Performance of a Wi-Fi Network as a function of Distance and Number of Connections in a School Environment

Group 1

Rajdeep Bhullar - 301187037
Kamal Ezz - 301213002
Daniel Quon – 301170142

http://www.sfu.ca/~rsa83/

Abstract

With the popularity of wireless devices and the decline of wired Ethernet connections, the importance of a reliable Wireless Local Area Network (WLAN) is crucial to the productivity of the school's faculties and students. In a university environment and with a high number of users, the wide areas along with high concentration in certain areas results in challenges for reliable wireless connection. In this report, we are going to use Network Simulator 2 (ns2) and the 802.11g wireless protocol to simulate these different school environments. Finally, the simulation analysis will be used to determine the optimal configuration for a typical university environment.

Table of Contents

List of Figures

List of Tables

1. Introduction

In a university, many students and faculty members use several personal wireless devices to connect to the school's wireless network on a daily basis. With the numerous simultaneous connections and multiple access points provided by the school, a specific convention needs to be followed to determine the optimum network setup in order to maintain the offering of a reliable wireless network. The goal of this project is to determine an ideal standard for universities to follow when improving or creating new wireless network infrastructure.

In order to establish the ideal setup, this paper will demonstrate the different wireless scenarios simulated using ns2 and the simulation metrics recorded to analyze the performance of the network. These scenarios include topologies of varying distances between the access points as well as altering the number of simultaneous connections. Due to the combination of its current popularity in modern networks, its benefits over older wireless standards and its availability in ns2; the wireless standard utilized for the wireless simulations is the 802.11g and packets are sent at a rate of 1Mbps [2].

The network performance will be assessed by evaluating different metrics that will include throughput, packet loss, end-to-end delay and jitter for each of the scenarios [3]. These metrics will then be analyzed and compared within different wireless standards to determine an ideal configuration that provides students and faculty members with the best Quality of Service (QoS).

2. Technology

Several technologies will be mentioned during throughout this paper that may need preliminary technical knowledge. Accordingly, this section will include some of these technologies as well as a brief description for each.

WI-FI

Wireless connectivity, also known as Wi-Fi, is a mean of communication between devices without the physical need of a wired connection. It allows data transmission between Wi-Fi enabled devices such as computers, mobile phones and tablet devices with the internet server or other wireless devices [5].

Figure 1: Wi-Fi Certified Logo

WI-FI Standards

Different WI-FI standards were introduced throughout the last two decades. A common family of Wi-Fi standards is 802.11, which is used to define an interface between a wireless client and a base station. This family includes 802.11 a/b/g/n/ac standards [2]. Table 2.1 shown below gives a further description of each of the standards and their specifications.

Table 1: Wi-Fi 802.11 Family Standards

Standard	Max Data Rate (Mbps)	Throughput (Mbps)	Channel Bandwidth (MHz)	Operating Frequency Band (GHz)
802.11b	11	6.5	22	2.4
802.11a	54	25	20	5
802.11g	54	25	20	2.4
802.11n	600	100	20, 40	2.4, 5

Protocols

Protocols are rules for which both endpoints in a Wi-Fi connection agree upon to allow successful communication and interface between them [6]. Such set of rules define which parts of the transmitted information should be sent at what time. Two common communication protocols are the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) [6].

UDP

User Datagram Protocol (UDP) is a connectionless communication protocol that is used for establishing low-latency connections [7]. In comparison with TCP where data transmission is tracked and controlled, UDP is mainly concerned with the speed [8].

QoS

Quality of Service (QoS) is the umbrella where the overall performance of a computer network is measured upon. Measuring the QoS includes assessing a number of parameters and metrics such as Throughput, Packet-Loss, End-to-end Delay and Jitter [9]. These metrics will be discussed further in the Performance and Metrics section of this paper.

3. Scenarios and Topologies

Wireless Access Point

Wireless access points have many settings and parameters that can be configured in several ways. One of the most important parameters is the transmission power, which relates to the distance in which the wireless signal can be transmitted. Typical power settings range from 1 to 8, with 1 being the highest power which result in the maximum the signal may travel [10].

For our simulation, we set our transmission power such that we provide 20 meters of coverage with an omnidirectional antenna. Accordingly, a single access point can provide coverage for a classroom while providing connectivity in the hallways. While it may appeal to the user to set the access point at its maximum power, this should be avoided as it may lead to the degradation of the wireless signal [11]. The over saturation of wireless signals introduces undesired noise in the transmission media and therefore the performance of the wireless signal is significantly declined [12]. Below in Figure 2 we show the layout of our simulation for a class room environment.
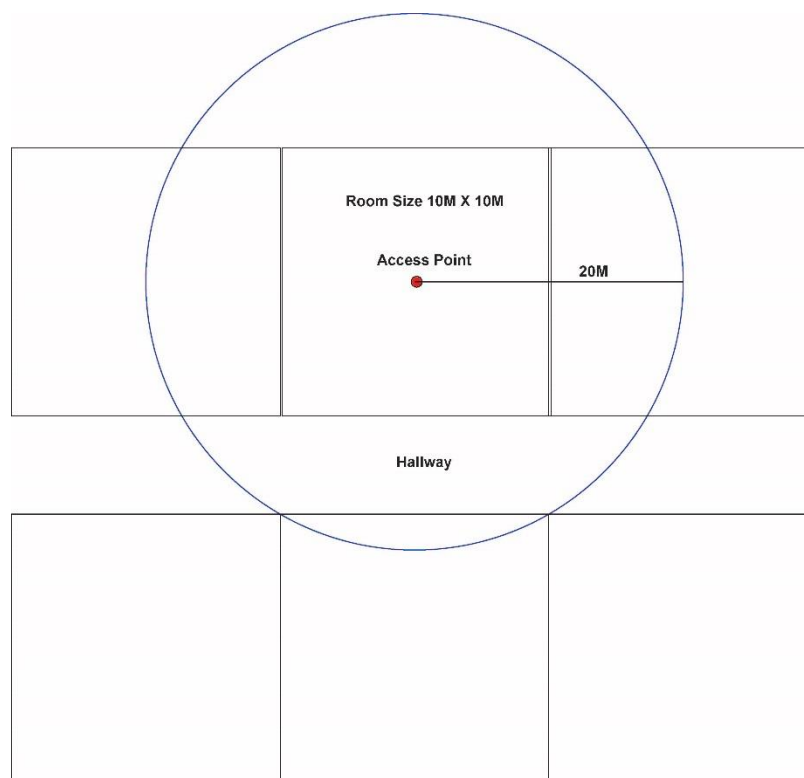

Figure 2: Wireless access point coverage and topology dimensions

Topology Setup

A realistic topology setup was considered throughout the simulated scenarios. The wireless local area network (WLAN) classroom environment was implemented using two wired nodes, a

single or multiple access points and connected wireless clients. Figure 3 shows the wired connections used for all the simulated scenarios. The wired nodes 0 and 1 represent the server and router that the access points send and receive data from. These nodes supply services for the access points represented by node 2 via Ethernet. Ethernet connections are applied using duplex cable links that allow data to be transmitted and received at a rate of 1Gbps, which is typical of a gigabit network. A drop-tail queue is also embedded in the simulation to allow handling of incoming packets.
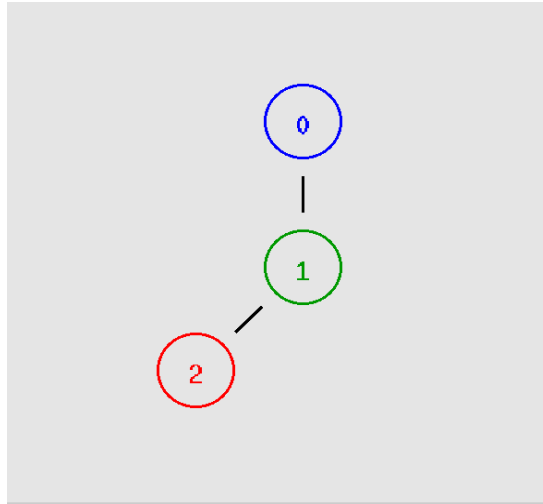

