

Evaluasi *Carrier Sense Multiple Access/Collision Avoidance* Berbasis *Opportunistic Random Access*

Reza Firsandaya Malik

*Jurusan Sistem Komputer
Universitas Sriwijaya
Palembang, Indonesia
rezafm@unsri.ac.id*

Erick Okvanty Haris

*Jurusan Sistem Komputer
Universitas Sriwijaya
Palembang, Indonesia
erickokvantyharis@gmail.com*

Abstrak-Makalah ini memaparkan mengenai pengiriman data dengan proses CSMA/CA berbasis *Opportunistic Random Access*. Adapun yang menjadi latar belakang penulisan ini karena sering adanya proses pengiriman data yang hilang sebelum sampai atau *packet lost*. Evaluasi yang dilakukan menggunakan tools Omnet++. Skenario yang dilakukan dengan menggunakan 2 jenis node dan 2 jenis variasi kecepatan. Parameter – parameter *Quality of Service* seperti *Delay*, *Throughput*, *Jitter*, dan *Routing Overhead* digunakan dalam evaluasi kinerja algoritma *Opportunistic Random Access* dalam proses CSMA/CA. Hasil simulasi didapatkan bahwa proses CSMA/CA menggunakan *Opportunistic Random Access* memberikan efisiensi terhadap waktu dan efisiensi paket terhadap proses yang hanya menggunakan CSMA/CA saja.

Kata kunci : CSMA/CA; *Opportunistic Random Access*; *Quality of Service*; Omnet++

I. PENDAHULUAN

Teknologi nirkabel berkembang pesat pada era sekarang ini, proses teknologi nirkabel ini juga memiliki kelemahan dan kekurangan pada saat melakukan proses pengiriman data, seperti halnya tabrakan paket, pengiriman yang salah dan sebagainya [1]. Dua hal penting yang utama pada jaringan nirkabel adalah komunikasi akhir dari koneksi nirkabel pada perangkat pengguna yang merubah node tambahan pada jaringan. Ditambahkan oleh Tang, C dkk [2], bahwasanya jaringan nirkabel saat ini sangat cepat, memiliki jangkauan yang luas, dan lebih dapat diandalkan dari teknologi sebelumnya karena akan dapat dengan mudah dilakukan peretasan, menyebabkan kesalahan pada bermacam bentuk dari tampilan antarmuka, dan dapat menolak (drop) perangkat yang belum dilakukan pemasangan yang

semestinya terhadap nirkabel. Implementasi dari jaringan nirkabel salah satunya adalah IoT (*Internet of Things*). IoT (*Internet of Things*) adalah suatu contoh dari penerapan internet yang terdiri dari banyak teknologi yang terinterkoneksi, seperti menurut [3], IoT ini dapat disimpulkan secara garis besar merupakan suatu gabungan menjadi suatu kesatuan diantara sensor sebagai pembaca data, dan merupakan koneksi internet dengan satu atau beberapa jenis topologi dalam jaringan yang digunakan. Sebelumnya, Seperti yang di paparkan oleh [4], IEEE merumuskan suatu mekanisme dimana sebelum suatu node akan melakukan pengiriman data, pertama-tama akan dipastikan terlebih dahulu bahwa media sedang tidak dipakai (tersedia) dengan melakukan pengiriman frame broadcast secara berkala dan kontinyu untuk memeriksa kondisi apakah media mungkin siap melakukan transfer data, jika dari hasil media telah dianggap clear maka node akan memulai mekanisme handshaking dengan melakukan pengiriman RTS (*Request To Send*) ke node tujuan setelah menerima CTS (*Clear To Send*) dari node tujuan maka proses transfer data akan dilakukan. Frame-frame tersebut tersebut yang berisi data akan mulai dikirim ke tujuan.

