

# **ENSC 833-3: NETWORK PROTOCOLS AND PERFORMANCE**

## **WIRELESS ETHERNET PERFORMANCE**

Spring 2001

FINAL PROJECT

Jim Chuang	(jchuang@sfu.ca)
Tim Lee	(leetim@pmc-sierra.com)
Marion Sum	(marion_sum@pmc-sierra.bc.ca)

# 1 ABSTRACT

Wireless LAN technology allows network users to access resources without restrictions on the physical location of the network interface (workstation). The IEEE 802.11b Wireless Ethernet specification provides for a high-speed, 11 Mbps system allowing workstations to roam freely a network spanning hundreds of metres without need for a physical connection to the network. In this project, a Wireless Ethernet (WLAN) configuration, using trace-driven traffic data, was simulated. The network performance was measured, analyzed, and compared with performance after modifications to the medium access (MAC) protocols. Modifications were made to the way data collisions and transmission backoff are handled. The goal was to observe the effects of MAC protocol variants on network performance.

## TABLE OF CONTENT

1	ABSTRACT .....	ii
2	INTRODUCTION .....	1
3	INTRODUCTION OF WIRELESS ETHERNET .....	2
4	IMPLEMENTATION.....	6
4.1	Trace-driven Network Traffic.....	6
4.2	Trace-driven Traffic in Complex Network:.....	8
4.3	Variations of MAC Protocol.....	12
4.3.1	Original Network.....	12
4.3.2	Linear Growth.....	13
4.3.3	Exponential Distribution.....	15
4.3.4	P-persistent.....	17
5	DISCUSSION AND CONCLUSIONS.....	20
6	REFERENCES .....	21
7	APPENDIX A – CODE LISTING .....	22

## LIST OF FIGURES

Figure 1 — Simple Network .....	2
Figure 2 — Complex Network.....	3
Figure 3 — OPNET WLAN Node Model for Workstation.....	4
Figure 4 — OPNET WLAN Node Model for Access Point.....	4
Figure 5 — OPNET WLAN Process Model for MAC.....	5
Figure 6 — Setup for Passing Trace-driven Traffic .....	6
Figure 7 — Sent/Received Traffic Plots .....	8
Figure 8 — Top Level Network Setup .....	9
Figure 9 — Network Setup for Subnet 1 .....	9
Figure 10 — Network Setup for Subnet 2 .....	9
Figure 11 — Simulation Result at wkstn22 (source in subnet 2).....	11
Figure 12 — Simulation Result at wkstn21 (destination).....	11
Figure 13 — Comparison of Starwars and Dino, for Original Network.....	13
Figure 14 — Comparison of Original and Linear .....	14
Figure 15 — Comparison of Starwar and Dino Traces, for Linear Growth .....	14
Figure 16 — Comparison of Original and Exponential (mean = $10 * \text{max\_backoff}$ ).....	15
Figure 17 — Comparison of Original and Exponential (mean = $\text{max\_backoff} / 16$ ).....	16
Figure 18 — Comparison of Starwars and Dino Traces, for Exponential Distribution (mean = $\text{max\_backoff} / 16$ ).....	16
Figure 19 — Comparison of Original and P-persistent (P = 90).....	18
Figure 20 — Comparison of Starwars and Dino Traces, for P-persistent (P = 90).....	19

## 2 INTRODUCTION

The proposed project involves implementing a multi-user, wireless Ethernet (IEEE 802.11b) network and simulating the network performance when video traffic is transmitted across the network. The goal of the project is to learn about the relatively new and popular short-range wireless networking technology, and what the practical applications are for its bandwidth and range limitation. Once the system has been implemented attempts will be made to tweak the performance and determine factors that affect Quality of Service (QoS.)

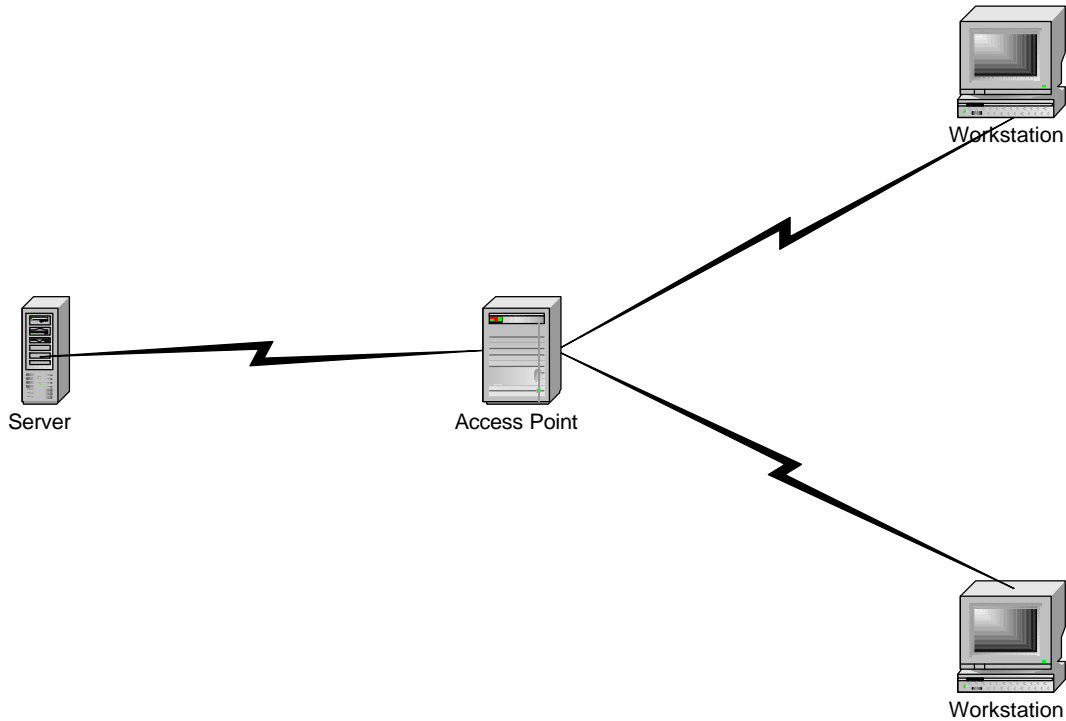
OPNET is the simulation tool of choice as there is an existing Wireless Ethernet implementation in OPNET. Our task includes determining what is already implemented and what additional features can be added.

Using the existing Wireless Ethernet implementation (WLAN), a simple and a more complex sample network will be created. Trace-driven traffic will be sent over the networks to observe the performance of the WLAN. The next step of the project involves modification of the MAC layer and comparing the resulting network performance with the original MAC implementation. The planned modifications are:

1. Linear growth of backoff slots with original uniform distribution.
2. Original binary exponential growth of backoff slots with exponential distribution.
3. Original binary exponential growth of backoff slots with p-persistent probability.