Figure 3: Nam topology showing wired connections and a single access point

Data Transmission and Packet Size

The wireless connections were configured based on the IEEE 802.11g parameters shown in Table 1. To simulate typical internet traffic being transmitted over Wi-Fi, we created Constant Bit Rate (CBR) Traffic at a rate of 1Mbps with a packet size of 620 bytes and interval of 5ms [10]. This traffic is attached to a UDP agent which travels between the wireless nodes and the destination. The following configurations were set in ns2 in order to simulate the 802.11g standard:

```
#*****WIFI Settings*********
set opt(wifi_dr) 54Mb        ; # 802.11g standard wifi BW
Phy/WirelessPhy set Pt_ 9.5872e-6    ;# 20m transmit power
Phy/WirelessPhy set L_ 1.0 ;#System loss factor
Phy/WirelessPhy set bandwidth_ opt(wifi_dr)
Phy/WirelessPhy set freq_ 2.472e9 ;#channel-13. 2.472GHz
Phy/WirelessPhy set CPThresh_ 10.0 ;#reception of simultaneous packets .. collision threshold
Phy/WirelessPhy set CSThresh_ 1.559e-11 ;#carrier sensing threshold/power
Phy/WirelessPhy set RXThresh_ 3.652e-10 ;#reception threshold/power
Mac/802_11 set dataRate_ $opt(wifi_dr)
Mac/802_11 set basicRate_ 24Mb       ;#typical wireless g datarate
```

Simulated Scenarios

For our simulations, different scenarios were tested to determine the maximum number of connected users allowed per access point, as well as the maximum area coverage for each of the access points. These scenarios can be classified into 3 main categories:

Single Access Point – Single Node

To begin with, a simple scenario was tested with 1 access point and 1 connected node. During transmission, the node is moving away from the access point at a constant speed. This was to ensure that the node is behaving as expected and to set a benchmark for how the nodes should behave for later simulations. Shown in Figure 4 is our typical topology with an addition of a single wireless node connection represented in the figure by node 3.
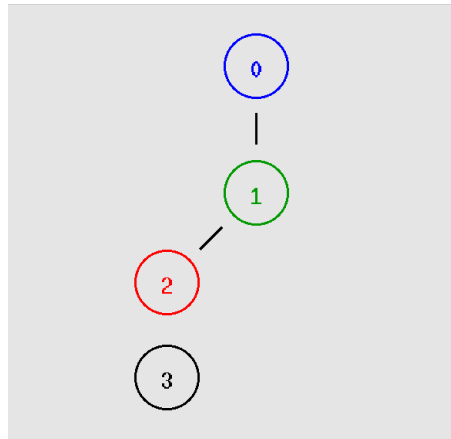
Figure 4: Nam topology with a single access point and a single connected node

Single Access Point – Multiple Nodes

To analyze further impacts on the network, more wireless connections were added in the second scenario. Figure 5 shows the same topology with more than 1 wireless node connected.
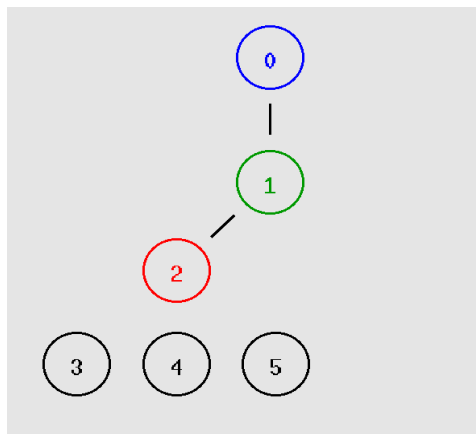
Figure 5: Nam topology with a single access point and multiple connected nodes

Multiple Access Points – Multiple Nodes

We now start simulating adding more access points so that the number of users are split between 2 access points. For the access point configuration in ns2, we also enable *mobileIP*. This setting allows the wireless nodes to connect between access points on different domains. Figure 6 shows the nam topology with 2 access points and users distributed between the access points.

Figure 6: Nam topology with 2 access points and multiple connected nodes

4. Performance and Metrics

In order to assess the performance of our wireless network, a number of parameters and metrics were recorded and monitored. In ns2, we used the LossMonitor agent which records the number of packets received, the time, and number of packets loss. We used these values to calculate various parameters which assist with determining network performance. Below is a description of each of these parameters and their effect on the wireless network performance.

Throughput

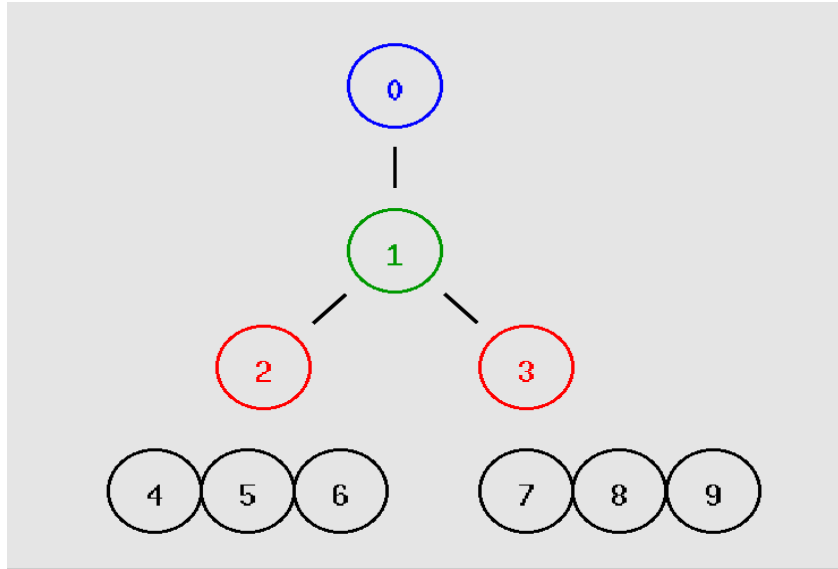Throughput is defined as the amount of data successfully transmitted per unit time over a communication channel [13]. Throughput is limited by the available bandwidth. Bandwidth is marked as the maximum throughput supported by the wireless standard channel [13]. In ns2, the Loss monitor records the number of bytes received therefore we take that amount divided by the time and multiply that value by 8/1000000 to convert to Mbps.

$$Throughput = \frac{Number\ of\ bytes\ received}{time} \times \frac{8}{1000000} = throughput\ in\ Mbps$$

Packet Loss

Packet loss is self-explanatory. It is a measure of the amount of data that failed to reach the targeted destination [13]. Packet loss is usually measured relative to the total sent packets and is typically caused by either link congestion. This occurs when packets arrive at a particular network element at a rate higher than it can handle and therefore the packets are automatically discarded [13].

$$Packet\ Loss = \frac{Number\ of\ packets\ loss}{time} = Packets/time$$

End-to-end Delay

End-to-end delay is the time it takes for a packet to be transmitted from the source node to the destination [13]. In order to calculate this, we take the time for the last packet (lpkttime), subtract the current packet time (timesent) then divide the total by the difference between the current number of packets received by the last number of packets received [14].

$$Avergage\ End - to - end\ Delay = \frac{lpkttime - timesent}{PacketsRecv - LastPktsRecv} = delay\ in\ s$$

Jitter

Jitter is the variation in the delay of the received packets. Although packets are usually sent at a consistent rate, yet delay between packets may be forced to vary due to congestions and limitations of queuing mechanisms [13].

