ENSC 835: COMMUNICATION NETWORKS

SPRING 2011

FINAL PROJECT

PERFORMANCE EVALUATION OF KEY 802.11 MAC PROTOCALS

Yitao Wu

Student ID, 301127569

Email Address, yitaow@sfu.ca

Biao Li

Student ID, 301154307

Email Address, biaol@sfu.ca

Team 8

Supervised by Professor Ljiljana Traikovic

http://www.sfu.ca/~yitaow/ensc835/index.html

http://www.sfu.ca/~biaol/ensc835/index.html

1. Abstract

The Wireless LAN (IEEE 802.11) has gradually become a popular internet access method. It is more convenient comparing to the wired Ethernet or other access network. However due to the error-prone characteristics of the physical layer (wireless media) – the WLAN has introduced some key features (CSMA/CA) in the MAC protocol layer, In this project we will use OPNET to evaluate the performance of some of these key mechanism such as RTS/CTS – a mechanism to improve performance in wireless environment with Hidden Nodes, Fragmentation - a mechanism to improve the throughput by avoiding retransmission of large block of data in fading and interference wireless environment, Back-off algorithm – a mechanism to avoid collision.

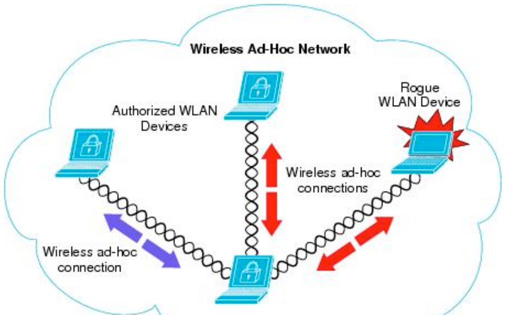
2. Introduction

The wireless LAN (IEEE 802.11) and the wired Ethernet are based on different physical media. Unlike the fibre or coax channel of the Ethernet, the air of the wireless channel of WLAN is a fading channel with interference. So the channel of wireless LAN (IEEE 802.11) is a bit error-prone channel. Also in a wireless network, it is likely that the node at the far end of the access point's range can see the access point, but it cannot see a node on the opposite end of the access point's range, these characteristic of channel make the MAC protocol of Ethernet does not work well. Some new features have been introduced to the MAC layer of IEEE 802.11. These features will effectively solve the problems of the wireless channel. The key features are RTS/CTS, Fragmentation and Back-off Algorithm. The goal of this project is to evaluate the performance of these key functions of IEEE 802.11 MAC layer protocol in ideal (no bit error) wireless environment and bit error-prone or Hidden Node wireless environment. Hidden Node and interference wireless environment will be simulated in this project.

3. Wireless Ethernet (IEEE 802.11) and the MAC protocol

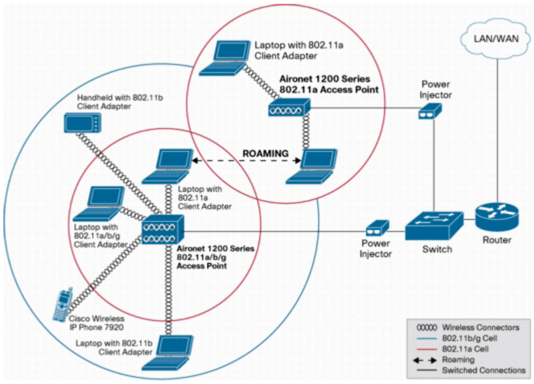
The Wireless LAN (IEEE 802.11) protocol is an alternative to existing local area network access technologies (Ethernet, FDDI, etc.). It allows multiple wireless users (fixed or mobile) within a limited range (300m-1000m) to communicate each other or with the internet using a radio connection.

The nodes in the domain of WLAN can work as an Ad Hoc network without an Access Point. However in most cases an 802.11 WLAN is based on a cellular architecture where the system is divided into cells, where each cell (called Basic Service Set or BSS, in the 802.11 nomenclature) is controlled by a Base Station (called Access Point, or in short AP), the other nodes communicate through the Access Point, which connect through backbone to the internet. Below is an example for Ad Hoc WLAN and Infrastructure WLAN.



The workstations can have peer-to-peer connections with other stations in the BSS, but communications is limited to within the BSS

Ad-hoc Network


Laptop with 802.11a Client Adapter
Aironet 1200 Series 802.11a Access Point
LAN/WAN
Power Injector
Handheld with 802.11b Client Adapter
ROAMING
Laptop with 802.11a Client Adapter
Laptop with 802.11a/b/g Client Adapter
Aironet 1200 Series 802.11a/b/g Access Point
Power Injector
Switch
Router
Cisco Wireless IP Phone 7920
Laptop with 802.11b Client Adapter
Wireless Connectors
802.11b/g Cell
802.11a Cell
Roaming
Switched Connections

The workstations communicate via the AP

Example of CISCO Infrastructure Network

As any 802.x protocol, the 802.11 WLAN protocol covers two layers (the MAC and Physical Layer) and it compliant to the OSI model. The 802.11 Standard currently defines a single MAC which supports three PHYs (FSS, DS, and IR). Please see the OSI model of IEEE 802.11 below.

802.2			Data Link Layer
802.11 MAC			
FH	DS	IR	PHY Layer

Protocol stack of IEEE 802.11

The MAC layer of IEEE 802.11 does not only perform the standard functionality usually performed by MAC Layers of Ethernet, but also deploy some unique functions that are typically related to upper layer protocols, such as RTS/CTS, Fragmentation, Packet Retransmission and Acknowledge to adapt to the characteristic of the wireless channel.

The most important differences between the wireless LAN and the MAC protocol of most wired networking applications is that the IEEE 802.11 cannot detect collisions in the wireless channel. With the receiving and sending through the same physical antenna or side by side antennas, a station is unable to see any signal but its own. As a result, the complete packet will be sent before the receiver aware a collision or data loss has happened. So to minimum the collisions and reducing the retransmission packet size are very important in the wireless environment.

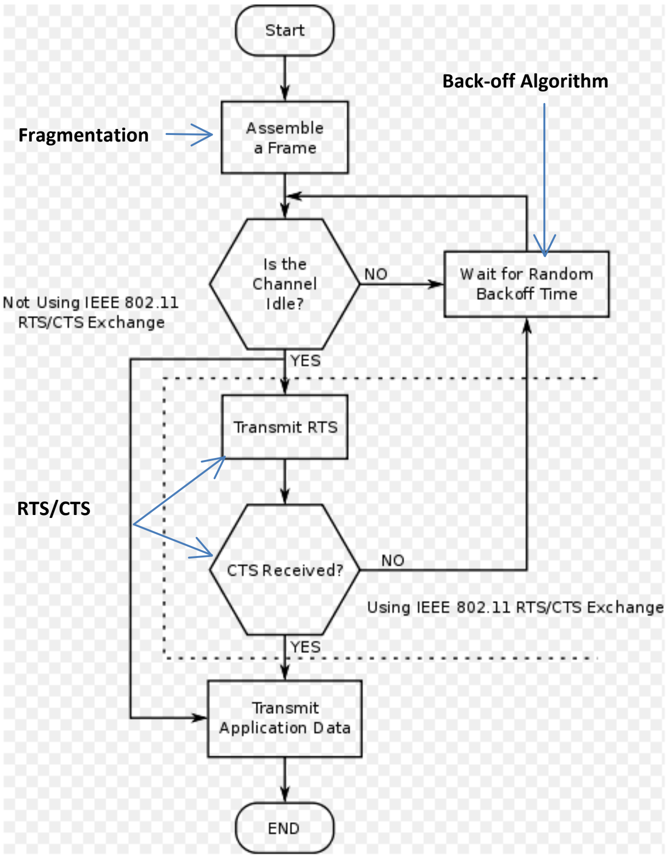
The main IEEE 802.11 MAC protocol is called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Unlike CSMA/CD (Carrier Sense Multiple Access/Collision Detect) which deals with transmissions after a collision has occurred, CSMA/CA acts to prevent collisions before the collisions happen.

In CSMA/CA, as soon as a node receives a packet that is to be sent, it checks if the channel is idle (no other node is transmitting at the time). If the channel is clear, then the packet is sent. If the channel is not clear, the node waits for a randomly back-off time, and then checks again to see if the channel is clear. The back-off time is counted down by a back-off counter. If the channel is clear when the back-off counter reaches zero, the node transmits the packet. If the channel is not clear when the back-off counter reaches zero, the back-off factor is reset again, and the process is repeated. One of the popular back-off algorithms is the Exponential Algorithm. We will evaluate and compare this Exponential and another back-off algorithm in this project.

As an optional feature, the 802.11 standard includes the RTS/CTS (Request to Send/Clear to Send) function to control station access to the medium. This feature is intended to solve the problem of Hidden Node. When an 802.11 device intends to transmit data, it will first sense whether another station is already transmitting (Carrier Sense). If no other transmissions are sensed, the 802.11 device will send a small request-to-send (RTS) packet to its intended recipient. Since the RTS is very small, it will not waste too much time even the collision happen in the RTS sending time. If the recipient senses that the medium is clear, it sends a clear-to-send (CTS) packet in reply and reserves the channel for the sender. Once the station wishing to transmit receives the CTS packet, it sends the actual data packet to its intended recipient. If the transmitting station does not receive a CTS packet in reply, it begins the RTS procedure over again. If an IEEE 802.11 user does sense another transmission when it wants to send, the device will apply a back-off time. After the random back off time is expired, it will sense the medium again to see if it can start transmitting.

