
Evaluasi Kinerja IEEE 802.11e HCCA untuk Dukungan QoS pada WLAN Menggunakan NS-2

Performance Evaluation of IEEE 802.11e HCCA for WLAN QoS Support in NS-2

Amry Daulat Gultom

*Puslitbang Sumber Daya dan Perangkat Pos dan Informatika
Jl. Medan Merdeka Barat No.9 Jakarta 10110*

amry002@kominfo.go.id

Naskah diterima: 16 Mei 2014; Direvisi: 12 Juni 2014; Disetujui: 20 Juni 2014

Abstract—Wireless Local Area Network (WLAN) is used by the multimedia application traffic that should have better network requirements for delay, jitter and packet losses. IEEE 802.11 Task Group E introduced the 802.11 MAC protocol improvements, Hybrid Coordination Function (HCF), which there are two access mechanisms: Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA), which supports Quality of Service (QoS) for multimedia traffic. The purposes of this study are to understand the algorithm IEEE 802.11e HCCA in WLAN networks, and analyze performances of 802.11e HCCA in multimedia applications by simulation in NS-2. The performance metrics are jitter and throughput. The results showed that HCCA can provide QoS guarantees more than DCF, due HCCA jitter more stable than DCF jitter. Neither throughput of HCCA is stable, unlike DCF that has large fluctuations.

Keywords— WLAN, QoS, HCCA, DCF

Abstrak—Wireless Local Area Network (WLAN) digunakan oleh trafik multimedia yang seharusnya memerlukan persyaratan jaringan yang lebih baik terhadap *delay*, *jitter* dan *packet losses*. IEEE 802.11 Task Group E memperkenalkan perbaikan protokol MAC 802.11, yaitu *Hybrid Coordination Function* (HCF), yang terdiri dari dua mekanisme akses: *Enhanced Distributed Channel Access* (EDCA) dan *HCF Controlled Channel Access* (HCCA) yang memberikan dukungan Kualitas Layanan/*Quality of Service* (QoS) bagi trafik multimedia. Tujuan dari penelitian ini adalah untuk memahami algoritma protokol MAC IEEE 802.11e HCCA pada jaringan WLAN, dan menganalisis kinerja protokol MAC 802.11e HCCA pada aplikasi multimedia dengan menggunakan metode simulasi pada NS-2. Metrik kinerja yang diukur adalah *jitter* dan *throughput*-nya. Hasil penelitian ini menunjukkan bahwa protokol MAC HCCA dapat memberikan jaminan QoS dibanding protokol MAC DCF, dimana *jitter* HCCA lebih stabil dari pada *jitter* DCF. Begitupula dengan *throughput* HCCA yang tidak berubah selama trafik berlangsung, tidak seperti pada DCF yang masih mengalami fluktuasi yang besar.

Kata kunci— WLAN, QoS, HCCA, DCF

I. PENDAHULUAN

Dalam dekade terakhir ini, teknologi jaringan komputer berkembang begitu pesat. Apalagi dengan adanya jaringan komputer tanpa kabel atau yang lebih dikenal sebagai *Wireless Local Area Network* (WLAN) yang memungkinkan pengguna untuk dapat berkomunikasi dan mengakses informasi tanpa mempermasalahkan posisi geografis, membuat teknologi ini semakin diminati.

WLAN merupakan jaringan komputer yang menggunakan medium *wireless* (tanpa kabel) sebagai pengganti kabel untuk menghubungkan komputer-komputer. Karena kemudahan dalam penerapannya, WLAN diharapkan dapat digunakan pada berbagai aplikasi termasuk aplikasi multimedia seperti *audio* dan *video*. Aplikasi multimedia ini memerlukan persyaratan jaringan yang lebih baik terhadap *delay*, *jitter* dan *packet losses* sehingga pengiriman data pada aplikasi ini dapat berjalan dengan baik. Untuk mendapatkan dukungan Kualitas Layanan/*Quality of Service* (QoS) ini, protokol *Media Access Control* (MAC) harus diperbaiki. Hal inilah yang menjadi tantangan untuk mendesain dan membangun serta mengevaluasi protokol MAC yang dapat memberikan dukungan QoS pada aplikasi multimedia.

Untuk mendukung penggunaan aplikasi *audio* dan *video* pada jaringan WLAN yang membutuhkan dukungan QoS, *Institute of Electrical and Electronics Engineers* (IEEE) 802.11 Task Group E telah memperkenalkan perbaikan protokol MAC 802.11, yaitu *Hybrid Coordination Function* (HCF). HCF terdiri dari dua mekanisme akses: *Enhanced Distributed Channel Access* (EDCA) dan *HCF Controlled Channel Access* (HCCA).

Tujuan dari penelitian ini adalah untuk memahami algoritma protokol MAC IEEE 802.11e HCCA pada WLAN, dan mengevaluasi kinerja protokol MAC 802.11e HCCA pada aplikasi *audio* dan *video* dengan menggunakan *Network Simulator* (NS). Kinerja ini dibandingkan dengan IEEE 802.11 *Distributed Coordination Function*.

II. KAJIAN LITERATUR DAN GAMBARAN UMUM

A. Standar IEEE 802.11

Spesifikasi WLAN didasari pada standar IEEE 802.11 yang dikembangkan pada tahun 1997 (Stallings, 2004). Awalnya standar ini menggunakan transmisi data dengan kecepatan hingga 2 Mbps. Dengan berkembangnya waktu, implementasi dari standar ini semakin populer dan meluas. Penambahan ekstensi di belakang 802.11 dipergunakan untuk mengenal beberapa perbaikan dan tambahan fitur dari standar yang telah ditentukan 802.11.

Pada tahun 1999, IEEE telah meratifikasi dua standar, yaitu standar 802.11a dan 802.11b. Standar 802.11a didasarkan pada skema modulasi *Orthogonal Frequency Division Multiplexing* (OFDM) yang bekerja pada frekuensi 5 GHz dengan kecepatan transmisi data mencapai 54 Mbps. Sedangkan standar 802.11b menggunakan skema modulasi *Direct Sequence Spread Spectrum* (DSSS) dengan kecepatan transmisi data mencapai 11 Mbps pada frekuensi 2,4 GHz.

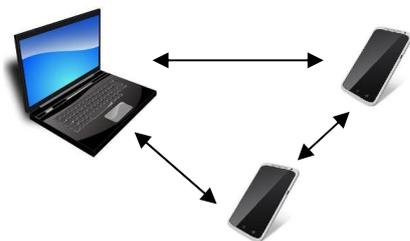
Spesifikasi 802.11g diratifikasi pada bulan Juni 2003, dimana spesifikasi ini beroperasi pada frekuensi 2,4 GHz dengan teknik modulasi yang sama dengan standar 802.11a yaitu OFDM. Standar ini memberikan kecepatan transmisi data hingga 54 Mbps.

Secara paralel, beberapa standar 802.11 yang lain juga dikembangkan. Standar 802.11h ditujukan untuk memperbaiki 802.11a dengan penambahan regulasi lisensi *outdoor* dan *indoor* pada frekuensi 5 GHz di Eropa. Standar 802.11n menggunakan antenna *Multiple-Input Multiple-Output* (MIMO) dan skema modulasi adaptif OFDM. Standar ini dapat mendukung kecepatan transmisi data hingga 100 Mbps.

B. Arsitektur IEEE 802.11 WLAN

Arsitektur jaringan WLAN mengacu pada standar 802.11, yang mendukung tiga topologi dasar untuk WLAN. Ketiga topologi tersebut adalah *Independent Basic Service Set* (IBSS), *Basic Service Set* (BSS), dan *Extended Service Set* (ESS) (Mittal & Anand, 2014).

IBSS merupakan topologi yang paling sederhana, dimana *node-node* yang independen akan saling berkomunikasi secara *peer to peer* atau *point to point*. Topologi ini juga dikenal sebagai *ad-hoc*.

