

SIMULASI PENINGKATAN TCP FAIRNESS MENGGUNAKAN PENDEKATAN CROSS LAYER DESIGN PADA JARINGAN AD-HOC IEEE802.11

IMPROVING TCP FAIRNESS SIMULATION USING CROSS LAYER DESIGN APPROACH IN IEEE 802.11 AD-HOC NETWORKS

Sahrul Mulia¹
(sahrul.siregar@gmail.com)

Dr. Rendy Munadi, Ir., MT.²
(rnd@telkomuniversity.ac.id)

Imam Hedi Santoso, ST., M.T.³
(ihs@telkomuniversity.ac.id)

Fakultas Elektro – Universitas Telkom
Jl. Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia

ABSTRAK

Exposed terminal merupakan masalah yang terjadi pada jaringan *ad-hoc* yang dapat mengakibatkan *unfairness* pada *host-host* yang saling berkaitan. Permasalahan *exposed terminal* muncul akibat dari pengadopsian protokol TCP yang didesain untuk memberikan layanan pada jaringan wired, diimplementasikan pada jaringan wireless. Pada protokol TCP, saat sebuah *host* mengirim paket kepada *host* lain, maka *host* pengirim akan menunggu *acknowledgement* (ACK) dari *host* penerima. Saat paket ACK tidak diterima oleh *host* pengirim, maka *host* penerima akan menganggap adanya kongesti dalam jaringan sehingga *host* pengirim akan menjalankan mekanisme *congestion control*. Terlebih lagi pada jaringan *wireless* menggunakan mekanisme CSMA/CA pada mekanisme *congestion control*-nya, yaitu dengan mengirimkan paket pemberitahuan saat akan melakukan pengiriman dan membuat *host* lainnya diam dan menunggu pengiriman paket dan ACK nya. Hal ini menyebabkan penurunan *throughput* dari jaringan.

Penanggulangan masalah *exposed terminal* yang diusulkan pada tugas akhir ini adalah dengan memotong beberapa mekanisme *congestion control* seperti pengabaian paket RTS/CTS, penghilangan NAV, dan pemotongan beberapa fungsi *backoff*. Dengan desain tersebut, setiap *node* diharapkan mendapatkan kesempatan lebih untuk mengirimkan paket tanpa terganggu dengan banyaknya waktu menunggu. Hal tersebut cukup efektif pada topologi tertentu. Namun hal tersebut tidak cukup untuk menanggulangi masalah *exposed terminal* pada topologi-topologi lainnya. Sehingga dirancanglah skema *cross layer design* untuk melengkapi desain usulan penanggulangan masalah *TCP fairness* pada jaringan *ad-hoc*. Layer MAC akan berfungsi setelah layer network memberikan informasi-informasi yang diperlukan seperti jumlah *hop* dan jarak antar *hop*. Dengan *cross layer design* ini, maka penanggulangan terhadap *exposed terminal* dilakukan saat *node* benar-benar berada dalam kondisi *ter-exposed*.

Kata kunci: IEEE 802.11, *Exposed Terminal*, *TCP Fairness*, *Cross Layer Design*

ABSTRACT

Exposed terminal is a serious problem in *ad-hoc* network that causes *unfairness* between *hosts*. TCP protocol adoption is the root of the *exposed terminal* problem. TCP was developed for wired network, and became a problem when it implemented in wireless network. TCP protocol which was designed for wired environments will always assume that the absence of acknowledgment from the receiver as a sign of congestion in the network. Moreover, wireless network uses CSMA/CA for congestion control mechanism. In CSMA/CA, a node sends a "notification" packet that the transmitter node wants to send packets and makes another nodes that receive the notification packet waiting for amount of time until the packets and the ACK packets are sent. This mechanism will reduce transmission throughput.

The simulation proposed in this Final Project is to cut some part of the congestion control mechanism such as ignore RTS/CTS packets, eliminate NAV, and cut through some backoff timer. For more option, the solution equipped with *cross layer design* to make it more adaptable in many topologies. MAC layer interrogate informations reside at network layer before take any decisions. Information from network layer comes in form of number of hop to receiver as well as another nodes.

Key words: IEEE 802.11, *Exposed Terminal*, *TCP Fairness*, *Cross Layer Design*

BAB I. PENDAHULUAN

1.1 Latar Belakang

Kebutuhan manusia akan komunikasi terus meningkat. Hal tersebut merupakan hal yang wajar dan pasti terjadi karena manusia merupakan makhluk sosial yang tidak bisa hidup sendirian. Seiring bertambahnya kebutuhan berkomunikasi, teknologi komunikasi juga berkembang dengan sangat pesat.

Dari konsep *ad-hoc*, muncullah sebuah teknologi yang menggabungkan beberapa *host wireless* ke dalam satu jaringan tanpa menggunakan *access*

point, yang biasa dikenal sebagai *Wireless Mesh Networks*. Hal ini memungkinkan setiap *host* melakukan koneksi ke *host* yang ada dalam jangkauannya untuk saling berhubungan langsung. Koneksi ini jauh lebih praktis dibandingkan dengan jaringan kabel ataupun *wireless LAN* yang masih menggunakan *access point*.

Teknologi WMN bukanlah teknologi baru, namun teknologi ini memang tidak sepopuler induknya. Karena itu perkembangan teknologi ini masih terus berjalan dan masih banyak yang harus disempurnakan. Misalnya adalah masalah yang akan

dibahas pada tugas akhir ini, yaitu menanggulangi *exposed terminal problem*.

BAB II. DASAR TEORI

2.1 WLAN dan IEEE 802.11

Teknologi ini dimaksudkan untuk menggantikan kabel UTP. Teknologi yang populer untuk jaringan LAN adalah WiFi. Banyak orang mencampuradukkan antara IEEE dengan 802.11 nya dengan WiFi, dan menganggap semua produk *wireless* adalah WiFi, bahkan ada juga orang yang mengira WiFi merupakan singkatan dari *wireless*. [7].

IEEE 802.11 adalah sebuah standard WLAN yang dirumuskan oleh lembaga IEEE.

2.2 Wireless mode ad-hoc

Ad-hoc merupakan bentuk jaringan *wireless* yang paling sederhana. *Ad-hoc* disebut juga sebagai jaringan *peer-to-peer* dan kadang-kadang dinamakan IBSS (*Independent Basic Service Set*). Dengan jaringan *ad-hoc*, kita bisa menghubungkan beberapa komputer ke dalam sebuah jaringan tanpa menggunakan peralatan tambahan seperti *Access Point* (AP).