The other optional function “fragmentation” enables an 802.11 station to divide data packets into smaller frames. This is done to avoid retransmitting large frames in the presence of RF interference or in a severe fading wireless environment. The bits errors resulting from RF interference or fading are likely to affect a single frame, and it requires less overhead to retransmit a smaller frame rather than a larger one. Users can set a maximum frame length threshold. If the frame size from upper layer is larger than the threshold, the radio MAC layer will break the packet into multiple frames, with each frame no larger than the threshold value.

A flow chart of the key features of the IEEE 802.11 MAC layer shows how these features work.


Start
Back-off Algorithm
Fragmentation
Assemble a Frame
Is the Channel Idle?
NO
Wait for Random Backoff Time
Not Using IEEE 802.11 RTS/CTS Exchange
YES
Transmit RTS
RTS/CTS
CTS Received?
NO
Using IEEE 802.11 RTS/CTS Exchange
YES
Transmit Application Data
END

Flow chart of the IEEE 802.11 MAC layer

4. OPNET and wireless modeler introduction

OPNET Modeler is the industry's leading network simulation commercial software. Modeler supports all major network types and technologies. The application areas include:

1. Network planning (both LAN and/or WAN) and analysis of performance and problems prior to actual implementation

2. Wireless and Satellite communication schemes and protocols

3. Microwave and Fiber-optic based Network Management

4. Protocol Development and management

5. Routing algorithm evaluation for routers, switches, and other connecting devices.

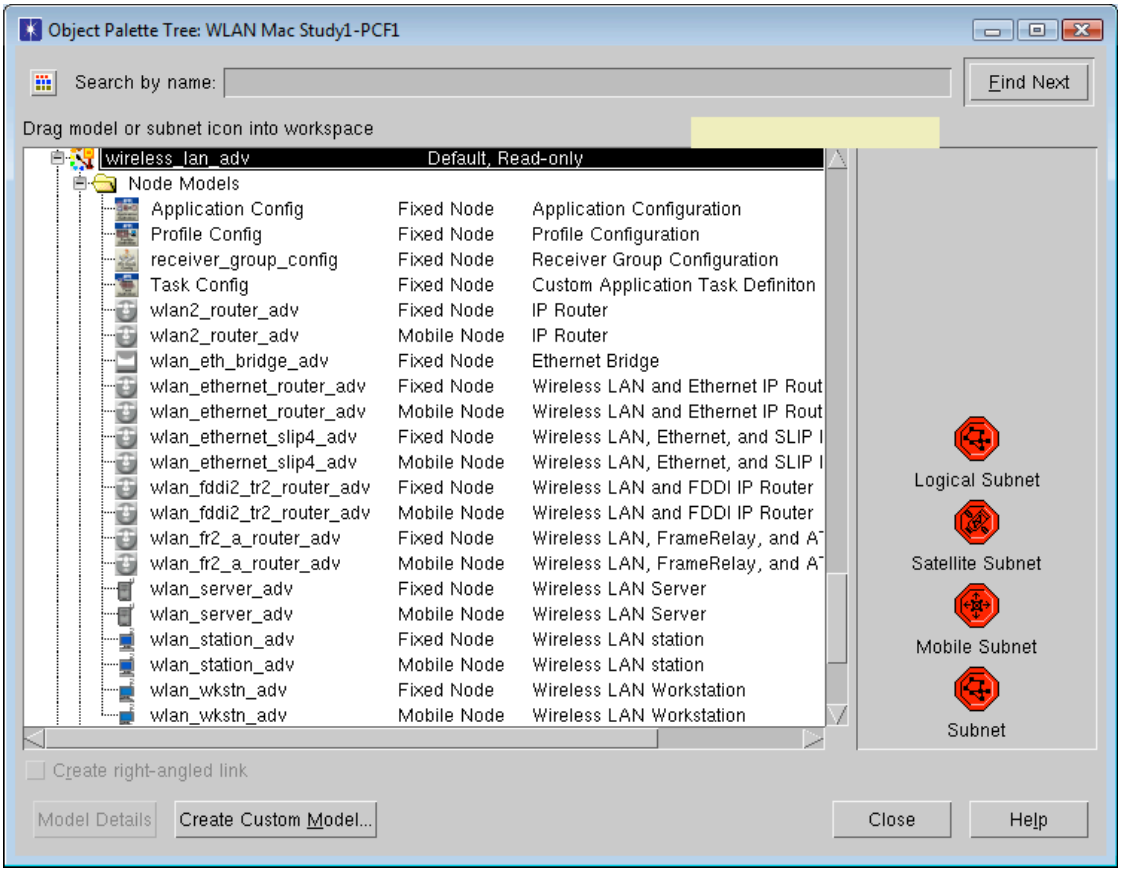
The OPNET Modeler® Wireless Suite provides high fidelity modeling, simulation, and analysis of a broad range of wireless networks. Modeler Wireless Suite supports any network with mobile devices, including cellular (GSM, CDMA, UMTS, IEEE802.16 WiMAX, LTE, etc.), mobile ad hoc, wireless LAN (IEEE 802.11), personal area networks (Bluetooth, ZigBee, etc.) and satellite.

The OPNET WLAN model provide high-fidelity modeling, simulation, and analysis of wireless LAN networks, including the RF environment, interference, transmitter/receiver characteristics, and full protocol stack, including MAC, routing, higher layer protocols and applications. Furthermore, the ability to incorporate node mobility and interconnection with wire-line transport networks provide a rich and realistic modeling environment.

The SFU Communication Network class students have been provided remotely access the SFU OPNET (Version 14.0.A) server. So our project study is based on the OPNET Version 14.0.A.

OPNET WLAN node models chosen:

This project is evaluating some of the key features of the IEEE 802.11 MAC layer. The high level protocol and application (IP, TCP, UDP, Roaming, interoperation, security, authentication, etc) is out of our study scope. To avoid interference from the high layer protocols and make the study simple, we should only implement the MAC and physical layer in our network. The router and server model, which incorporate high layer protocols will not be considered in our project. From the Wireless_lan_adv Node Models category, we will only compare the wlan_station_adv model and the wlan_wkstn_adv model to decide which model we will use in our created network. Please see the Object Palette Tree below for all the node models of wireless_lan_adv category.



Object Palette Tree: WLAN Mac Study1-PCF1

wireless_lan_adv		Default, Read-only
Node Models		
Application Config	Fixed Node	Application Configuration
Profile Config	Fixed Node	Profile Configuration
receiver_group_config	Fixed Node	Receiver Group Configuration
Task Config	Fixed Node	Custom Application Task Definiton
wlan2_router_adv	Fixed Node	IP Router
wlan2_router_adv	Mobile Node	IP Router
wlan_eth_bridge_adv	Fixed Node	Ethernet Bridge
wlan_ethernet_router_adv	Fixed Node	Wireless LAN and Ethernet IP Rout
wlan_ethernet_router_adv	Mobile Node	Wireless LAN and Ethernet IP Rout
wlan_ethernet_slip4_adv	Fixed Node	Wireless LAN, Ethernet, and SLIP I
wlan_ethernet_slip4_adv	Mobile Node	Wireless LAN, Ethernet, and SLIP I
wlan_fddi2_tr2_router_adv	Fixed Node	Wireless LAN and FDDI IP Router
wlan_fddi2_tr2_router_adv	Mobile Node	Wireless LAN and FDDI IP Router
wlan_fr2_a_router_adv	Fixed Node	Wireless LAN, FrameRelay, and A
wlan_fr2_a_router_adv	Mobile Node	Wireless LAN, FrameRelay, and A
wlan_server_adv	Fixed Node	Wireless LAN Server
wlan_server_adv	Mobile Node	Wireless LAN Server
wlan_station_adv	Fixed Node	Wireless LAN station
wlan_station_adv	Mobile Node	Wireless LAN station
wlan_wkstn_adv	Fixed Node	Wireless LAN Workstation
wlan_wkstn_adv	Mobile Node	Wireless LAN Workstation

Object Palette Tree for the wireless_lan_adv category

The difference between the wlan_ station_adv model and the wlan_wkstn_adv modle is that the wlan_ station_adv has only the WLAN MAC without a higher-layer stack (such as TCP/IP and applications). The higher layers protocols of wlan_ station_adv model are simulated by the source and sink models. So the wlan_ station_adv model is more suitable for studies that focus only on MAC and physical layers.

Some of the application of the wlan_ station_adv model is listed as follows.

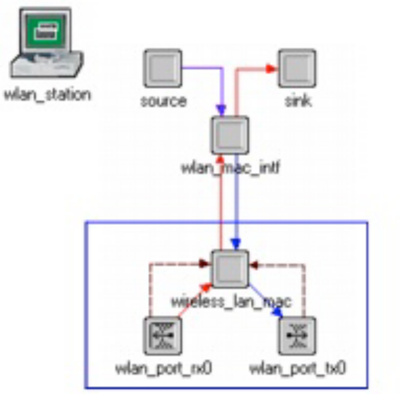
* Generate controlled traffic in the WLAN network and evaluate the performance of the MAC

* Simulate the effect of WLAN attributes independent of the higher layer

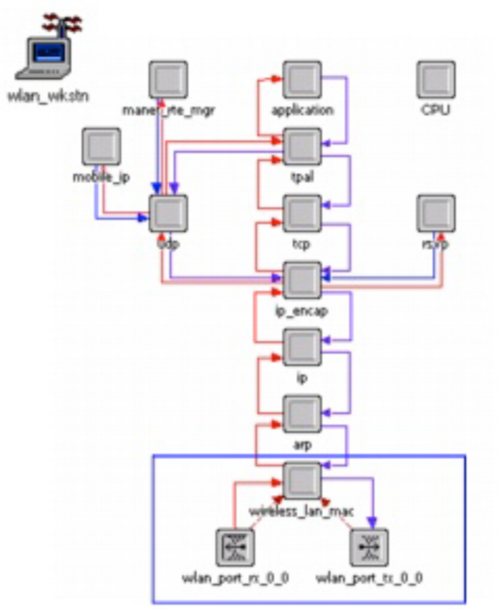
* Obtain shorter simulation time for large networks

The wlan_station_adv modeling of the MAC and the physical layer is comprised of the wireless_lan_mac process, transmitter, receiver, and the channel streams. The wlan_wkstn_adv has an ARP (address resolution protocol) interface which connects the MAC and the higher layers. It was excluded in our simulation networks.

