paket data yang gagal, maka nilai *packet delivery ratio* tidak akan sampai 100,00% dan hal ini di pastikan dengan nilai parameter *packet delivery ratio* pada tabel 2 sebesar 93,00%.

Nilai yang berbeda kemudian didapatkan pada tabel 3 dengan data pengiriman 24 *node client* menuju *node server 27* dengan parameter yang sama yang juga digunakan pada tabel berikut agar memudahkan proses analisa yang dilakukan. Untuk lebih jelas mengenai perbedaan data tersebut, berikut ditampilkan tabel 3 dengan data dari tiga parameter pengujian.

Tabel 3. Nilai rata-rata pengukuran *node server 27* Algoritma MACA.

No	Parameter	Nilai Rata-rata
1	Throughput	5,532 Kbps
2	Packet Loss	9,58%
3	Packet Delivery Ratio	90,42%

Nilai rata-rata yang didapatkan pada tabel di atas menunjukkan penurunan performansi jika dibandingkan dengan keadaan nilai tabel sebelumnya. Hal tersebut terjadi karena adanya kegagalan pengiriman data dari beberapa *node client* yang ditunjukkan dari catatan aliran data yang menuju *node server 27*. *Throughput* yang tercatat di dalam tabel sebelumnya adalah 5,681 Kbps, sedangkan pada tabel di atas nilai *throughput* yang didapat adalah 5,532 Kbps. Terdapat selisih angka sebesar 0,149 Kbps yang membuktikan perbedaan performansi jaringan yang digunakan. Terdapat pula perbedaan nilai data dari parameter *packet loss* sebesar 2,58%, peningkatan yang terjadi pada data *packet loss* adalah gambaran dari bertambahnya paket data yang hilang selama pengiriman berlangsung.

Parameter ketiga juga menegaskan bahwa adanya paket data yang tidak sampai ke penerima. Berkurangnya jumlah paket data yang diterima oleh *node server 27* dilihat dari nilai *packet delivery ratio* yang tidak sampai 100,00%, yaitu ada pada nilai 90,42%. Nilai ini lebih kecil 2,58% dari nilai *packet delivery ratio* yang diperoleh *node server 22*. Turunnya performansi jaringan ini di akibatkan adanya *collision* dan bertambahnya beban jaringan di dalam satu topologi. Penurunan data tersebut juga terjadi pada *node server 72* yang akan diuraikan pada tabel berikut ini.

Tabel 4. Nilai rata-rata pengukuran *node server 72* Algoritma MACA.

No	Parameter	Nilai Rata-rata
1	Throughput	4,768 Kbps
2	Packet Loss	22,00%
3	Packet Delivery Ratio	78,00%

Jika diperhatikan pada tabel di atas, maka dapat diketahui bahwa jumlah paket data yang hilang atau tidak sampai pada penerima bertambah banyak. Oleh karena hal tersebut yang mengakibatkan turunnya nilai rata-rata dari parameter *throughput* dan *packet delivery ratio* serta meningkatnya nilai *packet loss*. Nilai rata-rata *throughput* yang diperoleh dari tabel 4 adalah sebesar 4,768 Kbps.

Berbeda dengan nilai *packet loss* yang meningkat menjadi 22,00% jika dibandingkan dengan data informasi dari dua tabel sebelumnya. Jika nilai *packet loss* meningkat, maka nilai *packet delivery ratio* akan menurun dan data rata-rata yang diperoleh untuk parameter ini adalah 78,00%.

Tabel 5. Nilai rata-rata pengukuran *node server 77* Algoritma MACA.

No	Parameter	Nilai Rata-rata
1	Throughput	2,105 Kbps
2	Packet Loss	68,00%
3	Packet Delivery Ratio	32,00%

*Collision* yang terjadi pada jaringan yang terhubung pada *node server 77* diakibatkan oleh banyaknya paket data yang berlalu-lalang di dalam topologi yang dibangun. Dengan adanya *hidden node* membuat pengiriman data terganggu dan menyebabkan kegagalan pengiriman data. Hal ini menyebabkan rendahnya nilai *throughput* dari hasil pengukuran yang dilakukan, meningkatnya nilai *packet loss*, dan juga rendahnya nilai dari *packet delivery ratio*. *Throughput* yang diperoleh pada tabel 5 adalah sebesar 2,105 Kbps.

Data ini menurun dan merupakan data terkecil jika dibandingkan dengan data pada tabel-tabel sebelumnya, yaitu dengan selisih angka 3,576 Kbps, 3,427 Kbps, dan 2,663 Kbps pada tabel 2, 3, dan 4. Sebaliknya dengan penurunan data tersebut, peningkatan terjadi pada parameter *packet loss* yang merupakan representasi persentase kehilangan paket data.