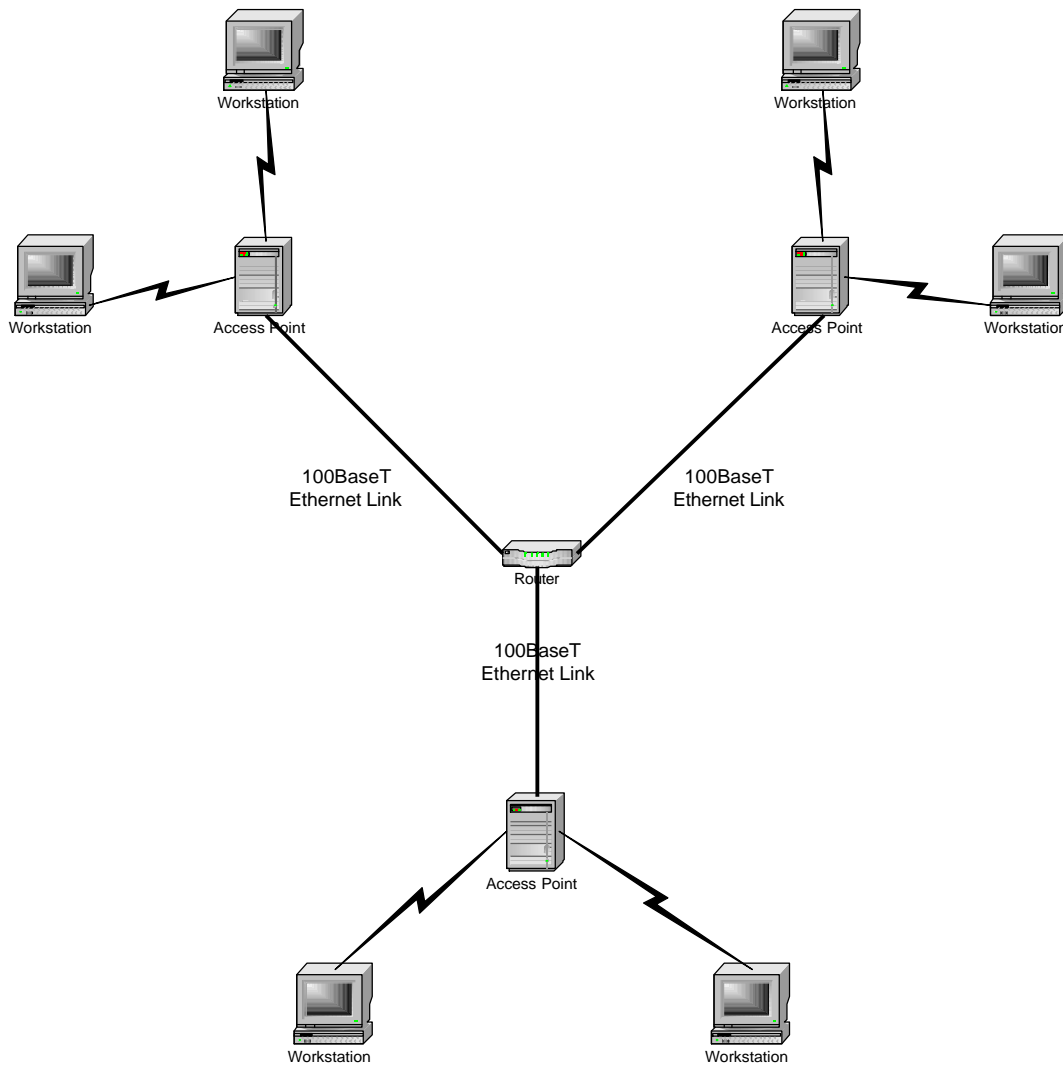
### 3 INTRODUCTION OF WIRELESS ETHERNET

Wireless Ethernet is defined by the IEEE 802.11b standard body. Currently it supports wireless links of up to 11 Mbps and 300-ft radius. It supports three different physical layer protocols: Frequency Hop Spread Spectrum, Direct Sequence Spread Spectrum, and Baseband Infrared.



**Figure 1 — Simple Network**

Figure 1 shows a basic network topology, where two network workstations and one server communicate with each other via a wireless access point. This setup emulates a possible client/server setup that could exist in an office environment where wired network links are more costly or impossible. The access point acts as a router for the local subnet, and is not necessary in simple network with few clients and small range.



**Figure 2 — Complex Network**

A more practical use of Wireless Ethernet is to combine the wireless access capabilities to extend the range of an existing hard-wired network (Figure 2). The standard does not specify the protocol used to connect the access points in the network backbone, and 100-BaseT Fast Ethernet is just one of the possibilities.

OPNET comes with an extensive model library for Wireless Ethernet applications. Figure 3 shows a node model for the wireless workstation and this model implements all seven layers of the OSI protocol stack. Figure 4 shows the node model for the access point. Figure 5 shows the process model for the Wireless LAN MAC.

The Wireless LAN MAC works as the following:

- Stations wait until medium is free (defer).
- After deferral, select random time to transmit (backoff).
- Binary exponential growth of backoff window.
- Backoff timer elapses only when medium is free.