Both wireless station and workstation node model has been shown below.



wlan_station_adv model



wlan_wknstn_adv model

As we indicated early, this project focuses only on MAC and physical layer. We decided to choose the wlan_station_adv for our study to avoid interference from the higher layer protocols and other features.

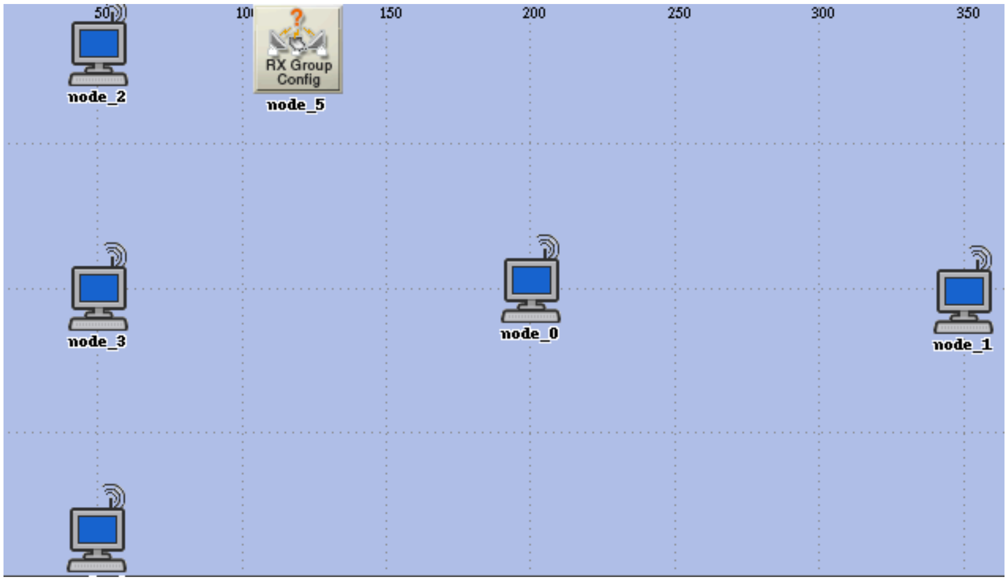
5. Implementation of the simulation

In this project we will evaluate the three key features of the IEEE 802.11 MAC layer – RTS/CTS, Fragmentation, and back-off Algorithm, which we have discussed in previous sections.

5.1. Evaluation of RTS/CTS performance

5.1.1. Network implementation

In this project we will evaluate the performance of the RTS/CTS under no Hidden Node environment and with Hidden Node environment. A small and simple wireless LAN network with 5 nodes has been created. Please see the Topology of the RTS/CTS network. The Node 0 in the center acts as AP and receives data from the Nodes 1-4. The Nodes 1 acts as Hidden Node to the Node 2-4. The "Rx Group Config" is introduced here to make the Node 1 as Hidden Node. In the Rx Group Config, we define the Distance Threshold to 220m, so the Nodes 2-4 can hear each other and become a receive group. The Nodes 1 is 300m away, it is beyond the communication distance and it will not hear the Nodes 2-4. From the attribute of the "Rx Group Config" we can also define the Channel Match Criteria or Pathloss Threshold to make the Nodes 1 as the Hidden Node. The distance parameter is the easiest way. So we use this method to define our Hidden Node. Below is the network topology.



RTS/CTS Network Topology

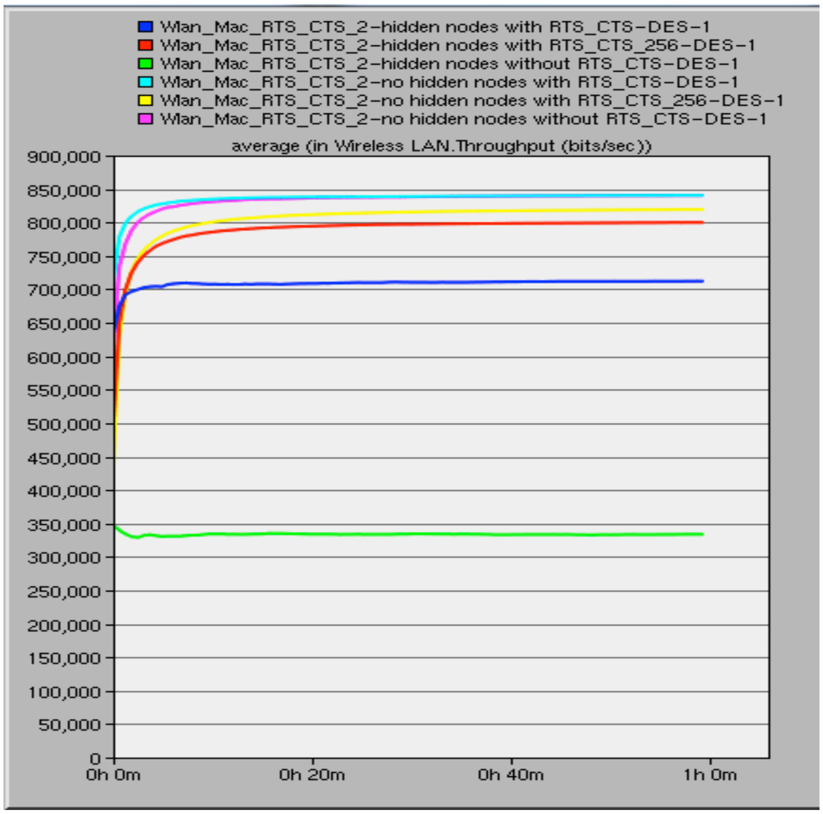
Six scenarios have been simulated in our project based on Hidden Node introduced or not, RTS/CTS on or off and different threshold size. (1). No Hidden Node without RTS/CTS. (2) No Hidden Node with RTS/CTS (threshold 1024). (3) Hidden Node without RTS/CTS. (4) Hidden Node with RTS/CTS (default threshold 1024). (5) Hidden Node with RTS/CTS (threshold 256). (6) No Hidden Node with RTS/CTS (threshold 256). Please see the below parameters table for the detail of the six scenarios.

Params	Scenarios					
	1	2	3	4	5	6
Dest Address	0	0	0	0	0	0
Start Time	exp(15)	exp(15)	exp(15)	exp(15)	exp(15)	exp(15)
ON State Time	exp(10)	exp(10)	exp(10)	exp(10)	exp(10)	exp(10)
OFF State Time	Off	Off	Off	Off	Off	Off
Pack Interarrival	exp(0.02)	exp(0.02)	exp(0.02)	exp(0.02)	exp(0.02)	exp(0.02)
Packet Size	exp(2048)	exp(2048)	exp(2048)	exp(2048)	exp(2048)	exp(2048)
Data Rate	1 Mbps	1 Mbps	1 Mbps	1 Mbps	1 Mbps	1 Mbps
Rts Threshold	None	1024	None	1024	256	256
RX Group - Distance	N/A	N/A	220 m	220 m	220 m	N/A