Di paparkan sebelumnya oleh [1], bahwasanya salah satu cara mengatasi tabrakan data tersebut adalah menggunakan metode Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) ini merupakan metode multiple access yang diterapkan pada jaringan nirkabel untuk mendeteksi tabrakan data dan menghindarinya, sehingga meningkatkan kinerja. Dalam pemaparannya tersendiri, [3] menjelaskan mengenai prinsip kerja CSMA/CA itu sendiri bahwasanya CSMA/CA disini digunakan dalam ranah metode *Opportunistic Random access*, yang mana proses ini di diusulkan

oleh [4], yang menerapkan metode mengenai keanekaragaman node dalam WLAN . Proses dengan metode ini berfokus kepada penjadwalan yang menguntungkan untuk traffic pada node untuk setiap kondisi node lainnya . Dengan proses ini maka akan didapatkan node dengan kondisi terbaik untuk dapat dijadwalkan untuk dikirim [1,3], skema ini memungkinkan pengguna dengan kondisi channel yang terbaik memiliki kemungkinan terbesar untuk mengakses channel. Dalam jangka Panjang setiap pengguna memiliki kemungkinan memperoleh *Throughput* yang cocok atau sebanding dengan kondisi channelnya [6]. Kemudian hasil akhir dari penerapan bahwasanya Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) ini dapat membantu memberikan gambaran mengenai performa jaringan dengan menampilkan dengan jelas *Throughput, Delay, dan jitter* dari protokol CSMA/CA.

II. CARRIER SENSE MULTIPLE ACCESS / COLLISION AVOIDANCE

Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) adalah metode multiple access yang diterapkan di jaringan *Nirkabel* untuk mendeteksi tabrakan data dan menghindarinya. CSMA/CA sering digunakan dibanyak jaringan nirkabel karena bertugas dapat mengurangi benturan data dan meningkatkan kinerja secara keseluruhan. Selain itu [5] pun menyebutkan hal ini memungkinkan seluruh pengguna memiliki peluang yang sama dalam mengakses saluran nirkabel.

Proses CSMA/CA dimulai dengan mempersiapkan *frame data* kemudian mengirimkan broadcast yang berguna untuk mengetahui apakah kanal tersedia untuk melakukan pengiriman data, apabila kanal tidak tersedia maka mengirimkan informasi berupa SIFS dan random backoff time untuk menunggu sampai proses pengiriman siap dilaksanakan namun apabila kanal tersedia maka akan dikirimkan sinyal yang berupa RTS untuk sebagai sinyal broadcast yang akan dilanjutkan dengan menerima umpan balik berupa sinyal CTS , apabila sinyal CTS telah sampai dan diterima maka akan langsung dikirimkan setelah sampai ,maka akan dikirimkan kembali sinyal SIFS kepada node pengirim kepada node penerima sebagai tanda paket telah selesai dikirimkan.

A. Opportunistic Random Access (Opportunistik akses acak)

Dari proses yang menggunakan Opportunistic Random Access ini digunakan tiga algoritma berdasarkan mekanisme dari *contention*

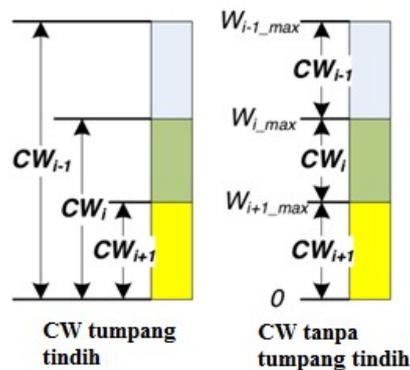
pada CSMA/CA, [3][5] menyebutkan dua dari ketiga algoritma ini ditargetkan untuk mencapai perhitungan contention window dengan tujuan untuk masing masing node berdasarkan kondisi kanal sesaat.

1) Overlaped Contention (Tumpang tindih)

Pada pendekatan pertama, *Contention Window* dari semua node menggunakan batas bawah yang sama dari "0" seperti CSMA/CA. namun memiliki batas awal dan batas akhir yang berbeda yang ditentukan oleh kondisi saluran dalam hal kecepatan bit yang dicapai [7]. Oleh karena itu contention window dari semua node terjadi tumpang tindih atau *overlapped* pada range, rumus 1 dibawah ini ditunjukkan sebuah node baru yang siap untuk disaingkan pada kanal transmisi baru.

$$CW = \left[\alpha \times \frac{Rb}{Ri} \times CWbase \right] \quad [1]$$

Dari formula yang tertera maka bit rate tinggi yaitu kondisi dimana kanal yang baik dan mengarah ke CW dan dengan demikian kemungkinan yang lebih besar untuk memenangkan perselisihan kanal dengan backoff yang seragam. Kemudian CW yang dihitung bisa digunakan untuk prosedur *Binary Exponential back-off* di CSMA/CA untuk memenuhi akses *opportunistic* [10].



Gambar 1. Ilustrasi *Contention Window* [3]

Seperti yang telah disebutkan oleh [5]. Dalam *overlapped contention* sebuah simpul dengan bit rate rendah tetap memiliki kesempatan untuk mengalahkan node lainnya dengan *bitrate* yang tinggi.