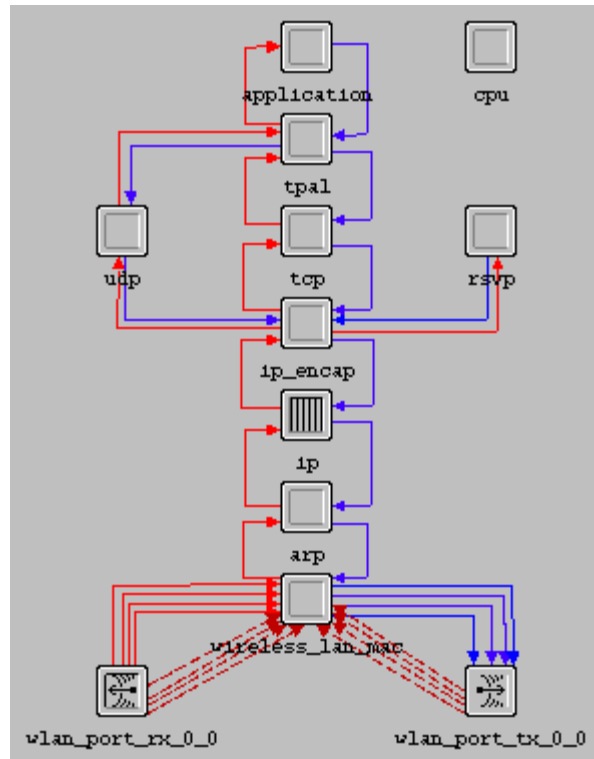


Figure 3 — OPNET WLAN Node Model for Workstation

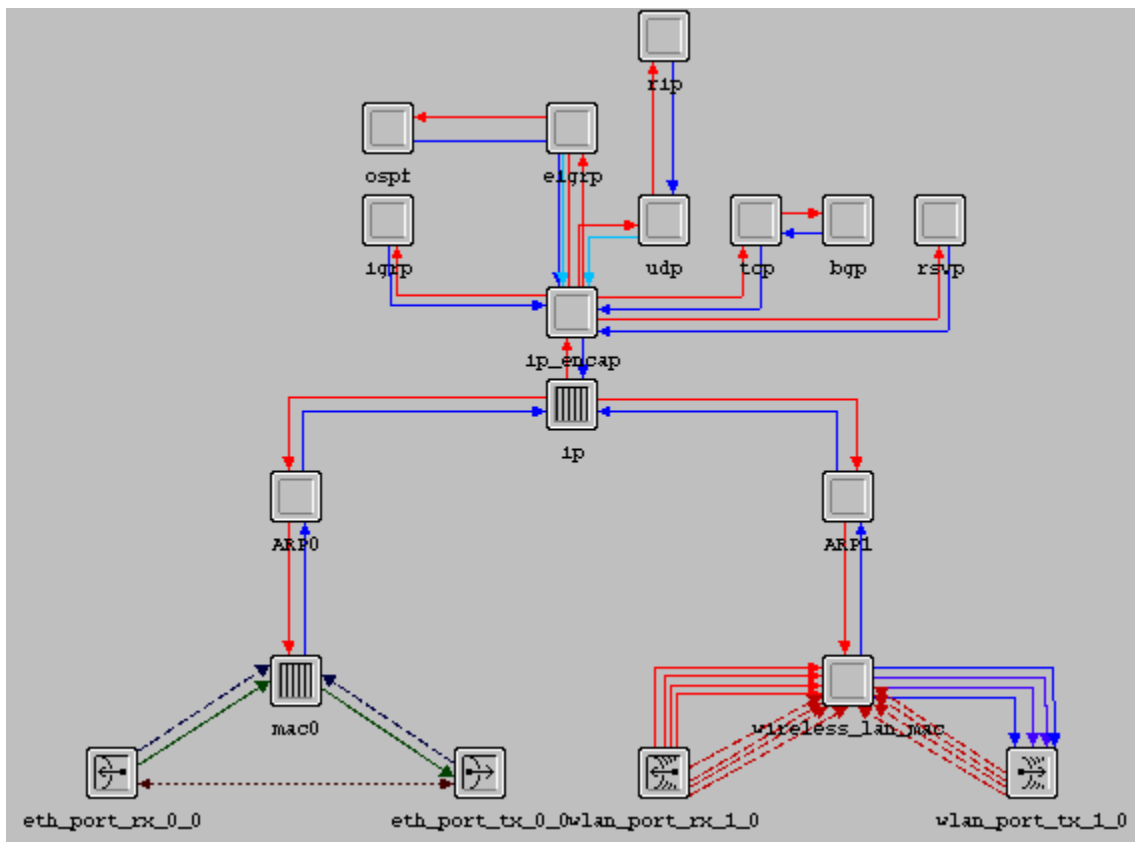


Figure 4 — OPNET WLAN Node Model for Access Point



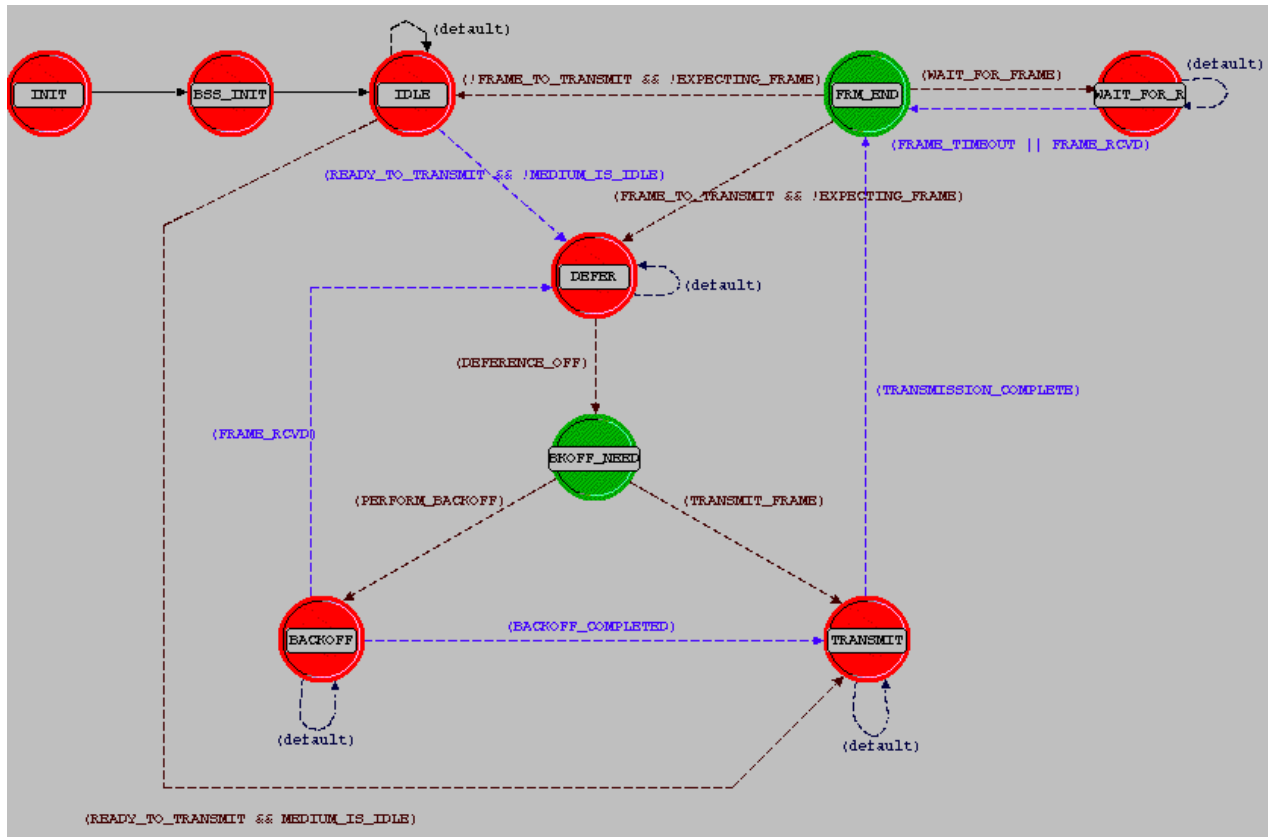


Figure 5 — OPNET WLAN Process Model for MAC

## 4 IMPLEMENTATION

The actual implementation of this project involves three major steps. The first step was to construct a very simple network using the OPNET WLAN models and transmitting trace traffic. The second step constructed a more complex network which contains two access points, one for each local subnet, and a Fast Ethernet router. The third step used the complex network from step 2 and varied the backoff behaviour of the MAC layer.

### 4.1 Trace-driven Network Traffic

The first step in using real-world traffic traces in our network simulations was to take two Wireless Ethernet workstations (with no Access Point) running at 11 Mbps and attempt to transfer a trace of the Star Wars movie between them. The simulation parameters for this and all later simulations are listed in the following table:

Item	Value
Data Rate	11 Mbps
Physical Characteristics	Direct Sequence
Buffer Size	256000 bits
Max Receive Lifetime	0.5 sec
MAC Address	assigned as needed

The simulation parameters shown are defaults from OPNET. Since the focus for this project is not in investigating the affect of these parameters on the performance of the WLAN network, these values are left untouched.

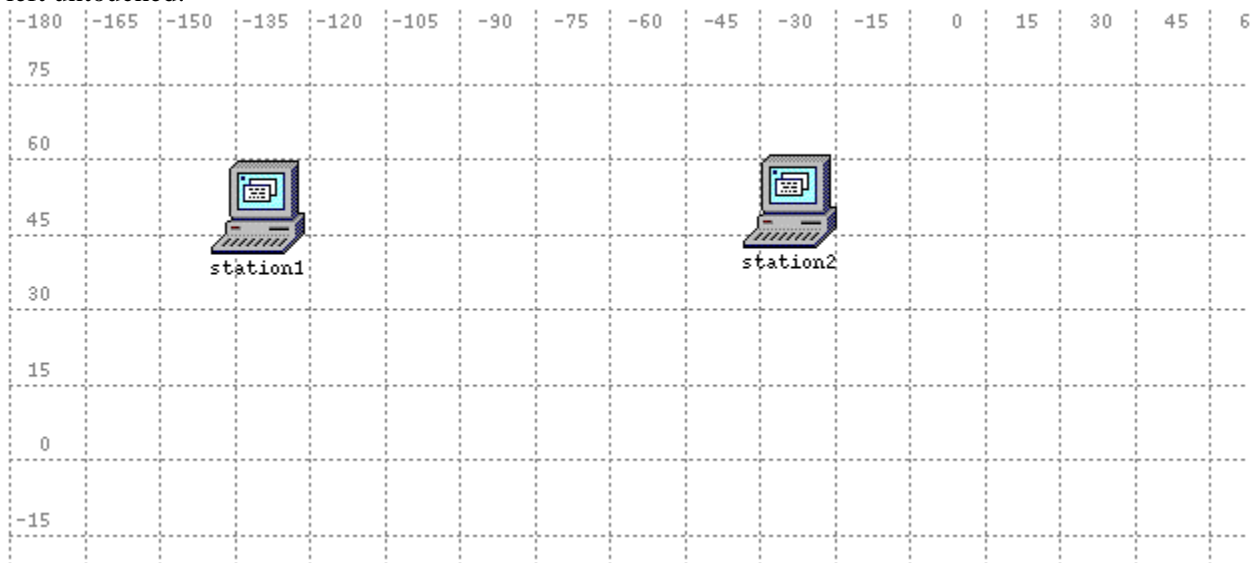


Figure 6 — Setup for Passing Trace-driven Traffic

Figure 6 shows the setup used for our trials to incorporate trace-driven traffic into our simulations. The procedure used to enable a workstation to transmit using a trace file is as follows:

- 1) open *station attributes*
- 2) open *traffic generation parameter* attribute
- 3) open *packet generation arguments* table
- 4) from the *packet size option*, choose **scripted**
- 5) for the *filename box*, enter the **trace name**

Note: The trace name must have the same name as the filename, which should reside in the model directory. Also, this file can ONLY have a *.gdf* or *.csv* extension.

The data traffic transmitted and received by each workstation was monitored during simulation. The plots in Figure 7 show that there was successful communication between the two workstations. Both station1 and station2 were able to send the Star Wars trace data to each other. In this simulation, both stations are transmitting and receiving at the same time. Therefore, even when the trace being used by both stations is the same, depends on the different collision and backoff rate, the data sent by each station may look different.