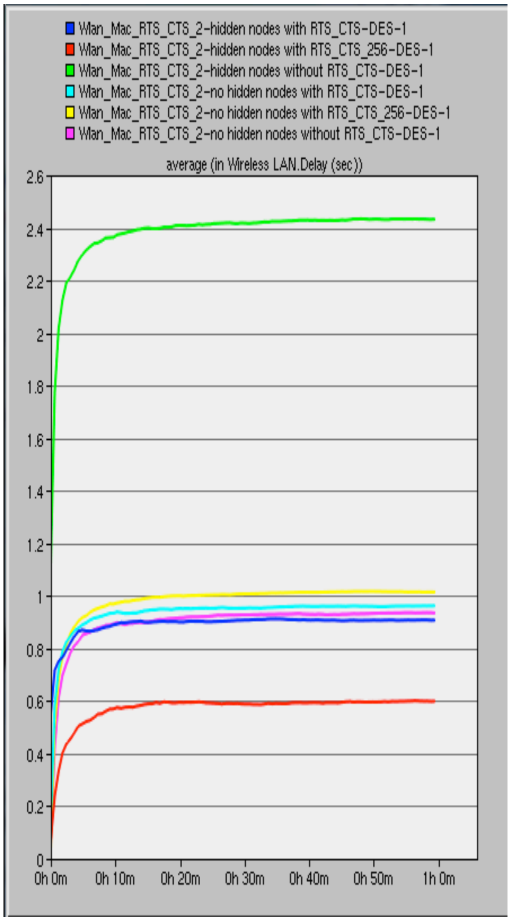
Parameters table of Scenarios

5.1.2. Simulation, results and analysis

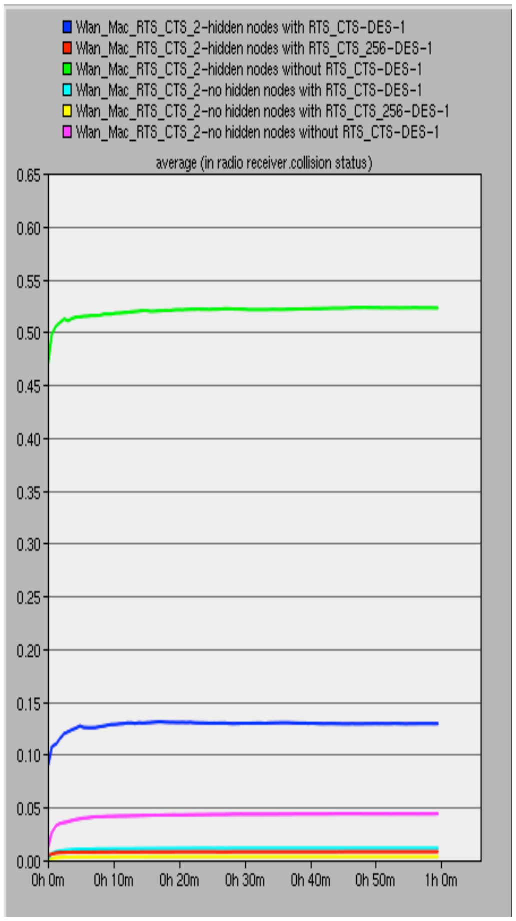
The important statistic parameters such as throughput and collision of the AP, the delay of the sending nodes have been collected to evaluate and analyze the different scenario. The simulation time is 60 minutes. Different random seeds have been chosen for each scenario.



Throughput



Delay of the sending nodes



Collision of the AP

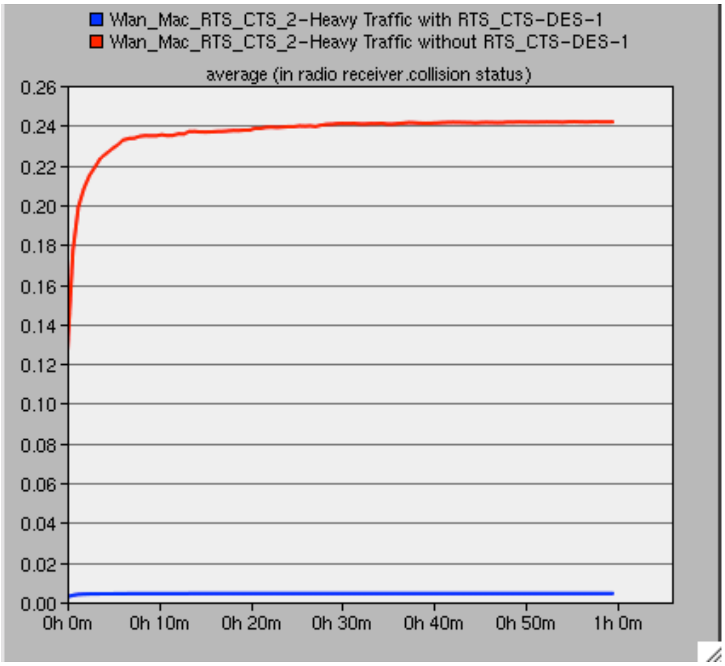
From above simulation results we find that the Hidden Nodes has dramatically decreased the network throughput (green line in throughput), increases the collision (the value is the average number of sum of collision status of rx channel, in our project all collision points this value) and average delay. By turning the option feature RTS/CTS on with the default threshold, the performance is increased (blue line in throughput). When we further reduce the threshold of the RTS/CTS at presentence of Hidden Node, The performance is further improved (red line in throughput).

As we can expect without hidden node, the network has highest performance (pink line in throughput). The option of RTS/CTS and will bring some negative affect to the network due to the overhead bring by the RTS, CTS and ACK frames. However this affection is very limited (pink line and cyan line in throughput).

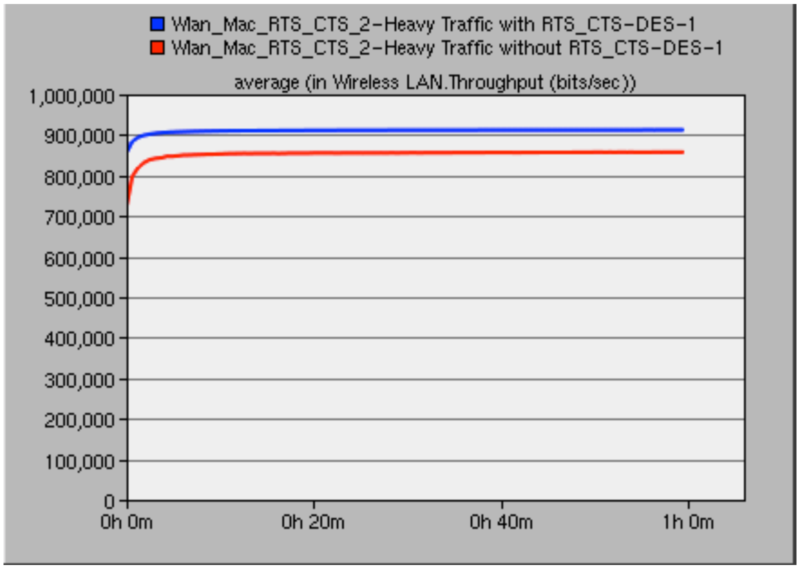
5.1.3. Further study under heavy traffic

We further study the RTS/CTS feature under heavy traffic environment. 20 nodes were created to build an Ad hoc network. Each node sends packets larger than 2000 bytes to random destination in order to create more collision.

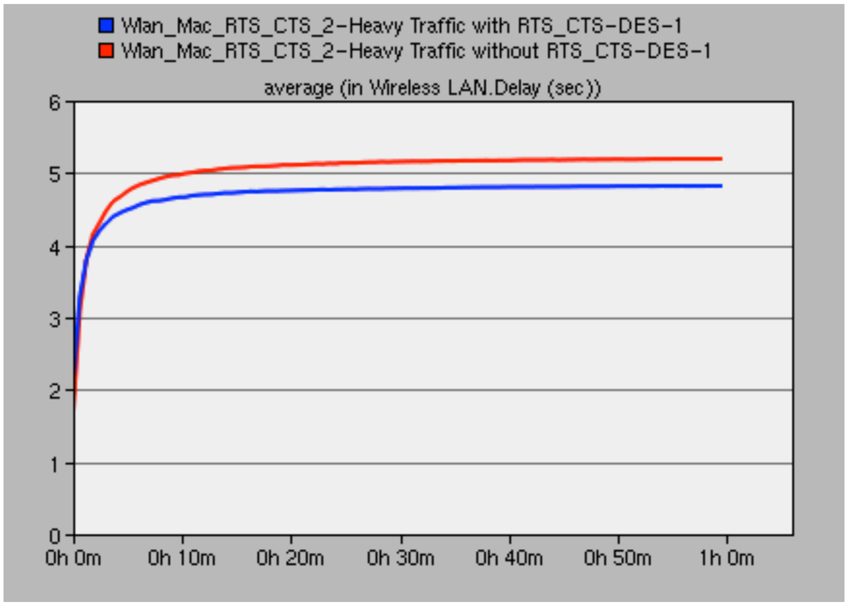
From the results, we notice that in a heavy big packet traffic network RTS/CTS can dramatically avoid the high chance of collision to improve performance like the throughput and delay.



Collision at the receiver port



Throughput

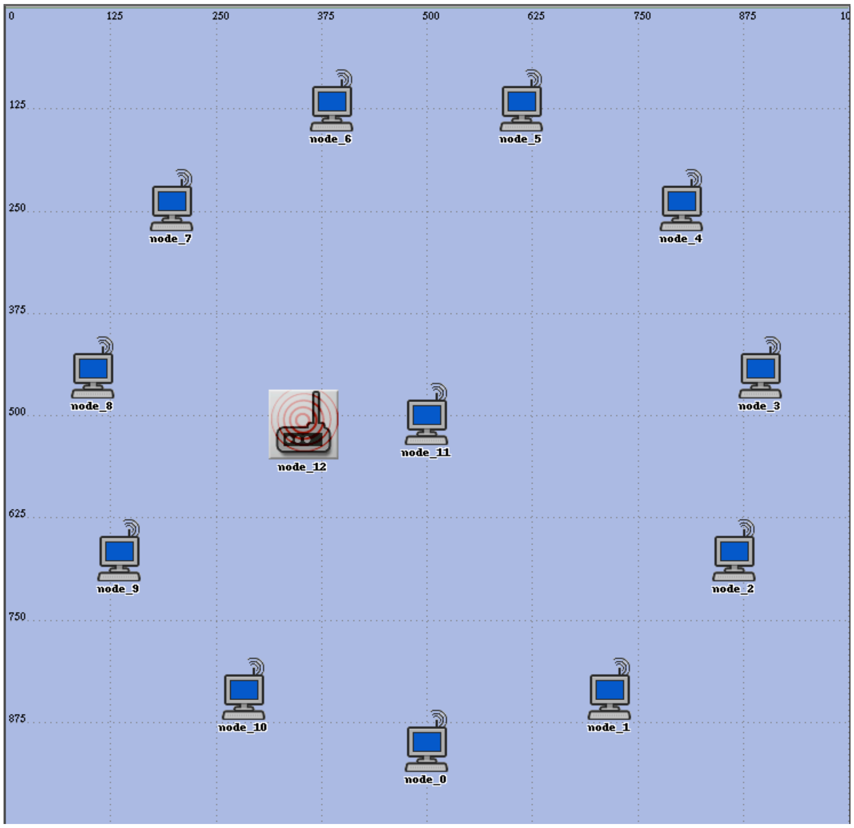


Delay

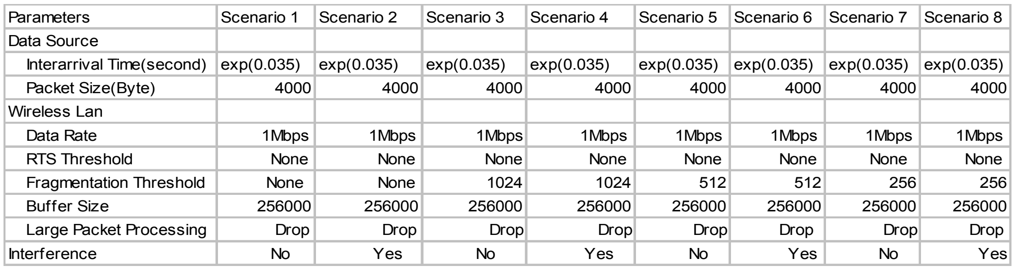
5.2. Evaluation of Fragmentation performance