5. Discussion

After collecting the data recorded in the trace files in ns2, we used MATLAB to parse and plot the results. The different scenarios simulated showed results that affected all the parameters discussed in the Scenarios and Topologies section. All simulations assume a constant data transmission as compared to random bursts of data.

Throughput

By simulating the first scenario in which we connected a single node to a single access point, a consistent throughput was maintained throughout the simulation. Figure 7 shows no throughput from 0 to 5 seconds as no data transmission occurs until 5 seconds into the simulation. From 5 seconds onwards, the packets are transmitted at a constant rate of 1 Mbps and the graph shows an uninterrupted throughput of 1 Mbps. After 12.5 seconds the node moves out of the range covered by the access point, resulting in a connection loss and accordingly throughput declines to zero.
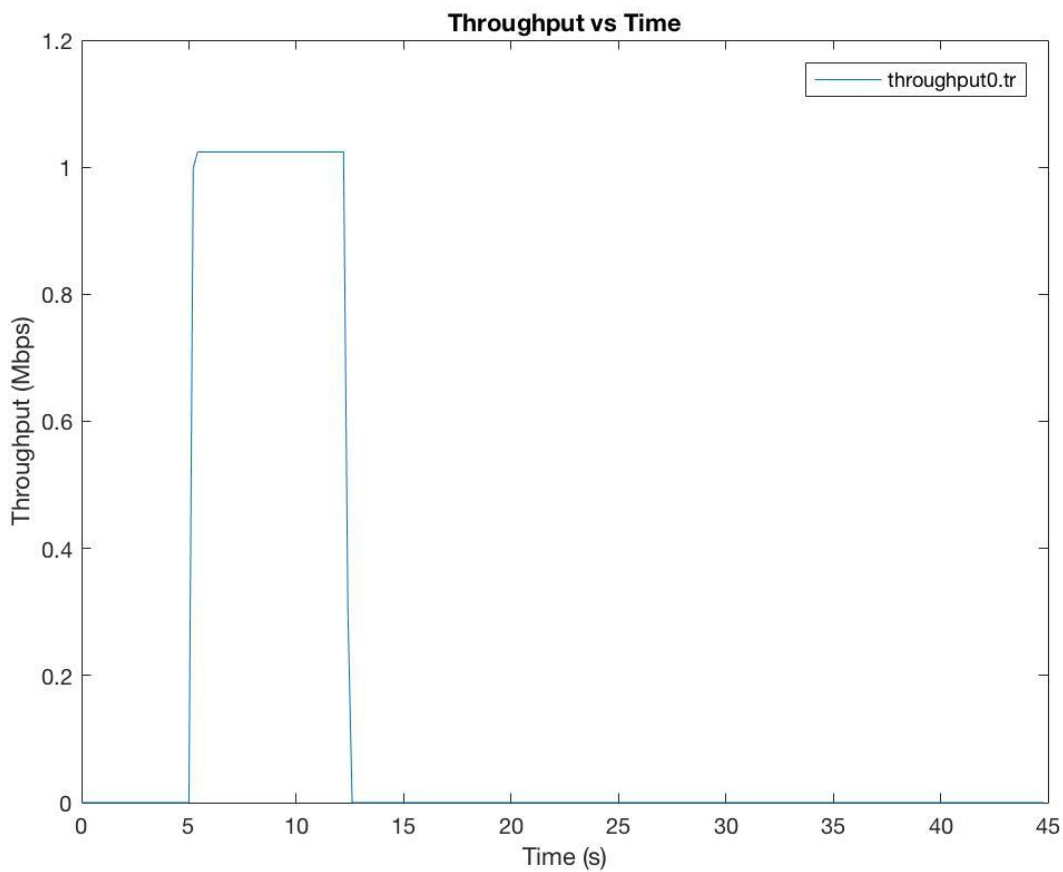

Figure 7: Graph showing throughput vs time for a single access point with 1 connected node

In our second scenario, we increased the number of wireless nodes gradually and analyzed the average throughput for the different number of nodes. Instead of graphing the throughput of each individual node, we used MATLAB to extract the throughputs from each node at each sampling time interval and averaged them. We then took the average throughputs v.s. time as shown in Figure 8. Firstly, we started with adding 7 nodes as seen from 5s to 10s. Next, we doubled the number of nodes for a total of 14 nodes as seen from 10s to 15s. Lastly we increased the number of nodes to 21 and saw that the throughput dropped to just below 0.2 Mbps which we considered to be our minimum throughput rate that is acceptable. This is since most internet webpages are around 1Mb therefore with this throughput it would take roughly 5s in order for all that data to be transmitted versus 1s in the original scenario.

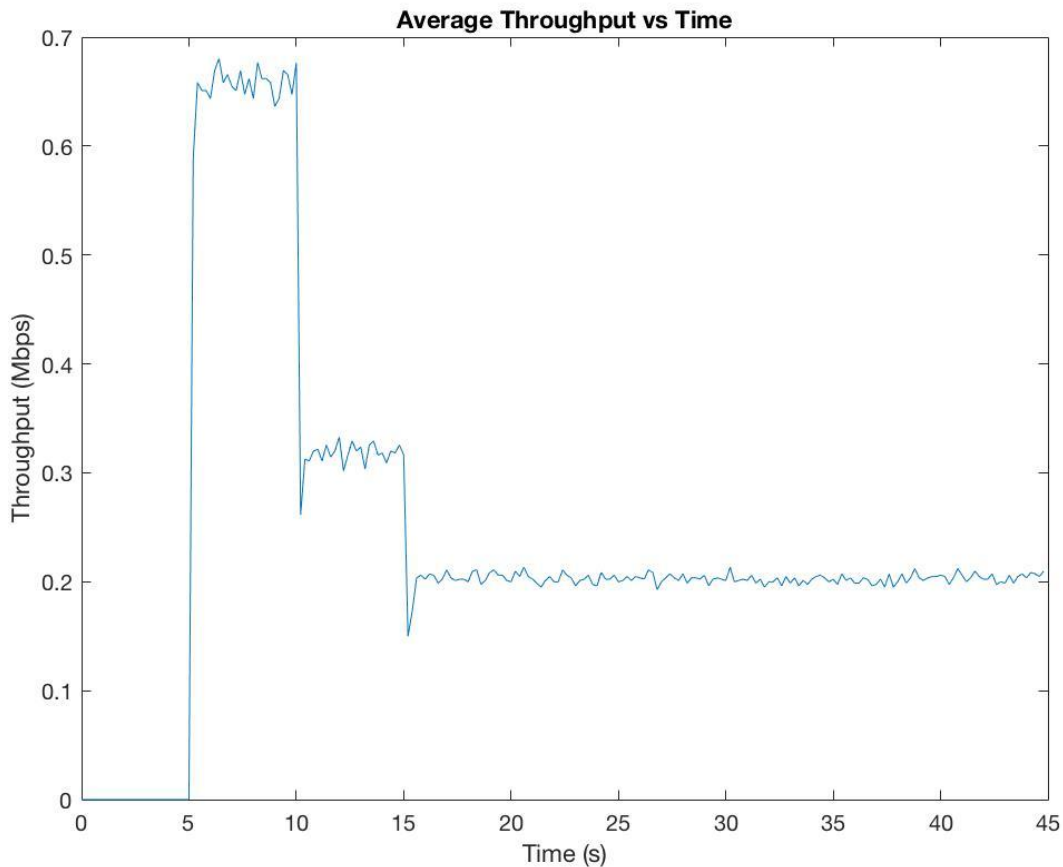*Figure 8: Throughput for multiple nodes connected to a single wireless access point*