*Packet loss* yang didapatkan dari pengukuran simulasi yang dijalankan adalah sebesar 68,00%. Nilai *packet delivery ratio* yang diperoleh dari pengukuran pada *node server 77* adalah sebesar 32,00%. Penurunan nilai data yang terjadi memperlihatkan kapasitas dari kemampuan jaringan yang dibangun, intervensi dari masing-masing *node*, *collision*, dan kondisi *hidden node* menjadi poin-poin dasar yang berakibat pada perbedaan nilai data hasil pengukuran di tiap parameter yang digunakan.

```

FLOW ID = 23 Src Address 192.168.1.3 Dst Address 192.168.1.23
Tx Packets = 10
Rx Packets = 10
Tx Bytes = 10520
Rx Bytes = 10520
Time Last Receive Packet = +583500164998.0ns
Time First Receive Packet = +570000000000.0ns
Packet Forwarded = 0
Packet Loss = 0
Sum Delay = 0.009573380Detik
Throughput = 6.08789 Kbps
-----
FLOW ID = 24 Src Address 192.168.1.41 Dst Address 192.168.1.23
Tx Packets = 10
Rx Packets = 10
Tx Bytes = 10520
Rx Bytes = 10520
Time Last Receive Packet = +584500165002.0ns
Time First Receive Packet = +571000000000.0ns
Packet Forwarded = 0
Packet Loss = 0
Sum Delay = 0.010364800Detik
Throughput = 6.08789 Kbps
    
```

Gambar 7. Data *hidden node* dan *client node* algoritma SOSMAC

Kedua sampel yang juga digunakan pada algoritma pertama ditampilkan pada gambar di atas dengan perbedaan nilai berdasarkan beberapa poin. Dimana kedua sampel di atas terlihat dapat mengirimkan kesepuluh data yang ditentukan dalam sistem yang dibangun. Tidak terjadi kegagalan pengiriman seperti pada percobaan sistem dengan algoritma pertama.

Tabel 6. Nilai rata-rata pengukuran node server 22 Algoritma SOSMAC.

No	Parameter	Nilai Rata-rata
1	Throughput	6,061 Kbps
2	Packet Loss	0,41%
3	Packet Delivery Ratio	99,58%

Nilai *throughput* yang didapatkan setelah diambil nilai rata-rata dari 24 *node client* adalah 6,061 Kbps, sedangkan nilai *packet loss* yang diperoleh adalah 0,41%, dan data parameter terakhir yang digunakan adalah *packet delivery ratio* dengan nilai ringkasan data sebesar 99,58%.

Tabel 7. Nilai rata-rata pengukuran node server 27 Algoritma SOSMAC.

No	Parameter	Nilai Rata-rata
1	Throughput	6,086 Kbps
2	Packet Loss	0,00%
3	Packet Delivery Ratio	100,00%

Berbeda dari data sebelumnya, nilai rata-rata dari 24 *node client* yang mengirimkan data ke *node server* 27 adalah sebesar 6,086%, dengan nilai *packet loss* 0,00%, dan nilai *packet delivery ratio* 100,00%. Hal ini menyatakan bahwa tidak adanya paket data yang tidak terkirim dari 24 *node client* yang mengirimkan ke *node server* 27.

Tabel 8. Nilai rata-rata pengukuran node server 72 Algoritma SOSMAC.

No	Parameter	Nilai Rata-rata
1	Throughput	5,833 Kbps
2	Packet Loss	4,16%
3	Packet Delivery Ratio	95,83%

Jika dibandingkan dengan dua tabel ringkasan data sebelumnya, maka pada data tabel di atas memiliki penurunan data yang cukup berbeda yaitu nilai *throughput* 5,833 Kbps didapatkan, dengan nilai *packet loss* sebesar 4,16%, dan *packet delivery ratio* 95,83%. Dengan demikian, pada *node server* 72 terdapat beberapa pengiriman yang gagal dikirimkan oleh *node client* sehingga menurunnya data hasil ringkasan yang didapatkan.

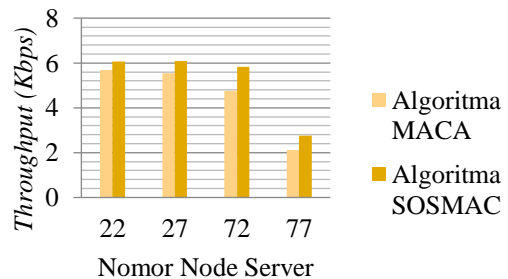
Tabel 9. Nilai rata-rata pengukuran node server 77 Algoritma SOSMAC.