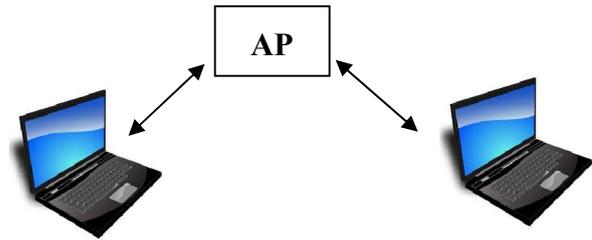


Gambar 1. Topologi Jaringan IBSS

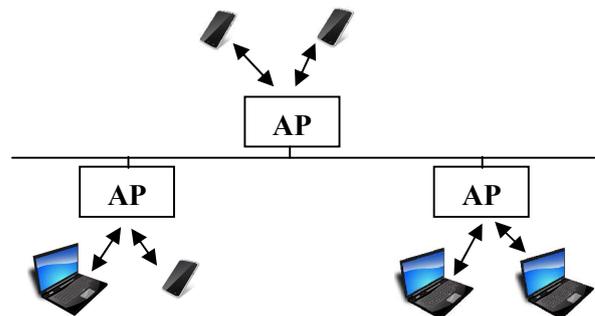
Topologi yang lebih kompleks adalah topologi infrastruktur, atau disebut juga sebagai *Basic Service Set* (BSS), terdiri dari paling sedikit satu *Access Point* (AP) yang bertindak sebagai *base station* dan sekumpulan *node-node wireless*. Komunikasi antara dua *node* melewati AP.

ESS merupakan topologi dimana beberapa AP dapat digunakan untuk membentuk daerah cakupan yang lebih luas. Topologi ini terdiri dari dua atau lebih BSS yang terkoneksi pada satu jaringan kabel. Setiap AP diatur dalam *channel* yang

berlainan untuk menghindari terjadinya interferensi. Metode ini akan membentuk sel-sel seperti pada jaringan selular.



Gambar 2. Topologi Jaringan BSS



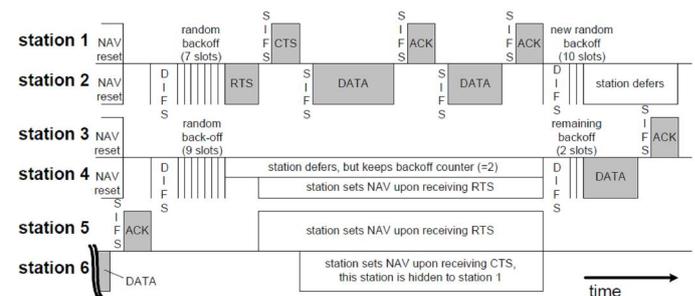
Gambar 3. Topologi Jaringan ESS

C. Protokol MAC pada WLAN

Pada awalnya, standar IEEE 802.11 memiliki dua protokol MAC, yaitu: *Distributed Coordination Function* (DCF) dan *Point Coordination Function* (PCF).

1) *Distributed Coordination Function*

DCF didasarkan pada *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) dengan skema *Request To Send* (RTS) dan *Clear To Send* (CTS).



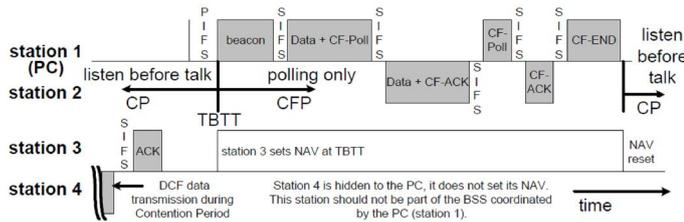
Gambar 4. Mekanisme Akses DCF (Didi, Labiod, Pujolle, & Feham, 2009)

Bila sebuah paket diterima pada antrian yang kosong dan bila medium *idle* untuk interval waktu yang lebih panjang dari *Distributed Interframe Space* (DIFS), stasiun sumber dapat mengirim paket segera. Sementara itu, stasiun yang lain menunda pengirimannya dengan menggeser NAV-nya, dan memulai proses *backoff* (Qiang Ni & Turletti, 2005).

2) *Point Coordination Function*

PCF merupakan mekanisme akses yang menerapkan skema akses *polling-based contention-free*, yang menggunakan AP sebagai *Point Coordination* (PC). Waktu akses kanal dibagi ke dalam interval periodik yang disebut

beacon interval, yang terdiri dari *Contention-Free Period* (CFP) dan *Contention Period* (CP). Selama CFP, PC menjaga daftar stasiun yang teregister dan mem-poll mereka sesuai dengan daftar tersebut (Qiang Ni & Turletti, 2005).



Gambar 5. Contoh skema PCF (Didi et al., 2009)

Standar 802.11 awalnya dikembangkan untuk mendukung aplikasi *best effort*. Pada saat ini, aplikasi *real time* seperti *audio* dan *video*, yang memiliki batasan kinerja yang lebih baik, semakin banyak digunakan. Oleh karena itu, untuk mendukung aplikasi tersebut agar dapat digunakan pada WLAN dengan tingkat QoS yang dapat diterima, maka dukungan pembedaan dan prioritas trafik diperlukan. Hal inilah yang merupakan faktor pemicu munculnya standar 802.11e yang diperlukan untuk memperbaiki kinerja jaringan WLAN.

Standar 802.11e memberikan *service differentiation* yang diperlukan dengan mengumpulkan tingkatan prioritas dari masing-masing paket. Paket-paket dengan prioritas yang lebih tinggi mendapatkan hak akses yang lebih tinggi ke medium *wireless*. Jadi skema ini memberikan *resource* ke paket berdasarkan tingkat kinerja yang diperlukannya. EDCA merupakan perbaikan dari DCF dengan mengelompokkan trafik ke dalam kumpulan prioritas tertentu yang masing-masing memiliki parameter-parameter akses medium yang berbeda-beda, sedangkan HCCA merupakan perbaikan dari PCF. Pada standar IEEE 802.11e ini, terjadi perubahan penamaan pada stasiun atau *node* yang menjadi cakupan jaringan serta AP yang menjadi koordinatonya. Setiap stasiun atau node pada jaringan IEEE 802.11e disebut sebagai *QoS-enabled Station* (QSTA) sedangkan AP-nya disebut *QoS-enabled Access Point* (QAP).

3) *Enhanced Distributed Coordination Access*

EDCA dirancang untuk memberikan prioritas QoS dengan memperbaiki DCF. Pada MAC layer, EDCA memperkenalkan empat antrian *First In First Out* (FIFO) yang berbeda, yang disebut *Access Categories* (AC). Nilai prioritas *user* ditentukan dalam spesifikasi *bridge 802.11d*.

TABEL 1. PEMETAAN ANTARA PRIORITAS USER DAN AC

Prioritas	Prioritas user	AC	Penandaan
Terendah	1	AC_BK	Background
	2	AC_BK	Background
0	3	AC_BE	Best Effort
	4	AC_BE	Best Effort
3	5	AC_VI	Video
	6	AC_VI	Video
Tertinggi	7	AC_VO	Voice

Sumber : (RUSCELLI, CECHETTI, ALIFANO, & LIPARI, 2012)

Tabel 1 memetakan jenis aplikasi yang berbeda seperti trafik background, trafik *best-effort*, trafik *video* dan *audio* ke antrian AC yang berbeda (contoh: AC_BK, AC_BE, AC_VI, AC_VO). Setiap AC memiliki entitas DCF sendiri dan tiap-tiap entitas memiliki parameter *Contention Window* (CW)-nya ($CW_{max}[AC]$, $CW_{min}[AC]$, $AIFS[AC]$, dan $TXOP_{Limit}[AC]$), masing-masing yang diberitahukan oleh QAP secara periodik melalui *frame beacon*. Pada dasarnya, nilai yang paling kecil dari $CW_{max}[AC]$, $CW_{min}[AC]$, $AIFS[AC]$, dan $TXOP_{Limit}[AC]$, *delay* akses kanalnya lebih tinggi untuk mengakses medium. EDCA memperkenalkan jenis baru *Interframe Space* (IFS), yang disebut *Arbitrary IFS* (AIFS). Masing-masing AIFS adalah interval IFS dengan panjang yang berubah-ubah dan ditentukan dengan persamaan (Poonguzhali, 2014):

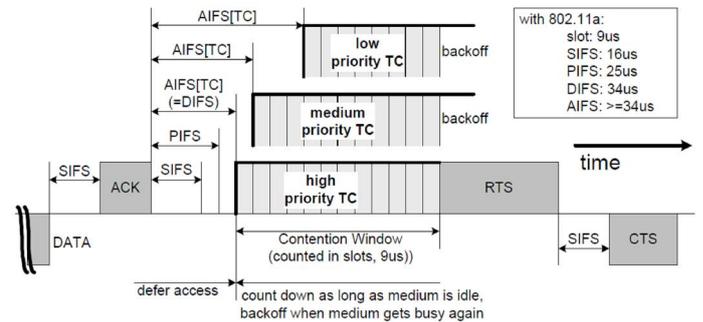
$$AIFS = AIFSN \times aSlotTime + aSIFSTime$$

Dimana $AIFSN[AC]$, disebut *Arbitrary Inter Frame Space Number*, yang menunjukkan jumlah slot time untuk sebuah AC.

