PERFORMANCE EVALUATION OF TCP OVER IEEE 802.11 WLANs

BY

SACHIN GOEL

B.E., SIKKIM MANIPAL INSITITUTE OF TECHNOLOGY (Sikkim), India (2001)

THESIS

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Master of Science

in

Computer Science

September 2006

This thesis has been examined and approved.

_____

Thesis Director, Dr. Radim Bartos
Associate Professor of Computer Science

_____

Dr. Elizabeth Varki
Associate Professor of Computer Science

_____

Mr. Benjamin Schultz
Managing Engineer, InterOperability Laboratory,
Research Computing Center

_____

Mr. Ankur Chadda
Member Technical Staff, Global Services,
Spirent Communications

_____

Date

DEDICATION

To my beloved *muskan* - *Deepti*.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Prof. Radim Bartos whose invaluable guidance has enabled me to complete this thesis. It has been very interesting and exciting working with him. I would also like to thank Mr. Benjamin Schultz who had been supportive right from the very first day of this work and enlightened me with new ideas and approaches. He took deliberate pain to peruse my work time and again and had constantly given me his feedback and opinions. I would like to thank Prof. Elizabeth Varki for being very encouraging and motivating. I would like to thank Mr. Ankur Chadda for assisting me with my doubts and providing invaluable feedback.

I am grateful to Dale Williams (Cisco Systems), Robert Levay (Ixia) and Jason Nutt (Anue Systems) for resolving my technical issues. I would also like to extend my thanks to Eric Ely, Nathan Bourgoine, James Swan and Lincoln Lavoie for helping me out with Azimuth, Rohde & Schwarz and Anue Systems and have been of great help. I am also grateful to the Wireless and VoX Consortium, InterOperability Laboratory for providing me the resources and supporting me in my work.

Finally, to my friends Ajay, Arpita, Sayantan and Vrushali who have always been supportive and encouraging to me in my darkest hours. Thank you for reminding me that success was just around the corner.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

PERFORMANCE EVALUATION OF TCP OVER IEEE 802.11 WLANs

by

Sachin Goel

University of New Hampshire, September 2006

Transmission Control Protocol (TCP) is a communication protocol that is used to provide reliable data delivery between hosts. As TCP is the most highly used transport-layer protocol, many have worked on addressing the issue of performance. Performance issues have been studied in various environments, especially when using 802.11 Wireless Local Area Networks (WLANs). Wireless networks are prone to a higher number of packets loss and corruption. 802.11 WLANs have an equivalently fast acknowledgement mechanism as TCP to ensure reliability of traffic over it. This duplication of functionality between TCP and 802.11 WLAN creates unexpected behaviors that can result in high costs in terms of overall performance. A significant amount of analytical and simulation work has been done in the past to study the behaviour of TCP over 802.11 WLANs. The main contribution of this work is the analysis of TCP interaction in an 802.11 WLAN topology by using real commercial-grade equipments.

A testing methodology is designed to do the quantitative performance evaluation in a network topology consisted of wired as well as a wireless connection. The

methodology contains test scenarios with different configurable settings on an Access Point (AP) and various controlled impairments in the network topology such as latency, packet drop, noise interference, etc. The performance of TCP is measured in terms of the throughput.

This work provides a comprehensive set of experiments to study the behaviour of TCP over 802.11 WLANs. The results can provide insight into the performance cost associated with TCP traffic on 802.11 WLANs under different network environments and configurations on the AP. The results of this work thus have a value to equipment manufacturers and network operators.

CHAPTER 1

INTRODUCTION

Internet traffic in a network is delivered end-to-end between hosts by Internet protocol (IP). IP is a connectionless protocol and therefore does not guarantee reliable delivery of data. TCP resides at the transport layer [8] in the Open System Interconnection (OSI) model. TCP is a connection-oriented protocol and thus ensures the reliable delivery of data between the hosts. TCP also implicitly assumes that the underlying layers do not participate in the reliable delivery of data. The situation can create a problem when TCP traverses on 802.11 WLAN network as the 802.11 data link layer also guarantees reliable delivery of data. This thesis analyzes the performance of TCP over 802.11 WLANs through practical testing to better understand this protocol interaction.

1.1 802.11 Wireless Local Area Network

A WLAN is a data transmission system that has the ability to provide location independent network access between communication devices. It uses high frequency radio waves for communication and operates in the unlicensed Federal Communications

Commission (FCC) 2.4 GHz and 5 GHz Industrial, Scientific, and Medical (ISM) frequency bands. The Institute of Electrical and Electronics Engineer (IEEE) 802 committee wrote the IEEE 802.11 standard that specifies the 802.11 MAC layer protocols. IEEE 802.11 standard specifies Medium Access Control (MAC) and Physical (PHY) layer functionality for fixed as well as mobile devices and defines Basic Service Set (BSS) as the building block of an 802.11 WLAN that consists of any number of 802.11 stations (STAs). IEEE 802.11 also specifies two types of network topologies for WLANs.

1. *Independent Basic Service Set* (IBSS): This is also called ad-hoc network and consists of at least two wireless devices, which communicate with each other directly in a BSS as shown in Fig. 1. These devices should be in the range of each other in order to communicate.

Figure 1. Independent Basic Service Set Topology.

2. *Infrastructure Basic Service Set*: In this communication mode, the wireless STA must be associated with the Access Point (AP) before communication between devices can occur. The AP may also provide communication of these STAs with devices present in the distributed system (DS), i.e., Ethernet networks as shown in Fig. 2. This type of topology is meant to cover a large network area and is the most commonly used in practice.

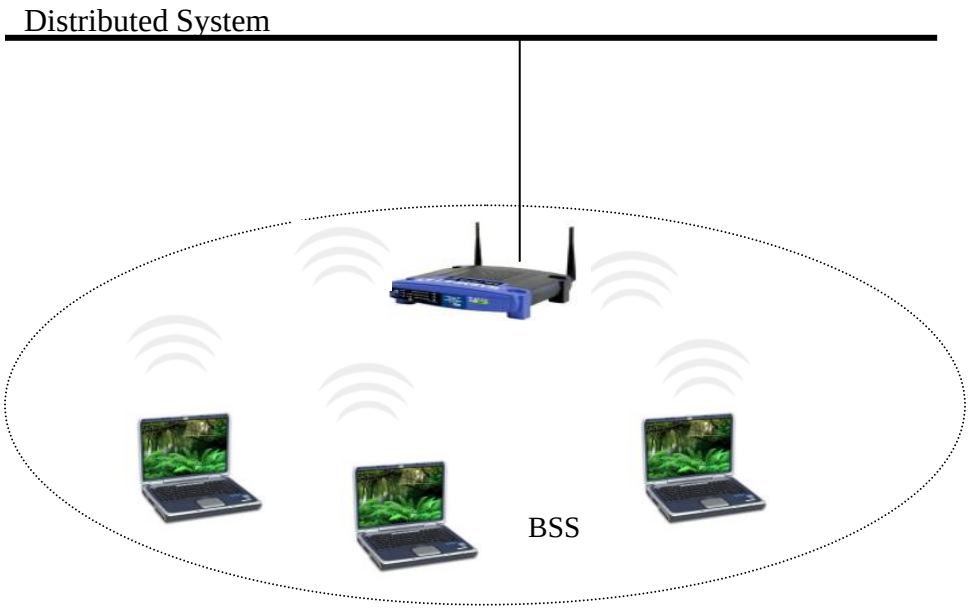

Figure 2. Infrastructure Basic Service Set Topology.

- A STA is any device that conforms to the 802.11 protocols, i.e., contains the functionality of PHY layer, MAC layer and an interface to the wireless medium [7].
- An AP is a STA and is used to route traffic from a wired to wireless network and vice versa.

The MAC layer resides in the lower half of the data link layer and provides some of the important functionalities to the upper half link layer such as:

- Layer 2 Addressing,
- Access Co-ordination,
- Frame Check Sequence,
- Recognition of frames.

A *radio frequency* (RF) wireless channel is characterized by high *bit error rate* (BER) that is defined as the ratio of the erroneous bits to the total bits transmitted. The primary reasons for this behavior are channel fading, interference from other sources operating in the same unlicensed band and/or mobility of users.