In our third scenario, we added an additional access point to see if we see an improvement in our throughput when 21 clients are connected. As seen in Figure 9 the throughput is now around 0.45 Mbps instead of 0.2 Mbps as now some of the mobile nodes are now connected to the other access point elevating the average throughput.
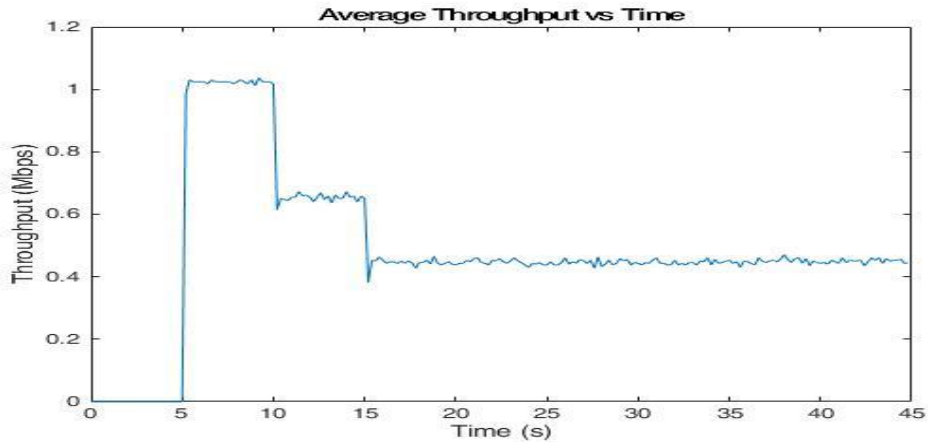
Figure 9: Average throughput vs time for multiple nodes distributed between 2 access points

Packet Loss:

In order to determine signal quality, we examined packet loss under our different scenarios, starting with our base situation of 1 access point with 1 node connected to it. As expected, due to minimal to no interference or noise, there is no packet loss experienced when enrolling one node to a single access point. After 12.5s when the node moves out of the range of the access point, there is no longer a connection between the node and the access point as the connection has dropped and therefore the simulation tool does not account for data loss.

To further investigate potential connection quality, we examine the situation where we increase the number of wireless nodes. As seen in Figure 11 the packet loss increase dramatically when a bunch of nodes all connect as seen at 5s. This loss further increases as the total number of wireless nodes are set to 21. In addition, some packet loss is introduced throughout as seen from 15s onwards.
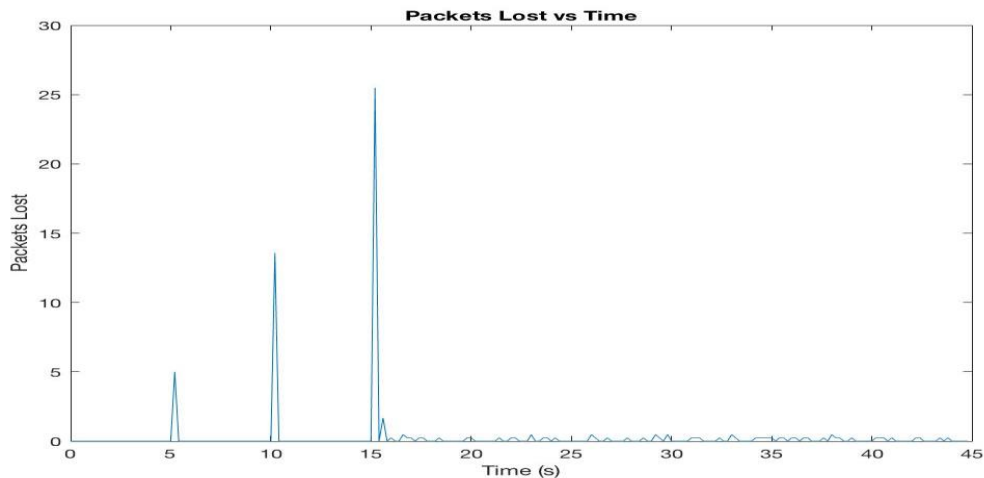
Figure 10: Packet loss for a single access point and multiple connected nodes

14

In comparison with our third scenario, adding another access point resulted in a lower number of simultaneous connections and therefore maximum packet loss dropped from 25 packets to only 11 packets. Packet loss results deducted from adding a second access point is shown in figure 11.
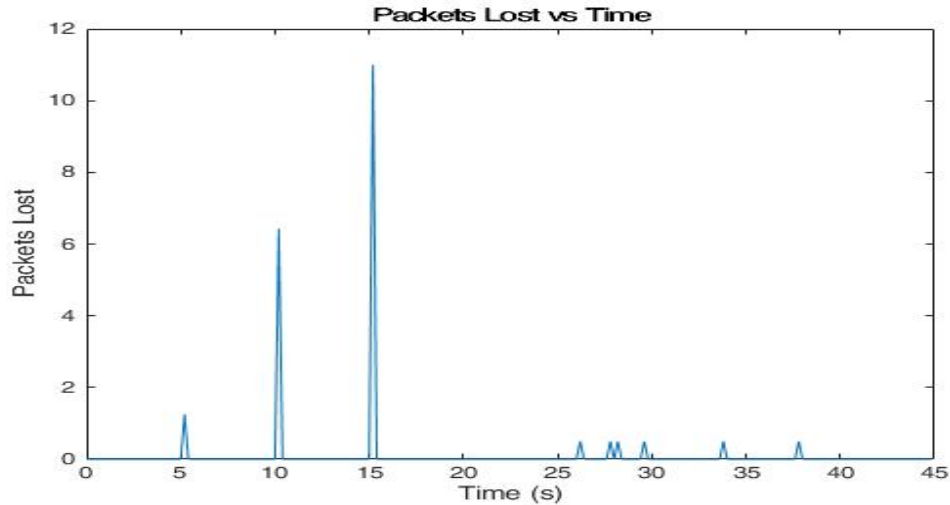
Figure 11: Packet loss for using 2 access points

End-to-end Delay:

Another parameter that is of concern when analyzing Wi-Fi is the amount of delay that occurs between packets being sent over the network. Once again, we look at our simple scenario where only 1 client is connected to the access point. As shown in Figure 12, we see a spike in the delay when the node starts to send data to its destination. After this initial delay of 0.2s there is no more delay.
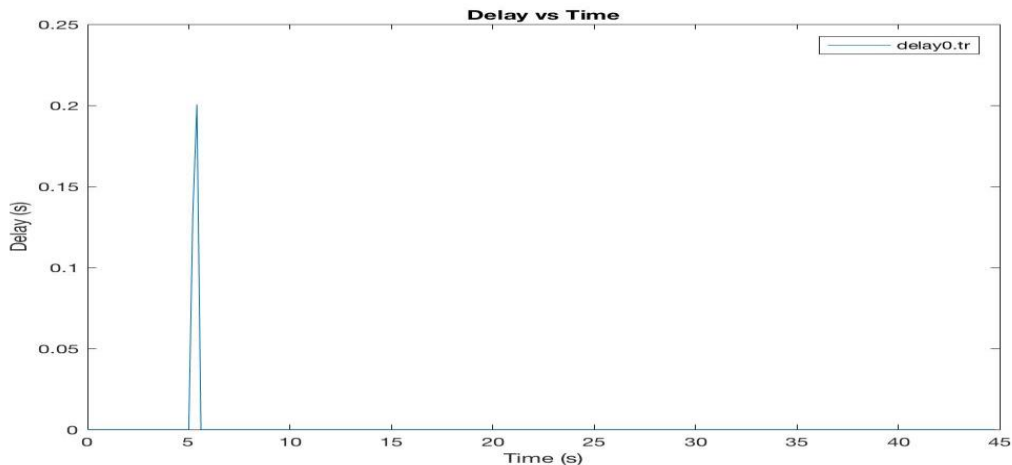
Figure 12: End-to-end delay for a single access point and 1 connected node

Alternatively, when we add more nodes the average delay time when the nodes are added increases as seen in Figure 13 at 10s and 15s. Also in addition to the initial delay, there is some delay throughout the simulation.

15
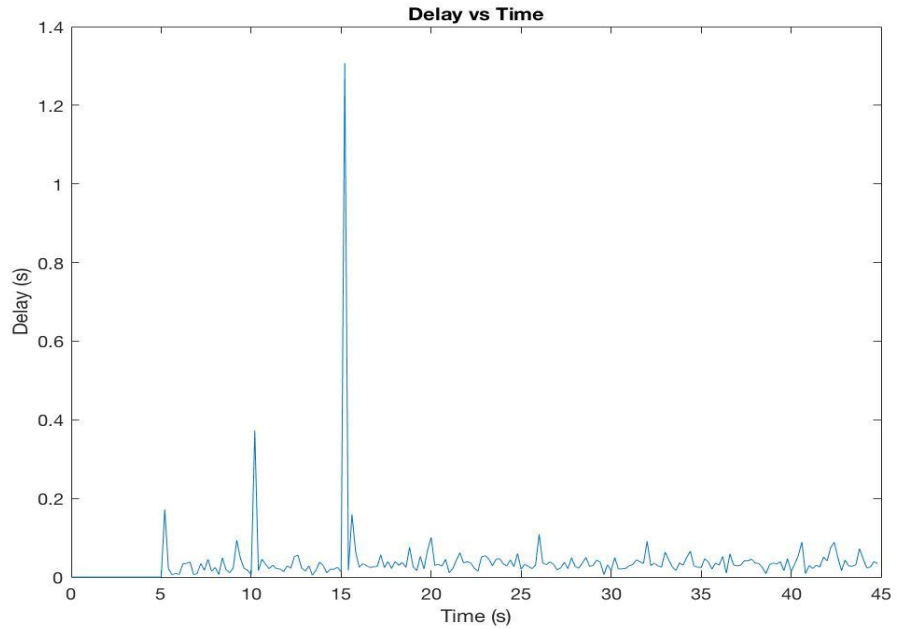
Figure 13: Delay for a single access point and multiple connected nodes

When we consider adding a second access point, the delay has significantly declined from a maximum peak of 1.3s to 0.45s as shown in Figure 14.
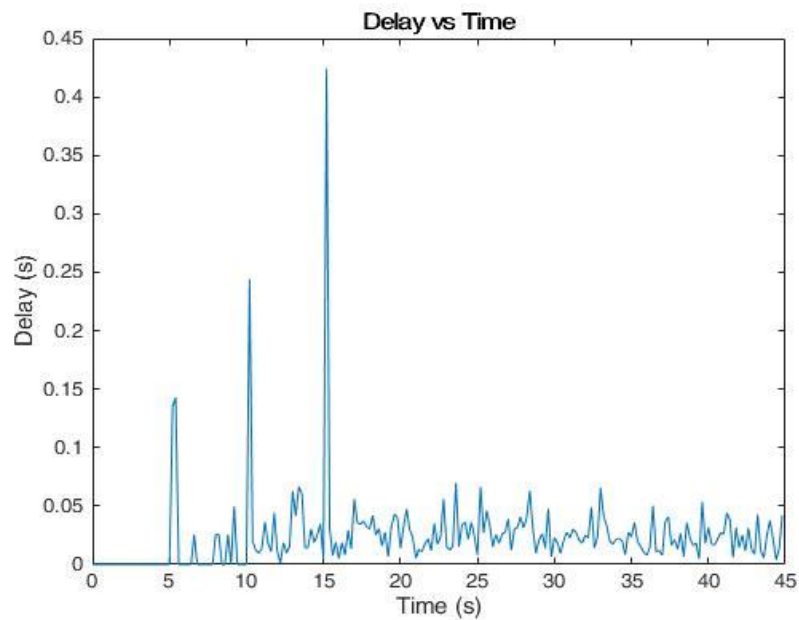
Figure 14: Delay for multiple nodes connected to 2 access points

Jitter:

In order to analyze the delay in more delay we look at jitter. We see that Jitter has dropped significantly when 2 access points served the nodes compared to when a single access point was used as shown in figures 16 and 15 respectively.
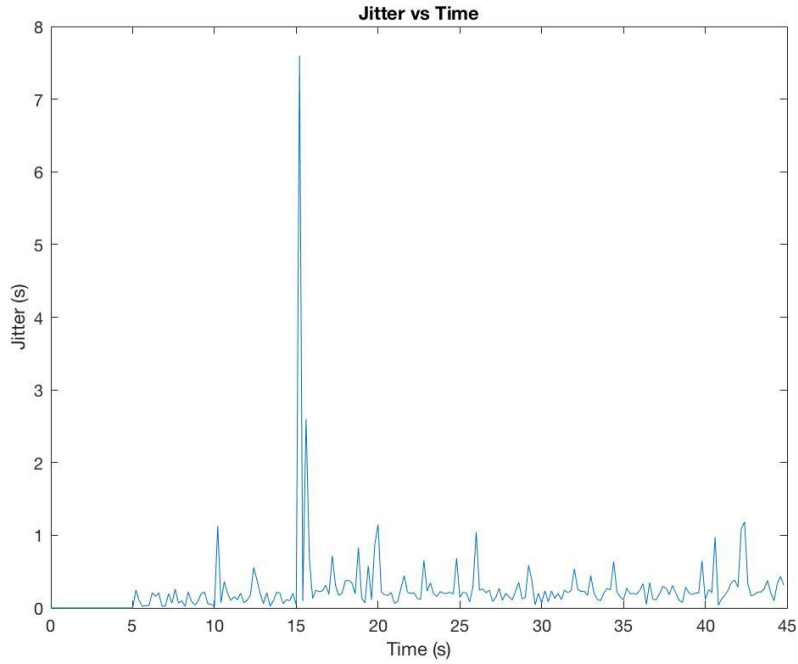
Figure 15: Jitter for connecting multiple nodes to a single access point
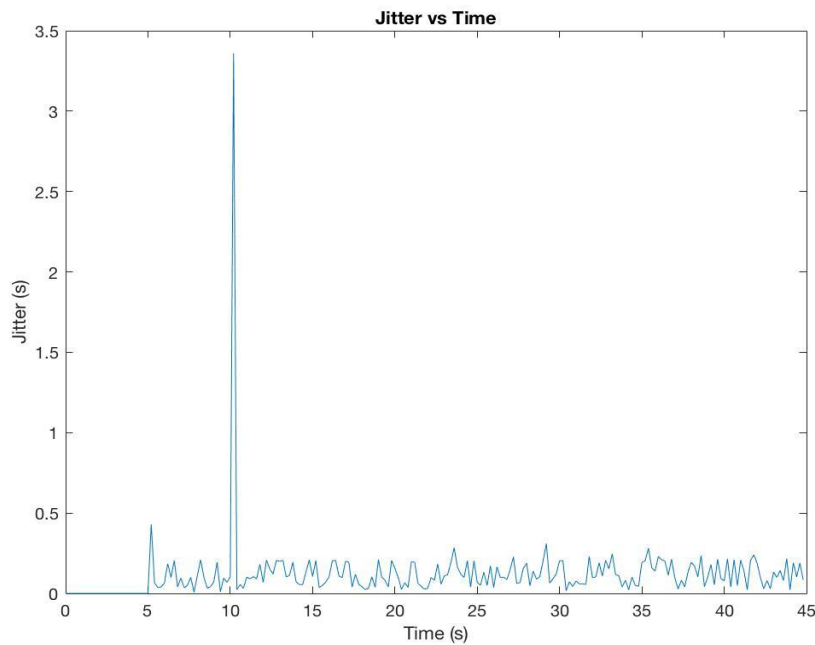
Figure 16: Jitter for connecting multiple nodes to 2 access points

6. Conclusion and Future Work

Conclusion

In this project, we have simulated a wireless network in a university environment with network simulator 2 (ns2). Using ns2 along with MATLAB, we successfully simulated, graphed and analyzed data from various scenarios. Through the results and graphs of throughput, delay, loss, and jitter we have seen a decline in performance when more wireless clients were added. In addition, we saw an improvement when an addition access point was added to the configuration. We have also seen clients lose connection if they displaced outside the wireless network range. When all devices simultaneously send data to one access point at 1Mbps and the minimum acceptable throughput rate is 200Kbps, 20 is the maximum recommended number of devices that should be connected without experiencing unacceptable decline in performance. We can now safely suggest this ideal setup for a classroom and scale the setup accordingly to a university environment. This optimum network setup ensures the best Quality of Service (QoS) for faculty members and students in a university environment.

Difficulties

Throughout this project, many difficulties emerged which we had to overcome before obtaining the results presented. One of the first hurdles was to learn the syntax ns2 in order to setup our topologies and scenarios. Due to the tool's limitation and simplicity, following a strict syntax is required and accordingly significant time needs to be invested. Another obstacle was the complexity and difficulty involved in setting up a realistic test environment. We had several issues trying to specify the throughput packet size. In ns2, there is the ability to customize not only what is happening in the physical layer in terms of the connection of nodes, but also what is going on layers on higher levels. Many of these settings and parameters are in multiple C++ files, which made it hard to determine which variables needed to be altered.

Furthermore, we wanted to simulate the 802.11n standard, but due to the difficulties in implementing a Multiple Input Multiple Output (MIMO) wireless configuration we were unable to accomplish this and we utilized the 802.11g standard. Also worth mentioning, adding an additional access point accompanied several constraints. In order to add the extra access point, we needed to specify and use domains and clusters in order to divide and address the different nodes appropriately. This task required significant amounts of time to be invested.

Future Work

In the future, we would like to implement the 802.11n standard with MIMO in which we were unable to accomplish in order to get results that reflect the current widely used hardware. We would also like to add random traffic patterns where packets would be sent in stages and bursts as compared to a consistent stream of data transmission. This would help get an even more accurate representation of what happens in a typical university environment.

7. References

[1] M. Afanasyev *et al*. Usage patterns in an urban WiFi network. *IEEE/ACM Transactions on Networking (TON) 18(5),* pp. 1359-1372. 2010. DOI: 10.1109/TNET.2010.2040087.

[2] P. P and M. D. H. Simulation and performance analysis of vertical handoff between WiFi and WiMAX using media independent handover services. *International Journal of Computer Applications 87(4),* 2014. DOI: 10.5120/15196-3578.

[3] S. Bhosale and R. D. Daruwala. Performance evaluation of QoS parameters during WiMAX to WiMAX handoff using NS2 and QualNet. *International Journal of Computer Applications 64(9),* 2013. DOI: 10.5120/10664-5435.

[4] A. I. Al-Alawi. WiFi technology: Future market challenges and opportunities. *Journal of Computer Science 2(1),* pp. 13-18. 2006.

[5] K. C. Neel. WiFi grows beyond cellular shadow. *Multichannel News 35(39),* pp. 8. 2014.

[6] A. A. Abdulaziz, A. S. Mohmad and Abdul Nasir Bin Zulkifli. QoS for virtual reality software based on RTCP over the protocols of IP/UDP/RTP. *International Journal of Computer Science and Information Security 9(11),* pp. 58-62. 2011.

[7] H. Zheng and J. Boyce. An improved UDP protocol for video transmission over internet-to-wireless networks. *IEEE Transactions on Multimedia 3(3),* pp. 356-365. 2001. DOI: 10.1109/6046.944478.

[8] R. Sharma and K. Gupta. Comparison based performance analysis of UDP/CBR and TCP/FTP traffic under AODV routing protocol in MANET. *International Journal of Computer Applications 56(15),* 2012. DOI: 10.5120/8969-3194.

[9] O. F.A, O. M.O and O. O.B. cWRR: A scheduling algorithm for maximizing the performance of quality of service network router. *International Journal of Computer Applications 44(16),* pp. 1-5. 2012. DOI: 10.5120/6344-7438.

[10] A J. Florwick, J. Whiteaker, A. Amrod and J. Woodhams, "Wireless LAN Design Guide for High Density Client Environments in Higher Education," 10 July 2015. [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1250-series/design_guide_c07-693245.html#wp9001157.

[11] J. G. Andrews *et al*. A primer on spatial modeling and analysis in wireless networks. *IEEE Communications Magazine 48(11),* pp. 156-163. 2010. DOI: 10.1109/MCOM.2010.5621983.

[12] F. Baccelli, N. Bambos and N. Gast. Distributed delay-power control algorithms for bandwidth sharing in wireless networks. *IEEE/ACM Transactions on Networking (TON) 19(5),* pp. 1458-1471. 2011. DOI: 10.1109/TNET.2011.2163079.

[13] F. kaboudvand and n. kaboodvand. Quality of service in wireless network. *International Journal of Computer Science and Communication Networks 2(1),* pp. 129-133. 2012.