TABEL 2. PARAMETER STANDAR EDCA

AC	CW_{min}	CW_{max}	AIFSN	$TXOP_{Limit}$ (802.11b)	$TXOP_{Limit}$ (802.11a/g)
BK	CW_{min}	CW_{max}	7	0	0
BE	CW_{min}	CW_{max}	3	0	0
VI	$(CW_{min}+1)/2-1$	CW_{min}	2	6.016ms	3.008ms
VO	$(CW_{min}+1)/4-1$	$(CW_{min}+1)/2-1$	2	3.008ms	1.504ms

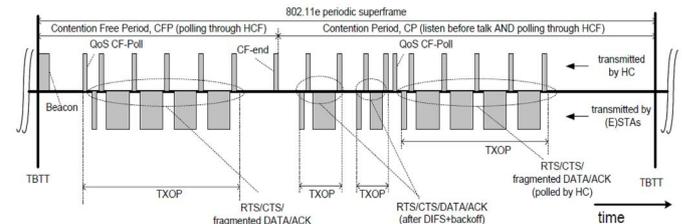
Sumber : (POONGUZHALLI, 2014)



Gambar 6. Mekanisme IEEE 802.11e EDCA (Xiao, Li, & Choi, 2004)

4) *HCF Controlled Channel Access*

HCCA menggunakan mekanisme *poll-and-response* yang sama dengan PCF, tetapi ada banyak perbedaan antara kedua mekanisme tersebut. Selama CP, QAP diijinkan untuk memulai beberapa *burst contention-free*, yang disebut *Controlled Access Period* (CAP), pada sembarang waktu setelah mendeteksi kanal *idle* selama interval waktu PIFS.



Gambar 7. Contoh *Interval Beacon* yang digunakan pada Algoritma Penjadwalan HCF (Didi et al., 2009)

III. METODOLOGI PENELITIAN

Metode yang digunakan pada penelitian ini adalah simulasi menggunakan *simulator*. Ada banyak *simulator* yang dapat dipertimbangkan untuk mengevaluasi kinerja protokol MAC pada WLAN. Namun pada penelitian ini, penulis menggunakan *simulator* NS-2. Beberapa pertimbangannya adalah NS-2 telah digunakan secara luas dan para peneliti di dunia telah memberikan kontribusi terhadap pengembangannya. Ini berarti bahwa *simulator* ini cukup lengkap dan memiliki banyak ekstensi untuk aplikasi-aplikasi, protokol-protokol dan model trafik yang berbeda-beda (NS-2, 2011).

A. Network Simulator

NS-2 merupakan *open source simulator*, yang dirancang untuk mensimulasikan *Local Area Network* (LAN) maupun *Wide Area Network* (WAN). Versi pertama dari NS dirilis pada tahun 1995 dan merupakan varian dari *Realistic and Large (REAL) Simulator*, yang ditulis oleh Keshev pada tahun 1988. REAL itu sendiri dikembangkan dari *Network Testbed (NEST)*. NS pertama-tama dikembangkan sebagai bagian proyek *Virtual Inter-Network Test-bed (VINT)* dan versi keduanya, NS-2 dirilis pada tahun 1996. Namun saat ini, pengembangan NS-2 didukung melalui *Defense Advanced Research Projects Agency (DARPA)* oleh proyek *Simulation Augmented by Measurement and Analysis for Network (SAMAN)* dan melalui *National Science Foundation (NSF)* oleh proyek *Collaborative Simulation for Education and Research (CONSER)*, yang kedua-duanya berkolaborasi dengan peneliti yang lain termasuk *ICSI Center for Internet Research (ICIR)* dan *Carnegie Mellon University (CMU)*. NS dibangun dengan menggunakan dua bahasa pemrograman: *simulator* yang berorientasi objek, yang ditulis dalam bahasa pemrograman C++ dan OTcl (ekstensi yang berorientasi objek dari Tcl) *interpreter*, yang digunakan untuk menjalankan *script* perintah-perintah *user*.

NS memiliki *library* jaringan dan objek-objek protokol yang sangat banyak. Ada dua kelas hirarki: hirarki *compiler* C++ dan hirarki OTcl *interpreter*, yang keduanya saling berhubungan. Hirarki *compiler* C++ memungkinkan kita untuk mendapatkan efisiensi di dalam simulasi dan waktu eksekusi yang lebih cepat, meskipun simulasi melibatkan jumlah paket dan sumber data dalam jumlah yang besar. Sedangkan dengan *script* OTcl yang diberikan oleh *user*, kita dapat mendefinisikan topologi jaringan tertentu, protokol-protokol dan aplikasi-aplikasi tertentu yang ingin kita simulasikan (yang karakteristiknya sudah didefinisikan dalam hirarki *compiler*) dan bentuk keluaran yang ingin kita dapatkan dari *simulator*. OTcl dapat membuat penggunaan objek-objek yang di-*compile* di dalam C++ melalui OTcl *linkage* yang membuat kesesuaian objek OTcl untuk tiap C++ (The VINT Project, 2011).

NS merupakan *discrete event simulator*, dimana kenaikan waktu tergantung pewaktuan *event* yang datur oleh *scheduler*. *Event* adalah suatu objek di dalam hirarki C++ dengan ID yang unik, waktu penjadwalan dan *pointer* ke objek yang menangani *event*. *Scheduler* menjaga struktur data yang berurutan dengan *event-event* yang dijalankan dan menjalakkannya satu per satu.

B. Implementasi IEEE 802.11e HCCA pada NS-2

Distribusi standar dari NS-2 tidak mendukung protokol MAC IEEE 802.11e HCCA. Tambahan dan perbaikan untuk mendukung protokol tersebut diberikan oleh *Computer Networking Group* dari *University of Pisa*. Tambahan dan perbaikan tersebut terdiri dari tiga modul (Cicconetti, Lenzi, Mingozi, & Stea, 2005), yaitu:

1. *Classifier*, digunakan untuk menandakan paket-paket yang datang dari *transport agent* dengan *Traffic Stream Identifier (TSID)* yang berhubungan. Fungsi dari modul *classifier* adalah untuk menandakan paket-paket secara benar yang dimiliki oleh deretan trafik yang dibentuk dengan *Traffic Identifier (TID)*. Hanya paket-paket dari *link layer* ke MAC yang ditandakan, karena paket-paket *uplink* hanya dilewatkan ke *layer* yang lebih tinggi tanpa mekanisme penjadwalan. Masing-masing stasiun menjalankan *classifier* yang terpisah, yang dapat dikategorikan ke dalam dua jenis:
 - a. *Classifier* untuk QSTA: aturan penandaan didasarkan pada kumpulan aturan tertentu dari terminal.
 - b. *Classifier* untuk QAP: aturan penandaan didasarkan pada aturan di atas dan pengidentifikasian dari QSTA tujuan.
 - c. Modul *scheduler* MAC dan HCCA dapat mencari TID dari sembarang paket.
2. HCCA *scheduler*, yang digunakan pada QAP dan QSTA untuk mengatur paket-paket HCCA *downstream* sesuai parameter-parameter QoS dari deretan trafiknya. Tidak seperti operasi pada *network layer*, *scheduler* beroperasi pada *MAC layer* sangat tergantung pada *layer* di bawahnya. Jadi, kita harus mendefinisikan *interface* umum yang cukup untuk mengadopsi urutan algoritma *scheduling*. Bila dalam keadaan aktif, baik penjadwalan QSTA dan QAP berpindah-pindah antara dua *state* utama:
 - a. *IDLE*. Merupakan *passive state*: paket-paket diantri dari *downlink Traffic Stream (TS)* tidak dikirimkan menggunakan HCCA dan bukan *poll* yang dibangkitkan (hanya QAP). Hanya dua aksi yang diikuti oleh keadaan ini: (i) mengantri paket *downlink* baru dari TS yang dibentuk, (ii) menambahkan TS baru ke kumpulan TS yang dibentuk.
 - b. *BUSY*. Merupakan *active state*. Semua tindakan yang diijinkan pada *state* sebelumnya dapat berlangsung pada *state* ini. Selain itu, paket-paket *downlink* dari TS yang dibentuk dapat dikirimkan atau QAP dapat mem-*poll* QSTA.

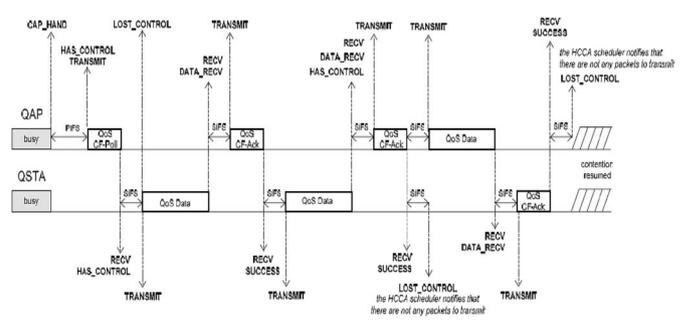
Sedangkan fungsi-fungsi berikut ditentukan untuk algoritma penjadwalan yang digunakan pada QAP dan QSTA:

- a. *Enque()*. Fungsi ini dipanggil oleh *link layer* dimana paket *downlink* baru telah diterima dari *layer* di atasnya. Paket yang datang harus diantri pada struktur data *scheduler* tertentu dari modul HCCA *scheduler*. Ada kasus dimana *scheduler* dapat membuang paket, sebagai contoh bila *buffer* dialokasikan ke paket *downlink* terbatas.
- b. *Deque()*. Fungsi ini dipanggil oleh MAC bila stasiun memiliki kontrol terhadap medium. Pengaruhnya tergantung kepada algoritma *scheduling* dan status saat ini dari *scheduler*. Tindakan yang diijinkan termasuk: (i) *scheduler* melewatkan ke MAC paket *downlink* yang dikirimkan, (ii) *scheduler*

berkomunikasi dengan MAC bahwa QSTA yang diberikan harus di-poll dengan durasi TXOP tertentu, (iii) sebuah poll terhadap dirinya sendiri harus dikirim, (iv) tidak ada tindakan yang dilakukan. Sudah tentu, tindakan (ii) dan (iii) hanya dilakukan oleh scheduler QAP.

- c. *Get_next_cap()*. [hanya QAP] Fungsi ini dipanggil oleh MAC bila scheduler QAP kehilangan kontrol terhadap medium. Digunakan oleh scheduler untuk memulai awal MAC dan CAP berikutnya. Sampai waktu itu, fungsi HCCA dari QAP adalah idle.
 - d. *addTSPEC()*. [hanya QAP] Fungsi ini dipanggil oleh MAC bila pemberian ijin deretan trafik baru diminta.
 - e. *Get_queue_size()*. [hanya QSTA] Fungsi ini dipanggil oleh MAC sebelum mengirimkan frame data. Ia mengembalikan ukuran antrian dari TS tertentu. Informasi ini di-piggybacked ke semua frame data yang dibangkitkan oleh QSTA yang memiliki TS tertentu.
3. Modul MAC yang dimodifikasi, yang memperbaiki MAC class standar IEEE 802.11. Modul ini menggambarkan struktur data utama, fungsi-fungsi dan event-event pada modul MAC. Sedangkan beberapa event yang diturunkan yang menggambarkan kondisi state MAC:
- a. *HCCA_HAS_CONTROL*. Event ini menyatakan bahwa stasiun memiliki kontrol terhadap medium. QSTA membangkitkan event ini bila ia di-poll oleh QAP. QAP membangkitkan event ini bila ia men-sense medium idle untuk periode waktu yang lebih besar dari PIFS atau bila medium menerima frame terakhir dari QSTA selama burts TXOP.
 - b. *HCCA_LOST_CONTROL*. Event ini menyatakan stasiun bahwa ia tidak memiliki kontrol terhadap medium lagi. Ia dibangkitkan bila HCCA scheduler memiliki paket yang akan dikirimkan. Juga, QAP membangkitkan event ini bila QSTA merespon frame polling dan QSTA membangkitkan event ini bila QAP secara benar menjawab frame terakhir dari TXOP burst.
 - c. *HCCA_DATA_RECV*. Event ini menyatakan stasiun bahwa sebuah frame yang membawa data yang dialamatkan ke stasiun ini telah diterima secara benar.
 - d. *HCCA_RECV*. Event ini menyatakan bahwa sembarang jenis frame yang dialamatkan ke stasiun ini telah diterima secara benar.
 - e. *HCCA_SUCCESS*. Event ini menyatakan stasiun bahwa frame data downlink telah dijawab oleh stasiun penerima.
 - f. *HCCA_TRANSMIT*. Event ini menyatakan stasiun bahwa frame downlink telah dikirim.
 - g. *HCCA_TX_END*. Event ini dibangkitkan oleh QAP bila TXOP yang dijamin ke QSTA telah berakhir.
 - h. *HCCA_CAP_HAND*. Event ini dibangkitkan oleh QAP bila inilah waktunya untuk memulai CAP yang baru, sesuai permintaan HCCA scheduler.

Gambar 8 menunjukkan event di atas dalam kasus pertukaran frame: pengiriman dua frame data uplink, diikuti dengan pengiriman frame data downlink, diasumsikan bahwa tidak ada tabrakan atau frame yang corrupt karena kanal yang jelek terjadi.



Gambar 8. Event-event Selama Deretan Pertukaran Frame (Ciconetti et al., 2005)

Pada simulasi MAC protokol IEEE 802.11e pada WLAN ini menggunakan physical layer IEEE 802.11b yang memakai modulasi Direct Sequence Spread Spectrum (DSSS) dengan kecepatan transmisi datanya mencapai 11 Mbps. Tabel 3 menunjukkan beberapa parameter MAC untuk physical layer IEEE 802.11b. Simulasi akan dijalankan menggunakan 3 (tiga) skenario, dan semua simulasi dijalankan selama 100 detik namun tidak ada data yang dikirimkan untuk 20 detik pertama.