1.2 Transmission Control Protocol

TCP is a communication protocol that ensures reliable delivery of data between hosts. It was designed for Department of Defense (DoD) [9] in early 1980's for traditional networks comprised of wired links and fixed hosts as shown in Fig. 3. The basic concept of TCP is that the two host systems first initialize a connection and then start communicating with each other. When communication is completed, a formal close process terminates the connection.

During data exchange, TCP uses its own recovery mechanism to prevent the loss of packets. The TCP sender assumes a packet loss when it does not receive an

acknowledgement for a data packet from the TCP receiver within a timer based timeout interval.

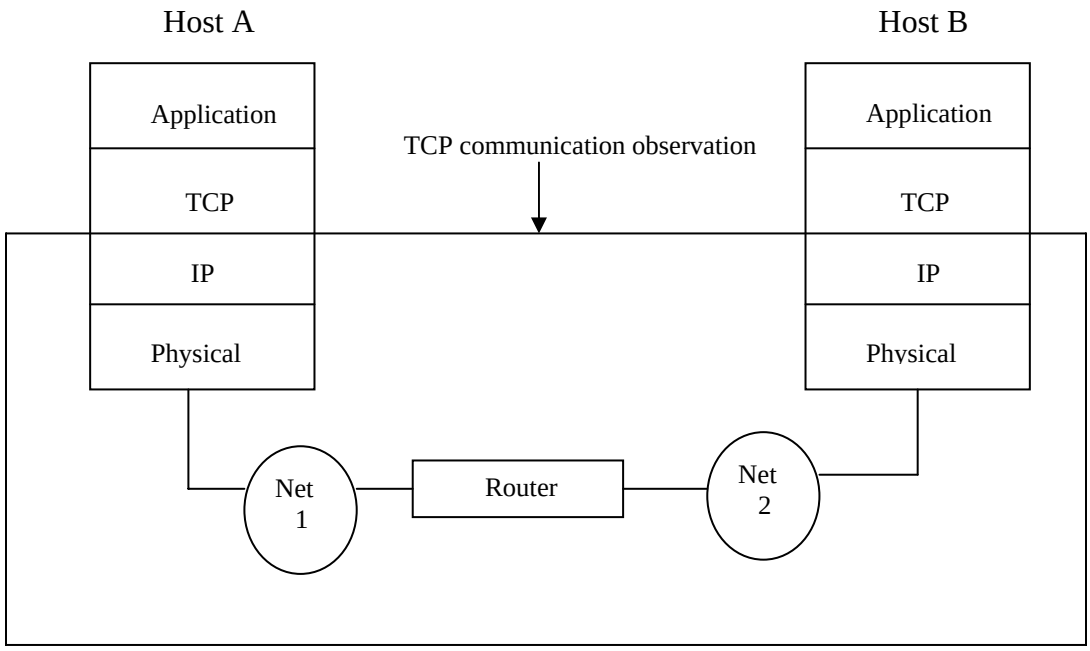

Figure 3. TCP Communication.

TCP was designed by considering some of the requirements as shown in Table 1. from the wired network in between the end hosts which are not met over a WLAN.

TCP requirements	**WLAN services**
Dedicated Access of Media	Shared Media
Full Duplex	Half Duplex
Low BER	High BER

Table 1. Characteristics of TCP and WLAN.

The study of TCP performance is of current interest due to the duplication of "connection-oriented" protocols on the same network. Both TCP and 802.11 MAC layer would retransmit the lost packet if they did not receive the proper acknowledgement. This leads to unnecessary TCP retransmissions and inefficient bandwidth utilization of the network.

1.3 Channel Access Mechanism in 802.11

The basic access method in 802.11 networks is the *Distributed Coordination Function* (DCF) in which 802.11 MAC layer uses the Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) mechanism before transmitting any frame in the air. This mechanism enables the 802.11 compliant devices to listen on the channel before transmission in order to reduce collisions. A STA will transmit in the air only if the channel is free for duration greater than or equal to *DCF Inter Frame Space* (DIFS). If the medium is busy, the STA shall follow these steps:

- Wait until the medium is clear.
- If the medium remains idle for a DIFS period of time, the STA shall perform a random backoff interval counter and starts decrementing it while the medium is clear.
- If the backoff counter reaches zero, the STA starts transmitting on the channel.
- If the medium becomes busy while the STA is decrementing the counter, the backoff counter is paused.

- Once the channel becomes clear again, the STA continues decrementing the counter from the previous value and transmits if the counter becomes zero.

The DCF method requires an acknowledgement for every packet from the receiver and the time interval between the reception of a packet and the transmission of its acknowledgement is separated by a *Short Inter Frame Space* (SIFS) period in 802.11 networks.

1.4 Error Recovery Mechanism of TCP & WLAN

The packet loss in wired networks is primarily caused by network congestion. TCP has a recovery mechanism to deal with packet losses. The TCP sender receives a cumulative acknowledgement from the receiver to determine which packets have reached the destination and which did not. If a TCP sender receives several duplicate cumulative acknowledgements or no acknowledgment at all for a packet from the receiver, the sender assumes that packet to be lost because of congestion. The TCP sender initiates its recovery procedure by reducing the transmissions of the packets by lowering the value of the congestion window and initiating its congestion control and avoidance mechanisms.

The error recovery mechanism in WLAN is much different from traditional wired networks. 802.11 networks have fast acknowledgement mechanism and the 802.11 MAC expects an 802.11 positive acknowledgment for every unicast packet sent to the destination. It is called *Automatic Repeat reQuest* (ARQ) [14] and is initiated by the STA

that started the communication. The source STA performs random backoff and waits for a random amount of time before contending for the channel again if an acknowledgement is not received from the receiver STA. The MAC coordination function in the MAC layer has the responsibility to retransmit the lost packet for a specified number of times for a packet loss. A WLAN is considered as a lossy network because of high probability of interference from other wireless STAs present in the neighborhood and operating in the same frequency band, channel fading due to mobility of users and multipath fading.

Therefore, when an application using TCP as its underlying protocol is subjected to a WLAN and a packet loss happens, it becomes difficult to identify if it was caused by a wireless or wired network. When an 802.11 packet carrying a TCP segment is lost, the 802.11 MAC layer will try to retransmit the packet. If an 802.11 link is experiencing a high packet loss, then the probability of the time taken by an 802.11 link to deliver the same packet may exceed the round trip time (RTT) of the TCP endpoints and cause the TCP source to resend the data packet again. This results in the unnecessary duplication of data packets on the TCP receiver side. This whole thing happens because the TCP structure has been based on the notion that the link layer will drop the packets and not delay it when congestion occurs. 802.11-link layer, however, delays the delivery of data packets instead of dropping them. Therefore, TCP performance in WLANs suffers considerable throughput degradation, even when there is a wireless packet loss. TCP assumes that loss was from network congestion and unnecessarily reduces its congestion window. Simultaneously, 802.11 uses its own local recovery mechanism and retransmits

the same data packets again. The main issue is that there is no way to explicitly inform the TCP source about the reason of packet loss in the network.

The experiments presented in this thesis are performed in the Infrastructure basic service set and in 2.4 GHz frequency band. A typical open-air experiment setup is as shown in Fig. 4.

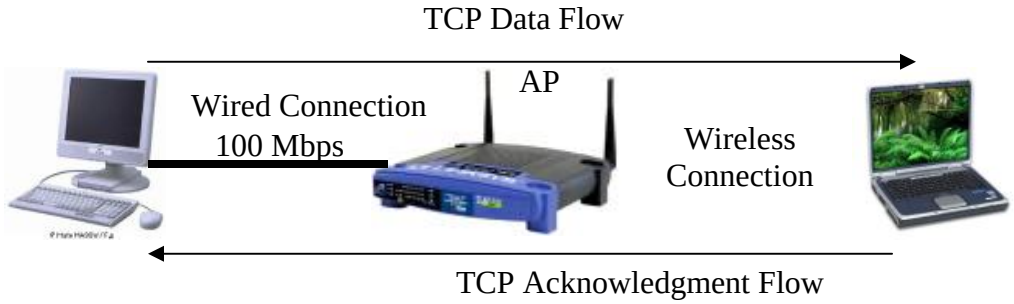

Figure 4. Typical TCP Over Wireless Experimental Setup.