5.2.1. Network implementation

In this section we will evaluate the performance of the optional feature of the Fragmentation under ideal wireless environment and the real error-prone wireless channel with fading and interference. A wireless LAN network with 11 nodes (wlan_station_adv) will be created. Node 11 in the center of the topology act as a receiving only node and the Nodes 1-10 act as sending nodes. A Jammer Node is introduced in the network to simulate the interference in the fading and interference wireless environment. To simulate the Fragmentation function, the packet size from the upper layer has been set to 4000bytes/packet, which is much larger than the default Fragmentation threshold. 8 scenarios have been studied in this project based on the interference on/off and different Fragmentation thresholds. (1) Scenario 1, no Fragmentation without interference. (2) Scenario 2, no Fragmentation with interference. (3) Scenario 3, Fragmentation default threshold 1024bytes without interference. (4) Scenario 4, Fragmentation default threshold 1024bytes with interference. (5) Scenario 5, Fragmentation threshold 512 bytes without interference. (6) Scenario 6, Fragmentation threshold 512 bytes with interference. (7) Scenario 7, Fragmentation threshold 256 bytes without interference. (8) Scenario 8, Fragmentation threshold 256 bytes with interference. Below is the network topology.



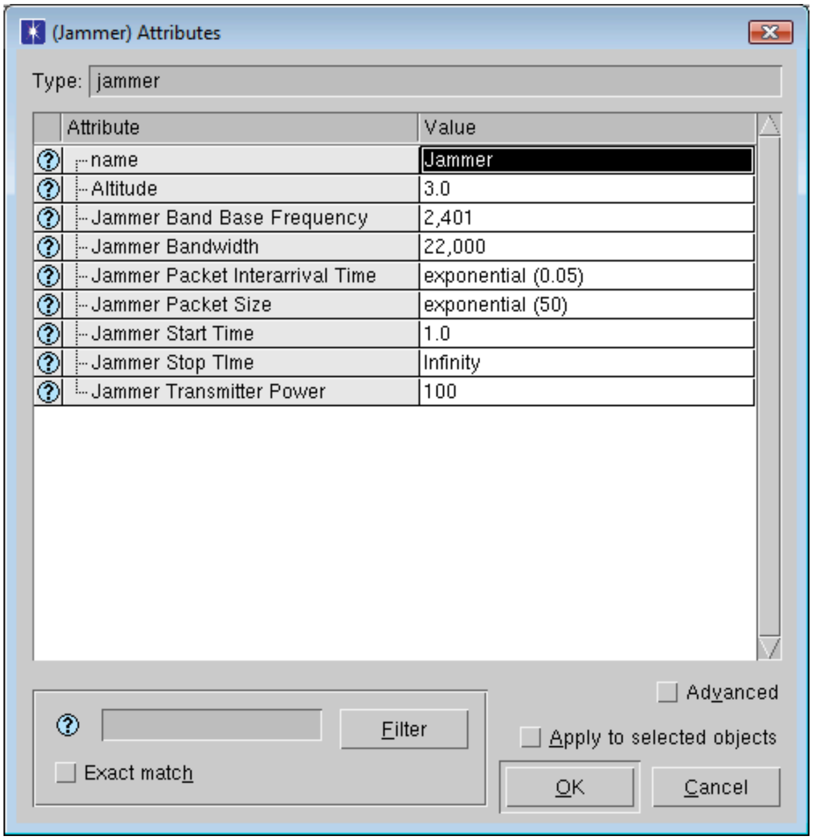
RTS/CTS Project Network Topology



Parameters	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8
Data Source								
Interarrival Time(second)	exp(0.035)	exp(0.035)	exp(0.035)	exp(0.035)	exp(0.035)	exp(0.035)	exp(0.035)	exp(0.035)
Packet Size(Byte)	4000	4000	4000	4000	4000	4000	4000	4000
Wireless Lan								
Data Rate	1Mbps	1Mbps	1Mbps	1Mbps	1Mbps	1Mbps	1Mbps	1Mbps
RTS Threshold	None	None	None	None	None	None	None	None
Fragmentation Threshold	None	None	1024	1024	512	512	256	256
Buffer Size	256000	256000	256000	256000	256000	256000	256000	256000
Large Packet Processing	Drop	Drop	Drop	Drop	Drop	Drop	Drop	Drop
Interference	No	Yes	No	Yes	No	Yes	No	Yes

RTS/CTS Project parameters table

The Jammer Node has been set to the same Band Base Frequency and Bandwidth as the Sending Nodes in the network to interfere the data sending. Every minute the Jammer sends 20 packets 50 Bytes size data out. The parameters are set in below Jammer Node parameter table.



(Jammer) Attributes

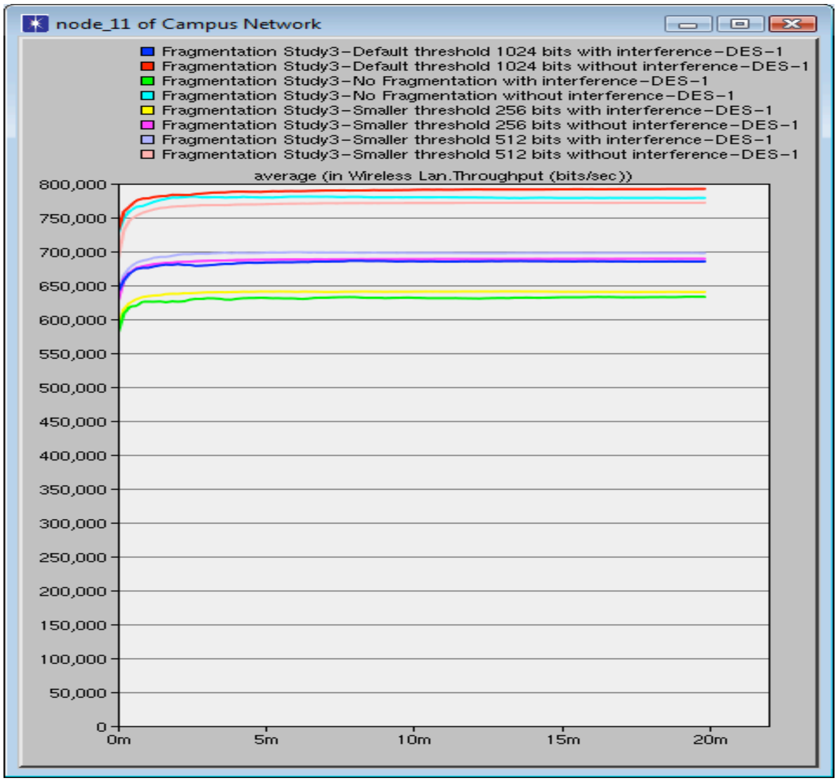
Type: jammer

Attribute	Value
name	Jammer
Altitude	3.0
Jammer Band Base Frequency	2,401
Jammer Bandwidth	22,000
Jammer Packet Interarrival Time	exponential (0.05)
Jammer Packet Size	exponential (50)
Jammer Start Time	1.0
Jammer Stop TIme	Infinity
Jammer Transmitter Power	100