2.3 Protokol MAC pada IEEE 802.11

Sub layer MAC mendefinisikan *Distributed Coordination Function* (DCF) dan *Point Coordination Function* (PCF) sebagai fungsi akses medium. Namun, PCF tidak akan dibahas karena digunakan untuk jaringan IEEE 802.11 yang berbasis infrastruktur. [5].

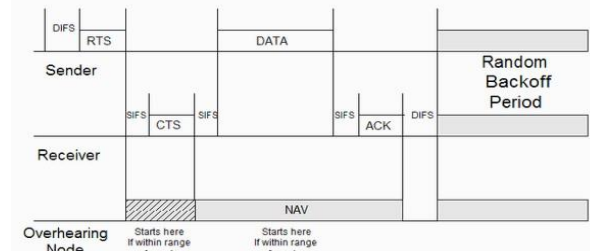
DCF adalah mekanisme *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) bersama dengan *positive acknowledgement*. DCF dapat bekerja di dalam dua mode yaitu *basic access mode* dan mode yang melibatkan frame-frame *Request to Send* (RTS) serta *Clear to Send* (CTS).

Pada *basic access mode*, sebuah stasiun yang akan mengirimkan frame akan “merasakan” (*sense*) medium. Jika medium *idle* selama suatu selang waktu yang disebut *Distributed Inter-Frame Space* (DIFS) maka stasiun tersebut akan mengirimkan frame. Jika medium tidak *idle*, maka pengiriman frame akan ditunda lalu proses *backoff* akan dimulai. Pada fase *backoff* ini, suatu stasiun akan memilih suatu waktu acak yang terdistribusi antara nol sampai dengan ukuran *Contention Window* (CW) lalu menjalankan *backoff flow*. *Flow* ini akan dikurangi terus secara periodik bila medium terasa *idle* untuk jangka waktu yang melebihi DIFS. *Backoff flow* akan dihentikan jika suatu transmisi terdeteksi. Skema *backoff* yang didefinisikan pada IEEE 802.11 ini disebut *Binary Exponential Backoff* (BEB).

Mode akses yang melibatkan frame RTS dan CTS diterapkan untuk menghindari masalah *hidden terminals*. Pada metoda ini, *carrier sense* dapat dilakukan secara fisik maupun *virtual*. *Virtual Carrier Sense* menggunakan *medium reservation* dengan cara saling mempertukarkan frame RTS dan CTS sebelum mengirimkan data. Frame RTS dan CTS mengandung

informasi tentang node tujuan serta durasi pengiriman baik data maupun ACK. Penggunaan RTS dan CTS dikendalikan oleh *RTS threshold*.

Setelah merasakan *mediumidle* selama DIFS, pengirim mengirimkan RTS ke penerima untuk “memesan” medium (*medium reservation*). Penerima akan menanggapi dengan CTS jika siap menerima frame dan *mediumidle* untuk paling tidak selama SIFS. Seluruh stasiun yang “mendengarkan” RTS, CTS, atau keduanya, menggunakan informasi durasi untuk mengupdate alokasi vektor jaringan (*Network Allocation Vector* (NAV)) untuk melakukan *virtual carrier sensing*. Terminal-terminal yang *hidden* akan mendengar CTS dan mengurungkan transmisi untuk mencegah *collision*. Setelah menerima CTS dan menunggu *mediumidle* selama SIFS, pengirim akan memulai transmisi frame seperti pada *basic DCF*. Jika stasiun tidak menerima CTS, maka ia akan memasuki fase *backoff* dan meretransmisi RTS. Gambar II.2 memperlihatkan mekanisme pertukaran pesan pada DCF.

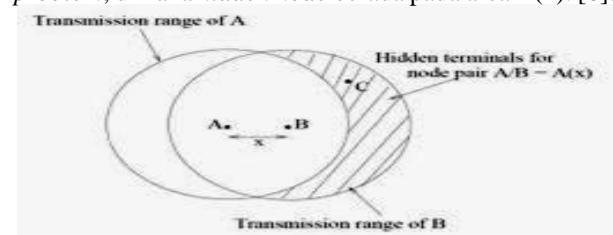


Gambar 2.1 Pertukaran pesan pada DCF. [6].

2.4 Hidden dan Exposed Terminal Problem

2.4.1 Hidden terminal problem

Saat node A akan mengirimkan data ke node B yang berjarak x dari node A. dengan hanya *medium sensing*, node A tidak akan dapat mendengar transmisi dari node lain (misalnya node C) yang berada pada area yang diarsir. Kemudian saat node A mengirim paket, sedangkan node C juga sedang mengirimkan paket, maka akan terjadi paket *collision* pada node B. hal ini dikenal sebagai *hidden terminal problem*, dimana *hidden node* berada pada area $A(x)$. [6].

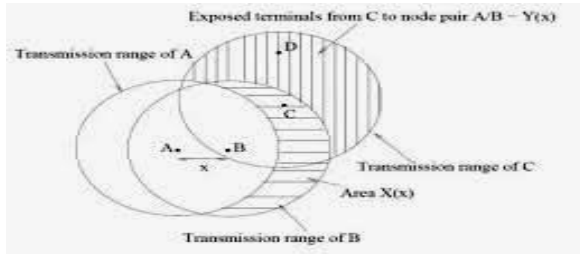


Gambar 2.2 Hidden terminal

2.4.2 Exposed terminal problem

Antena node A mengirim RTS dan menunggu node B untuk mengirim balasan CTS. Pada saat yang sama, node D yang berada di daerah $Y(x)$ akan mengirim data ke node C yang ada di area $X(x)$, dan D mengirim RTS dari D, C mengirim CTS dan paket CTS ini tertangkap juga oleh B yang kemudian node B akan melakukan *backoff* dan membatalkan pengiriman

CTS ke A. maka, semua transmisi dari *node* di daerah Y(x) ke daerah X(x) akan menggagalkan transmisi data dari A ke B, meskipun transmisi simultan dari daerah Y(x) ke X(x) tidak akan mengganggu transmisi dari A ke B. kita mendefinikan terminal di daerah Y(x) sebagai *exposed terminal* untuk pasangan A *node* / B. untuk sebagian besar topologi, *exposed terminal problem* akan mendominasi pada jaringan *ad-hoc*, dengan catatan, jaringan yang terbentuk haruslah cukup besar.