2) Opportunistic Scheduling

Opportunistic scheduling menurut [5], ini dengan algoritma yang berguna menjadwalkan sebuah node untuk mentransmisikan pada suatu slot waktu

tertentu sesuai dengan *signal-to-noise-ratio* (SNR) dari channellnya. Untuk menghindari node dengan SNR yang sama dengan slot waktu yang sama maka di berlakukan slot waktu yang berbeda secara acak. Algoritma ini memprioritaskan suatu penjadwalan dari node yang menguntungkan terhadap node terhadap node yang lain, traffic ke node sebelahnya dengan kondisi kanal yang terbaik maka dijadwalkan pertama untuk dikirim [8].

3) Opportunistic Transmission

Transmisi ini berlandaskan pada perkalian atau peningkatan rate pada demonstrasi untuk hasil mengenai kemajuan yang signifikan pada kemampuan jaringan IEEE 802.11, dimana sebuah node secara oportunistik mentransmisikan beberapa frame jika bit rate-nya tinggi, bukan secara tradisional satu frame seperti yang di tuliskan[3]. Pada penjabarannya mengenai transmisi, [5] menjelaskan, dalam algoritma ini mengandalkan adaptasi tingkat seperti RBAR untuk memperkirakan kondisi kanal, dimisalkan bitrate dan kemudian pengirim mengkalkulasikan jumlah dari frame yang terkirim untuk menjaga kestabilan temporal.

4) Opportunistic Routing

Proses ini berfokus pada beberapa alur simultan dalam jaringan mesh nirkabel dengan memanfaatkan keragaman node dengan memilih jalur opportunistic kemudian juga menurut [5] juga memprioritaskan *forwarding* berbasis waktu untuk mendukung pengguna dengan memberikan jalur terbaik, lalu secara adaptif menentukan tingkat pengiriman berdasarkan kondisi kanal saat ini.

a) Contention Window

Merupakan besarnya jangkauan atau range dari nilai *backoff interval* yang akan di panggil oleh terminal. Besarnya *contention window* ini bergantung pada berapa kali proses pengiriman data berlangsung. Dapat di jabarkan apabila proses pertama pengiriman data dilakukan maka nilai $w = CW_{\text{minimum}}$ dan pada proses retransmisi yang mana paket akan meningkat sampai pada nilai maksimum [11]. Seperti yang di tunjukan pada rumus 2 dibawah ini.

$$CW_{\text{max}} = 2m \cdot CW_{\text{min}} \quad [2]$$

b) Distribusi normal berbasis Back-off selection

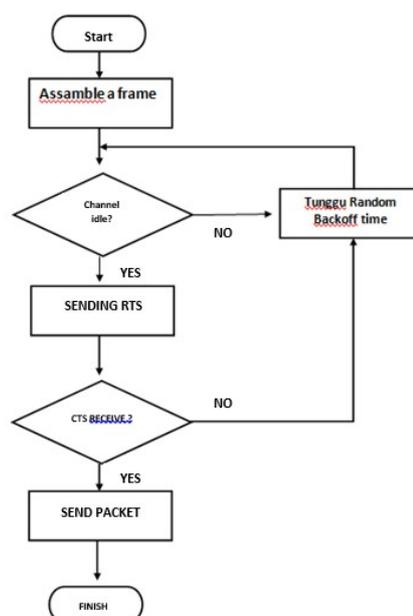
Pada saat terjadi pertumbukan tumpeng tindih, meskipun dengan kondisi kanal yang berbeda untuk mendapatkan *contention*

window yang berbeda, masing-masing node menggunakan proses pengiriman yang seragam untuk mendapatkan nilai *back-off selection* dari *contention window* yang ada [12].

Pada tujuan dari *opportunistic access* tersebut menggunakan pendekatan yang normal, dan juga seperti pengiriman yang seragam, pada saat menapatkan nilai back-off seperti contention windows yang ditentukan seperti pada pertumbukan tumpeng tindih. Sebagai contoh pada distribusi normal yang di sesuaikan dengan nilai dalam contention window dan sesuai dengan selisih nilainya [13]. Node dengan nilai bit rate yang lebih tinggi akan berkemungkinan memenangkan pertumbukan data dibanding node untuk mendapatkan nilai yang lebih kecil dari bit rate yang lebih rendah.