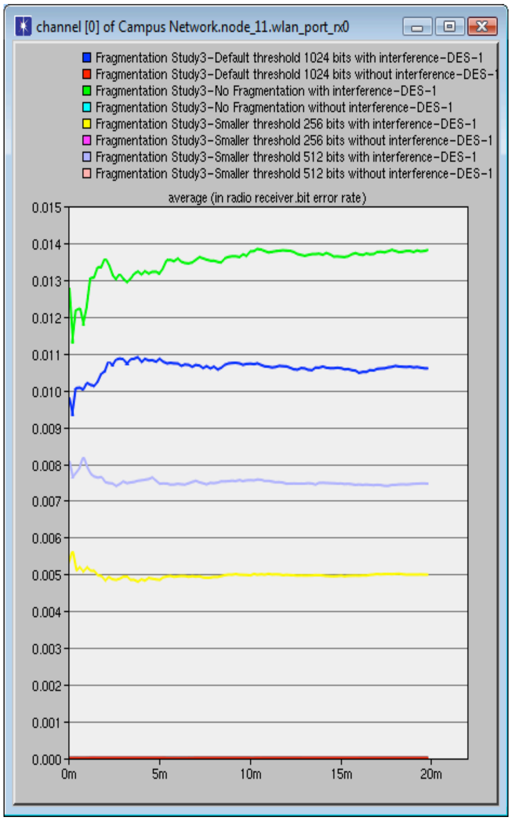
Jammer Node parameters table

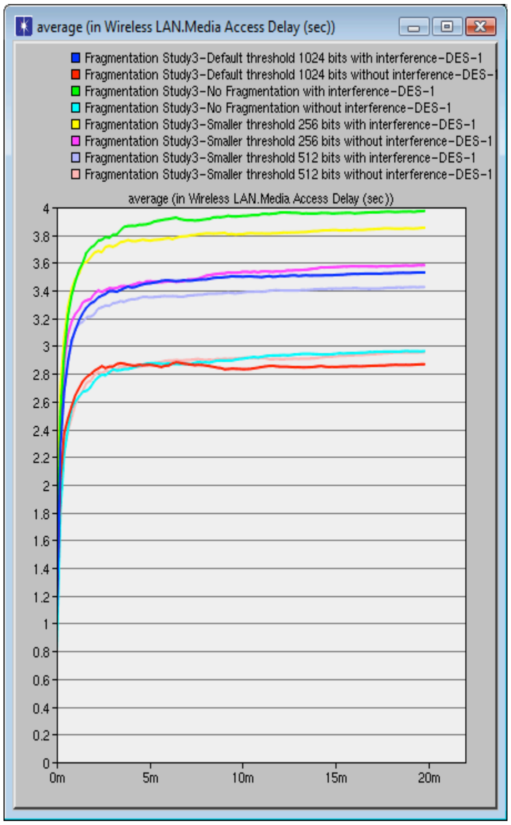
5.2.2. Simulation, results and analysis

The important statistic parameters have been collected for our evaluation and analysis such as the receiving Node throughput, the sending Node media access delay, and the packet collision at the receiver port, etc.20 minute's simulation data has been collected in our project. Random seeds were chosen to for simulation.



Throughput of Node 11





Bit Error Rate at Node 11 receiver port **The Sending Nodes Media Access Delay**

Throughput – In the ideal no interference wireless environment, the default 1024 bytes threshold scenario (red line) has the highest throughput. When the threshold reduces, the throughput reduces (pink and orange lines). However when we increase the threshold or leave the Fragmentation feature off with the large packets passing through to the MAC layer, the throughput decreases (light blue line). In the real wireless environment with fading and interference, large threshold creates large packet retransmission, therefore decreases the performance of the network (blue and green lines). The 512 bytes threshold scenario has the highest throughput (grey). The no Fragmentation scenario has the worst throughput (green). When we further decrease the threshold, the overhead of the small packets decrease the throughput. The Fragmentation feature works very well in the error-prone channel in our simulation.

Bit error in the receiver of Node 11 – As we can expect, in the ideal wireless environment there is no bit error. When the interference presents in the network, the bigger the packet size the higher the bit error rate.

Media Access Delay – In the ideal wireless environment, the threshold 1024 bytes scenario has least delay. When the interference presents in the network, the bigger the packet size the more delay introduced by the retransmission. However if the threshold is too small, the overhead will increase the delay (yellow line).

5.3. Evaluation of Back-off Algorithm performance

5.3.1. Network Implementation

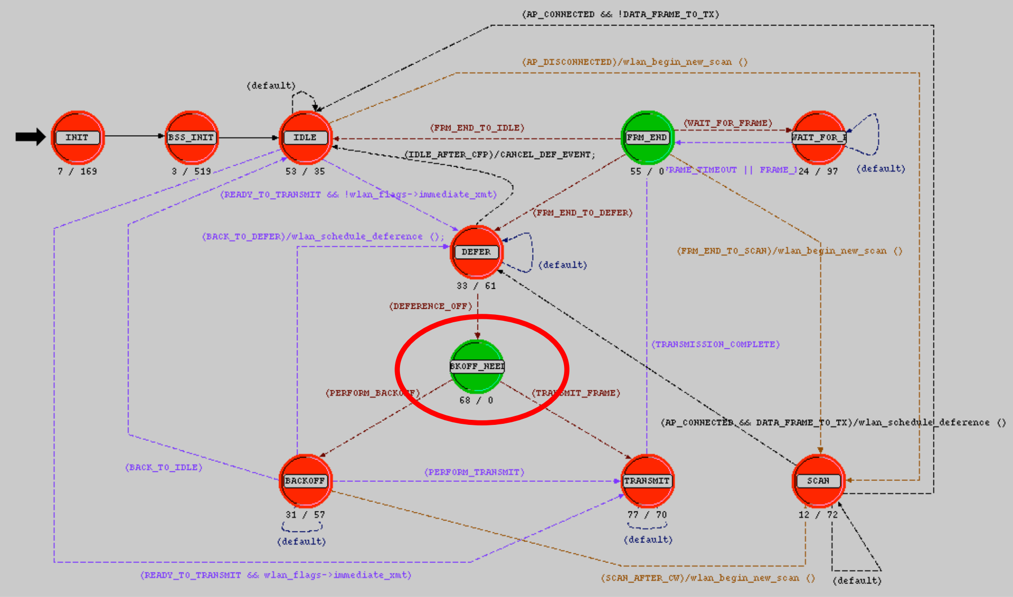
In previous two sections we have evaluated two features of 802.11 MAC. In this section we will evaluate two different back-off algorithms under different traffic to understand better of their advantages and disadvantages. One algorithm is the default algorithm of 802.11 MAC layer - BEB (Binary Exponential Back-off). Another algorithm comes from Chatzimisios' paper [7], called DIDD (Double Increment Double Decrement).

As WLAN document stated, the station will wait for a period before next retrial if it detects a collision. But how much time the station should wait is determined by back-off algorithm. BEB method is very simple. Each station has a minimum congestion window at the beginning. If the collision is detected, the congestion window is doubled until it reaches a maximum value. If the packet is successfully delivered, then the congestion window is reset to minimum value. When the station has to wait for a period before next deliver, it chooses a random value from the 1 to the value of the congestion window. The algorithm works very well, but there are some situations the algorithm haven't concern yet. First it causes unfair. The station's congestion window is reset to minimum value when it sends packet successfully, so its waiting time is shorter than others. It is more possible that the same station can access media again because it can wake up soon from shorter waiting status. Second resetting congestion window ignore the recent change of network situation; the successful deliver doesn't mean the congestion is gone. So the station will encounter another collision soon.

DIDD algorithm uses a very simple method to resolve the problems. Instead resetting congestion window it halves congestion window when successfully deliver. On one hand the successfully deliver maybe mean the worse situation of network is changing better, on other hand we don't know how much the network recover. So the new method shortens the congestion window but not reset it to minimum value.

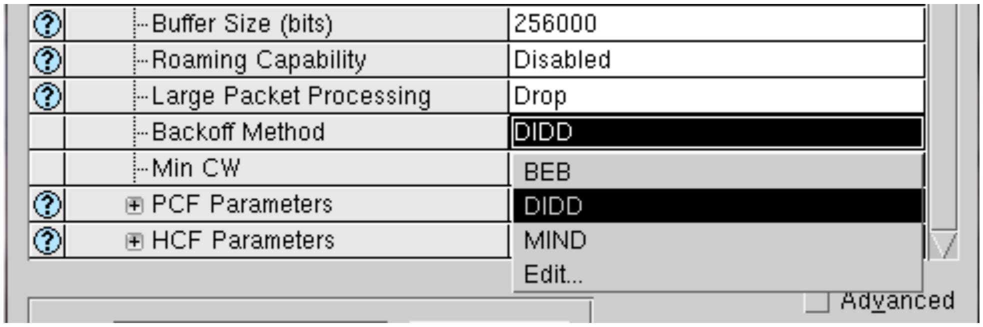
In order to test functionality of the two algorithms, we use 20, 40 and 60 nodes to build ad hoc network. All nodes are set to same parameter except the Back-off method. So the different nodes number can create different traffic.

For adding the DIDD algorithm, we have changed some code in the "back-off need" module.


(AP_CONNECTED && !DATA_FRAME_TO_TX)
(AP_DISCONNECTED)/wlan_begin_new_scan ()
(default)
INIT
7 / 169
BSS_INIT
3 / 519
IDLE
53 / 35
(FRM_END_TO_IDLE)
(IDLE_AFTER_CFP)/CANCEL_DEF_EVENT;
FRM_END
55 / 0
(WAIT_FOR_FRAME)
WAIT_FOR_F
24 / 97
(default)
FRAME_TIMEOUT || FRAME_
(READY_TO_TRANSMIT && !wlan_flags->immediate_xmt)
(FRM_END_TO_DEFER)
(BACK_TO_DEFER)/wlan_schedule_deference ();
DEFER
(default)
33 / 61
(FRM_END_TO_SCAN)/wlan_begin_new_scan ()
(DEFERENCE_OFF)
(TRANSMISSION_COMPLETE)
BKOFF_NEED
68 / 0
(PERFORM_BACKOFF)
(TRANSMIT_FRAME)
(AP_CONNECTED && DATA_FRAME_TO_TX)/wlan_schedule_deference ()
(BACK_TO_IDLE)
BACKOFF
31 / 57
(default)
(PERFORM_TRANSMIT)
TRANSMIT
77 / 70
(default)
SCAN
12 / 72
(default)
(READY_TO_TRANSMIT && wlan_flags->immediate_xmt)
(SCAN_AFTER_CW)/wlan_begin_new_scan ()

The changed process module in the state machine (the circled module)

One new parameter “Back-off Method” is added in the node attribute. It controls which back-off method is used in current scenario.