Gambar 2.3 Exposed Terminal

2.5 Cross Layer Design

Di dalam kerangka kerja arsitektur *layered*, perancang memiliki dua pilihan pada saat merancang protokol. Pilihan pertama adalah protokol dapat dirancang berdasarkan aturan arsitektur referensi. Di dalam arsitektur *layered* hal ini memiliki makna bahwa perancangan protokol seperti itu hanya menggunakan layanan layer-layer yang lebih bawah dan tidak peduli tentang rincian penyediaan layanan tersebut. Mengikuti arsitektur juga menimbulkan konsekuensi bahwa protokol tidak memerlukan interface apapun yang tidak tersedia pada arsitektur referensi. [5].

Pilihan kedua bagi perancang adalah protokol dirancang dengan cara melanggar arsitektur referensi. Sebagai contoh, protokol dirancang untuk memperbolehkan komunikasi langsung antar protokol pada layer-layer yang tidak berdekatan atau menggunakan secara bersama suatu variabel. Bentuk pelanggaran terhadap arsitektur *layered* tersebut merupakan *cross layer design* dengan arsitektur referensi yang berbeda. Dengan demikian *cross layer design* dapat didefinisikan sebagai protokol yang dirancang dengan melanggar arsitektur komunikasi *layered*.

2.6 TCP Fairness dan Fairness Index

Defenisi *fairness* paling dasar adalah suatu kondisi dimana semua *wireless node* (lebih tepatnya adalah setiap *flow*) di dalam jaringan memperoleh alokasi *bandwidth* yang sama. [8] [2].

Kemudian akan diukur dengan *fairness index*, yaitu bilangan real yang mengukur berapa adil atau tidak adil sumber daya yang dibagi di antara *host*.

$$Fairness_index = g(y_1, y_2, \dots, y_n) = \dots(2)$$

Pada persamaan tersebut, n menyatakan jumlah individual (*node, flows, dsb*) dan y_i adalah bagian *resource* yang diterima oleh individual ke-i. bila *resource* di-share secara *fair* maka nilai indeks adalah

1. Bila satu individual menerima seluruh bagian *resource* sedangkan yang lain tidak mendapat bagian maka indeks adalah 1/n. nilai indeks di antara 1 dan 1/n menunjukkan derajat *fairness* dari alokasi *resource*. [1] [3].

BAB III. PERANCANGAN DAN MPLEMENTASI SISTEM

3.1 Tahap Perancangan

Parameter yang digunakan dalam menjalankan simulasi ini: Tabel 3.1 Parameter penelitian

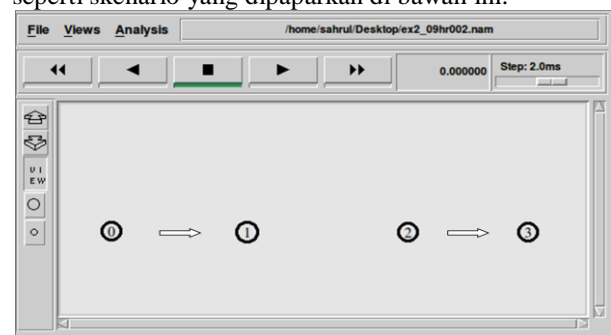
Parameter	Nilai
Simulator	NS 2.34
Tipe trafik	ftp
Jumlah node	4
Jumlah data source	2 ftp connection
Waktu simulasi	100 detik
Luas area simulasi	2000 x 2000 m
Transmisi rate	1Mbps dan 5 Mbps
Range transmisi	250 m

Tabel 3.2 Parameter *network interface*

Parameter	Nilai
Channel type	Wireless channel
Propagation model	Two ray ground
Network interface model	Wireless
MAC	IEEE 802.11
Network interface queue type	Drop tail/pri Queue
Antenna model	Omni Directional

3.2 Algoritma yang Diusulkan pada Cross Layer

Pada penelitian ini diusulkan solusi untuk menanggulangi masalah *exposed terminal*. Prinsip dasar dari algoritma baru yang diusulkan salah satunya seperti skenario yang dipaparkan di bawah ini:

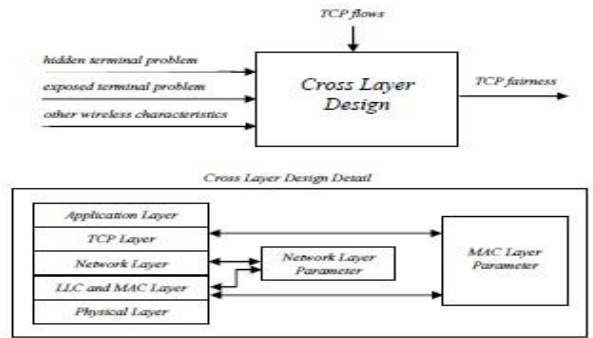
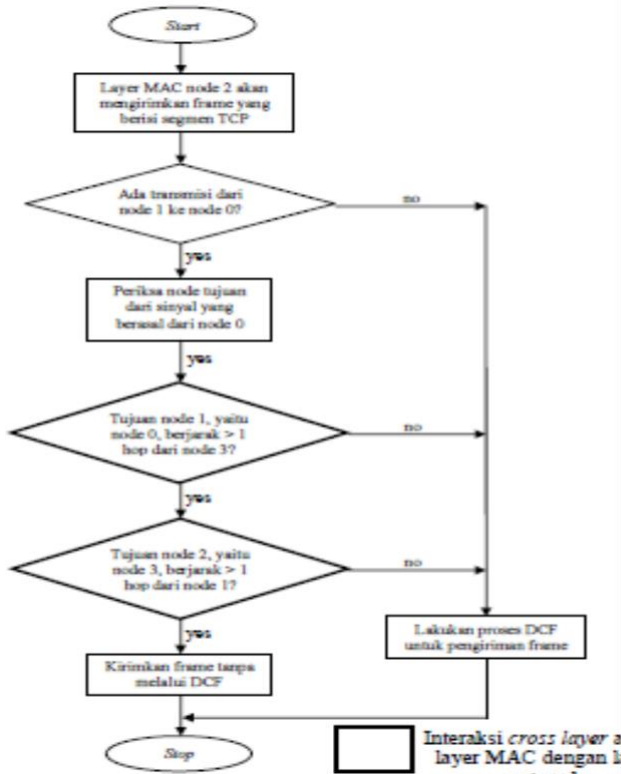