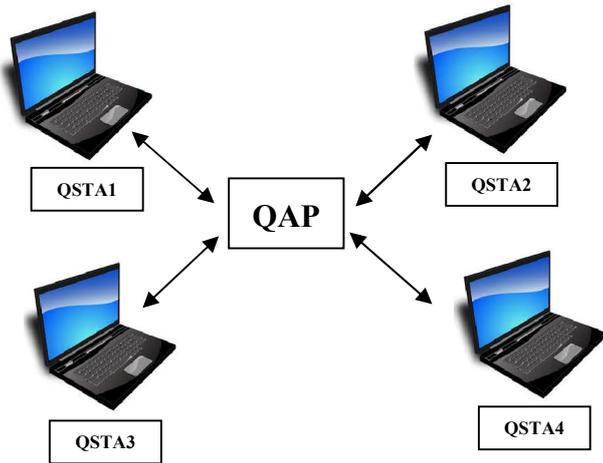
TABEL 3. PARAMETER MAC UNTUK PHYSICAL LAYER IEEE 802.11B

Karakteristik	Nilai
Slot Time	20 μs
SIFS Time	10 μs
Data Rate	11 Mbps
Basic Rate	1 Mbps
Preamble Length	144 bit
PLCP Header Length	48 bit
PLCP Data Rate	1 Mbps
CWmin	31
CWmax	1023

Ada tiga jenis trafik yang diberikan pada masing-masing skenario yaitu trafik audio, video dan data. Trafik audio menggunakan generator trafik Constant Bit Rate (CBR) yang membangkitkan data secara kontinu dengan bit rate konstan. Trafik audio CBR ini dipasang pada agent transport Universal Datagram Protocol (UDP), dibangkitkan dengan laju bit yang konstan 64 kbps, ukuran paket 160 bytes dan waktu antar kedatangan paket 20 ms. Trafik video juga menggunakan generator trafik CBR dengan agent transport yang sama dengan trafik audio yaitu UDP. Model trafik video ini dibangkitkan dengan laju bit yang konstan 80 kbps, dengan ukuran paket sebesar 1280 bytes dan dengan waktu interval paket 16 ms. Dan juga digunakan trafik best effort dengan ukuran paket 210 bytes dengan laju pengiriman 64 kbps. Sedangkan waktu kedatangannya adalah 25 ms, diberikan pada agent TCP.

1. Skenario 1

Skenario pertama ini digunakan untuk melihat dukungan protokol MAC HCCA pada aplikasi trafik audio dan video.



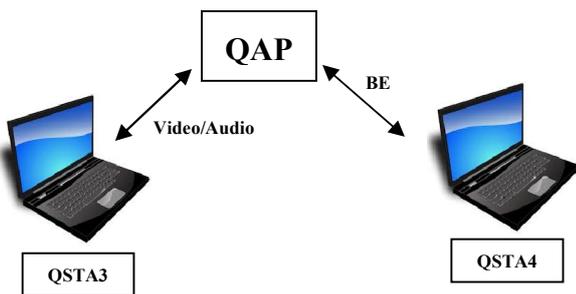
Gambar 9. Topologi Skenario 1

Dari topologi tersebut dapat dilihat bahwa ada empat stasiun yang masing-masing mengirimkan trafik *audio* ke AP. Trafik yang dikirimkan bisa secara *uplink* maupun *downlink* untuk memperlihatkan mekanisme seperti kondisi nyatanya. Selanjutnya, dengan topologi dan jumlah stasiun yang sama, trafik yang dikirimkan diganti dengan trafik *video*. Seperti trafik *audio*, trafik *video* ini juga dikirimkan secara *bidirectional*.

Untuk skenario yang sama juga akan dilakukan simulasi pada protokol MAC 802.11 mode DCF sebagai perbandingan terhadap kinerja protokol 802.11e HCCA.

2. Skenario 2

Berbeda dengan skenario 1, skenario 2 menghadirkan trafik *Best Effort*. Hal ini dilakukan untuk memperlihatkan pengaruh yang terjadi apabila pada suatu jaringan WLAN selain menggunakan trafik *audio* atau *video*, juga menggunakan trafik BE ini. Jumlah stasiun yang mengirimkan trafik BE divariasikan untuk memperlihatkan pengaruh yang terjadi. Untuk skenario ke-2 ini juga akan dilakukan simulasi menggunakan protokol MAC 802.11 mode DCF.

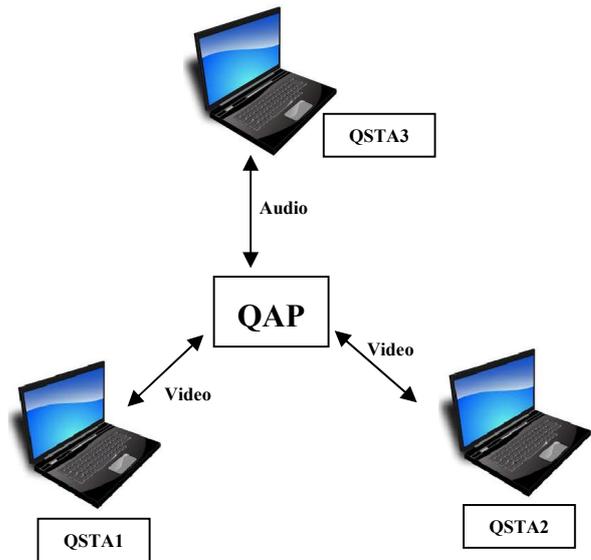


Gambar 10. Topologi Skenario 2

3. Skenario 3

Skenario 3 menggabungkan trafik *audio* dan *video* pada suatu jaringan WLAN. Pada skenario ini, ada dua stasiun mengirimkan trafik *video* pada arah *uplink* dan *downlink*. Sedangkan trafik *audio* dibangkitkan oleh satu stasiun yang *bidirectional*. Skenario 3 ini dilakukan untuk memperlihatkan mekanisme protokol MAC HCCA dalam menghadapi dua sumber trafik multimedia. Sebagai pembanding, juga akan

dilakukan simulasi menggunakan protokol MAC 802.11 mode DCF.



Gambar 11. Topologi Skenario 3

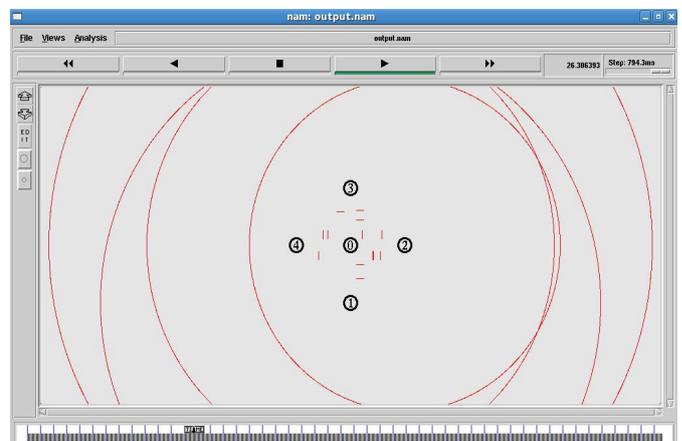
C. Metrik Kinerja

Di dalam simulasi ini, ada beberapa ukuran kinerja yang diambil sebagai acuan dalam membandingkan protokol MAC HCCA dengan protokol MAC standar DCF. Ukuran atau metrik kinerja yang pertama adalah *jitter*. *Jitter* merupakan nilai waktu dari satu paket ke paket berikutnya dalam sebuah trafik. Ukuran kinerja lainnya adalah *throughput*. *Throughput* merupakan jumlah paket yang sukses diterima dalam satu satuan waktu. Satuan yang dipakai di sini adalah bps (*bit per second*).

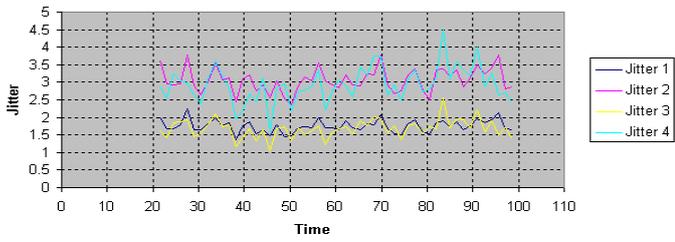
IV. HASIL PENELITIAN DAN PEMBAHASAN

A. Skenario 1

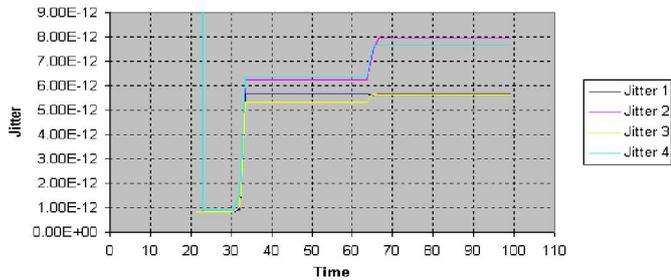
Gambar 12 menunjukkan tampilan topologi simulasi skenario 1 pada *Network Animator*. Hasil dari simulasi NS-2 pada skenario 1 ini terbagi menjadi 2 bagian, yaitu grafik *jitter* dan *throughput* bila pada jaringan WLAN tersebut diberikan trafik *audio bidirectional* dengan jumlah stasiun empat buah.