1.5 Fragmentation in 802.11 WLANs

802.11 networks have a peculiar feature to send fragment burst in the air based on a *fragmentation threshold* metric on the STA. A STA fragments a MAC Service Data Unit (MSDU) into MAC Protocol Data Units (MPDUs) based on the *fragmentation threshold*. The MPDUs are then transmitted to the destination in a fragment burst. Each fragment consists of a MAC layer header, *frame check sequence* (FCS), and data payload as shown in Fig. 5. Each MPDU consists of the same sequence number as the MSDU but different fragment numbers in order to be distinguished at the receiver side. It is the responsibility of the recipient STA to defragment the MPDUs into an MSDU. An MSDU is transmitted

successfully only, if all the MPDU's are received by the destination STA. If an MPDU is unable to reach the recipient STA inspite of being retransmitted its maximum number of retransmissions, then the MSDU is discarded by the STA along with the remaining MPDUs.

802.11 networks have fast acknowledgement mechanism and each fragment expects an acknowledgement from the destination receiver. The MPDUs and their acknowledgements are separated by a SIFS interval that is 10 µsec in case of 802.11b networks. A source STA releases the channel after the transmission of first fragment. The source STA then waits immediately to listen for the acknowledgement. If the source STA receives a positive acknowledgement, then it will transmit the next fragment after a SIFS interval. The source STA looses control of the channel if it did not receive an acknowledgment back and contends for the channel again to transmit the pending fragments.

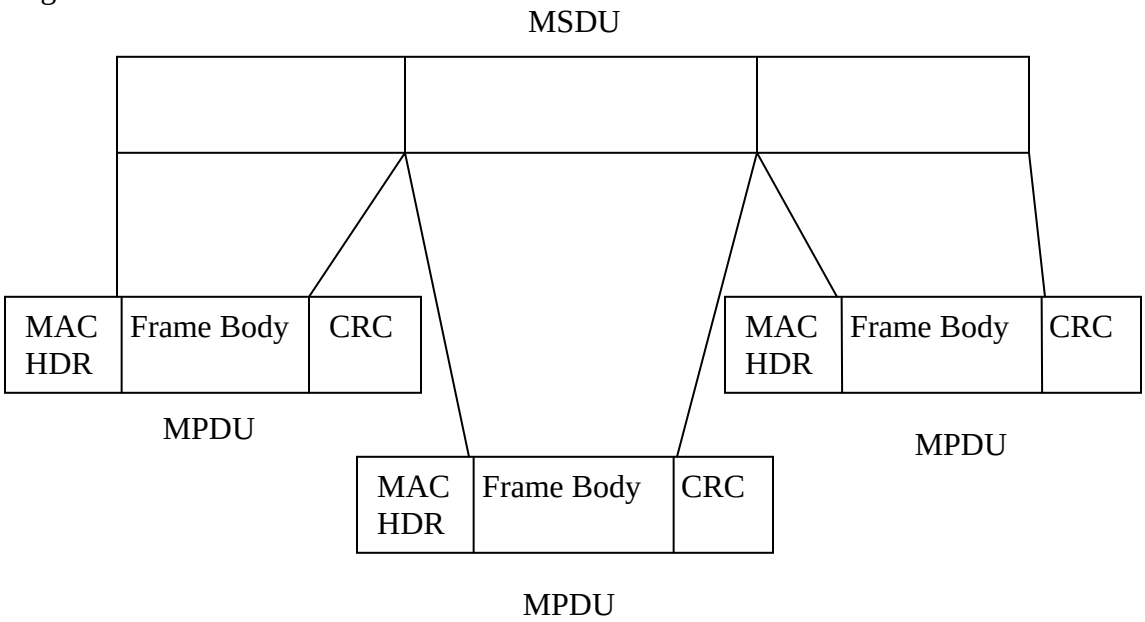

Figure 5. Fragmentation in 802.11 Networks.

1.6 Related Work

A considerable amount of work has been done in the past to evaluate the performance of TCP traffic over WLANs. Simulations and analytical models [17] were used for many of these experiments. Various alternatives have been proposed to enhance the performance such as TCP TULIP [1], Explicit Loss Notification [3], and Snoop Protocols [4]. All of the previous works were directed to propose changes in the TCP protocol structure or to provide cross layer approach, i.e., ILC-TCP [2]. Changes to any protocol structure involve a long, resource-intensive process in working with the standards bodies and their associated industry interests. None of these analyses provided an enhanced understanding of existing TCP problems and practical, configuration-based solutions.

The primary contribution of this thesis is that all experiments are implemented in a real network environment with real clients and traffic. The performance of TCP over WLANs has been evaluated in accordance with different AP configuration settings and other network topology impairment metrics such as latency, packet loss, duplication, reordering, bit errors, and noise interferences. A Local Area Network (LAN) emulator and vector signal generator are used to create different types of wired and wireless network impairments in order to study the behavior of TCP over WLANs in more detail. The work presented in this thesis gives a better understanding of the performance of TCP over WLANs and its relative performance degradation in different network environments.

1.7 Motivation

This work is driven by the following reasons and goals.

1 Known issues with network behavior and performance in 802.11 networks.

2 To understand and analyze the dynamics of TCP in wireless LAN environment when protocol functionalities are duplicated.

3 To discover practical solutions to this issue through testing and configuration of real equipment.

1.8 Challenges

This work involved significant challenges because of the nature of experiments being carried out with real devices. The first challenge was to ensure that the experiments were reproducible. To solve this, the experiments were run in an RF isolation chamber to maximize the reduction of external interference. The second challenge was to study the various devices used for the experiment. It took a considerable amount of time to learn how to operate, configure and understand supported features of each device. . The features provided by different devices were first tested in order to ensure proper functionality. A significant amount time was spent to understand the proper configuration. Some of the devices were found to have bugs in their firmware that affected the automation scripts used to run the experiments. All such issues were communicated to the participant companies and appropriate actions were taken to rectify the issues such as a firmware or model upgrade.

1.9 Summary

The remaining chapters deal with the following subject areas:

Chapter 2 contains the description of all the experiments that have been considered to test the TCP performance. It explains all the various metrics that are important to analyze the performance and their significance.

Chapter 3 explains the experiment set up and the configuration of devices.

Chapter 4 contains the experiment section with the results and conclusions.

Chapter 5 summarizes the work. It also lists the possibility of extending this work and outlines some of the future experiments that could be performed on the same subject.

CHAPTER 2

TYPES OF EXPERIMENTS

2.1 Objective

The goal of this thesis is to do the quantitative evaluation of the performance of TCP over 802.11 WLANs. Several planning steps are important to identify the types of experiments to test performance. The experiments are designed based on the metrics on devices as well as network environment that may affect the performance of TCP.

2.2 Identification of Performance Metrics

The metrics for the experiments are classified into two types:

- Optional configuration parameters on an Access Point.

  This includes the parameters that are present on an access point to

  1. Provide variable 802.11 fragment sizes.
  2. Provide protection mechanism to reduce collisions from other devices in the same frequency band.

- Network configuration parameters (wired and wireless).

This includes configuration of the network and is not device specific. It emulates different network conditions such as variable latency, packet drop, packet duplication, packet reordering, bit errors, channel fading, and noise interference.

2.2.1 Access Point Metrics

The AP provides a large number of optional configuration parameters for a network administrator or a user. The usage of such parameters is test specific and user dependent. The parameters chosen in this thesis are listed in the following subsections:

2.2.1.1 Fragmentation Threshold

This experimental parameter defines the maximum size of an 802.11 frame. It has a range from 256 bytes to 2346 bytes. Fragmentation threshold can be configured at two places:

- At the Access Point: The AP does not forward a frame or fragment to a STA on the wireless side with fragment length greater than the fragmentation threshold. The packet size is always less than or equal to the fragmentation threshold depending upon the original size of the packet.
- At the STA (radio based NIC): Similarly the wireless STA does not transmit a frame with packet size greater than the fragmentation threshold.

An 802.11 MAC layer fragments the MSDU destined for a unicast address based on the *fragmentation threshold* value configured on that device.

For example, an AP with fragmentation threshold set, as 512 bytes will fragment a packet of size 1000 bytes into two before forwarding it on the wireless network. The TCP data packets flow from the wired network to the wireless network and therefore *fragmentation threshold* has been configured on the AP only while STA's have a default configuration of the highest default fragmentation value. The reason to choose *fragmentation threshold* as an experimental parameter is to see the theoretical advantages and disadvantages of smaller and bigger 802.11 fragments on the dynamics of TCP in different network conditions.