No	Parameter	Nilai Rata-rata
1	Throughput	2,750 Kbps
2	Packet Loss	61,00%
3	Packet Delivery Ratio	39,00%

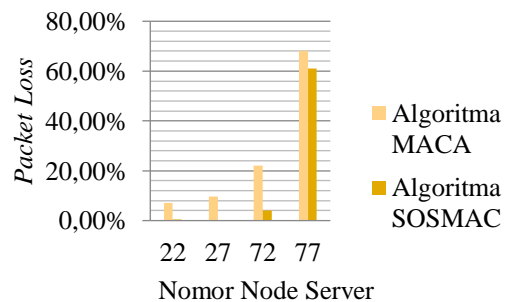
Data terakhir yang disajikan dalam tabel adalah data dari *node client* yang mengirimkan paket data ke *node server* 77. Terlihat bahwa nilai *throughput* menurun cukup jauh dengan hasil 2,75 Kbps, *packet loss* 61,00%, dan *packet delivery ratio* 39,00%. Perbedaan hasil yang sangat kecil dari data pada tabel sebelumnya dari *node server* 22, 27,

dan 72. Terdapat banyaknya aliran data yang membebani jaringan membuat performa jaringan semakin memburuk yang terlihat dari kecilnya data pada *node server* 77.

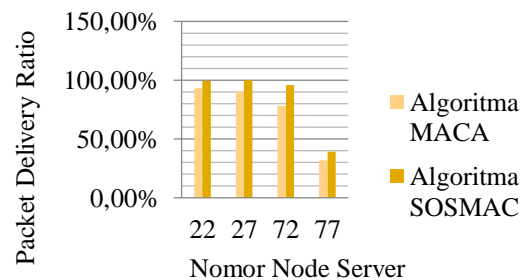
Setelah didapatkan data hasil pengukuran, maka data tersebut dapat dievaluasi sebagai dalam bentuk grafik perbandingan data. Grafik tersebut dapat dilihat sebagai berikut.



Gambar 8. Grafik Perbandingan Throughput .



Gambar 9. Grafik Perbandingan Packet Loss.



Gambar 10. Grafik Perbandingan Packet Delivery Ratio.

Dari tiga data perbandingan yang disajikan dalam gambar grafik di atas memperlihatkan perbedaan kualitas yang diberikan oleh algoritma SOSMAC lebih baik dari pada yang diberikan oleh algoritma MACA. Nilai *throughput* dan *packet delivery ratio* dari algoritma SOSMAC lebih tinggi dari algoritma MACA, sedangkan *packet loss* dari algoritma SOSMAC lebih rendah dari algoritma MACA.

## 5. Kesimpulan

Pengaruh *hidden node* terhadap jaringan nirkabel adalah ketika pengiriman data dilakukan secara bersamaan pada *node server* yang sama sehingga mengakibatkan penurunan nilai *throughput* dan *packet delivery ratio* serta peningkatan nilai *packet loss*. Protokol IEEE 802.11 ac sebagai standar komunikasi nirkabel memiliki karakteristik yang berbeda dengan standar IEEE 802.11 lainnya. Pada umumnya, algoritma MACA digunakan untuk mengatasi masalah *hidden node* pada jaringan nirkabel IEEE 802.11 dengan berdasarkan konsep CSMA/CA. Algoritma SOSMAC diusung sebagai pilihan berbeda untuk mengatasi masalah yang sama dengan kualitas yang lebih baik ketika diimplementasikan pada IEEE 802.11ac.

Pada penelitian ini, penerapan algoritma SOSMAC dan MACA dengan standar 802.11ac, masih terdapat paket data yang gagal dikirimkan dari *node client* menuju *node server*. *Node client* 40 yang mengirimkan paket data bersamaan *hidden node* melakukan komunikasi pada *node server* tercatat mengalami kegagalan 10 pengiriman paket data. Dari hasil simulasi yang diperoleh, algoritma SOSMAC yang juga digunakan untuk mengatasi masalah *hidden node* pada jaringan yang padat dapat mempertahankan 10 paket data pada *node client* 40 dan *hidden node* yang berkomunikasi dengan *node server* dalam topologi *grid* dibandingkan dengan algoritma MACA.