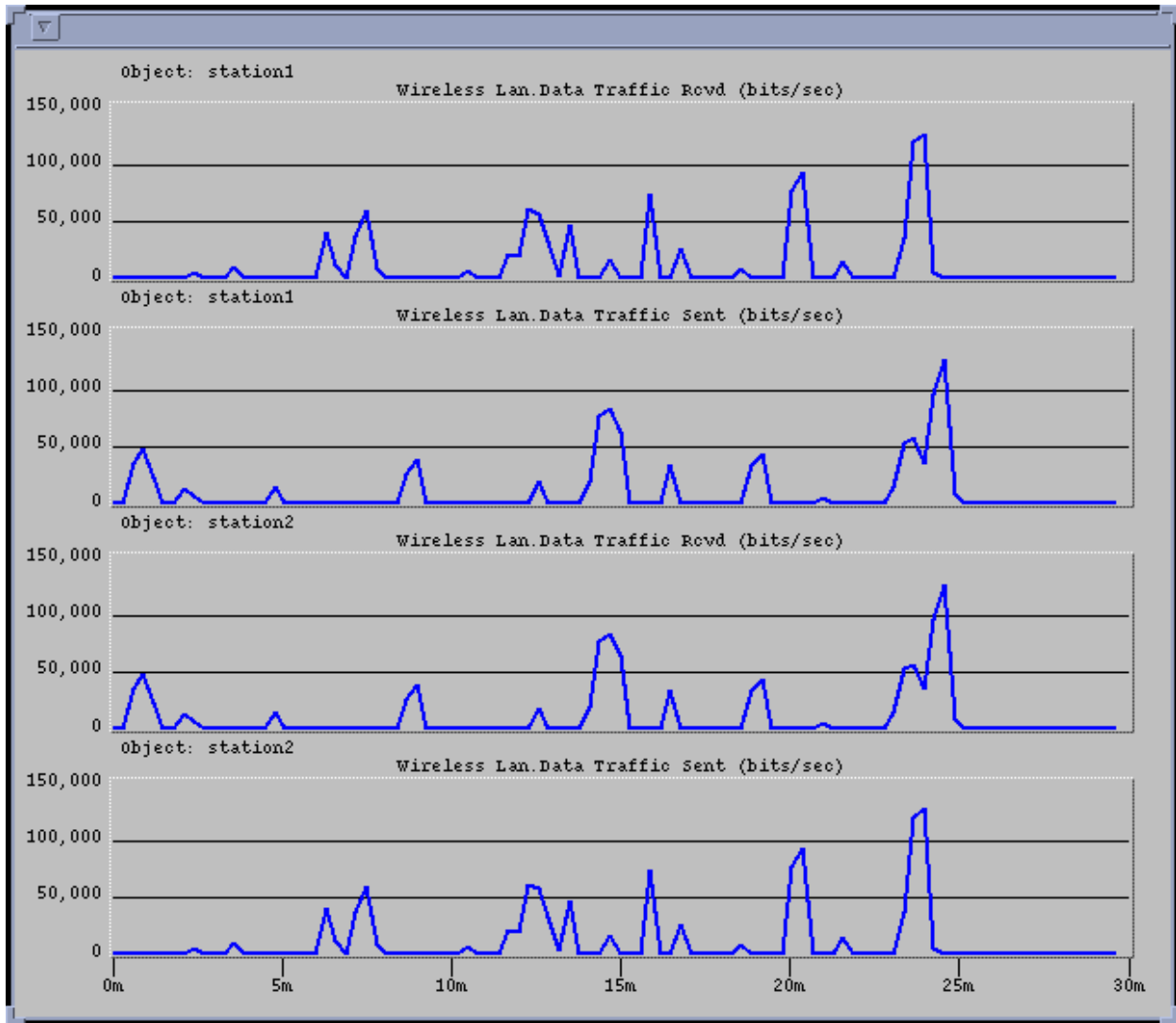


Figure 7 — Sent/Received Traffic Plots

#### 4.2 Trace-driven Traffic in Complex Network:

A more complex network is constructed to resemble practical application of the Wireless Ethernet technology. The network is composed to two separate subnets. Each subnet consists of one Access Point, one wireless node that can act as a server, and two wireless nodes that act as workstations. A regular wired Ethernet router connect to the two subnets at the two Access Points, via wired 100BT links. The network topology is shown in Figure 8, Figure 9 and Figure 10.

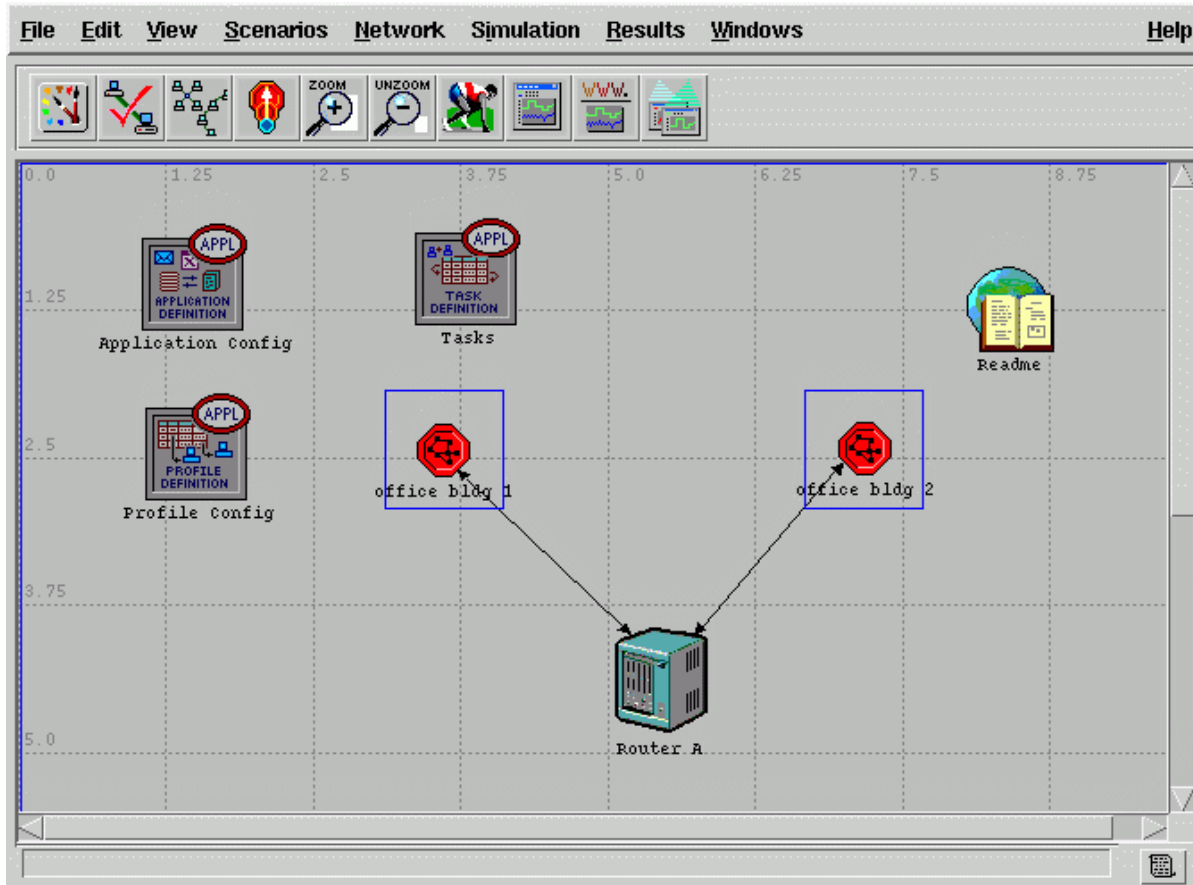


Figure 8 — Top Level Network Setup

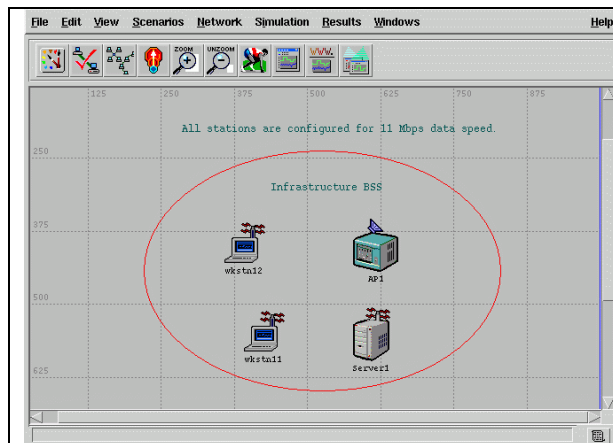


Figure 9 — Network Setup for Subnet 1

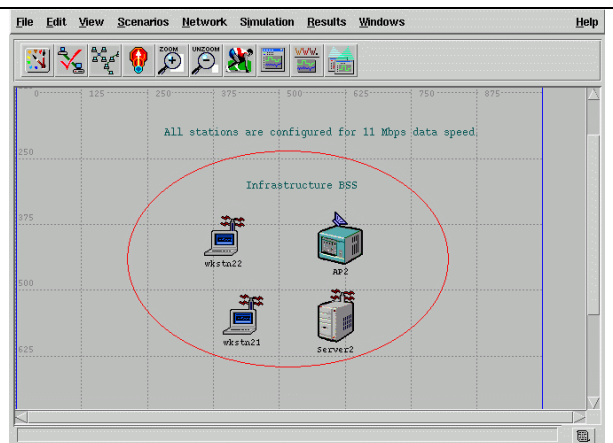


Figure 10 — Network Setup for Subnet 2

All wireless workstations were configured to run at 11 Mbps, with the direct sequence physical protocol. The Star War trace from the course website is used here. The project information and setup procedure are provided below:

Project Path: /ensc/grad1/msuma/op\_admin/mod\_dirs/e833\_wlan.prj  
Scenario: wlan\_network\_duplex\_2

Procedure:

- 1) Select task attributes from task definition.
- 2) Select "edit" for task specification attribute. Currently, there is only 1 task to be performed, namely "custom\_task."
- 3) Select "edit" from manual configuration for change.
- 4) In the current setup, "wkstn11" is the source and "wkstn21" is the destination. The traffic used is the starwar trace.
- 5) To change the trace that's being used, select click on the "source -> Dest" box.
- 6) Click on "Request Packet Size" value column to change to the desire trace.
- 7) If more tasks are to be performed, simply go back to the manual configuration table and change the number of rows to the number of desire task.
- 8) Refer to the following table for source name info:

Node Name	Source Name
wkstn11	source1
wkstn12	source2
wkstn21	source3
wkstn22	source4

The source name should match the application source preference symbolic name for each node.