2.2.1.2 Request to Send Threshold

Request to Send (RTS) threshold is used as one of the protection mechanisms by 802.11 compliant devices. It can also be configured at the AP and STA and is primarily used for two main reasons.

A. Solve hidden node problem:

In the hidden node problem, a STA is able to communicate with the AP but is invisible to another STA in the Basic Service Set (BSS) [6], which is also associated with the AP. Fig. 6 shows the hidden node problem. STA's A and E can hear each other and

communicate with the AP. Similarly, STA's C and D can listen to each other and AP but are invisible to STA's A and E. In this situation, the chances of collision increase because the STAs like A, E and C, D are not aware of each other's presence. RTS mechanism is used so that any STA, whose data packet size is greater than the RTS threshold size will first ask permission to send the packet. The AP then responds with a Clear to Send (CTS) frame that is listened to by all the STAs present in the BSS. This RTS-CTS exchange enables other hidden nodes to not send any data in such cases, thereby, minimizing collision.

In a similar way, when the data packet size to be transmitted by the AP is greater than the RTS threshold on it, it too sends the RTS frame in order to minimize collision. Though RTS/CTS involves a greater amount of overhead because of the extra number of frame exchanges, it is a good mechanism to reduce the chances of collisions between hidden STA's.

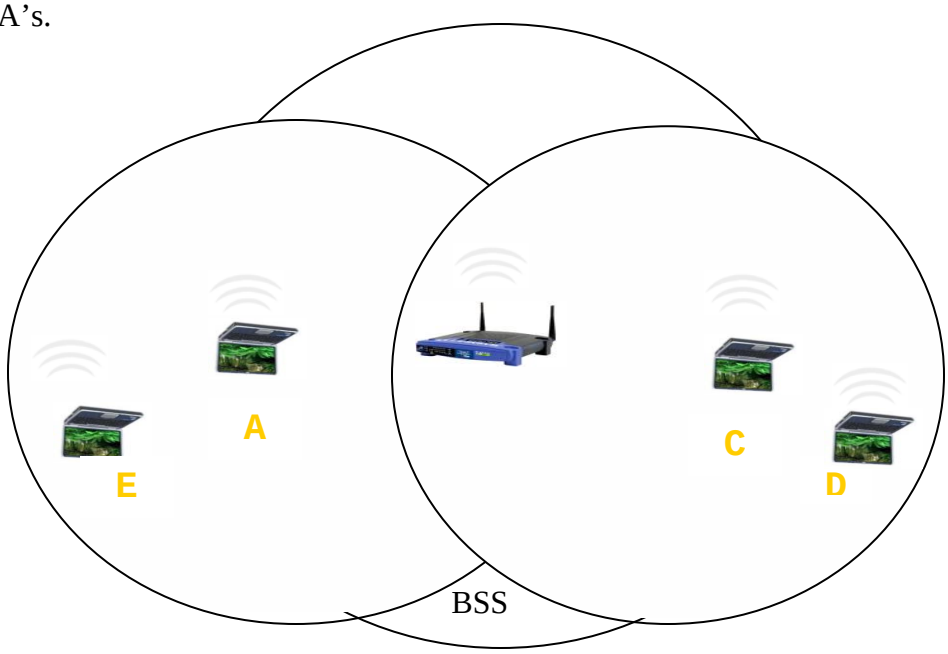

Figure 6. Hidden Node Problem.

B. To reserve the wireless medium before transmission.

RTS/CTS frame contains duration field, which is seen by the STA's to update their network allocation vector (NAV). NAV is used by the STAs to maintain the future traffic on the medium and is based on the duration field value of the frames. This is termed as virtual carrier sense (CS) mechanism in WLANs and is used to determine the state of the medium before any STA starts transmission. When an STA listens to an RTS/CTS, it will update its NAV and not send any data upto the duration of its NAV. This will help the initiator STA to reserve the medium before the actual transmission of data.

This has been chosen as one of the experimental parameters because RTS mechanism could influence the performance of TCP over WLANs because of the extra time and overhead involved to perform RTS-CTS exchange. Simultaneously, it could improve the performance in a scenario in the presence of hidden stations. Therefore, the performance cost associated with this parameter has been analyzed in the experiments.

2.2.2 Non - Access Point Metrics

Non-access point metrics are not device specific. These metrics cause network impairments to conduct experiments in different network conditions. These metrics are discussed in the following sections.

2.2.2.1 Variable Signal Strength

The transmission of packets between two STAs is dependent on the strength of the power signal between them. When a station moves away from an AP, the signal received by it decreases proportionally and the tendency to not receive packets or corruption of packets increases because of channel fading. This is exactly what happens in a hand off, when a station moves out of the BSS of one AP and may enter the BSS of another AP. Because of low signal strength and higher observation of packet loss, the 802.11 MAC layer would try to retransmit the lost packets at a lower transmission rate because of its dynamic rate adaptive algorithm. The 802.11 PHY layer has the ability to decrease the transmission rate to reduce bit error rates in the presence of high interference or low Signal to Noise ratio (SNR). 802.11b networks start transmitting the data frames at its highest possible rate, i.e., 11 Mbps, and then in the presence of bit errors or low SNR, reduces its speed to 5.5 Mbps, 2 Mbps, and 1 Mbps. This transmission rate jumps to a higher rate with low bit error rate or high SNR. The direct affect of this would be the triggering of the congestion control algorithms on the TCP host side. Consequently, the 802.11 link becomes the bottleneck in an end-to-end TCP connection because of variable transmission rate. The TCP congestion control algorithm and window management behavior in such a scenario becomes the point of interest. The 802.11 link gets fatter and skinnier for longer or shorter period of time because of rate adaptation and this affects the round trip time of the TCP algorithm. The performance of TCP in such a scenario is a motivation for this experiment.

This experiment is performed to observe the performance of TCP with varying fragment sizes on the link layer. The reduction in the transmit rate over the 802.11 PHY layer will increase the time to get back the TCP acknowledgements by the TCP host. This may result in a decreased TCP window on the TCP host size, which will result in the transmission of less data. 802.11 fragments of different sizes utilize different amounts of bandwidth and would take different times to get transmitted with varying 802.11-link speed. This experiment will analyze the affect this has on the TCP congestion algorithms and the performance on different 802.11 fragment sizes.

2.2.2.2 Packet Duplication

A duplicate packet is of two types in a network consisting of wired as well as 802.11 link: TCP duplicate and 802.11 duplicate packet.

1) TCP duplicate packet

A TCP duplicate packet is one, which has the same sequence number and expected acknowledgement number as its precedent. TCP protocols send a sequence number with each acknowledgement in order to avoid confusion between duplicate acknowledgments and new acknowledgments with the correct sequence packets. The valid range to retransmit a TCP data without receiving an acknowledgement is 0 – 4294967295 (decimal) before retransmission timeout. The default value of maximum retransmission in Windows XP and 2000 TCP/IP stack is 5 [16]. A TCP duplicate packet can be of two types.

a) Duplicate TCP data packet

b) Duplicate TCP acknowledgement packet

The experiments have been designed to create duplicate TCP data as well as TCP acknowledgements on the wired side in various ranges as described below. The duplication of TCP acknowledgements is categorized into two types.

- Duplicate TCP acknowledgement with range 0 – 2.

  This experiment creates at most two duplicate packets.

- Duplicate TCP acknowledgement with range 0 – 4.

  This experiment creates at most four duplicate acknowledgements (DupAcks). When a TCP source receives more than two duplicate acknowledgements without the arrival of any intervening packets, the TCP source invokes congestion recovery algorithm. The TCP source uses fast retransmission mechanism and sends the implied lost TCP data packet without waiting for the retransmission timer to expire [15].

2) 802.11 duplicate packet

An 802.11 duplicate packet is one, which has the same sequence number as its precedent MSDU but with retry bit set in it. The number of 802.11 duplicate or retry packets depend upon *802.11ShortRetryLimit* and *802.11LongRetryLimit* which are defined as:

802.11ShortRetryLimit: is defined as the maximum number of retransmission attempts of a frame for a lost frame on the wireless link before that frame is discarded by an STA. This limit is applicable to frames whose frame size is less than or equal to *dot11RTSThreshold* and has 7 as its default value.