Hasil perbandingan yang diperoleh adalah nilai rata-rata *throughput* 6,061 Kbps, *packet loss* 0,41%, dan *Packet Delivery Ratio (PDR)* 99,58% yang didapatkan dari algoritma SOSMAC sedangkan pada algoritma MACA diperoleh hasil *throughput* 5,681 Kbps, *packet loss* 7,00%, dan *Packet Delivery Ratio (PDR)* 93,00%. Secara keseluruhan dapat disimpulkan bahwa algoritma SOSMAC dapat memperbaiki performansi QoS di jaringan WLAN IEEE 802.11ac.

## Referensi

- [1] T. Sanada, X. Tian, T. Okuda, and T. Ideguchi, "Estimating the Number of Nodes in WLANs to Improve Throughput and QoS," *IEICE Transactions on Information and Systems*, no. 1, pp. 10–20, 2016.
- [2] H. A. Omar, K. Abboud, N. Cheng, K. R. Malekshan, A. T. Gamage and W. Zhuang, "A Survey on High Efficiency Wireless Local Area Networks: Next Generation WiFi," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2315-2344, Fourthquarter 2016.
- [3] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "Impact of IEEE 802.11n/ac PHY/MAC High Throughput Enhancements on Transport and Application Protocols-A Survey," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2050–2091, 2017.
- [4] B. Ramsey, J. Fuller and C. Badenhop, "Efficacy of physical layer preamble manipulation for IEEE 802.11a/ac," *IEEE Electronics Letters*, vol. 52, no. 8, pp. 669-671, 14 4 2016.
- [5] P. K. Esubonteng and R. Rojas-Cessa, "SOSMAC: Separated operation states in Medium Access Control for emergency communications on IEEE 802.11-like crowded networks," *2017 26th Wireless and Optical Communication Conference (WOCC)*, Newark, NJ, 2017, pp. 1-6.
- [6] A. S. AL-Khaleefa, M. R. Ahmad, R. C. Muniyandi, R. F. Malik, and A. A. M. Isa, "Optimized authentication for wireless body area network," *J. Telecommun. Electron. Comput. Eng.*, vol. 10, no. 2, 2018.
- [7] Yu Wang and J. J. Garcia-Luna-Aceves, "Performance of collision avoidance protocols in single-channel ad hoc networks," *10th IEEE Int. Conf. Netw. Protoc. 2002. Proceedings.*, pp. 68–77, 2002.
- [8] B. Mawlawi, J. Doré, N. Lebedev and J. Gorce, "Multiband CSMA/CA with RTS-CTS strategy," *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Larnaca, 2014, pp. 628-633.
- [9] I. Selinis, "Performance study of 802 . 11n WLAN and MAC enhancements in ns-3," *Master Thesis*, 2014.
- [10] R. F. Malik, M. S. Nurfatih, H. Ubaya, R. Zulfahmi and E. Sodikin, "Evaluation of greedy perimeter stateless routing protocol on vehicular ad hoc network in Palembang city," *2017 International Conference on Data and Software Engineering (ICoDSE)*, Palembang, 2017, pp. 1-5.
- [11] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation," *J. Netw. Comput. Appl.*, pp. 1–11, 2014.
- [12] K. Zafar, S., Tariq, H., & Manzoor, "Throughput and Delay Analysis of AODV, DSDV and DSR Routing Protocols in Mobile Ad Hoc Networks.," *Int. J. Comput. Networks Appl.*, vol. 3, no. 2, pp. 1–7, 2016.
- [13] Y. Yao, K. Zhang, and X. Zhou, "A Flexible Multi-Channel Coordination MAC Protocol for Vehicular Ad Hoc Networks," *IEEE Commun. Lett.*, vol. 21, no. 6, pp. 1305–1308, 2017.
- [14] D. N. M. Dang, V. D. Nguyen, H. T. Le, C. S. Hong, and J. Choe, "An efficient multi-channel MAC protocol for wireless ad hoc networks," *Ad Hoc Networks*, vol. 44, pp. 46–57, 2016.
- [15] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: A survey," *Ad Hoc Networks*, vol. 4, no. 3, pp. 326–358, 2006.
- [16] Haitao Wu, Shiduan Cheng, Yong Peng, Keping Long and Jian Ma, "IEEE 802.11 distributed coordination function (DCF): analysis and enhancement," *2002 IEEE International Conference on Communications*, New York, NY, USA, 2002, pp. 605-609 vol.1.
- [17] P. Karn, "MACA - A New Channel Access Method for Packet Radio." *9th ARRL Computer Networking*, London, Ontario, Canada, pp. 134–140, 1990.
- [18] J. Tourrilhes, "Packet frame grouping: improving IP multimedia performance over CSMA/CA," *ICUPC '98. IEEE 1998 International Conference on Universal Personal Communications*, Florence, Italy, 1998, pp. 1345-1349 vol.2.