- 9) The destination names are as follow: wkstn11, wkstn12, wkstn21, wkstn22. This name should match the client name for each node.
- 10) Program the start phase after column to either application start (in this case, all task starts simultaneously) or Previous Phase End (task 2 starts after task 1 is finished) or a numerical number to indicate the start delay time (in second).
- 11) All other attributes should set to be the same as the first task.

The data traffic sent and received at node "wkstn22" was monitored. Figure 11 shows that each node successfully communicated with each other and the desire packets were received at the destination.

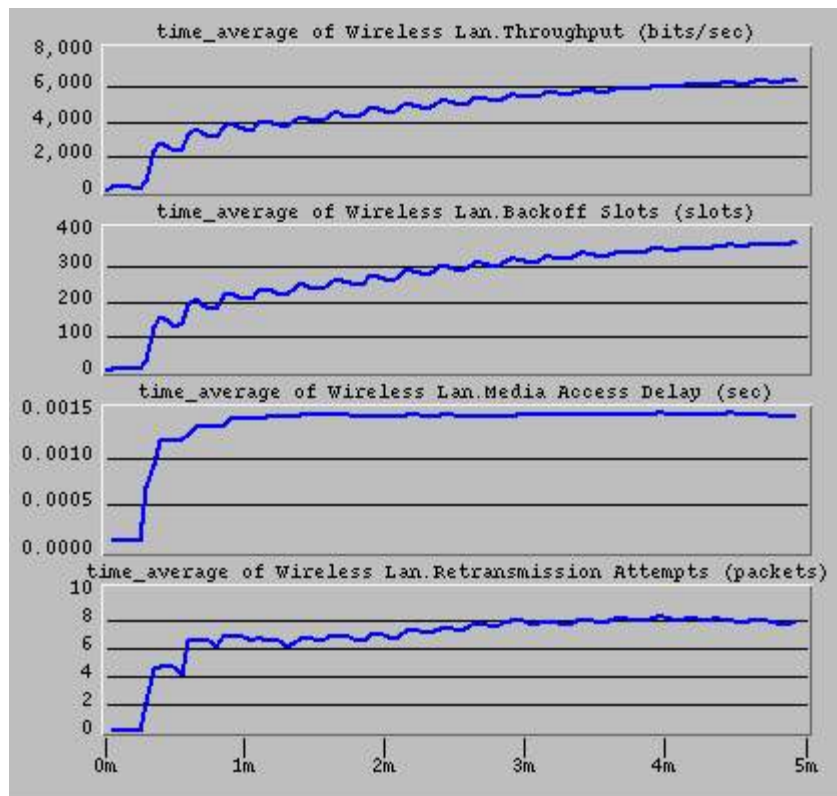


Figure 11 — Simulation Result at wkstn22 (source in subnet 2)

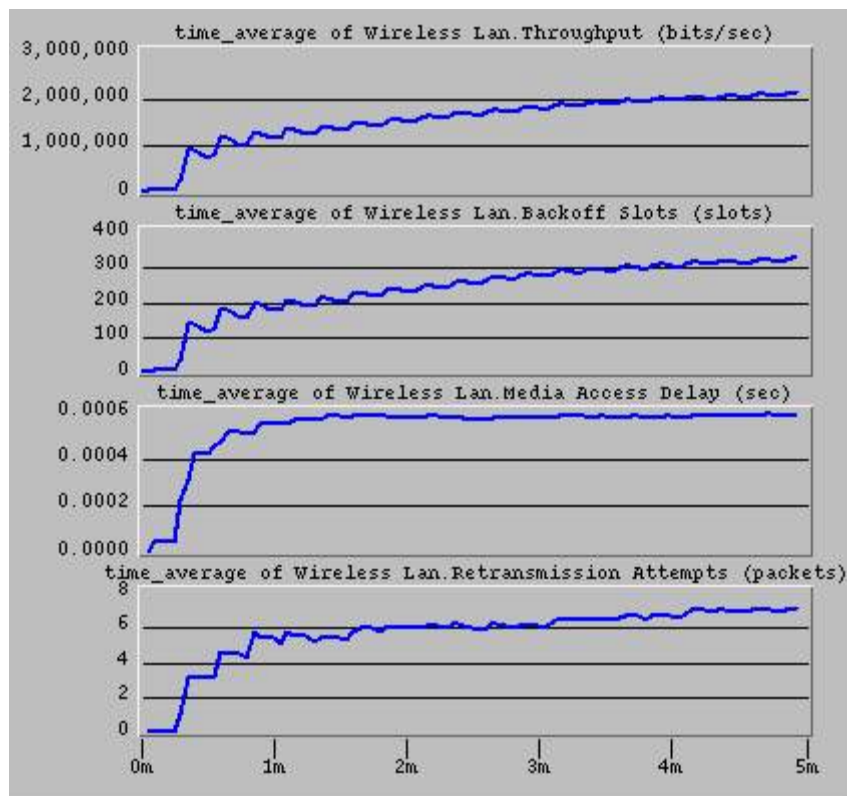


Figure 12 — Simulation Result at wkstn21 (destination)

### **4.3 Variations of MAC Protocol**

In this section, the different MAC backoff schemes were implemented and simulated, to see how network performance was affected. The different MAC backoff schemes are Original Network, Linear Growth, Exponential Distribution and P-persistent.

The network used for all simulation is the complex network from the previous section, where two workstations from Subnet 1 and one workstation from Subnet 2 are transmitting simultaneously to the second workstation in Subnet 2. Movie traces were used as the traffic, and for each MAC backoff scheme, two simulations were run. For the first simulation, the Star Wars (Starwars) movie trace was used by all three traffic sources, and for the second simulation, the Jurassic Park (Dino) movie trace was used by the source in Subnet 2 while the sources in Subnet 1 were using the Star War movie trace. The Starwars trace and the Dino trace were obtained from the the course web page (<http://www.ensc.sfu.ca/people/faculty/ljilja/ENSC833/>) and University of Wuerzburg (<http://nero.informatik.uni-wuerzburg.de/MPEG/>) respectively.

Results for each of the new MAC backoff schemes was compared with results from the original network. Each network was also analyzed as different traffic patterns were transmitted through the network. As some of the simulations took a long time to run, all simulations were run for the first five minutes. All simulation results plotted were using the time-averaged values.

For all simulations, four sets of data series were gathered and analyzed.

- The data throughput of the node.
- The actual number of backoff slots
- The media access delay
- The number of retransmission attempts.

#### **4.3.1 Original Network**

The backoff mechanism comes with the OPNET WLAN model library. The maximum backoff slots, the internal variable `max_backoff`, doubles every time the node needs to re-transmit. The selection of the actual backoff slots bases on uniformly distribution from zero to `max_backoff`.