[14] A. Saifullah *et al*. End-to-end communication delay analysis in industrial wireless networks. *IEEE Transactions on Computers 64(5),* pp. 1361-1374. 2015. DOI: 10.1109/TC.2014.2322609.

[15] J. Kim, J. Zheng, and P. Bertsch, "Video Streaming over WiFi". Available: http://www.sfu.ca/~jwk10/ENSC%20427%20Spring%202015%20Group%202%20Report.pdf

[16] S. Calzada, C. Rietchel, and T. Szajner, "Performance Analysis of a Wireless Home Network". Available: http://www.sfu.ca/~tszajner/FinalReport_Group4_Spring2014.pdf

[17] K. Fall and K. Varadhan, "The ns Manual (formerly ns Notes and Documentation)," 5 Novemeber 2011. [Online]. Available: http://www.isi.edu/nsnam/ns/tutorial/.

8. Appendix

```
# THIS SCRIPT SIMULATES 2 WIRED NODES CONNECTED TO 2 ACCESS POINT AND 21 MOBILE
NODES
# WE ARE GOING TO BE VISUALIZING DATA LOSS, THROUGHPUT, DELAY
# TRAFFIC BETWEEN ALL N MOBILE NODES TO THE ACCESS POINT (AP)
# NO MOVEMENT
# wireless_phase3.tcl
# ========================================================================
# Define options
# ========================================================================
set opt(chan)        Channel/WirelessChannel    ;# channel type
set opt(prop)         Propagation/TwoRayGround   ;# radio-propagation model
set opt(netif)        Phy/WirelessPhy          ;# network interface type
set opt(mac)          Mac/802_11             ;# MAC type
set opt(ifq)         Queue/DropTail/PriQueue    ;# interface queue type
set opt(ll)          LL                 ;# link layer type
set opt(ant)          Antenna/OmniAntenna        ;# antenna model
set opt(ifqlen)        10000                  ;# max packet in ifq
set opt(nn)         21                 ;# number of mobilenodes
set opt(adhocRouting)   DSDV                  ;# wireless routing protocol

#*****WIFI Settings***********
set opt(wifi_dr) 54Mb      ; # 802.11g standard wifi BW
Phy/WirelessPhy set Pt_ 9.5872e-6    ;# 20m transmit power
Phy/WirelessPhy set L_ 1.0 ;#System loss factor
Phy/WirelessPhy set bandwidth_ opt(wifi_dr)
Phy/WirelessPhy set freq_ 2.472e9 ;#channel-13. 2.472GHz
Phy/WirelessPhy set CPThresh_ 10.0 ;#reception of simultaneous packets .. collision threshold
Phy/WirelessPhy set CSThresh_ 1.559e-11 ;#carrier sensing threshold/power
```

```
Phy/WirelessPhy set RXThresh_ 3.652e-10 ;#reception threshold/power

Mac/802_11 set dataRate_ $opt(wifi_dr)

Mac/802_11 set basicRate_ 24Mb    ;#typical wireless g datarate


set opt(x)     100                 ;# x coordinate of topology

set opt(y)     100                 ;# y coordinate of topology

set opt(seed)   0.0                  ;# seed for random number gen.

set opt(stop)   45.0                   ;# time to stop simulation


set opt(ftp1-start)     1.0

set opt(ftp2-start)     30.0


set num_wired_nodes     2

set num_ap_nodes        2


# ============================================================================
# check for boundary parameters and random seed
if { $opt(x) == 0 || $opt(y) == 0 } {
        puts "No X-Y boundary values given for wireless topology\n"
}
if {$opt(seed) > 0} {
        puts "Seeding Random number generator with $opt(seed)\n"
        ns-random $opt(seed)
}


# create simulator instance
set ns_   [new Simulator]


#throughput, packet lost, packet delay
for {set i 0} {$i < $opt(nn)} {incr i} {
        set t($i) [open throughput$i.tr w]
        set l($i) [open lost$i.tr w]
```

```
        set d($i) [open delay$i.tr w]
}


#Initialize Values
for {set i 0} {$i < $opt(nn)} {incr i} {
        set timesent($i) 0
        set lpktsrec($i) 0
}


# set up for hierarchical routing
$ns_ node-config -addressType hierarchical


AddrParams set domain_num_ 2          ;# number of domains
lappend cluster_num 2 2           ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 11 10         ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain


#set up nam and trace file
set tracefd  [open wireless_phase3.tr w]
set namtrace [open wireless_phase3.nam w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)


# Create topography object
set topo   [new Topography]


# define topology
$topo load_flatgrid $opt(x) $opt(y)


# create God
create-god [expr $opt(nn) + 4]
```

```
#create wired nodes
set temp {0.0.0 0.1.0}        ;# hierarchical addresses for wired domain
for {set i 0} {$i < $num_wired_nodes} {incr i} {
   set W($i) [$ns_ node [lindex $temp $i]]
   # $ns_ initial_node_pos $W($i) 1
}

#set colour of wired nodes
$W(0) color blue
$W(1) color orange

# configuration for access point nodes
$ns_ node-config -mobileIP ON \
              -adhocRouting $opt(adhocRouting) \
         -llType $opt(ll) \
         -macType $opt(mac) \
         -ifqType $opt(ifq) \
         -ifqLen $opt(ifqlen) \
         -antType $opt(ant) \
         -propType $opt(prop) \
         -phyType $opt(netif) \
         -channelType $opt(chan) \
              -topoInstance $topo \
         -wiredRouting ON \
              -agentTrace ON \
         -routerTrace OFF \
         -macTrace ON

#create access point nodes
set AP(0) [$ns_ node 1.0.0]
set AP(1) [$ns_ node 1.1.0]
```

```
$AP(0) random-motion 0          ;# disable random motion
$AP(1) random-motion 0


#provide some co-ord (fixed) to access point node
$AP(0) set X_ 25.0
$AP(0) set Y_ 45.0
$AP(0) set Z_ 0.0
$AP(0) color red


$AP(1) set X_ 75.0
$AP(1) set Y_ 45.0
$AP(1) set Z_ 0.0
$AP(1) color red


#configure mobilenodes
$ns_ node-config -wiredRouting OFF


set temp {1.0.1 1.0.2 1.0.3 1.0.4 1.0.5 1.0.6 1.0.7 1.0.8 1.0.9 1.0.10 1.0.11 1.0.12 1.0.13 1.0.14
1.0.15 1.0.16 1.0.17 1.0.18 1.0.19 1.0.20 1.0.21 1.0.22}   ;# hier addresses for wireless domain


for {set i 0} {$i < $opt(nn)} {incr i} {
        set node_($i) [ $ns_ node [lindex $temp $i]]
        set APaddress [AddrParams addr2id [$AP(0) node-addr]] ;# provide hier address of its
base-station
        [$node_($i) set regagent_] set home_agent_ $APaddress
}


#Provide co-ordinates for node_(0), node_(1), node_(2),node_(3)
for {set i 0} {$i < 11} {incr i} {
        $node_($i) set X_ 15.0
        $node_($i) set Y_ 35.0
        $node_($i) set Z_ 0.0
}
```

```
for {set i 11} {$i < $opt(nn)} {incr i} {
        $node_($i) set X_ 85.0
        $node_($i) set Y_ 35.0
        $node_($i) set Z_ 0.0
}


#create links between wired and AP nodes


$ns_ duplex-link $W(0) $W(1) 1Gb 5ms DropTail

$ns_ duplex-link $W(1) $AP(0) 1Gb 5ms DropTail

$ns_ duplex-link $W(1) $AP(1) 1Gb 5ms DropTail


$ns_ duplex-link-op $W(0) $W(1) orient down

$ns_ duplex-link-op $W(1) $AP(0) orient left-down

$ns_ duplex-link-op $W(1) $AP(1) orient right-down


# setup UDP connections between wireless nodes and $W(1)

for {set i 0} {$i < $opt(nn)} {incr i} {
        set udp($i) [new Agent/UDP]
        $udp($i) set class_ 0
        set sink($i) [new Agent/LossMonitor]
        $ns_ attach-agent $node_($i) $udp($i)
        $ns_ attach-agent $W(1) $sink($i)
        $ns_ connect $udp($i) $sink($i)
        $udp($i) set fid_ 0


        set cbr($i) [new Application/Traffic/CBR]
        $cbr($i) set packetSize_ 620   ;#620 bytes
        $cbr($i) set interval_ 0.005   ;#rate of 5ms therefore data rate of 1Mb/s
        $cbr($i) set class_ 0
        $cbr($i) attach-agent $udp($i)
```

```
}

for {set i 0} {$i < 7} {incr i} {
        $ns_ at 5.0 "$cbr($i) start"
        $ns_ at 30.0 "$cbr($i) stop"
}

for {set i 7} {$i < 14} {incr i} {
        $ns_ at 10.0 "$cbr($i) start"
        $ns_ at 30.0 "$cbr($i) stop"
}

for {set i 14} {$i < $opt(nn)} {incr i} {
        $ns_ at 15.0 "$cbr($i) start"
        $ns_ at 30.0 "$cbr($i) stop"
}

#node movement for wireless node_(0) used with 1 AP
#$ns_ at 10.0 "$node_(0) setdest 90.0 90.0 10.0"

#**********record procedure**************
$ns_ at 0.0 "record"

proc record {} {
global opt sink t l d timesent lpktsrec

set ns [Simulator instance]

set time 0.2 ; #set Sampling time to 0.2

for {set i 0} {$i < $opt(nn)} {incr i} {
        set bw($i) [$sink($i)  set bytes_]
```

```
        set bwl($i) [$sink($i) set nlost_]


        set lpkttime($i) [$sink($i) set lastPktTime_]
        set pktsrec($i) [$sink($i) set npkts_]
}


set now [$ns now]

#record
for {set i 0} {$i < $opt(nn)} {incr i} {
        puts $t($i) "$now [expr $bw($i)/$time*8/1000000]" ; #record throughput
        puts $l($i) "$now [expr $bwl($i)/$time]" ; #packet lost
# Record Packet Delay
        if { $pktsrec($i) > $lpktsrec($i) } {
        puts $d($i) "$now [expr ($lpkttime($i) - $timesent($i))/($pktsrec($i) - $lpktsrec($i))]"
        } else {
        puts $d($i) "$now [expr ($pktsrec($i) - $lpktsrec($i))]" ;# if no pkts received this should
be zero
        }
}


#reset the values on the traffic sinks
for {set i 0} {$i < $opt(nn)} {incr i} {
        $sink($i) set bytes_ 0
        $sink($i) set nlost_ 0
        $sink($i) set npkts_ 0

        set timesent($i) $lpkttime($i)
        set lpktsrec($i) $pktsrec($i)
}


$ns at [expr $now+$time] "record"
```

```
}

# Define initial node position in nam

for {set i 0} {$i < $opt(nn)} {incr i} {

    # 10 defines the node size in nam, must adjust it according to your
    # scenario
    # The function must be called after mobility model is defined

    $ns_ initial_node_pos $node_($i) 10
}

# Tell all nodes when the simulation ends
for {set i } {$i < $opt(nn) } {incr i} {
    $ns_ at $opt(stop) "$node_($i) reset";
}

$ns_ at $opt(stop) "$ns_ nam-end-wireless $opt(stop)"
$ns_ at $opt(stop) "stop"
$ns_ at $opt(stop) "puts \"NS EXITING...\" ; $ns_ halt"

#**********stop procedure**************
proc stop {} {
  global ns_ tracefd namtrace t l d opt
   $ns_ flush-trace
   close $tracefd
   close $namtrace

#close trace files
for {set i 0} {$i < $opt(nn) } {incr i} {
```

```
        close $t($i)

        close $l($i)

        close $d($i)

}

exit 0

}

$ns_ run
```