802.11longRetryLimit: is defined as the maximum number of retransmission attempts of a frame for a lost frame on the wireless side before that frame is discarded by an STA. This limit is applicable to frames whose frame size is more than *dot11RTSThreshold* and has 4 as its default value.

dot11RTSThreshold and *dot11FragmentationThreshold* are the IEEE 802.11 management objects, i.e., Management Information Base (MIB) objects that have a predefined value and can be changed with the help of a management tool such as Simple Network Management Protocol (SNMP).

The main reasons to observe duplicate TCP packets in a network are:

- Data Loss

  When TCP data are lost on the way and the TCP sender does not receive an acknowledgment in its acknowledgement timeout interval.

- Acknowledgment (Ack) Loss

When an Ack is lost on the way for a TCP segment and the sender does not receive the acknowledgement in its timeout interval.

- Reordering of Packets

  Reordering causes generation of DupAcks in the network and the sender will transmit a duplicate TCP segment based on the reception of three or more DupAcks consecutively.

TCP when used on 802.11 networks may cause more duplicate packets to be observed in the network because of a higher probability of loss of 802.11 segments that encapsulate TCP data and TCP acknowledgements. Some cases where packet duplication can happen are shown as:

- TCP data loss

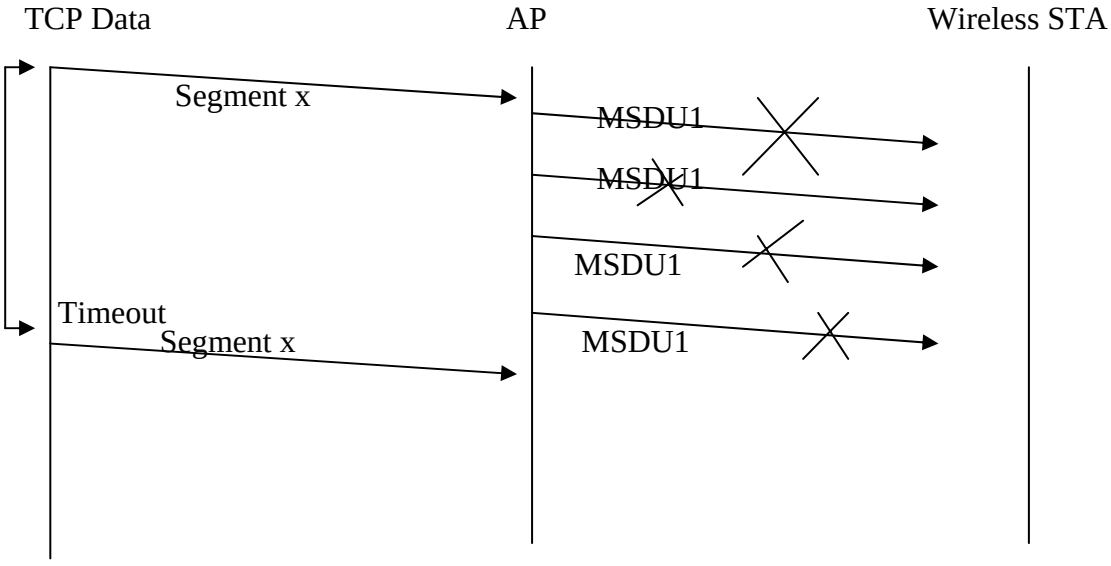

Figure 7. 802.11 Data Loss.

Fig. 7 shows a scenario where there is a packet loss of MSDUs on the wireless side. An AP will assume a transmission failure when it does not receive an acknowledgement for the MSDU sent. 802.11b networks do not reorder the MSDUs normally and forwards packets as and when received.

An AP will try to retransmit an MSDU or MMPU until the dot11ShortRetryLimit or dot11LongRetryLimit based on the MSDU size and the *dot11RTSThreshold*. Because of multirate support in 802.11 networks, the first MSDU will be tried at the maximum possible rate of the 802.11 network, i.e., 11 Mbps and all the retransmissions will be tried at a lower rate such as 5.5, 2 and 1Mbps. When the bandwidth speed changes on the wireless side, there is a possibility that a TCP acknowledgement does not reach in RTT and a retransmission timeout occurs. In this event, retransmissions of the lost segment will be observed in the network along with the TCP congestion control mechanism.

- 802.11 Ack Loss

Fig. 8 shows a scenario where there is a loss in the 802.11 on the wireless side. The AP will not receive an 802.11 Ack back for the MSDU transmitted and retry it until its *802dot11ShortRetryLimit or 802dot11LongRetryLimit*. In this event, MSDUs will retry at a lower rate. There is still a possibility that by the time an MSDU is able to make to the destination and receives its Ack, TCP retransmission timer times out. This would result in unnecessary retransmissions of the lost segment, which in actuality has been received by the TCP host. This negatively affects the bandwidth of the network.

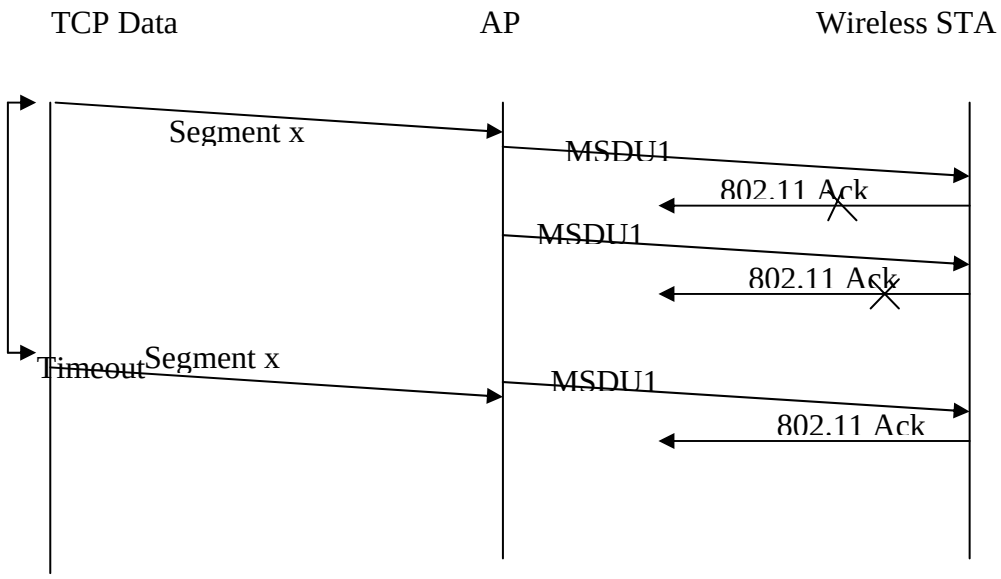

Figure 8. 802.11 Acknowledgement Loss.

- TCP segments arrive in different order

The function of an AP is to forward the packets from the distributed system to a station in a BSS or vice versa. Therefore, if TCP segments arrive in reverse order to the AP, it will just forward them to the destination address. In Fig. 9 segment w, x, y, z, a, b are supposed to arrive in order but segment w finds a different route and other segments arrive before it. The TCP receiver is expecting segment x but finds segment to be in some other order. In this case, TCP receiver immediately issues DupAck [16] for the assumed lost segment. TCP source assumes this duplicate acknowledgement because of a possible reordering and does not perform fast retransmission unless the source receives at least three same duplicate acknowledgements. Fig. 9 shows a case where TCP Ack as well as TCP data segments might get duplicated in the network. The experiments are designed to reflect the cases discussed in Figs. 7 and 9.