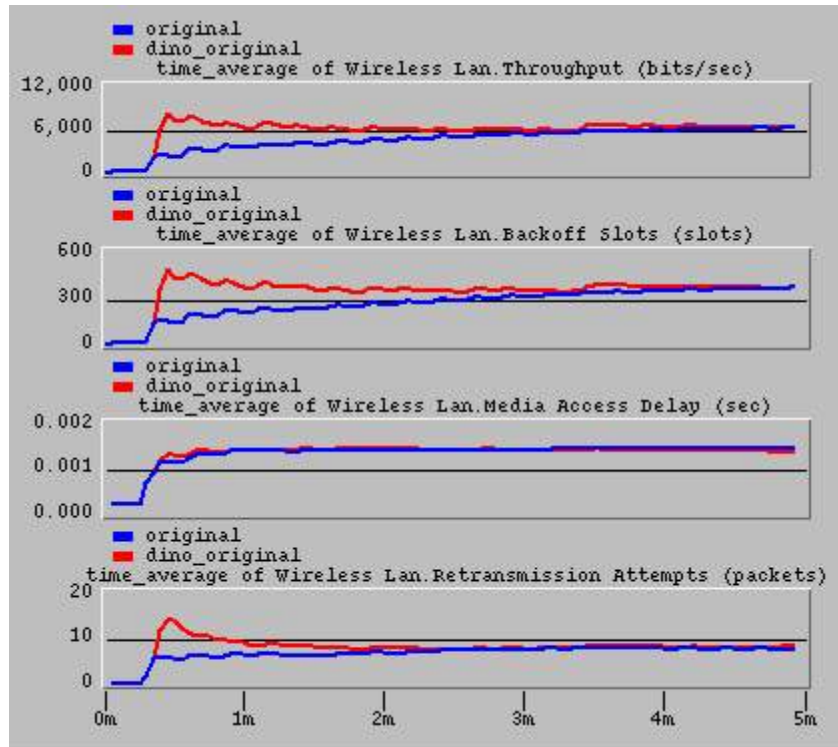


Figure 13 — Comparison of Starwars and Dino, for Original Network

Figure 13 plot the simulation results of the original network, without any changes to the MAC backoff scheme, for both Starwars-only traces and mixed Starwars-Dino traces. As can be seen from the first plot, when the Starwars trace was used, the time-averaged throughput slowly increases to slightly higher than 6000 bits/sec. When the Dino trace was used as well, the time-averaged throughput quickly jumped to around 8000 bits/sec and stabilized to the same level as the Starwars simulation. The actual backoff slots shown on the second plot and the number of retransmission attempts shown on the fourth plot show very similar behavior as the plot for time-averaged throughput, that the Dino simulation overshoots at the beginning and then stabilizes. The media access delay of the two simulations, shown on the third plot, are almost identical, with value of about 1.4 ms.

#### 4.3.2 Linear Growth

The only change from the original network is that the maximum backoff slots grow linearly, instead of binary exponentially as in the original network. The variable `max_backoff` increases by seventeen every time a retransmission is required.

Figure 14 plots the simulation results, comparing the original backoff mechanism to the new linear-growth backoff mechanism. The time-averaged throughput, actual backoff slots, and media access delay of the two backoff mechanisms are almost identical, while the waveforms for number of retransmission attempts are slightly different. This indicates that the particular modification to the backoff mechanism does not affect the network performance.

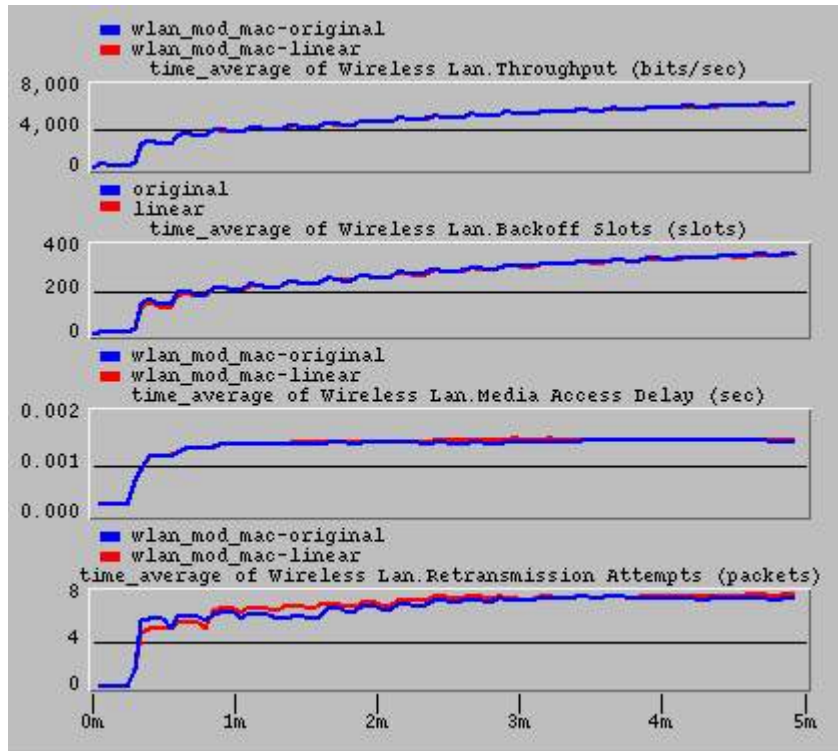


Figure 14 — Comparison of Original and Linear

It can be seen from Figure 15 that, as with the original backoff mechanism, switching to the Dino trace results in initial overshoot for time-averaged throughput, actual backoff slots and number of retransmission attempts. The media access delay was not affected as different traffic was transmitted.

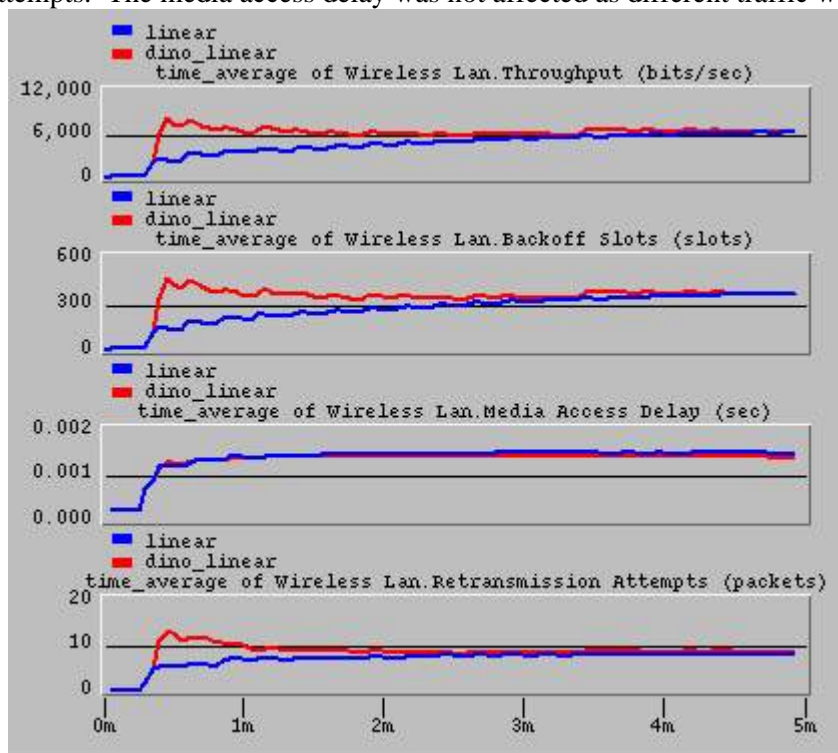


Figure 15 — Comparison of Starwar and Dino Traces, for Linear Growth

### 4.3.3 Exponential Distribution

In this section, the growth of maximum backoff slots is the same as the original network that it doubles every time a retransmission is required. The change is the way the actual number of backoff slots is selected. In the original backoff scheme, the uniform distribution function is used. Here the OPNET-defined exponential distribution function, with a specified mean, is used.

Figure 16 shows the simulation results, when a mean of  $(10 * \text{max\_backoff})$  was used. The time-averaged throughput was almost identical between the two backoff schemes. However, the exponential backoff resulted in greater number of actual backoff slots, longer media access delay and fewer retransmission attempts. These results matched the expectation, as a mean with really large value would cause larger actual values of backoff slots, thus each source waited longer before actually transmitting. This resulted in longer delay but fewer collisions.

Figure 17 shows the simulation results, when a mean of  $(\text{max\_backoff} / 16)$  was used. The time-averaged throughput was almost identical between the two backoff schemes. However, the exponential backoff resulted in smaller number of actual backoff slots, shorter media access delay and fewer retransmission attempts. These results matched the expectation, as a mean with really small value would cause smaller actual values of backoff slots, thus each source waited less time before actually transmitting. This resulted in shorter delay but fewer collisions.