Gambar 12. Skenario 1 Pada NAM



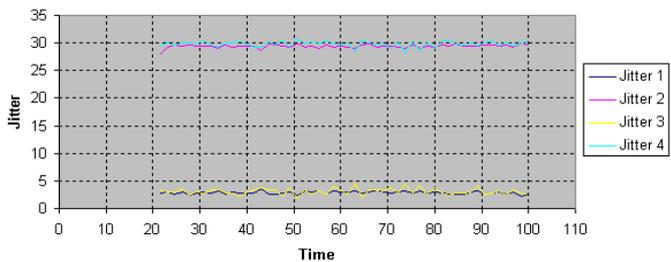
Gambar 13. Jitter Trafik Audio DCF



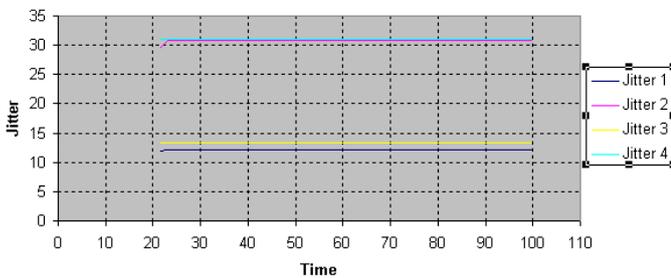
Gambar 14. Jitter Trafik Audio HCCA

Gambar 13 memperlihatkan *jitter* pada saat menggunakan protokol MAC DCF, sedangkan gambar 14 merupakan hasil pada saat menggunakan protokol MAC HCCA.

Dari gambar 13 dan 14 dapat kita lihat bahwa *jitter* trafik *audio* dengan menggunakan protokol MAC HCCA cenderung konstan. Hasil ini berbeda bila kita menggunakan protokol MAC DCF. Rata-rata *jitter* 1 dengan menggunakan protokol MAC HCCA adalah sebesar $6,16307 \times 10^{-12}$, *jitter* 2 sebesar $2,5 \times 10^{-6}$, *jitter* 3 sebesar $7,84345 \times 10^{-12}$, sedangkan dengan DCF untuk *jitter* 1-4 masih fluktuatif. Di sini dapat dilihat bahwa *jitter* HCCA lebih konstan dibanding *jitter* DCF.



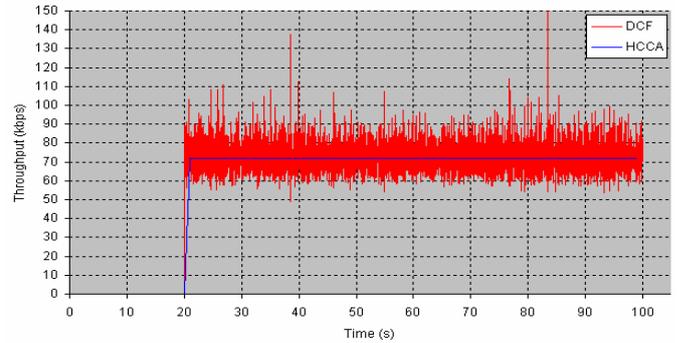
Gambar 15. Jitter Trafik Video DCF



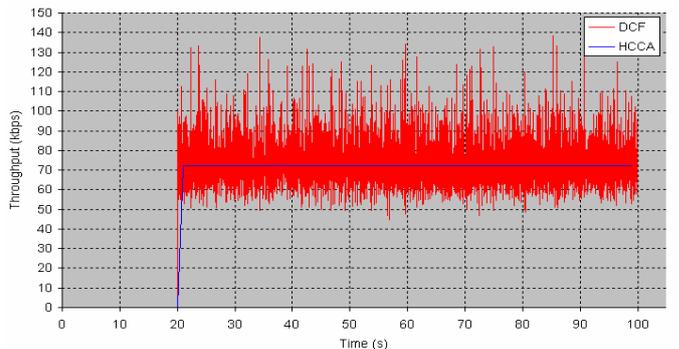
Gambar 16. Jitter Trafik Video HCCA

Gambar 15 memperlihatkan *jitter* trafik *video* pada protokol MAC DCF dan gambar 16 merupakan *jitter* menggunakan protokol MAC HCCA. Dari gambar 15 dan 16 tersebut juga dapat kita katakan bahwa *jitter* dengan menggunakan protokol MAC HCCA cenderung konstan bila dibandingkan dengan menggunakan protokol MAC DCF yang cenderung fluktuatif.

Hasil ini sama seperti pada gambar 13 dan 14 untuk trafik *audio*. Hal ini dikarenakan adanya mekanisme *polling* pada HCCA yang menyebabkan masing-masing *node*/stasiun hanya dapat transmit selama durasi TXOP-nya, dimana jarak antara dua TXOP yang berurutan dijaga konstan sesuai dengan *Service Interval*-nya. Sedangkan pada DCF, mekanisme pengiriman pakatnya yang secara *contention* menyebabkan paket-paket dikirim tidak pada interval waktu yang tetap. Durasi pengiriman untuk tiap *node* pun dapat berbeda tergantung pada waktu *node* tersebut men-*sensing* mediumnya, dan hanya dapat mengirimkan paket apabila medium *idle*.



Gambar 17. Throughput Audio Terhadap Waktu Pada Flow 1

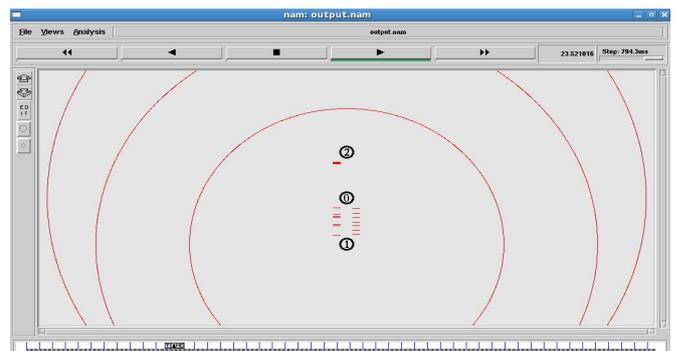


Gambar 18. Throughput Audio Terhadap Waktu Pada Flow 2

Gambar 17 menunjukkan *throughput* yang dihasilkan pada *flow* 1 (*downlink*) untuk trafik *audio*. Sedangkan gambar 18 memperlihatkan hasil untuk trafik *flow* 2 (*uplink*). Gambar tersebut dapat menunjukkan perbedaan *throughput* pada protokol MAC HCCA yang cenderung konstan di angka 72 kbps, dibandingkan protokol MAC DCF yang cenderung fluktuatif.

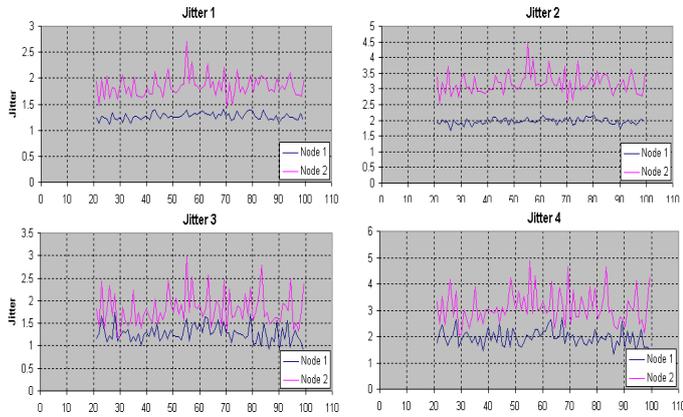
B. Skenario 2

Gambar 19 menunjukkan tampilan topologi simulasi skenario 2 pada *Network Animator*.