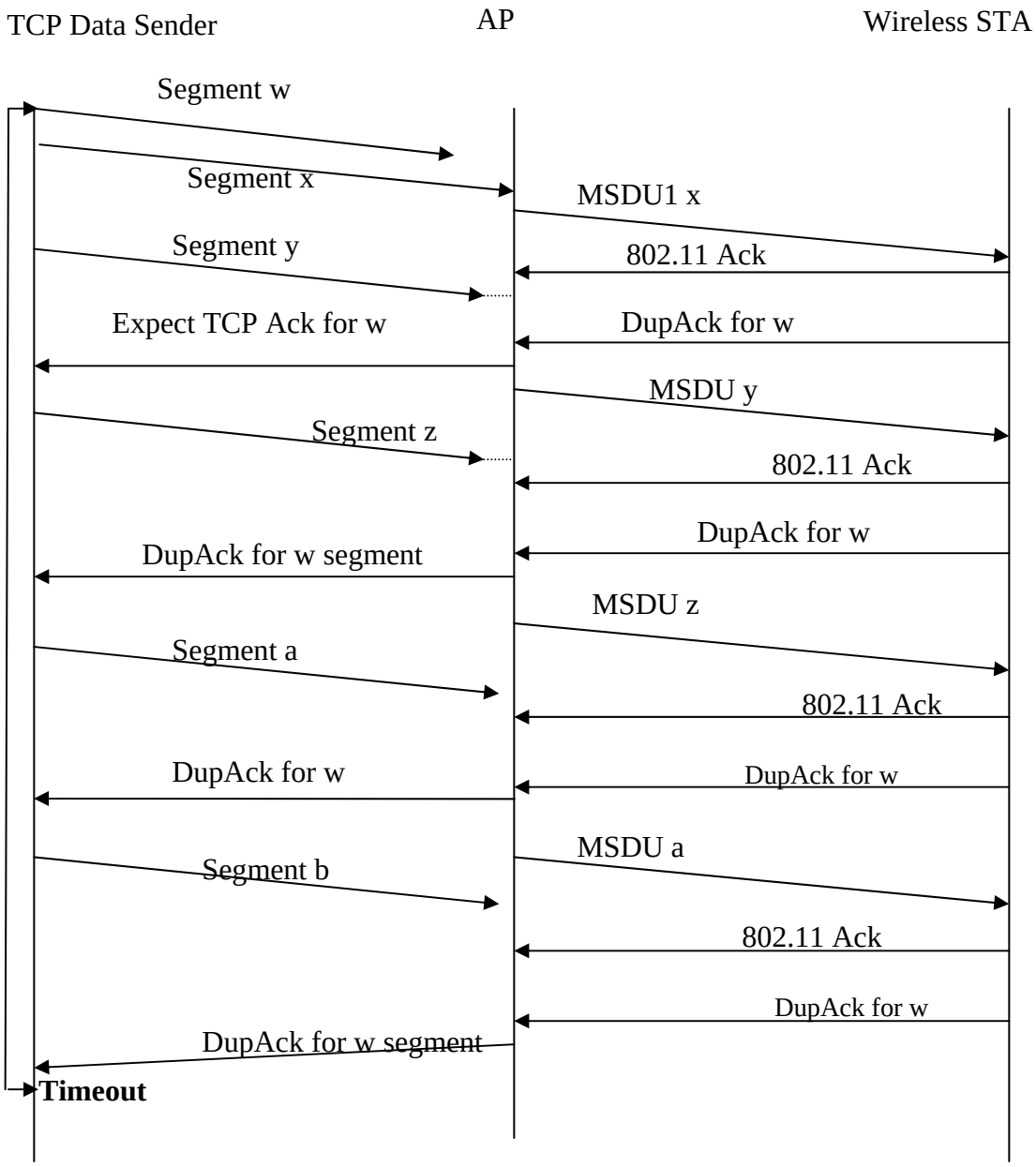

Figure 9. Out of Order Delivery of Packets.

2.2.2.3 Packet Drop

A packet drop occurs when a packet is unable to reach the destination or gets corrupted. In wired networks, this is attributed to congestion and collisions. In 802.11 networks, this

is primarily because of the lossy nature of the 802.11 links. Packet drop triggers congestion prevention and error recovery mechanism by TCP hosts. When a TCP data packet is lost, TCP receiver would not receive the segment it was expecting and issue a DupAck to the TCP sender. TCP host on the reception of more than three DupAcks would trigger fast retransmission in order to process fast recovery of the network.

A TCP source waits for a retransmission timeout period to get an Ack back before retransmitting the data frame. But in a scenario when TCP source receives an Ack for the next higher sequence data packet, the source sends the next higher sequence data and ignores the transmission of frame for which Ack was not received. A packet drop may occur in the following scenarios as shown in Figs. 10, 11, 12, & 13.

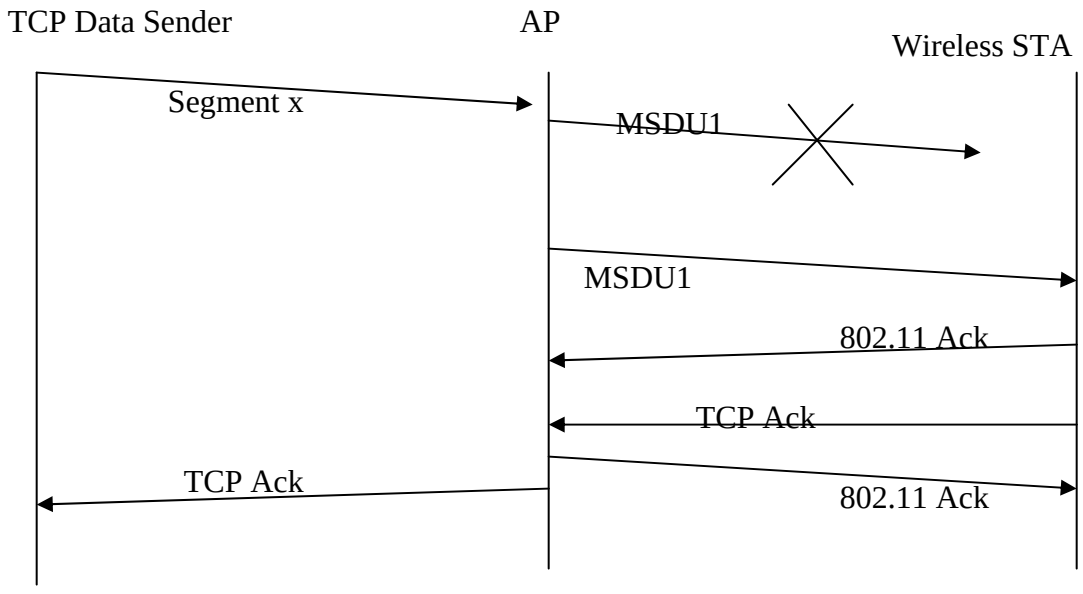

Figure 10. Lost MSDU.

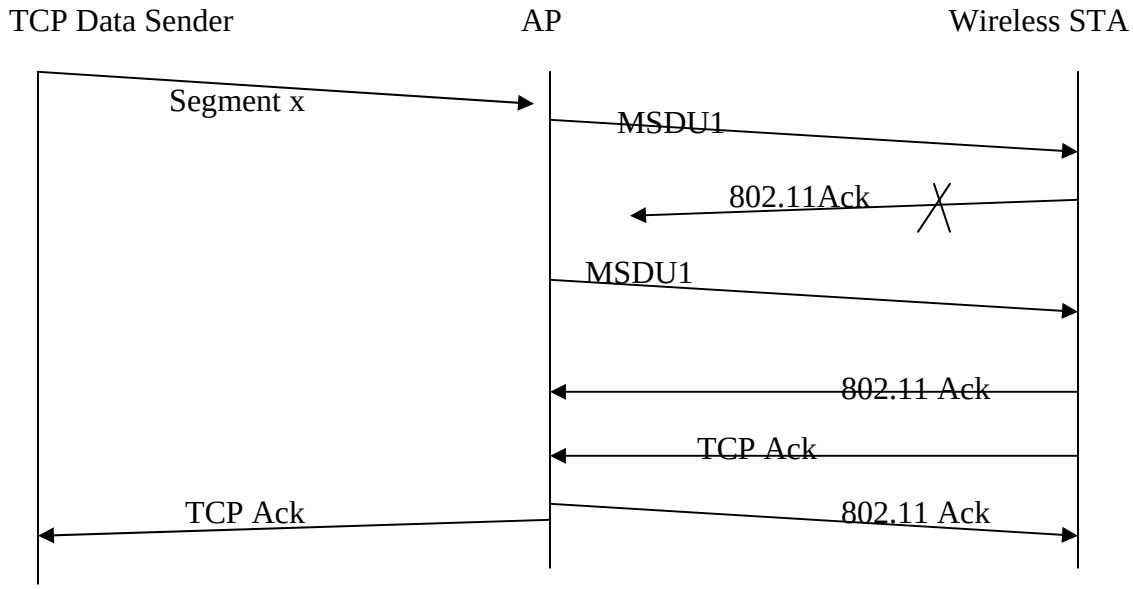

Figure 11. 802.11 Acknowledgement Drop.