III. METODOLOGI PENELITIAN

CSMA/CA memiliki 2 proses atau 2 variasi pengambilan data, yaitu posisi indoor dan outdoor, proses akan melalui proses pengiriman broadcast terhadap penerima, dan pengirim mengirimkan RTS dan mendapat feedback berupa CTS atau proses dimana DIFS dan CIFS yang mana akan memberikan hasil berupa parameter QoS yaitu *Delay*, *Throughput*, *Jitter* dan *Routing Overhead*. Proses algoritma CSMA/CA akan ditunjukan pada gambar 3 dibawah ini.



Gambar.3 Proses CSMA/CA

Pada penulisan ini,terdapat parameter variasi kondisi yang menjadi fundamental penulisan ini. Dalam parameter kondisi ini yaitu memposisikan simulasi saat dijalankan. Pada parameter ini juga yang menjadi parameter terdiri dari node dengan bitrate yang rendah ataupun tinggi, dalam proses pengiriman data tidak terjadi kemacetan data ataupun tabrakan data.

Tabel 1. Kriteria pengujian dalam ruangan dan luar ruangan

Pada parameter ini terdiri dari skenario pada

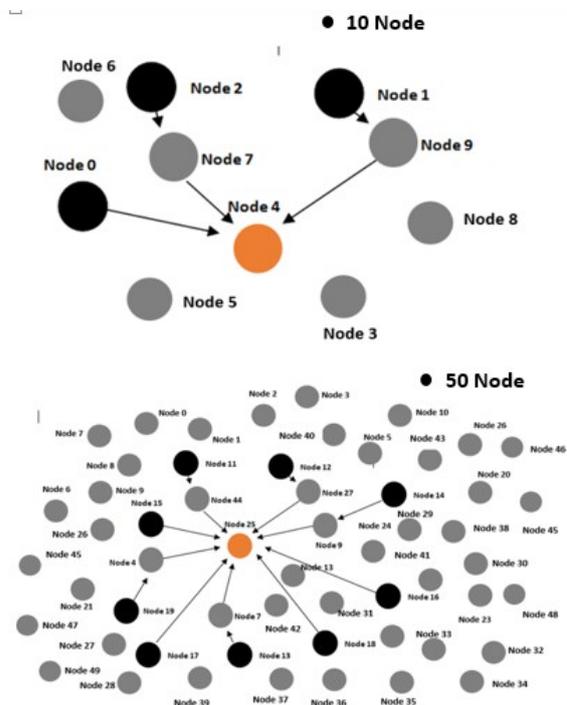
No	Kriteria pengujian	Spesifikasi pengujian
1	Jumlah Node	10,50
2	Kecepatan Maksimal Node (km/jam)	7,2 km/jam dan 28,8 km/jam
3	Jumlah Percobaan	1x percobaan
4	Pergerakan node	Acak
5	Posisi Awal Node	Acak

pengujian yang dilakukan pada simulator. Dibawah ini di merupakan parameter pengujian yang akan dilakukan:

Tabel 2
Parameter skenario

Routing Protocol	AODV
Jumlah Node	10,50 Node
Kecepatan Node	7,2 Km/jam dan 28,8 Km/jam
Ukuran Paket	10 Kbps
Jenis Paket	TCP
Data Rate	5 Mbps
MAC Protokol	IEEE : 802.11
Variasi Pengujian	Indoor dan Outdoor

digunakan 2 variasi node sebanyak 10 dan 50 node. Dari node yang digunakan, hanya menggunakan beberapa node sebagai sampel . Proses akan dijelaskan pada gambar 4 dibawah ini.



Gambar.4 Variasi Node

Proses penelitian menggunakan dua topologi terhadap dua variasi node,yang pertama adalah proses pengambilan data terhadap jumlah 10 node kemudian dengan menggunakan 50 node , data tersebut dilakukan menggunakan topologi *peer-to-peer* atau Adhoc , proses menggunakan 10 node namun di sederhankan menggunakan 3 node sebagai *sample* node pengirim yaitu node 0,1,2 sebagai pengirim, node selain itu sebagai hub yang telah ditentukan dan node 4 sebagai node penerima, sedangkan pada percobaan menggunakan 50 node menggunakan node 11,12,13,14,15,16,17, 18,19 sebagai pengirim kemudian node 25 sebagai penerima dan node selain itu sebagai node yang bertugas sebagai hub penjelasan dapat dilihat pada gambar.4.