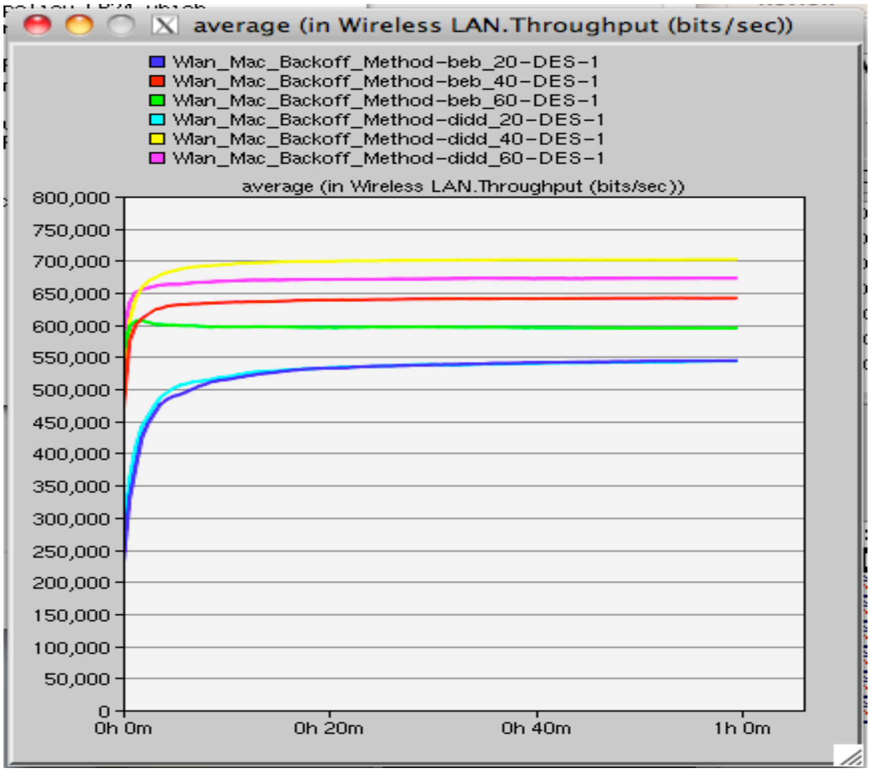


Buffer Size (bits)	256000
Roaming Capability	Disabled
Large Packet Processing	Drop
Backoff Method	DIDD
Min CW	
⊞ PCF Parameters	
⊞ HCF Parameters	

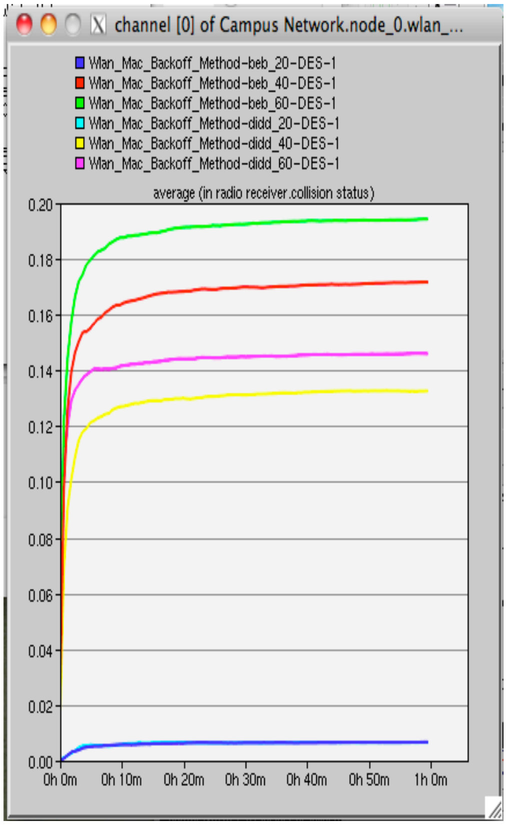
The new back-off algorithm in the Node attributes

5.3.2. Simulation result and analysis

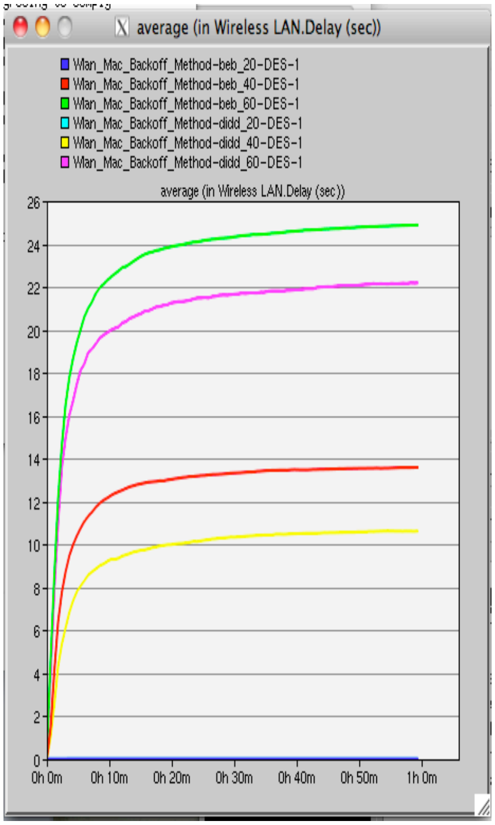
The important statistic parameters have been collected for our evaluation and analysis such as the average throughput, the average delay, and the packet collision at the receiver port, etc. 60 minute’s simulation data has been collected in our project. Random seeds were chosen to for simulation.



Throughput



Collision



Delay

When nodes number is 20, the traffic doesn't exceed the link bandwidth. So the throughput, delay and collision are close in two back-off algorithms. But when nodes number is greater than 40, the difference between two algorithms can be found. The DIDD can get better performance. The interesting thing is the performance doesn't grow up with number of nodes; it is dropped on the contrary (red line and green line in throughput). Because more workstation causes more collision, so it affects the performance. But the new algorithm can compensate the drop. From the result we find the DIDD can reduce the performance drop (yellow line and pink line in throughput).

6. Discussion and conclusion

From the simulation we mentioned above, the key features such as RTS/CTS, Fragmentation, and back-off Algorithm have effectively improved the performance of the IEEE 802.11 network in a wireless network. However it also shows that these features do not work well in some environment. Reasonable implementing these features and optimizing the parameters is a key work to run the IEEE 802.11 network in high performance.

The RTS/CTS handshaking provides positive control over the use of the shared air medium. The main function of this feature is to minimize collisions among hidden stations. The increase in performance using RTS/CTS is the net result of introducing overhead (i.e., RTS/CTS frames) and reducing overhead (i.e., fewer retransmissions). If there is no or slight hidden nodes problem, then the use of RTS/CTS will only increase the overhead, which reduces throughput. One of the best ways to determine if the RTS/CTS should be set on is to monitor the wireless LAN for collisions. If there is a large number of collisions and the users are relatively far apart and likely out of range, and then try enabling RTS/CTS.

Our simulation results also showed that the Fragmentation feature is a way to improve performance in an interference wireless environment. Similar to RTS/CTS, the large quantities of collision is a sign of high interference. In very few collisions (less than 5 percent) network, the Fragmentation will add additional headers to each fragment, which would dramatically increase the overhead on the network, reducing throughput. The threshold is a very important parameter. Too small value would create too much overhead cost.

In most cases, it is difficult to judge the collisions are caused by Hidden Node, the network interference or the channel fading. The best practice is to jointly consider the use of RTS/CTS and fragmentation. For our future work we will evaluate the RTS/CTS and Fragmentation joint performance under the Hidden Node and interference wireless environment.

We have also brought a new back-off algorithm in our project. Compared this DIDD algorithm to the basic access BEB Back-off algorithm, we found that the DIDD is more efficient in a busy high traffic network and it has no performance drop in light traffic against BEB. BEB is a memoryless algorithm; the feature makes it get worse performance when wireless stations are busy in sending or receiving packets like video conference. When more and more wireless devices appear in the small area like campus, airport, the network congestion is very easy to occur. Finding a more efficiency algorithm may be is an emergent requirement with WLAN growth. .

The key features of MAC layer of the 802.11 we have evaluated in this project can effectively improve the network performance in the wireless error-prone network. However we should be very careful to implement these features. Otherwise they will decrease the network performance.