Gambar 3.1 Ilustrasi untuk menjelaskan algoritma yang diusulkan

Berdasarkan gambar di atas, *node 2* sebagai pengirim pada *ftp2*, terekspose oleh *node 1*. Berdasarkan layer *transport*, *node 2* akan mengirimkan segmen TCP ke *node 3*. Gejala *unfairness* muncul akibat ketidakmampuan *node 2* mengirimkan segmen data TCP (TCP_DATA) kepada *node 3* akibat terekspose oleh *node 1*. Terhambatnya pengiriman TCP_DATA ini menyebabkan penurunan *throughput* kepada aliran *ftp2*, karena *node 2* mengirimkan segmen TCP_DATA yang ukuran segmennya lebih panjang daripada segmen yang berasal dari *node 1* yang mengirimkan *acknowledgement* terhadap segmen TCP dari *node 0* (TCP_ACK). [9].

Untuk menanggulangi masalah *exposed terminal* di atas, penelitian ini diusulkan algoritma baru. Algoritma baru ini modifikasi dari algoritma DCF (algoritma yang dijalankan pada sublayer MAC) yang diterapkan pada jaringan *wireless ad-hoc* berbasis IEEE 802.11 b. Pada DCF, *node 2* yang berada dalam kondisi terekspose harus diam selama *node 1* melakukan transmisi. Padahal *node 3*, yang merupakan tujuan *node 2*, memiliki posisi di luar jangkauan sinyal *node 1*.

Melalui pendekatan *layered*, *node 2* tidak mampu mengetahui bahwa *node 1* dan *node 3*, serta *node 0* dan *node 3*, tidak berada dalam satu cakupan transmisi sinyal. Oleh karena itu pada penelitian ini diusulkan suatu algoritma baru dengan pendekatan *cross layer*. Algoritma baru ini memungkinkan suatu *node* yang terekspose melakukan transmisi walaupun sedang berada di dalam keadaan terekspose asalkan *node* tujuan dari *node* terekspose tersebut berada di luar jangkauan *node* yang mengekspose *node* yang bersangkutan.



Gambar 3.2 Algoritma *cross layer design* yang diusulkan. [5].

3.3 Perancangan MAC-802_11

Pada penelitian ini, dilakukan beberapa perubahan terhadap standard MAC 802.11 yang sudah ada. Seperti yang ditunjukkan di bawah ini, mekanisme transmit pada MAC 802.11 adalah sebagai berikut:

```

Recv(down)-> send()-> sendData() and
sendRTS()-> start defer timer-> deferHandler()->
check_pktRTS()-> transmit()-> rcv()-> receive timer
started-> rcv_timer()-> rcvCTS()-> tx_resume()->
start defer timer-> rx_resume()-> deferHandler()->
check_pktTx()-> transmit()-> rcv()-> receive timer
started -> rcv_timer()-> rcvACK()-> tx_resume()->
callback_-> rx_resume()-> done!
    
```

Sedangkan mekanisme 802.11 yang diusulkan adalah sebagai berikut:

```

Recv()-> send()-> sendData()-> start defer
timer-> rx_resume()-> deferHandler()->
check_pktTx()-> transmit()-> rcv()-> receive timer
started-> rcv_timer()-> rcvACK()-> tx_resume()->
callback_-> rx_resume()-> done!
    
```

Backoffhandler adalah *handler* yang dipanggil saat waktu *backoff* telah habis. Fungsi ini mengecek paket yang ada, kemudian melakukan *defer*. Sedangkan *deferhandler* adalah fungsi yang dipanggil, maka *node* telah melakukan cukup waktu menunggu untuk mengurangi kemungkinan *collision* (tabrakan paket). Fungsi ini akan mengecek paket, apakah paket berbentuk paket *control*, RTS maupun paket data, kemudian fungsi akan memanggil fungsi *check* dari paket tersebut yang kemudian fungsi *check* tersebut akan memanggil fungsi *transmit*. Dengan memotong fungsi *backoff* maka akan memperbesar kemungkinan dapat mengirim paket. Lama dari waktu *defer* juga dapat diatur untuk menaikkan *TCP fairness*.

Kemudian juga dicoba untuk menghilangkan NAV (*network allocation vector*), yaitu pada fungsi *capture* dan *rcv_timer*. Pada fungsi *capture* penghilangan NAV mengakibatkan paket yang ditangkap namun tidak cukup *power* (*transmitter* di luar jangkauan) akan langsung dibuang tanpa menunggu selama waktu NAV.

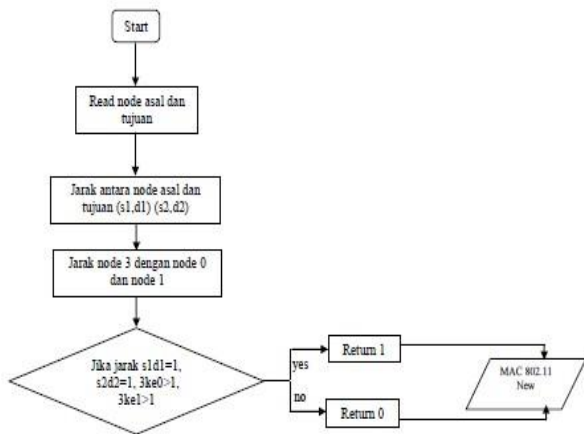
Fungsi *rcv_timer* adalah fungsi *handler* yang dipanggil saat waktu *receive* (*mhRcv*) berakhir. Di akhirnya *timer receive* berarti bahwa paket telah diterima sepenuhnya, dan sekarang dapat ditindaklanjuti. Pada fungsi *rcv_timer* penghilangan NAV menyebabkan pembebasan paket saat tabrakan paket,

paket *error* ataupun paket salah alamat dilakukan tanpa menunggu selama NAV. Sehingga meskipun *node* akan tetap mencoba mengirim paket meskipun medium dalam keadaan sibuk, meskipun tidak selalu berhasil mengirimkan paket karena masih banyak variabel-variabel lain.