IV. HASILL DAN DISKUSI

Berikut ini adalah hasil dari percobaan yang dibuat dengan menggunakan omnet++ yang menggunakan routing AODV yang mana routing ini melakukan pengiriman *broadcast message* pada setiap jaringan atau perangkat yang terhubung. Proses yang dilakukan pada penelitian ini mencari nilai dari parameter *Delay*, *Throughput*, *jitter* dan *Routing Overhead* yang mana hasil dari parameter Qos tersebut akan ditampilkan pada table dibawah ini.

Tabel.1 Pengujian dengan menggunakan 10 node (dalam ruangan)

No	Parameter	Nilai
1	<i>Delay</i>	0,134 ms
2	<i>Throughput</i>	1034 bps
3	<i>Jitter</i>	0,127 ms
4	<i>Routing Overhead</i>	82 byte

Tabel.2 pengujian dengan menggunakan 10 node (Luar ruangan)

No	Parameter	Nilai
1	<i>Delay</i>	0,2627 ms
2	<i>Throughput</i>	740 bps
3	<i>Jitter</i>	0,19 ms
4	<i>Routing Overhead</i>	70,6 byte

Tabel 3 Tabel pengujian dengan menggunakan 50 node (dalam ruangan)

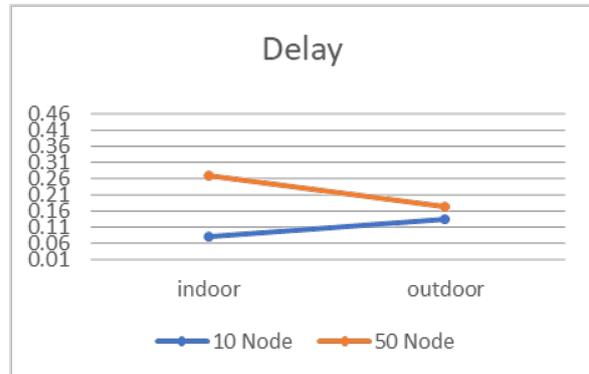
No	Parameter	Nilai
1	<i>Delay</i>	0,172 ms
2	<i>Throughput</i>	2102 bps
3	<i>Jitter</i>	0,199 ms
4	<i>Routing Overhead</i>	94,6Byte

Tabel 4 Tabel pengujian dengan menggunakan 50

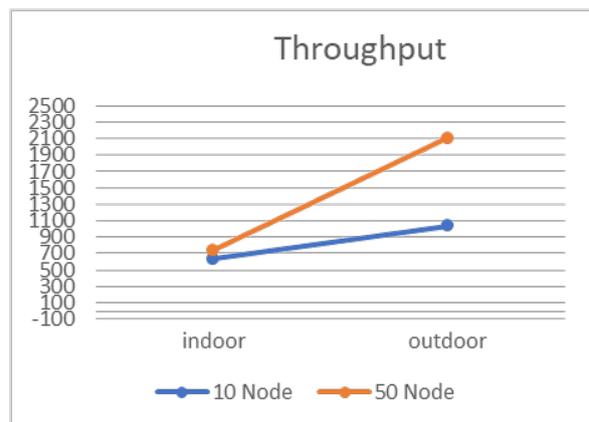
No	Parameter	Nilai
1	<i>Delay</i>	0,0787 ms
2	<i>Throughput</i>	633 bps
3	<i>Jitter</i>	0,045 ms
4	<i>Routing Overhead</i>	60,6 byte

node (luar ruangan)

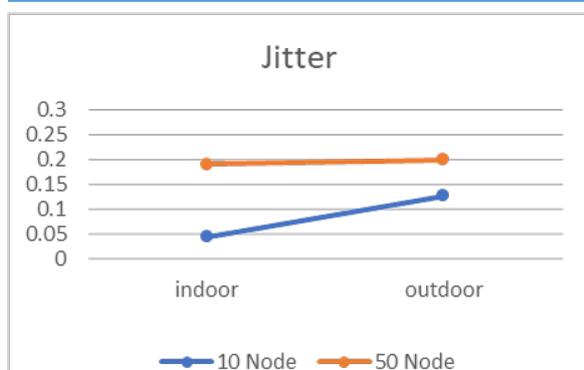
Proses dari hasil yang didapat akan ditampilkan pada grafik dibawah ini.