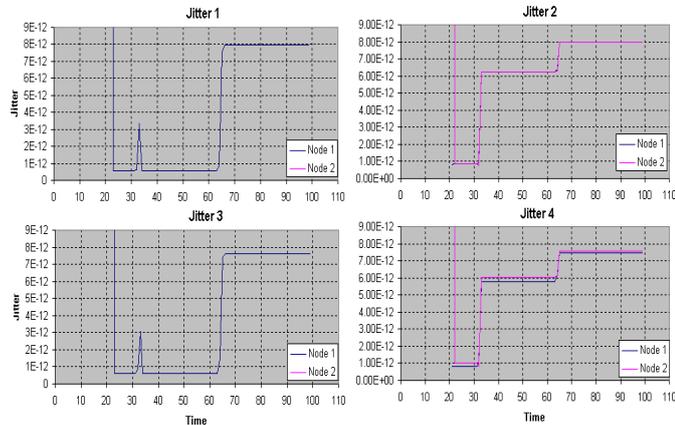


Gambar 19. Skenario 2 Pada NAM

Hasil dari simulasi NS-2 pada skenario 2 ini, yaitu skenario dengan pemberian trafik *Best Effort* pada jaringan WLAN dimana satu *node*/stasiun mengirimkan trafik *audio* atau *video bidirectional* sedangkan *node* yang lain memancarkan trafik *Best Effort bidirectional* digambarkan oleh gambar 20 dan gambar 21.

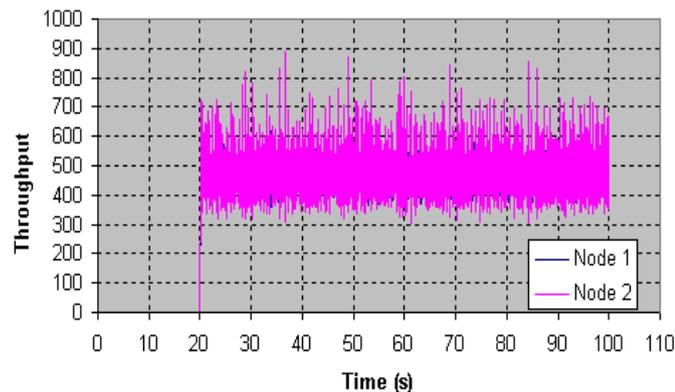


Gambar 20. *Jitter* Pada Protokol MAC DCF

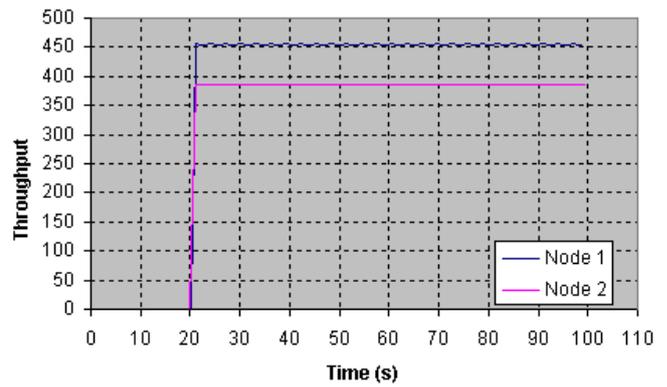


Gambar 21. *Jitter* Pada Protokol MAC HCCA

Gambar 21 memperlihatkan *jitter* pada protokol MAC HCCA. Dari gambar tersebut didapat bahwa *jitter* dengan menggunakan protokol MAC HCCA tidak berubah (konstan) walaupun ada perubahan pada *jitter* 2 dan 4. Sementara untuk DCF yang diperlihatkan pada gambar 20, terjadi perubahan yang signifikan pada setiap *jitter* baik untuk *node* 1 maupun *node* 2.



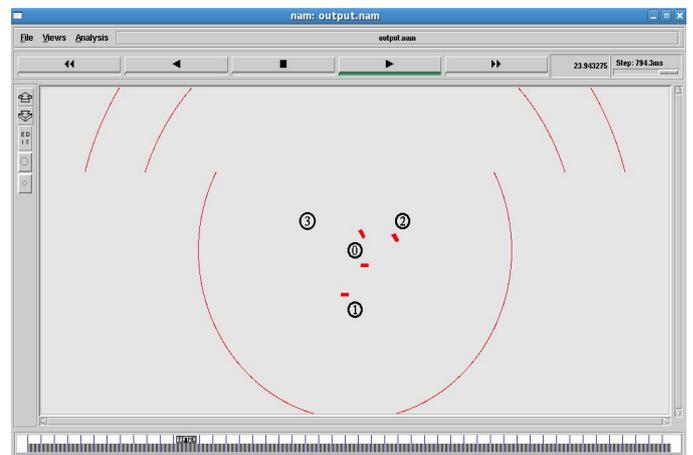
Gambar 22. *Throughput* Pada Protokol MAC DCF



Gambar 23. *Throughput* Pada Protokol MAC HCCA

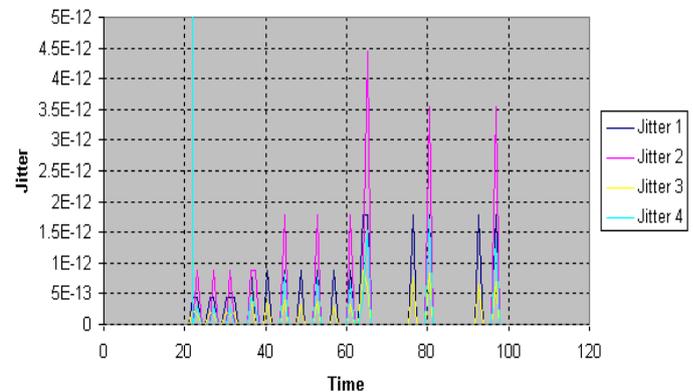
Untuk ukuran kinerja *throughput*-nya, dapat dilihat pada gambar 22 dan 23, dimana *throughput* rata-rata dengan menggunakan HCCA lebih konstan dibandingkan DCF yang sangat fluktuatif.

Skenario 3

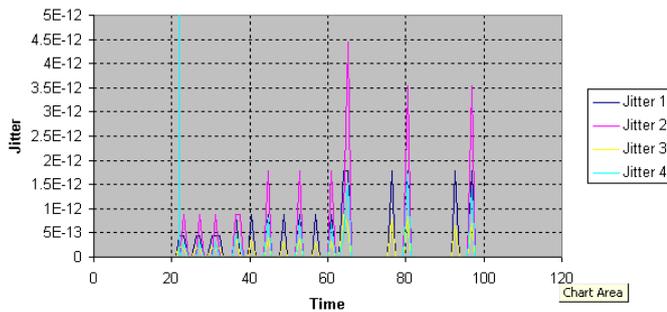


Gambar 24. Skenario 3 Pada NAM

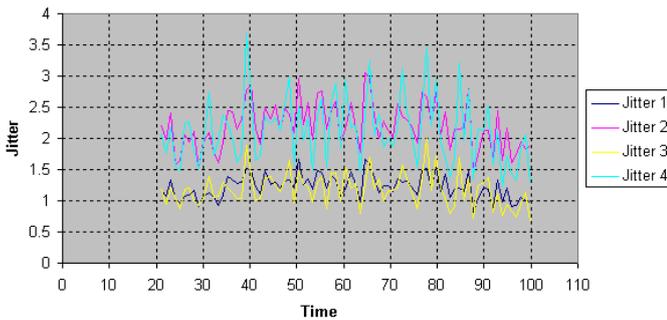
Gambar 24 menunjukkan tampilan topologi simulasi skenario 3 pada *Network Animator*. Hasil dari simulasi NS-2 pada skenario 3 ini, yaitu skenario dengan pemberian trafik *video* pada dua stasiun dengan arah *uplink* dan *downlink*, sedangkan trafik *audio* dibangkitkan oleh satu stasiun yang *bidirectional*.