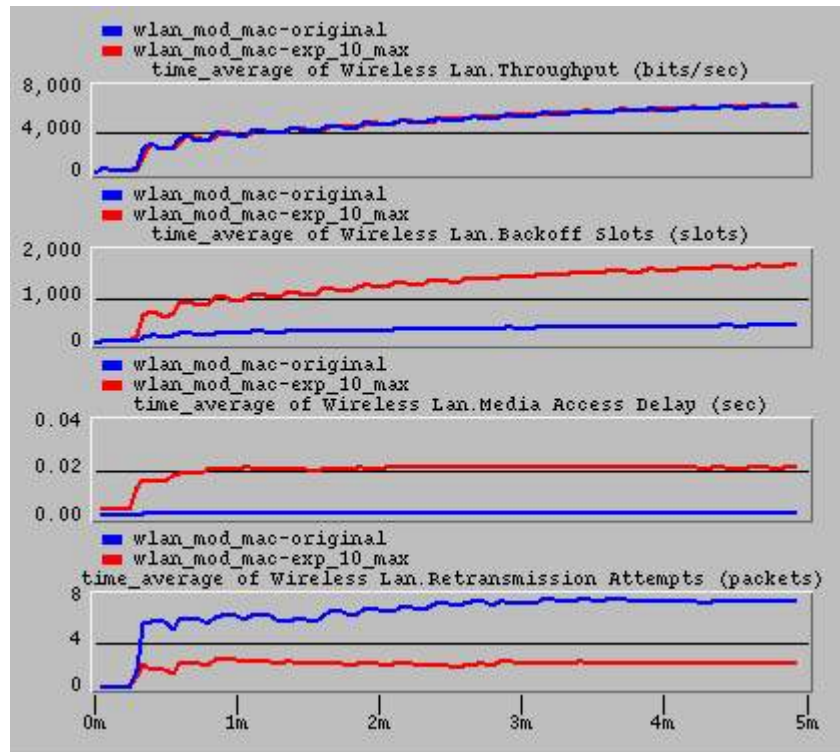


Figure 16 — Comparison of Original and Exponential (mean =  $10 * \text{max\_backoff}$ )

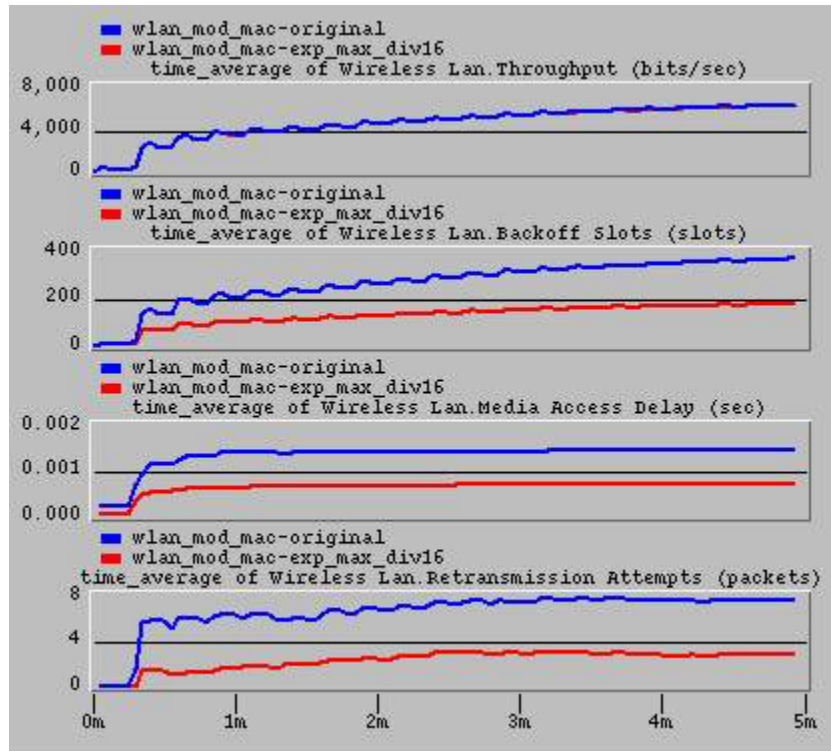


Figure 17 — Comparison of Original and Exponential (mean = max\_backoff / 16)

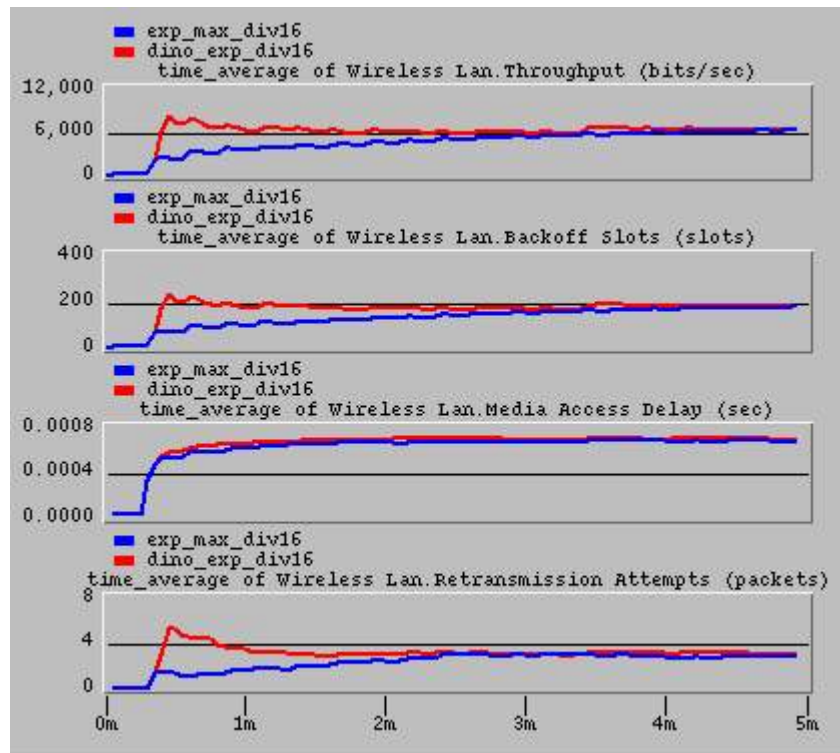


Figure 18 — Comparison of Starwars and Dino Traces, for Exponential Distribution (mean = max\_backoff / 16)

For the case where  $\text{mean} = \text{max\_backoff} / 16$ , the network was simulated again with the Dino movie trace transmitted from one of the source. Only the  $\text{max\_backoff} / 16$  case is chosen for resimulation with Dino trace because from the results of Figure 16 and Figure 17,  $\text{max\_backoff} / 16$  method shows a smaller MAC delay and is believed to have a better performance over the  $10 * \text{max\_backoff}$  case. Figure 18 shows the simulation results of both Starwars-only traffic and Starwars-Dino traffic. Very similar to the original backoff and the case of linear growth, other than almost identical media access delay; the Starwar-Dino traffic caused initial overshoot for throughput, actual number of backoff slots, and number of retransmission attempts.

#### 4.3.4 P-persistent

In their paper “IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism,” Cali, Conti and Gregori discussed the P-persistent backoff scheme where a geometric distribution with parameter P is used. This backoff scheme was implemented and the effects were analyzed in this project.

Description for P-Persistent MAC

- Station still defer while medium is busy
- When medium is free station transmits with probability p and defers with probability (1-p)
- Repeated in next backoff slot

This algorithm allows the average back time to be set via the 'p' value. By varying the probability, an optimal backoff interval can be obtained. However, the optimal setting depends on network configuration, load, traffic, and acceptable network performance.

A smaller backoff interval allows for smaller medium access delays; however, the number of retransmissions may increase. A larger backoff interval may help reduce the number of retransmissions, but the medium access delay may increase and overall bandwidth utilization may suffer.

Changes to WLAN MAC Process Model include:

Modified code for the Exit Executives of the IDLE state is in `idle_exit_execs.c`.

1. The following variable declarations need to be added into the STATE VARIABLES block inside the MAC PROCESS MODEL:

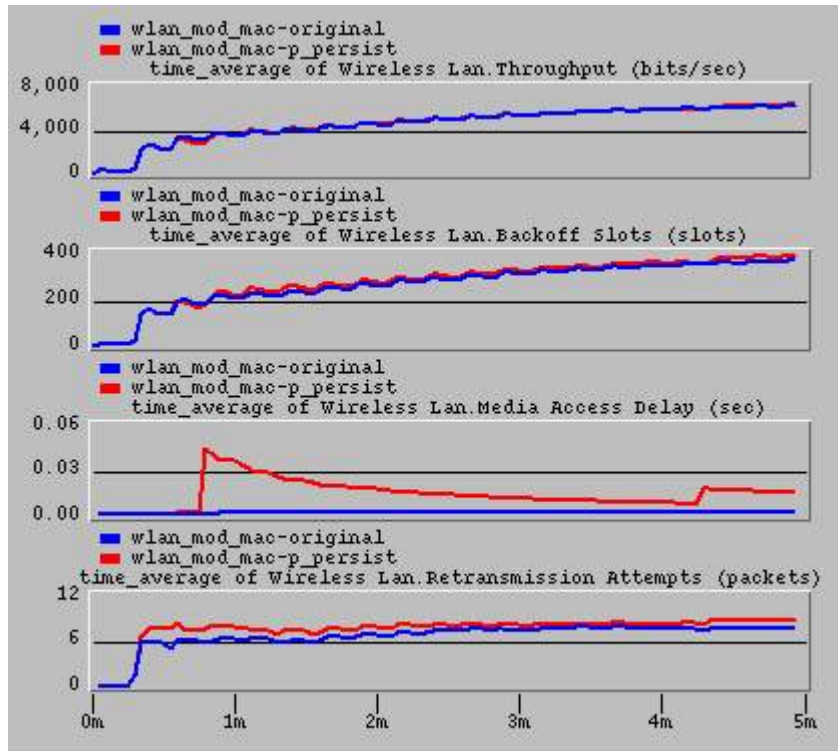
- `double xmit_probability;`
- `double p_persist;`
- `Boolean MEDIUM_IDLE_NOXMIT;`