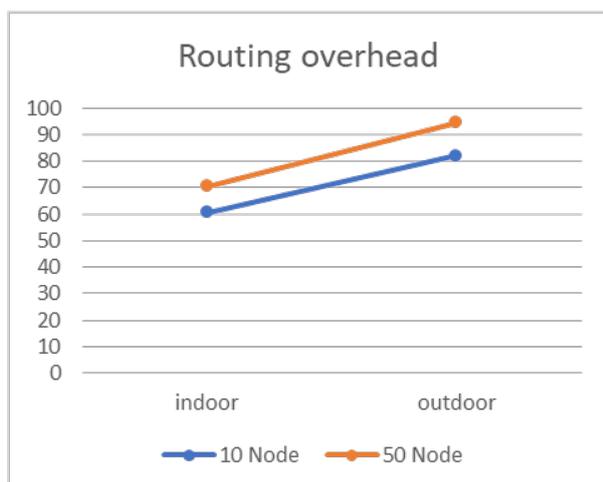
Gambar 5. Grafik parameter *Delay*



Gambar 6. Grafik parameter *Throughput*



Gambar 7. Grafik parameter *Jitter*



Gambar 8. Grafik parameter *Routing Overhead*

Hasil yang di dapat pada percobaan dalam ruangan atau *indoor* ini digunakan kecepatan bervariasi sebagai perbandingan yaitu 7,2 km/jam dan 28,8 km/jam . seperti yang dapat dilihat pada tabel 5 yaitu pada percobaan menggunakan 10 node. Pada kolom *Delay* mengalami peningkatan pada tabel dengan kecepatan 7,2 km/jam dengan nilai 0,078 ms menjadi 0,134 pada kecepatan 28,8 km/jam. Kemudian hasil dari percobaan pada parameter *Throughput* juga mengalami peningkatan nilai dari 633 bps menjadi 1034 bps pada table 6 saat di naikan kecepatan pengiriman menjadi 28,8 km//jam dari kecepatan 7,2 km/jam. Pada hasil dari *Jitter* mengalami penambahan nilai dari 0,045 ms menjadi 0,127 ms , hal ini terjadi karena pada saat melakukan proses pengiriman dengan

menggunakan kecepatan yang berbeda, paket yang terkirim akan semakin memerlukan waktu untuk terkirimnya paket pada penerima. Kemudian pada proses *Routing Overhead* yang dapat dilihat pada tabel 8 Nilai dari *Routing Overhead* mengalami peningkatan pada percobaan 10 node ini yaitu dari 60,6 byte pada kecepatan 7,2 km/jam atau pada kecepatan dalam ruangan menjadi 82 byte.

Kemudian pada percobaan 50 node, hal yang memiliki variasi selain jumlah node yaitu kecepatan paket dikirimkan , sama halnya dengan percobaan menggunakan 10 node diatas, nilai nilai dari parameter kali ini mengalami penurunan dan peningkatan *value* , hal ini terjadi karena pada saat proses pengiriman dalam banyak data dan banyak node serta proses pengiriman yang berulang terus menerus tanpa proses *feedback* , maka proses melakukan broadcast yang menyebabkan paket secara cepat tersebar untuk mengetahui kemana paket akan dikirimkan. Nilai dari *Delay* pada tabel 4 menunjukkan nilai rata rata sebesar 0,2674 ms, kemudian pada *outdoor* mengalami penurunan menjadi 0,1721 ms. Kemudian *Throughput* pada kecepatan dalam ruangan memiliki nilai 740 byte , namun pada *outdoor* menjadi 2102 byte hal ini terjadi karena pada saat kecepatan luar ruangan terdapat *obstacle* yang mempengaruhi pada saat terjadinya proses pengiriman, maka jarak node ke node mengalami sistem acak atau pergerakan yang *random*, hal ini yang menyebabkan *real bandwidth* atau *Throughput* ini menjadi meningkat..