7. References

1. IEEEE "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" 2005

2. Crow, B.P. , Widjaja, I. , Kim, L.G. ,Sakai, P.T. . IEEE 802.11 Wireless Local Area Networks. IEEE Communications Magazine, Vol. 35 Issue 9, Sep 1997, pp.116-126

3. Manshaei, M.H. , Cantieni, G.R. , Barakat, C. , Turletti, T.. Performance analysis of the IEEE 802.11 MAC and physical layer protocol. World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a Digital Object Identifier: 10.1109/WOWMOM.2005.76

4. Eustathia Ziouva, Theodore Antonakopoulos. CSMA/CA performance under high traffic conditions: Throughput and delay analysis. Computer Communications Vol: 25 Issue: 3 ISSN: 0140-3664 Date: 02/2002 Pages: 313 - 321

5. Matthew S. Gast. 802.11 Wireless Networks: The Definitive Guide, O'Reilly Press. Chapter 2,Chapter 3, Chapter 4, Page 12-113

6. Wu HaiToa, LIN Yu, ChENG ShiDuan, PENG Yong, Long KePing. IEEE 802.11 Distributed Coordination Function: Enhancement and Analysis. J. Comput. Sci. & Technol, Sept. 2003, Vol.18, No. 5, pp.607-614

7. P. Chatzimisios, V. Vitsas, A. C. Boucouvalas, M. Tsoulfa. Achieving performance enhancement in IEEE 802.11 WLANS by using the DIDD backoff mechanism. Int. J. Commun. Syst. 2007; 20:23-41

8. M. Heusse, A. Duda, F. Rousseau, R. Guillier. Idle sense- An optimal access method for high throughput and fairness in rate diverse wireless LANs. COMPUTER COMMUNICATION REVIEW Vol: 35 Issue: 4 ISSN: 0146-4833 Date: 10/2005 Pages: 121 – 132

9. Jing Deng, Pramod K. Varshney, Zygmunt J. Haas. A new backoff algorithm for the IEEE 802.11 distributed coordination function. Electrical Engineering and Computer Science. Paper 85, http://surface.syr.edu/eecs/85

10. OPNET documentation: "Model Library: Wireless Lan" and "Modules: Wireless"

11. OPNET Wireless module: node module “wlan_station_adv”, process module “wlan_dispatch”, “wlan_mac”.

8. Appendix

8.1. Code List

In header block of “wlan_mac_dispatch” process model we define the enum “WlanT_Mac_Backoff_Method” to include the back-off algorithms we added. The code is list below.

```
            #include "wlan_support.h"


/** Define the WLAN global variables, which are declared in wlan_support.h.     **/


/* Global linked list of AP position info.                                                    */

WlanT_AP_Position_Info*                    global_ap_pos_info_head = OPC_NIL;


/* Global variable to keep note of the nature of the subnet.                          */

/* This variable is initialized to not set.                                                    */

WlanT_Bss_Identification_Approach       bss_id_type = WlanC_Not_Set;


/* Read-only array of the minimum frequencies of the 12 operational 802.11a      */
/* WLAN channels.                                                                                       */

double  WLANC_11a_CHNL_MIN_FREQ_ARRAY [WLANC_11a_OPER_CHNL_COUNT] =
{5170.0, 5190.0, 5210.0, 5230.0, 5250.0, 5270.0, 5290.0, 5310.0, 5735.0, 5755.0, 5775.0, 5795.0};


/* Read-only arrays for mandatory 802.11a and 802.11g data rates.                    */

double  WLANC_11a_MANDATORY_DRATE_ARRAY [3] = {24000000.0, 12000000.0,
6000000.0}; double  WLANC_11g_MANDATORY_DRATE_ARRAY [7] = {24000000.0,
12000000.0, 11000000.0, 6000000.0, 5500000.0, 2000000.0, 1000000.0};
```

```
/* Reset one of the packet field index global variables so its value can be     */
/* checked to determine whether all of those variables are initialized or        */
/* not.                                                                          */

int         WLANC_DATA_TYPE_FD = OPC_FIELD_INDEX_INVALID;

int         WLANC_DATA_HEADER_FD, WLANC_DATA_QOS_FD,
WLANC_DATA_BODY_FD, WLANC_DATA_ACCEPT_FD, WLANC_DATA_PKID_FD;

int         WLANC_CNTL_TYPE_FD, WLANC_CNTL_HEADER_FD, WLANC_CNTL_BA_FD,
WLANC_CNTL_ACCEPT_FD;

int         WLANC_BEACON_BODY_FD;

int    WLANC_ACT_MGMT_CAT_ACT_FD, WLANC_ACT_MGMT_TID_FD,
WLANC_ACT_MGMT_PARAMS_FD;


/* Aded By Yitao Wu at Apr 01, 2011 */

/* Define Backoff method                    */

typedef enum WlanT_Mac_Backoff_Method

     {

     WlanC_Backoff_BEB,              /* Binary Exponential back off, default */

     WlanC_Backoff_DIDD,              /* Double increament double decrement */

     WlanC_Backoff_MIND,              /* Set max number if colision occures */

     } WlanT_Mac_Backoff_Method;

/* End Add Mar 20,2011  */
```

Based on new node parameter “Backoff Method”, we change the “Enter Executive” of state “Spawn” of “wlan_dispatch”, the new code call different sub process by back-off method.

```
/* Find out whether the surrounding WLAN MAC module                  */

/* supports Hybrid Coordination Function (HCF),                      */
/* specified in the IEEE 802.11e standard. Access the      */

/* WLAN configuration attribute.                                        */

op_ima_obj_attr_get (op_id_self (), "Wireless LAN Parameters", &comp_attr_objid);

comp_attr_row_objid = op_topo_child (comp_attr_objid, OPC_OBJTYPE_GENERIC, 0);


/* Read backoff method */

op_ima_obj_attr_get (comp_attr_row_objid, "Backoff Method", &backoff_method);


/* Read the value of the corresponding attribute under     */

/* HCF Parameters.
*/

op_ima_obj_attr_get (comp_attr_row_objid, "HCF Parameters", &comp_attr_objid);

comp_attr_row_objid = op_topo_child (comp_attr_objid, OPC_OBJTYPE_GENERIC, 0);

op_ima_obj_attr_get (comp_attr_row_objid, "Status", &hcf_support_int);


/* Create the appropriate MAC process model.                        */

/* mac_prohandle = (hcf_support_int == OPC_BOOLINT_ENABLED) ?

                                        op_pro_create ("wlan_mac_hcf", OPC_NIL) :

                                        op_pro_create ("wlan_mac"    , OPC_NIL); */


if (hcf_support_int == OPC_BOOLINT_ENABLED)

{

mac_prohandle =op_pro_create ("wlan_mac_hcf", OPC_NIL);

}

else if (backoff_method == WlanC_Backoff_DIDD)

{
```

```
mac_prohandle =op_pro_create ("wl_wlan_mac_backoff_didd", OPC_NIL);
}
else if (backoff_method == WlanC_Backoff_MIND)
{
mac_prohandle =op_pro_create ("wl_wlan_mac_backoff_mind", OPC_NIL);
}
else
{
mac_prohandle =op_pro_create ("wl_wlan_mac_backoff_beb", OPC_NIL);
}

/* Make the child process the recipient of the                    */
/* interrupts of the module.                                             */
op_intrpt_type_register (OPC_INTRPT_STRM,   mac_prohandle);
op_intrpt_type_register (OPC_INTRPT_STAT,   mac_prohandle);
op_intrpt_type_register (OPC_INTRPT_REMOTE, mac_prohandle);

/* Spawn the MAC child process.                                                */
op_pro_invoke (mac_prohandle, OPC_NIL);
```

We also change the process model “wlan_mac”. In “Enter Executive” of state “BKOFF_NEED” there is code to recalculate congestion window. In DIDD the resetting function is replaced by halve operation.

```
/** In this state we determine whether a back-off is necessary for the    **/
/** frame we are trying to transmit. It is needed when station              **/
/** preparing to transmit frame discovers that the medium is busy or      **/

/** when the the station infers collision. Backoff is not needed when      **/

/** the station is responding to the frame. Following a successful                 **/

/** packet transmission, again a back-off procedure is performed for a  **/

/** contention window period as stated in 802.11 standard.                                  **/

/**
                                                  **/

/** If backoff needed then check whether the station completed its                  **/

/** backoff in the last attempt. If not then resume the backoff              **/

/** from the same point, otherwise generate a new random number                **/

/** for the number of backoff slots.
**/



/* Checking whether backoff is needed or not.
*/

if (wlan_flags->backoff_flag == OPC_TRUE || wlan_flags->perform_cw == OPC_TRUE)

{

if (backoff_slots == BACKOFF_SLOTS_UNSET)

          {

          /* Compute backoff interval using binary exponential process. */

          /* After a successful transmission we always use cw_min.                  */

          if (short_retry_count + long_retry_count == 0 || wlan_flags->perform_cw == OPC_TRUE)

                    {

                    /* If retry count is set to 0 then set the maximum backoff         */

                    /* slots to min window size.
*/

                    //max_backoff = cw_min;

                    /* Added By Yitao Wu, Apr 01,2011 */

                    /* double decrease cw_min */
```

```
        if ((max_backoff -1 )/2> cw_min)

            {

            max_backoff = (max_backoff - 1)/2;

            }

        else

            {

            max_backoff = cw_min;

            }

        }

    else

        {

        /* We are retransmitting. Increase the back-off window                */

        /* size.
        */

        max_backoff = max_backoff * 2 + 1;

        }


    /* The number of possible slots grows exponentially until it      */

    /* exceeds a fixed limit.
*/

    if (max_backoff > cw_max)

        {

        max_backoff = cw_max;

        }


    /* Obtain a uniformly distributed random integer between 0 and         */

    /* the minimum contention window size. Scale the number of             */

    /* slots according to the number of retransmissions.                   */
```

```
        backoff_slots = floor (op_dist_uniform (max_backoff + 1));
        }

/* Set a timer for the end of the backoff interval.                              */
intrpt_time = (current_time + backoff_slots * slot_time);

/* Scheduling self interrupt for backoff.                                                */
if (wlan_flags->perform_cw == OPC_TRUE)
        backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_CW_Elapsed);
else
        backoff_elapsed_evh = op_intrpt_schedule_self (intrpt_time, WlanC_Backoff_Elapsed);

/* Reporting number of backoff slots as a statistic.                       */
op_stat_write (backoff_slots_handle, backoff_slots);
}
```