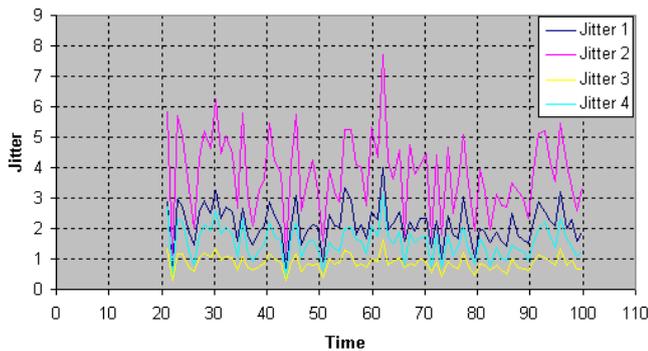
Gambar 25. *Jitter* Trafik *Video* HCCA



Gambar 26. Jitter Trafik Audio HCCA

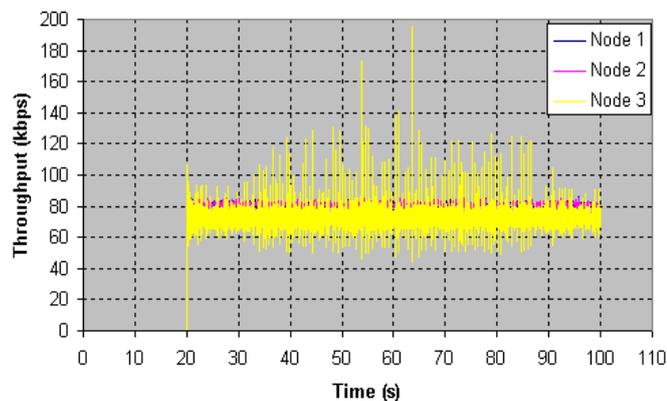


Gambar 27. Jitter Trafik Video DCF

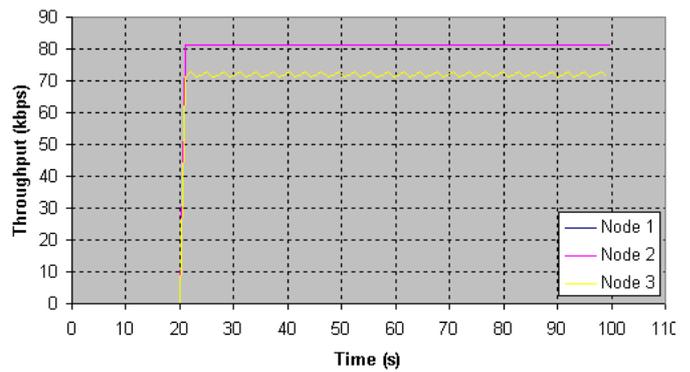


Gambar 28. Jitter Trafik Audio DCF

Sebagai hasil dari simulasi NS-2 pada skenario 3 ini, Gambar 25 dan 26 memperlihatkan *jitter* untuk trafik *audio* dan *video* pada MAC HCCA, meskipun ada fluktuasi namun pada waktu tertentu menjadi konstan. Sementara dengan DCF, hal tersebut tidak terjadi, seperti yang ditunjukkan pada gambar 27 dan 28, dimana *jitter* selalu fluktuatif.



Gambar 29. Throughput Pada Protokol MAC DCF



Gambar 30. Throughput Pada Protokol MAC HCCA

Untuk kinerja *throughput* sendiri gambar 29 dan 30 menunjukkan secara jelas pada DCF, trafik masing-masing *node* mengalami fluktuasi, sedangkan pada HCCA hal tersebut tidak terjadi. Dapat dilihat bahwa trafik pada setiap *node* mengalami *throughput* yang konstan dari awal hingga akhir simulasi.

Dari serangkaian simulasi yang dilakukan berdasarkan skenario 1 – 3, dapat dilihat bahwa pada penggunaan protokol MAC HCCA, *jitter* lebih stabil bila dibandingkan dengan protokol MAC DCF. Hal ini disebabkan HCCA menjaga agar tiap *node* hanya dapat transmit pada durasi TXOP-nya dengan jarak antar TXOP yang berhasil ditentukan oleh *Service Interval*-nya, sedangkan DCF menggunakan metode *contention*. Untuk kasus adanya trafik *Best Effort* yang membebani jaringan, protokol MAC HCCA tetap memberikan *throughput* yang stabil pada masing-masing nilai *throughput* nya sesuai dengan prioritas jenis trafik yang diberikan, dan ini menjelaskan setiap jenis trafik mendapatkan kualitas layanan yang baik. Sementara DCF menghasilkan *throughput* yang masih mengalami fluktuasi yang besar, yang berarti setiap jenis trafik masih saling bersaing untuk mendapatkan bandwidth. Pada kasus dua jenis trafik yang berbeda, seperti pada *video* dan *audio*, HCCA melayani masing-masing trafik, tanpa mempengaruhi kondisi trafik lain. Berbeda pada DCF, dimana masing-masing trafik saling bersaing.

V. SIMPULAN DAN REKOMENDASI

Hasil dari penelitian perbandingan kinerja protokol MAC IEEE 802.11e HCCA dan IEEE 802.11 DCF yang dilakukan adalah sesuai dengan teori yang menyatakan bahwa protokol MAC HCCA dapat memberikan dukungan QoS pada trafik aplikasi multimedia sesuai dengan prioritas jenis trafik yang diberikan.

Direkomendasikan untuk diambil kebijakan terhadap penerapan standar protokol MAC IEEE 802.11e HCCA pada setiap perangkat WLAN, baik di sisi AP maupun di sisi *user*, sehingga pada pertukaran data melalui jaringan WLAN, diperoleh kualitas layanan yang baik untuk setiap jenis trafik yang berbeda.

DAFTAR PUSTAKA

Cicconetti, C., Lenzini, L., Mingozzi, E., & Stea, G. (2005). A Software Architecture for Simulating IEEE 802.11e HCCA 2 . HCF Controlled Channel Access Descrip-. *IPS-MoMe'05: Proceeding from the 3rd Workshop on Internet Performance, Simulation, Monitoring and Measurement*, 97–104.

- Didi, F., Labiod, H., Pujolle, G., & Feham, M. (2009). Mobility and QoS of 802.11 and 802.11e Wireless LAN Standards. *The International Arab Journal of Information Technology*, 6(4), 403–410.
- Mittal, I., & Anand, A. (2014). WLAN Architecture. *International Journal of Computer Trends and Technology*, 8(3), 148–151.
- NS-2. (2011). NSNAM. Retrieved September 16, 2014, from http://nslam.isi.edu/nslam/index.php/Main_Page
- Poonguzhali, A. (2014). Performance Evaluation Of IEEE 802.11e MAC Layer Using Cell Processor. *International Journal of Scientific & Technology Research*, 3(1), 255–261.
- Qiang Ni, & Turletti, T. (2005). QoS support for IEEE 802.11 wireless LAN. In Y. Xiao & Y. Pan (Eds.), *Wireless LANs and Bluetooth (Wireless Networks and Mobile Computing, Vol. 4)* (pp. 23–50). Nova Science Publishers.
- Ruscelli, A. L., Cecchetti, G., Alifano, A., & Lipari, G. (2012). Enhancement of QoS Support of HCCA Schedulers Using EDCA Function in IEEE 802.11e Networks. *Ad Hoc Networks*, 10(2), 147–161. doi:10.1016/j.adhoc.2010.09.014
- Stallings, W. (2004). IEEE 802.11: Wireless LANs from a to n. *IT Professional*, 6(5), 32–37. doi:10.1109/MITP.2004.62
- The VINT Project. (2011). *The ns Manual (formerly ns Notes and Documentation)*. (K. Fall & K. Varadhan, Eds.). Retrieved from <http://www.isi.edu/nslam/ns/ns-documentation.html>
- Xiao, Y., Li, H., & Choi, S. (2004). Protection and guarantee for voice and video traffic in IEEE 802.11e wireless LANs. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (Vol. 3)* (Vol. 3, pp. 2152 – 2162). Hong Kong: IEEE. doi:10.1109/INFOCOM.2004.1354622