3.4 Perancangan Cross Layer Design

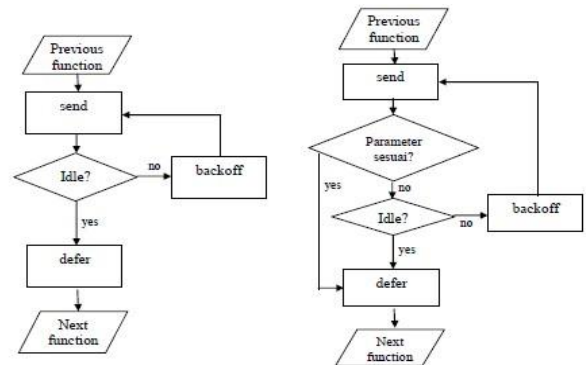
Skema *cross layer design* adalah menggabungkan *tabel routing* yang ada pada layer 3 dengan mekanisme MAC 802.11. Pada penelitian ini, *cross layer design* menggunakan sebuah tabel dinamis dari *routing protocol* OLSR. Dari *routing tabel* tersebut diambil parameter jumlah *hop* dan jarak antar *hop*, yang kemudian dijadikan parameter yang akan mengatur parameter-parameter di MAC 802.11 diusulkan. Kemudian parameter-parameter di MAC 802.11 yang diusulkan akan menjalankan proses sesuai fungsi-fungsi yang ada. Jika parameter yang dijalankan sesuai dengan ketentuan yang ada pada *file Readtable.cc* maka, pada MAC 802.11 yang diusulkan akan menjalankan skema seperti yang ada pada perancangan MAC, sedangkan jika tidak, maka MAC akan menjalankan 802.11 seperti biasa, namun tetap tanpa RTS/CTS.

Hal pertama yang dilakukan adalah menentukan asal dan tujuan dari *node*. Informasi *node* sumber (*source*), *node* tujuan (*destination*), *hopcount*, dan jarak antar *node* terdapat pada *routing table* OLSR. Kemudian dilakukan proses penentuan parameter. Jika parameter sesuai, bernilai 0. Namun jika tidak, bernilai 1. Kemudian hasil tersebut akan dipakai pada MAC 802.11new.cc.

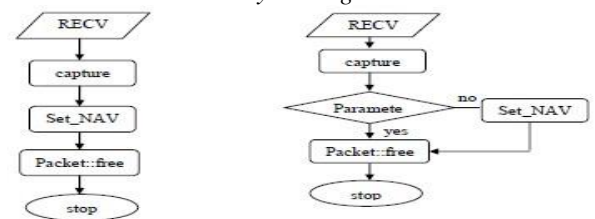


Gambar 3.8 Flowchart algoritma ReadTable.cc

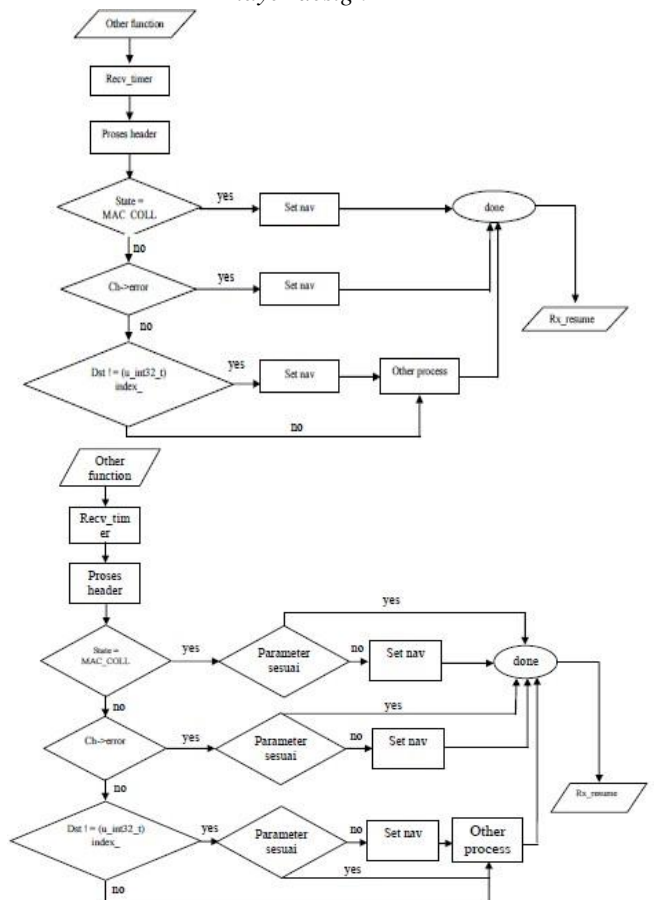
Berikut adalah fungsi *send()*, fungsi *capture()*, dan fungsi *recv_timer()* setelah ditambahkan parameter *cross layer design*. Kesesuaian parameter diambil dari *Readtable.cc*, kesesuaian parameter mengakibatkan pemotongan proses dari *backoff* ke *defer timer*, pemotongan fungsi *set_NAV* yang merupakan fungsi menunggu selama jumlah waktu tertentu, dan pengabaian fungsi *set_NAV*.



Gambar 3.9 Flowchart perbandingan algoritma fungsi send() pada MAC 802.11 dengan MAC 802.11 baru yang diusulkan yang sudah ditambah parameter *cross layer design*



Gambar 3.10 Flowchart perbandingan algoritma fungsi capture() pada MAC 802.11 dengan MAC 802.11 baru yang diusulkan yang sudah ditambah parameter *cross layer design*



Gambar 3.11 Flowchart perbandingan algoritma fungsi recv_timer() pada MAC 802.11 dengan MAC 802.11 baru yang diusulkan yang sudah ditambah parameter *cross layer design*

BAB IV. PENGUJIAN DAN ANALISIS

4.1 Pengujian

Skenario simulasi yang dilakukan terdiri dari 3 topologi pengujian:

- pengujian 1: 1 ke 0 dan 2 ke 3
- pengujian 2: 0 ke 1 dan 3 ke 2
- pengujian 3 :0 ke 1 dan 2 ke 3