MATLAB Code

Plot all Trace Files:

```matlab
i = 20; %Number of (nodes-1) to graph
graphs = {'throughput', 'lost', 'delay'}; %graph names
for g = 1:length(graphs)

    figure
    Legend = cell(i+1,1)
    for c = 0:i
        filename = [graphs{g} num2str(c) '.tr'];
        Legend{c+1} = strcat(filename);
        A = importdata(filename);
        length(A(:,2))
        %plot(A(:,1),M(2)*ones(length(A(:,2))))%Plot average of results
        plot(A(:,1),A(:,2)) %Plot results
         if g == 1
            title('Throughput vs Time')
            ylabel('Throughput (Mbps)')
        elseif g == 2
            title('Packets Lost vs Time')
            ylabel('Packets Lost')
        elseif g == 3
            title('Delay vs Time')
            ylabel('Delay (s)')
        end
        xlabel('Time (s)')
        hold on
    end
    legend(Legend)
    hold off
end
```

Plot Average Trace Files:

```matlab
i = 9; %Number of (nodes-1) to graph
graphs = {'throughput', 'lost', 'delay'}; %graph names
for g = 1:length(graphs)
    mean = zeros(1,225)
    data = zeros(1,225)
    figure
```

30

```matlab
        Legend = cell(i+1,1)
        for c = 0:i
             filename = [graphs{g} num2str(c) '.tr'];
           % Legend{c+1} = strcat(filename);
            A = importdata(filename);
            mean = mean(:) + A(:,2)

            %plot(A(:,1),M(2)*ones(length(A(:,2))))%Plot average of results
            %plot(A(:,1),A(:,2)) %Plot results

        end
        data(1:51) = mean(1:51)/4    %number of nodes in first time interval
        data(52:76) = mean(52:76)/7  %number of nodes in second time interval
        data(77:225) = mean(77:225)/10 %number of nodes in last time interval
        plot(A(:,1),data)
        if g == 1
                title('Average Throughput vs Time')
                ylabel('Throughput (Mbps)')
            elseif g == 2
                title('Packets Lost vs Time')
                ylabel('Packets Lost')
            elseif g == 3
                title('Delay vs Time')
                ylabel('Delay (s)')
            end
            xlabel('Time (s)')
        %legend(Legend)
        hold off
end


Plot Jitter for Averages:
i = 9; %Number of (nodes-1) to graph
graphs = {'delay'}; %graph names
max_delay = zeros(1,225)
B = ones(1,225)
jitter = zeros(1,225)

jitter = zeros(1,225)
figure
Legend = cell(i+1,1)
for c = 0:i
    filename = ['delay' num2str(c) '.tr'];
    A = importdata(filename);
    max_delay = max(max_delay,A(:,2));
    clear min
    B = min(B,A(:,2));
end
jitter = max_delay - B
plot(A(:,1),jitter)
title('Jitter vs Time')
ylabel('Jitter (s)')
xlabel('Time (s)')
%legend(Legend)

hold off
```