2. The IDLE -> DEFER state transition condition needs to be changed to:

```
((READY_TO_TRANSMIT && !MEDIUM_IS_IDLE) ||  
(READY_TO_TRANSMIT && MEDIUM_IDLE_NOXMIT))
```

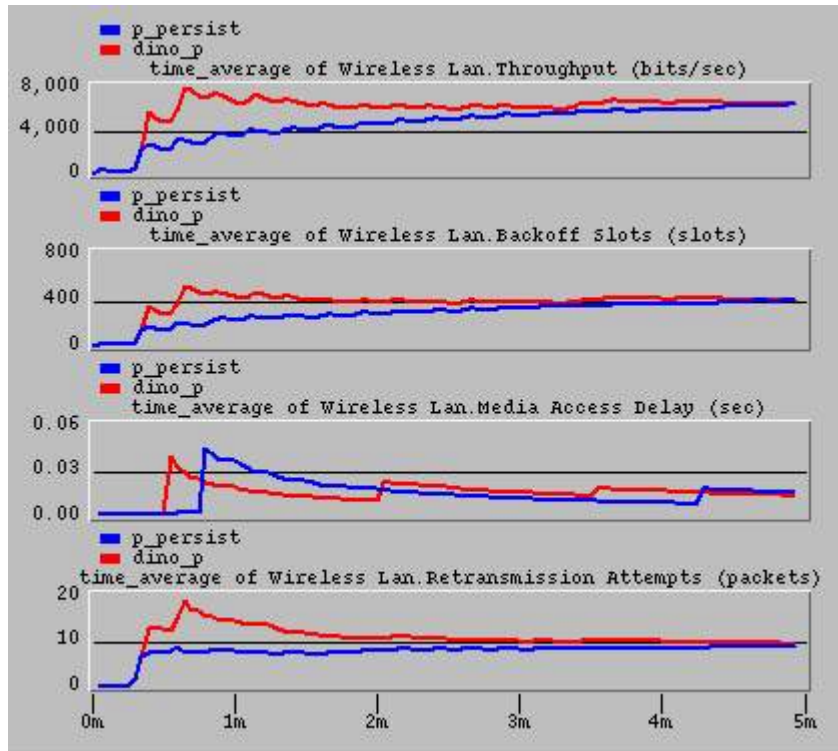
3. The IDLE -> TRANSMIT state transition condition needs to be changed to:

```
(READY_TO_TRANSMIT && MEDIUM_IS_IDLE && !MEDIUM_IDLE_NOXMIT)
```



**Figure 19 — Comparison of Original and P-persistent (P = 90)**

Figure 19 plots the simulation results of the P-persistent backoff scheme, with  $P = 90$ , overlapped with the results from the original backoff scheme. The time-averaged throughput and the actual numbers of backoff slots were almost identical compared to the original backoff scheme. The media access delay of the P-persistent backoff scheme was much larger, with the peak value appearing close to the beginning. The number of retransmission attempts was slightly larger. The MAC delay is worse in P-persistent scheme because large delays occurred a few times throughout the simulation. Since we are using a time-averaged plot, the effect of sporadic, large delays is exaggerated by the averaging.



**Figure 20 — Comparison of Starwars and Dino Traces, for P-persistent (P = 90)**

The network was simulated again with the Dino movie trace transmitted from one of the source. Figure 20 shows the simulation results of both Starwars-only traffic and Starwars-Dino traffic. Very similar to the other backoff schemes, Starwars-Dino traffic caused initial overshoot for throughput, actual number of backoff slots, and number of retransmission attempts. What is unique here is that instead of nearly identical media access delay, the Starwars-Dino traffic shifts the delay waveform earlier and has a slightly smaller peak value. The Dino trace has higher peak data rates at the beginning; therefore, the time-averaged plots shows increased (compared to Starwars) backoff slots, and retransmission attempts near the start of the simulation, and eventually settles to approximately the same levels.

## 5 DISCUSSION AND CONCLUSIONS

In this project, different Wireless Ethernet (WLAN) configurations, using trace-driven traffic data, were simulated. OPNET was the simulator used, with the OPNET WLAN model library. A simple network was constructed first and traffic traces transmission was enabled. A more complex network was later created and its simulation results were referenced for later experimentation. Various modifications to the MAC protocols were done and the network performance was measured, analyzed, and compared. The modifications were made to the way data collisions and transmission backoff are handled. We quantified network performance by measuring the MAC delay, backoff slots, and retransmission attempts. By changing the algorithm by which backoff time is chosen we were able to observe both degradation and improvement in these performance metrics.

The project difficulties include unstable development environment and insufficient documentation for the models. The development environment has not been very stable, as OPNET crashes occasionally because of the models or overloaded hardware. Documentation for the WLAN model library can be improved, as a lot of time was spent learning about it and understanding it. Application notes can also be provided for easier network construction and exploration. In particular, it has not been intuitive how the addresses for the various network protocols, implemented by the WLAN models, should be assigned or whether they need to be assigned at all.

One alternative approach to the project objective include constructing OPNET models from the ground up instead of being dependent on the existing model library. It would be very time consuming and difficult to build an extensive model library, but at the same time greater control and better understanding of the models become possible.

Suggested future work includes:

- Analysis of the PHY layer which the WLAN models in OPNET work do not allow
- Disable the models of certain layers (TCP/IP) as they affect network performance too
- Can tune the backoff selection algorithm based on network conditions, such as number of clients
- More comprehensive simulations
- Investigate of roaming effects
- Investigate of effect of different data traffic



## 6 REFERENCES

1. Al Petrick, "Standards & Protocols: IEEE 802.11b – Wireless Ethernet", [www.csdmag.com/main/2000/06/0006stand.htm](http://www.csdmag.com/main/2000/06/0006stand.htm)
2. OPNET technologies, Inc., "Wireless LAN Model Description", [http://www.opnet.com/products/library/WLAN\\_Model\\_Guide1.pdf](http://www.opnet.com/products/library/WLAN_Model_Guide1.pdf)
3. Editors of IEEE 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Std 802.11-1997", Institute of Electrical and Electronics Engineers, Inc. New York, 1997.
4. Rusty O. Baldwin, Nathaniel J. Davis IV, Scott F. Midkiff, "Implementation of an IEEE 802.11 Wireless LAN Model using OPNET", Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, 1998.
5. Giuseppe Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3, March 2000.
6. Federico Cali, Marcho Conti, and Enrico Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism", IEEE Journal on Selected Areas in Communications, Vol. 18, No. 9, September 2000.
7. Linda D. Paulson, "Exploring the Wireless LANscape", IEEE Journal of Computer, Vol. 33, Issue 10, Page 12~16, October 2000.
8. Ajay Chandra V. Gummalla, John O. Limb, "Design of an Access Mechanism for a High Speed Distributed Wireless LAN", IEEE Journal on Areas in Communications, Vol. 18, No. 9, September 2000.

## 7 APPENDIX A – CODE LISTING

### “idle\_exit\_execs.c”

```
/* Interrupt processing routine*/
wlan_interrupts_process ();

/* Schedule deference interrupt when there is a frame to transmit*/
/* at the stream interrupt and the receiver is not busy*/
if (READY_TO_TRANSMIT)
{
    /* If the medium was idling for a period equal or longer than*/
    /* DIFS time then we don't need to defer.*/

    MEDIUM_IDLE_NOXMIT = OPC_FALSE;

    if (MEDIUM_IS_IDLE)
    {
        /* When medium is IDLE, station transmits with a probability */
        /* equal to the value of "p_persist". Otherwise, the station */
        /* defers with a probability of (1 - persist). */
        xmit_probability = op_dist_uniform (100.1);
        p_persist = 50;

        if (xmit_probability <= p_persist )
        {
            /* We can start the transmission immediately.*/
            wlan_flags->immediate_xmt = OPC_TRUE;
            backoff_slots = 0;
        }
        else
        {
            MEDIUM_IDLE_NOXMIT = OPC_TRUE;
        }
    }
    else
    {
        /* We need to defer. Schedule the end of it.*/
        wlan_schedule_deference ();
    }

    /* If we are in the contention window period, cancel the self*/
    /* interrupt that indicates the end of it. We will reschedule*/
    /* if it will be necessary.*/
    if (intrpt_type == OPC_INTRPT_STRM && op_ev_valid (cw_end_evh) == OPC_TRUE)
    {
        op_ev_cancel (cw_end_evh);
    }
}
}
```