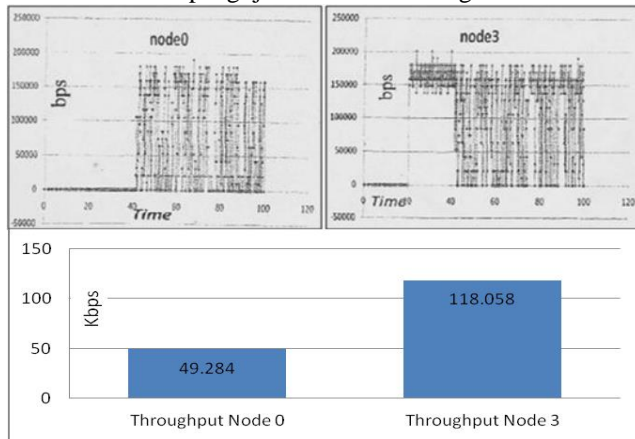
4.2 Pengujian pada data rate 1 Mb

Pengujian pada data rate 1 Mb dilakukan tanpa mengubah parameter-parameter yang sudah ada NS-2. Secara default, NS-2 telah menset data rate 802.11 pada 1 Mb.

a. Pengujian 1 pada 1 Mb

Percobaan pertama dilakukan dengan seluruh node menggunakan MAC 802.11 biasa.

Hasil dari pengujian ini adalah sebagai berikut



Gambar 4.2 Throughput pada node 0 dan 3 MAC 802.11 pada 1 Mb

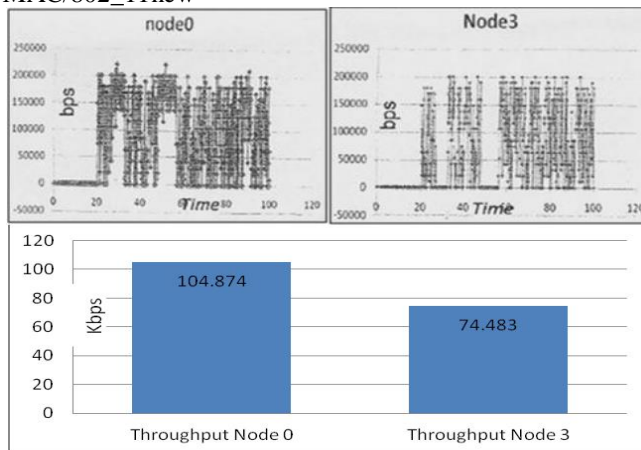
Merujuk pada persamaan (2).

$$\text{Fairness_index} = g(y_1, y_2, \dots, y_n) = \frac{1}{n} \sum_{i=1}^n \frac{y_i}{\max(y_1, y_2, \dots, y_n)}$$

Diketahui: $y_1 = 49.284$ kbps ; $y_2 = 118.058$ kbps ; $n = 2$

$$\text{Fairness_index} = \frac{1}{2} \left(\frac{49.284}{118.058} + \frac{118.058}{118.058} \right) = 0,855$$

Kemudian diubah menggunakan MAC 802.11 yang diusulkan dengan cara mengubah MAC/802_11new

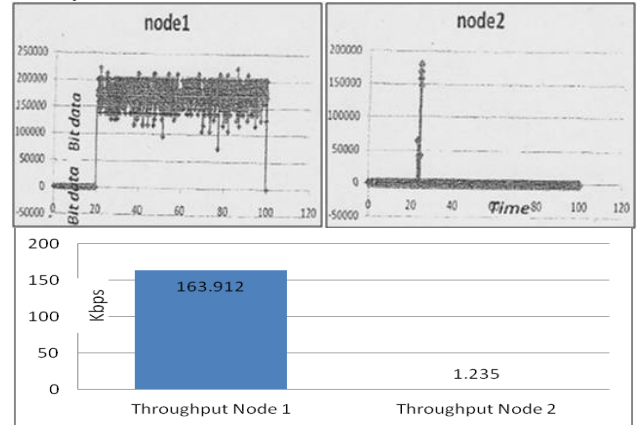


Gambar 4.3 Throughput pada node 0 dan 3 MAC 802.11new pada 1 Mb

$$\text{Fairness_index} = g(y_1, y_2, \dots, y_n) = \frac{1}{2} \left(\frac{163.912}{163.912} + \frac{1.235}{163.912} \right) = 0,9721$$

b. Pengujian 2 pada 1 Mb

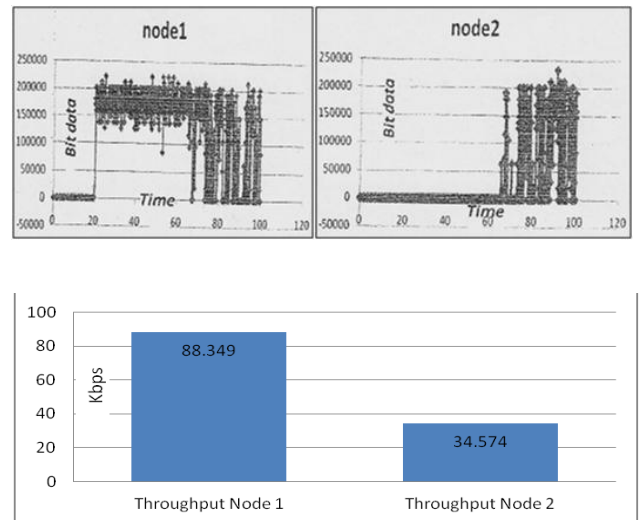
Pertama, seluruh node akan dipasang dengan MAC 802.11 standard sebagai pembanding. Dan hasilnya adalah:



Gambar 4.4 Throughput pada node 1 dan 2 MAC 802.11 pada 1Mb

$$\text{Fairness_index} = g(y_1, y_2, \dots, y_n) = \frac{1}{2} \left(\frac{88.349}{88.349} + \frac{34.574}{88.349} \right) = 0,507539$$

Kemudian menggunakan MAC 802.11 baru yang diusulkan. Hasil throughput-nya adalah sebagai berikut.