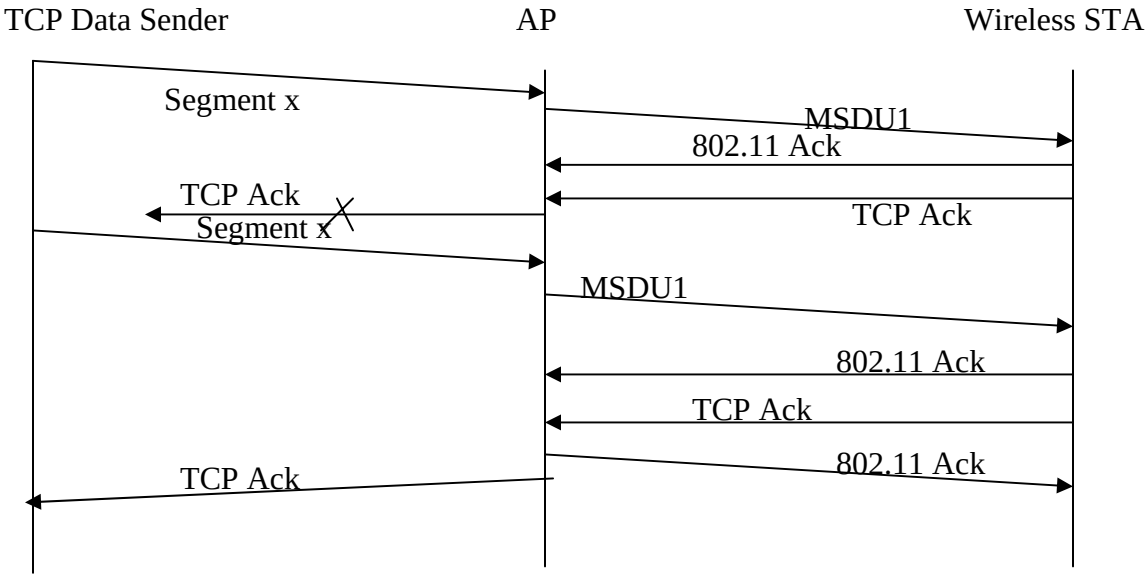

Figure 12. TCP Acknowledgement Drop.

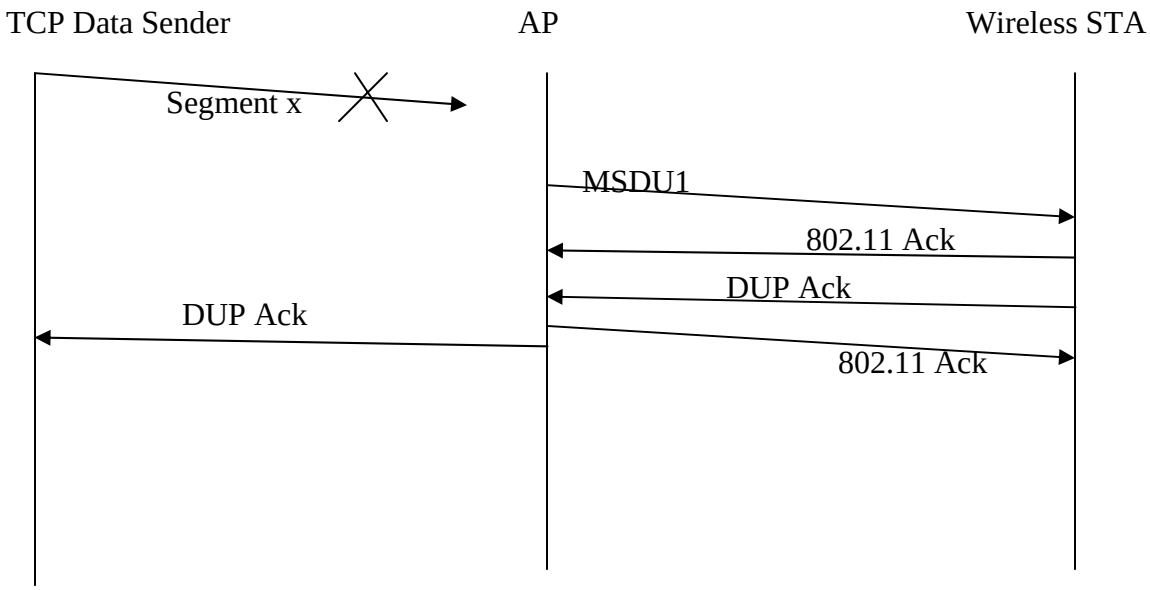

Figure 13. TCP Data Drop.

The experiments are designed to cover the cases of Fig. 12 and Fig. 13 because of the controlled network impairment on the wired side. A typical representation of a TCP Ack loss in the experiment setup using a LAN emulator is shown in Fig. 14.

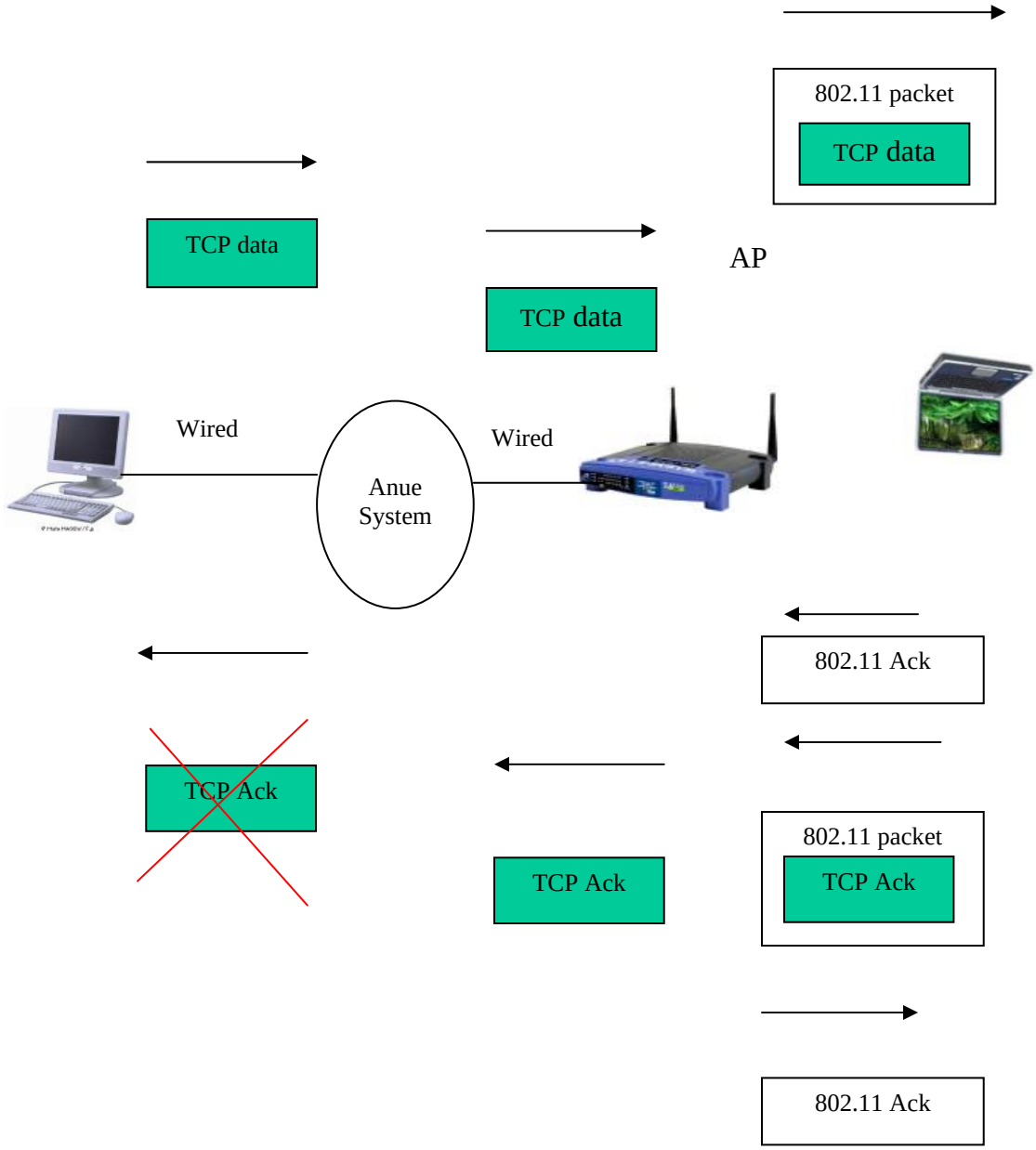

Figure 14. TCP Acknowledgement Loss by Anue System.