Nilai pada *Jitter* mengalami penurunan *value* dari 0,1900 ms pada percobaan menggunakan kecepatan dalam ruangan menjadi 0,1992 ms atau mengalami peningkatan. *Routing Overhead* menunjukkan penurunan nilai dari 70,6 byte menjadi 24,6 byte.

v. KESIMPULAN

Dari penelitian yang dilakukan , penulis mendapatkan kesimpulan sebagai berikut. Bahwasanya nilai dari parameter QoS dipengaruhi dengan nilai yang lainnya sehingga hasilnya saling berpengaruh. Sebagai contoh *jitter* dipengaruhi oleh nilai *Delay* dikarenakan proses dari *Delay* yang menjadi nilai dari variasi *Delay* pada *jitter*. Kemudian pada hasil yang didapat adalah semakin banyak jumlah node yang digunakan, grafik memperlihatkan sebuah peningkatan pada *Delay* dan *Throughput*, namun terjadi penurunan pada *Routing Overhead* pada percobaan menggunakan 50 Node dikarenakan proses tersebut mengalami proses *routing* yang terjadi pada banyak node, sehingga proses *Routing Overhead* harus menjangkau ke seluruh node untuk mengetahui kemana proses pengiriman akan terjadi. Proses

- pengiriman menggunakan *Opportunistic Random Acces* terjadi pada proses DIFS dan SIFS memberikan efisiensi terhadap waktu dan efisiensi paket.
- [1]. Sidhu, N., & Rani, S. (2013). Improved Optimal Slotted CSMA / CA Protocol. *International Journal of Computer Applications*, 79(October), pages 43–47. <https://doi.org/10.5120/13749-1569>
- [2]. Tang, C., Song, L., Balasubramani, J., Wu, S., Biaz, S., Yang, Q., & Wang, H. (2014). Comparative investigation on CSMA/CA-based opportunistic random access for *Internet of Things*. *IEEE Internet of Things Journal*, vol 1(2), pages 171–179. <https://doi.org/10.1109/JIOT.2014.2302534>
- [3]. Chatzimisios, P., Boucouvalas, A. C., & Vitsas, V. (2014). Optimisation of RTS / CTS handshake in IEEE 802 . 11 Nirkabel LANs for maximum performance, *International Journal of Computer Applications*, vol 1 ,pages 270–275. <https://doi.org/10.1109/GLOCOMW.2004.1417586>
- [4]. Balasubramani, J. (2013). Opportunistic Random Access in CSMA/CA based Nirkabel Networks. *IEEE Internet of Things Journal*, vol 1, pages 12–17. Retrieved from <http://hdl.handle.net/10415/3854>
- [5]. Hwang, C. S., & Cioffi, J. M. (2009). Opportunistic CSMA/CA for achieving multi-user diversity in nirkabel LAN. *IEEE Transactions on Nirkabel Communications*, vol 8(6), pages2972–2982. <https://doi.org/10.1109/TWC.2009.080134>
- [6]. Malik, R. F., Rahman, T. A., Ngah, R., Zaiton, S., & Hashim, M. (2012). The New Multipoint Relays Selection in OLSR using Particle Swarm Optimization, *10(2)*, 343–352.
- [7]. Climent, S., Sanchez, A., Capella, J. V., Meratnia, N., & Serrano, J. J. (2014). Underwater acousticnirkabel sensor networks: Advances and future trends in physical, MAC and routing layers. *Sensors (Switzerland)*, vol 14(1), pages 795–833. <https://doi.org/10.3390/s140100795>
- [8] Kamesh, & Sakthi Priya, N. (2014). Security enhancement of authenticated RFIDgeneration. *International Journal of Applied Engineering Research*, vol 9(22), pages 5968–5974. <https://doi.org/10.1002/sec.946>
- [9] Kumaravel, K., & Marimuthu, A. (2014). An optimal mesh MASH routing topology using mesh in nirkabel sensor networks. *Proceeding of the IEEE International Conference on Green Computing, Communication and Electrical Engineering, ICGCCEE 2014*, 1(2), 10–11. <https://doi.org/10.1109/ICGCCEE.2014.6922412>
- [10] Student, P. G., & Engg, C. (2017). Simulation and performance evaluation of AODV Protocol with QoS using Network Simulator 3 (NS3),vol 3(04), pages 3-5. <https://doi.org/10.3390/info8030088>
- [11] Katkar, P. S. (2016). Comparative Study of Network Simulator : NS2 and NS3. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(3), 608–612. <https://doi.org/10.5120/ijca2016910918>
- [12] Henry, A. J., & Burton, E. M. (2011). Author: Jerome Henry Editor: Marcus Burton, vol 1(November), pages 1–30. <https://doi.org/10.13140/RG.2.1.2546.7607>
- [13] Dipobagio, M. (n.d.) (2014). An Overview on Ad Hoc Networks, *IEEE Communications Magazine* vol 1(2) pages1–9. <https://doi.org/10.1109/MCOM.2014.6710069>