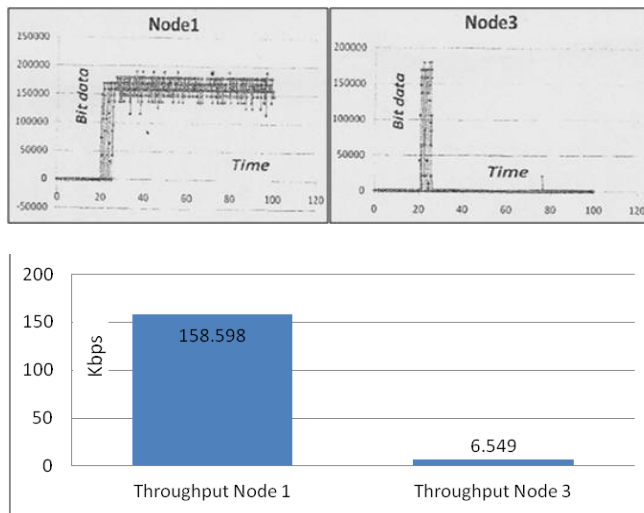


Gambar 4.5 Throughput pada node 1 dan 2 MAC 802.11new pada 1Mb

$$\text{Fairness_index} = g(y_1, y_2, \dots, y_n) = \frac{1}{2} \left(\frac{104.874}{104.874} + \frac{74.483}{104.874} \right) = 0,8396$$

c. Pengujian 3 pada 1 Mb

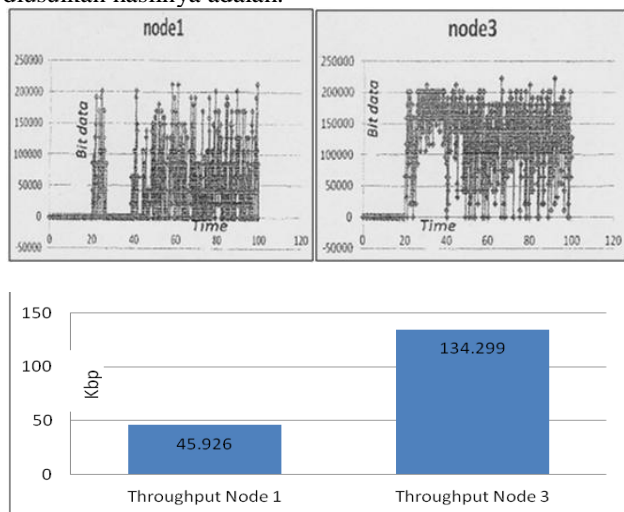
Pertama-tama, seluruh node dipasangkan dengan MAC 802.11 standard, hasil throughput-nya adalah sebagai berikut



Gambar 4.6 Throughput pada node 1 dan 3 MAC 802.11 pada 1 Mb

$$\text{Fairness_index} = g(y_1, y_2, \dots, y_n) = \frac{6.549}{158.598} = 0,041$$

Kemudian menggunakan MAC 802.11 baru yang diusulkan hasilnya adalah:



Gambar 4.7 Throughput pada node 1 dan 3 MAC 802.11 new pada 1 Mb

$$\text{Fairness_index} = g(y_1, y_2, \dots, y_n) = \frac{134.299}{45.926} = 2,924$$

4.3 Pengujian pada data rate 5 Mb

Pengujian berikutnya adalah dengan menggunakan data rate 5 Mb, yaitu dengan menambahkan baris fungsi pada file TCL MAC/802_1new set dataRate_ 5.0e6 ;# 5Mbps

Hal ini akan menaikkan data rate dari node menjadi 5 Mbps.

Kemudian pengujian 1, 2, dan 3 dilakukan pada pengaturan data rate 5 Mb ini.

4.4 Hasil Pengujian

Berikut adalah tabel hasil seluruh pengujian

No	Data Rate	Topologi Pengujian	Node	Throughput		Fairness Index	
				MAC 802.11 (Kbps)	MAC 802.11 Baru (Kbps)	MAC 802.11	MAC 802.11 Baru
1	1 Mb	Pengujian 1	Node 0	49.284	104.874	0.855	0.9721
			Node 3	118.058	74.483		
2	1 Mb	Pengujian 2	Node 1	163.912	88.349	0.507539	0.8396
			Node 2	1.235	34.574		
3	1 Mb	Pengujian 3	Node 1	158.598	45.926	0.541	0.806
			Node 3	6.549	134.299		
4	5 Mb	Pengujian 1	Node 0	0	300.098	0.501	0.99
			Node 3	424.542	318.754		
5	5 Mb	Pengujian 2	Node 1	438.326	412.575	0.5	0.73
			Node 2	0	154.668		
6	5 Mb	Pengujian 3	Node 1	369.943	322.314	0.6685	0.9743
			Node 3	64.215	232.249		

Tabel 4.1 Hasil seluruh pengujian

4.5 Analisis

4.5.1 Analisis Pengujian pada data rate 1 Mb

Pada pengujian pertama, pengiriman data lebih banyak dilakukan oleh node 1 dan node 2. Sedangkan node 0 dan node 3 kebanyakan hanya menerima paket atau mengirim ACK dan paket-paket lain seperti paket routing OLSR. Node yang teresponse adalah node-node transmitter, sehingga masalah exposed terminal tidak terlalu buruk. Dengan memasang MAC 802.11 standard saja, fairness index antar node dalam topologi tersebut cukup baik.

Pada pengujian dengan default rate mobile node pada NS-2, dapat dilihat bahwa throughput pada pengujian 1, dengan semua node menggunakan MAC 802.11, fairness index mencapai angka 0,855. Sedangkan dengan node 0 dan node 1 menggunakan MAC 802.11 baru yang diusulkan fairness index meningkat menjadi 0,9721. Terjadi kenaikan yang cukup signifikan dalam hal ini. Hal ini performa yang dijalankan oleh MAC 802.11 baru terlihat baik.

Pada pengujian ke 2, saat node yang mengirim adalah node-node ujung, yaitu node 0 dan node 3, maka masalah exposed terminal terjadi sangat buruk jika menggunakan MAC 802.11 standard. Fairness index mencapai angka 0.5 yang berarti kondisi terburuk bagi fairness 2 kubu. Node 3 tidak dapat mengirim data kepada node 2, karena setiap node 2 akan membalas atau memberi balasan paket dari node 3, ketika itu juga node 1 sedang melakukan pengiriman yang berarti paket dari node 3 akan di-drop. Sehingga node 2 tidak akan dapat memproses paket apapun dari node 3 akibat collision.

Namun dengan menggunakan MAC 802.11 new, fairness index dapat diperbaiki dengan sangat signifikan, yaitu mencapai angka 0,8396. Dengan menghilangkan mekanisme RTS/CTS, meniadakan proses menunggu NAV dan pemotongan terhadap beberapa proses backoff menjadikan proses node 3 dan 2 mendapat kesempatan untuk mengirimkan paket lebih banyak dari sebelumnya. Masalah exposed terminal problem teratasi dengan baik oleh MAC 802.11 yang baru.

Pada pengujian 3, dengan menggunakan MAC 802.11 standard *fairness index* tercatat pada angka 0,541. *Throughput* di *node* 1 mencapai 158,598 Kbps sedangkan *throughput* di *node* 2 hanya mencapai 6,549 Kbps. *Throughput* merupakan keberhasilan pengiriman paket ke penerima. Dengan menggunakan MAC 802.11 baru dapat meningkatkan *throughput* di *node* 3, yaitu mencapai angka 45,926 Kbps. Tetapi hal tersebut disertai penurunan *throughput* pada *node* 1 yaitu menjadi 134,299 Kbps. Sehingga *fairness index* menjadi 0,806.

4.5.2 Analisis Pengujian pada data rate 5 Mb

Pada pengujian-pengujian ini digunakan topologi yang sama dengan pengujian-pengujian pada *data rate* 1 Mb, namun dengan *data rate* yang lebih tinggi, yaitu 5 Mb. Hal ini mengakibatkan pemrosesan paket akan dilakukan jauh lebih cepat dari *data rate* 1 Mb. Dengan begitu, tanpa mekanisme RTS/CTS *collision* akan lebih besar kemungkinan terjadinya. Tabrakan-tabrakan paket yang biasa terjadi tidak terlalu mengganggu pengiriman, namun tidak menjamin paket tersebut utuh. Sedangkan tabrakan-tabrakan paket yang terjadi pada NS-2 terkadang akan berakibat buruk bagi berlangsungnya simulasi.

Pada pengujian pertama, dengan seluruh *node* menggunakan MAC 802.11, dengan pengujian normal, maka *fairness index* mencapai angka 0,501. Sedangkan dengan menggunakan MAC 802.11 yang baru, *fairness index*-nya meningkat menjadi 0,99. Terjadi kenaikan yang signifikan. Hal ini menandakan algoritma baru yang diusulkan berfungsi cukup baik.

Pada pengujian ke 2, masalah yang terjadi sama dengan pada pengujian dengan *data rate* 1 Mb, yaitu ada pasangan *node* yang tidak dapat mengirimkan data. Dapat dilihat bahwa masalah *exposed terminal* terburuk terjadi pada topologi ini, yaitu hingga menghasilkan angka *fairness index* terendah 0,5. Namun dengan mengganti MAC pada dengan MAC 802.11 yang baru, maka *fairness index*-nya meningkat menjadi 0,73.

Pada pengujian 3, *fairness index* mencapai angka 0,6685. Sedangkan dengan menggunakan MAC 802.11 baru yang diusulkan *fairness index* meningkat menjadi 0,9743. Terjadi kenaikan yang signifikan. Hal ini menandakan algoritma baru dengan menghilangkan RTS/CTS, dan pemotongan NAV dapat menanggulangi masalah *exposed terminal* yang menyebabkan *TCP unfairness*.

BAB V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Terjadi *unfairness* pada simulasi jaringan *ad hoc* menggunakan MAC 802.11 pada NS-2 dikarenakan adanya *exposed terminal problem*.
2. *TCP fairness* terendah terjadi pada topologi ke-2, yaitu saat *node* pengirim adalah *node-node* ujung sedangkan *node* penerima adalah *node* tengah, dan selalu terjadi saat menggunakan MAC 802.11.

3. Perubahan terhadap MAC 802.11 yaitu dengan menghilangkan mekanisme CTS/RTS, menghilangkan NAV *Timer* dan memotong beberapa *backoff* pada saat terjadi *exposed terminal problem* dapat menaikkan *TCP Fairness* pada topologi yang dirancang.
4. Untuk mengetahui apakah komunikasi yang berlangsung sedang terjadi *exposed terminal* atau tidak pada layer MAC, layer MAC membutuhkan informasi jumlah *hop* dan jarak antar *hop* yang terdapat pada layer Network.
5. Tidak ada teknologi yang bagus disegala kondisi, termasuk MAC 802.11 yang diusulkan tidak dapat melakukan performa yang baik di segala topologi, sehingga harus ditambah parameter baru yaitu parameter topologi.

5.2 Saran

1. Dibutuhkan beberapa penghalusan lagi pada fungsi-fungsi yang ada MAC 802.11 agar dapat lebih ditingkatkan lagi *TCP fairness*-nya.
2. Pengujian harus dicoba pada *simulator* lain yang memiliki *event scheduler* paralel sehingga dapat menangani *event-event* yang terjadi benar-benar secara bersamaan.
3. Pengujian juga secara implementasi pada dunia nyata, sehingga didapat data yang lebih realistis.

DAFTAR PUSTAKA

- [1] Deng, Jing, dkk. (2008): Fairness Index Based on Variational Distance.
- [2] Floyd, S. dan Fall, K. (1999) : Promoting the use of end-to-end congestion control in the internet, IEEE/ACM Transactions on Networking, 7(4), 458-472.
- [3] Jain, R.K., Chiu, D.W, dan Hawe, W.R. (1984): A quantitative Measure of Fairness and Discrimination for resource allocation in shared computer system, Technical Report DEC-TR-301, Digital Equipment Corporation, USA.
- [4] Jayasuriyana, Aruma, dkk. (2003): Hidden vs Exposed Terminal Problem in Ad hoc Networks. Institute for Telecommunication Research University of South Australia.
- [5] Juhana, Tuntun. (2010): Penanggulangan Masalah Exposed Terminal pada Jaringan Ad-Hoc Berbasis IEEE 802.11 Menggunakan Pendekatan Cross Layer. Disertasi ITB.
- [6] Sharma, S. (2003): Analysis of 802.11b MAC: A QoS, Fairness, and Performance Perspective, ECSL. Technical Report. CORR cs. CE/0411017.
- [7] S'to, (2007): Wireless Kung Fu Network & Hacking?. Jasakom.
- [8] Vaidya, N., Bahl, P., dan Gupta, S. (2000) : Distributed Fair Scheduling in a Wireless LAN, Proceedings of MOBICOM 2000, 167-178.
- [9] Xu, K., Gerla, M., dan Bae, S. (2003): Effectiveness of RTS/CTS Handshake in IEEE 802.11-Based Ad-hoc Network, Ad-hoc Net. J., Elsevier, 1(1) 107.