2.2.2.4 Latency

Latency is defined as the time it takes for a packet to reach from the source to the destination.

TCP has a tendency to adjust to the network capacity by adjusting its window size that is defined as the number of packets a sender can send before receiving acknowledgements. The TCP window size increases with the number of received acknowledgements and decreases with the packet loss. Because of the introduction of latency, the sender will spend more time waiting for acknowledgements than sending the packets, which in turn would lead to reduced bandwidth usage. TCP has a direct inverse relationship with the throughput and latency. The latency changes over an 802.11 link because of rate adaptation based on the link conditions by the 802.11 PHY. Therefore, this experiment will help the network administrator to analyze the performance of TCP over 802.11 links.

2.2.2.5 Reordering of TCP Segments

Reordering of TCP segments occurs when segments arrive out of order at the receiver. TCP uses cumulative acknowledgements using which the receiver acknowledges the highest in-order segment received. Reordering triggers fast retransmission mechanism on the TCP receiver side and results in the immediate transmission of the DupAck. This duplicate acknowledgement is indistinguishable from the duplicate acknowledgement produced due to the actual loss of packet. Reordering may take place in the network may be due to different amounts of latency present in the network, MAC retransmissions, router buffer management, different routing algorithms present in the router or transmission errors in the network.

Retransmission may have a negative impact on the performance of TCP because it triggers the generation of duplicate data packets by fast retransmission of TCP. It also causes burstiness in the TCP traffic because of out of delivery of data received by the receiver.

Reordering can happen in two ways.

1. Reordering of TCP data segments.
2. Reordering of TCP acknowledgements.

When a TCP receiver receives out of order data segment, it generates DupAck immediately for the segment it was expecting. This will let the TCP sender assume that there was a loss of a segment for which the DupAck was received. The TCP receiver will not send the acknowledgements for the next segments received unless it receives the segment it was expecting. After the reception of the out of order segment, the TCP receiver will send the acknowledgement for the next higher segment it would expect and ignores the acknowledgements for the rest of the out of order segments received as shown in Fig. 15.

Reordering of acknowledgements may cause burstiness in the TCP traffic because of unexpected acknowledgments received by the TCP source. The acknowledgements, which carry no new information are discarded by the host and new acknowledgments with new information may cause burst in the TCP traffic. This burstiness is caused because TCP source may open its congestion window to transmit more bytes based on the information present in the received acknowledgment frame.

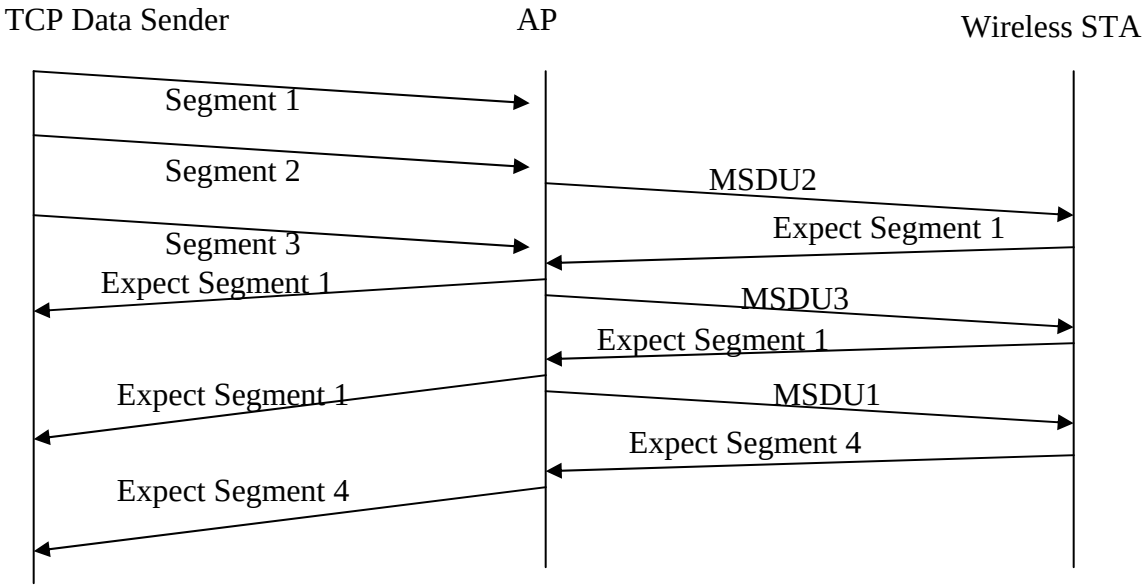

Figure 15. Packet Reorder.

2.2.2.6 Bit Error Rate

BER in wireless networks is much higher than the BER in wired networks because of which the sender has to transmit more number of packets than in the wired network. The cause of the BER is more probably the interference due to other devices in the same frequency band but the TCP source has no knowledge of the wireless loss and triggers its congestion control mechanism, thereby unnecessarily reducing the performance of the network. Therefore this experiment does an analysis of performance deterioration in the presence of varying BER.

2.2.2.7 Interference on 802.11 Link

The primarily reasons for an 802.11 device to observe interference is the thermal noise and the transmission of frames by other devices operating in the same or adjacent frequency band . The interference caused by the devices operating in the same frequency band is referred as *co-channel interference* (CCI) and due to the devices transmitting in the neighboring channels is *adjacent channel interference* (ACI). Devices running on the same frequency bands are bound to suffer more performance degradation than devices running in adjacent frequency bands.

An 802.11b channel represents the center frequency and occupies 20 MHz of the frequency spectrum. Each channel is at a difference of 5 MHz from the adjacent channel. Therefore, only three channels, i.e., 1, 6 and 11 are considered as the non-overlapping channels out of the currently assigned 1-12 channels for use by the USA.

Interference is one of the main reasons due to which, an 802.11 link is classified as a lossy link and is responsible for higher bit error rates and packet loss. This results in higher retransmissions of 802.11 segments, resulting in reduced bandwidth utilization. This is interesting in 802.11 networks because of the fragmentation mechanism that causes packets of different sizes to traverse over 802.11 link. A large packet can be sent in several fragments of smaller sizes. This is significant because each transmission of a fragment involves the transmission of a preamble, Physical Layer Convergence Protocol

(PLCP) encapsulation, a minimum inter-framing space between the fragments and expects a positive acknowledgment from the receiver.

The experiments are designed to analyze the performance of TCP with variable 802.11 fragment sizes in an environment with CCI and ACI.

CHAPTER 3

METHODOLOGY & EXPERIMENTAL PROCEDURE

3.1 Test Methodology

The *first* step in the test design methodology is to categorize the experiments into two types:

- **Baseline Experiments**

These tests cover the baseline experiments. Baseline experiments define the default configuration of the experimental parameters on the devices. The baseline configuration is shown in Table 2.The additional experiments in this thesis reference the baseline tests and assume a device has the baseline capabilities. This set of experiments is conducted without any introduction of latency, malformed packets, intentional packet loss, packet re-ordering, and interference in order to obtain the most optimal performance behavior.

Fragmentation Threshold	Maximum allowable value on an AP
RTS Threshold	Maximum allowable value on an AP

Table 2. Baseline Configuration.

- **Impairment Experiments**

These tests cover the impairment experiments. The impairment is done in two ways.

1. Wired Network impairment:

   This is done using a LAN Emulator to introduce latency, packet drop, packet duplication, BER and reordering.

2. Wireless Network:

   This is done using a Vector Signal Generator to inject noise in the 802.11 link.

The *second* step is to conduct experiments with the baseline configuration of the AP and no intentional network impairment. The results obtained from baseline experiments will serve as benchmark to evaluate the performance of other experiments with different impairment in the network.

The *third* step is to analyze the performance with different values of the configurational parameters on the AP, the results of which are then compared with the baseline experiments. All networks are susceptible to some form of network impairments in the real world and therefore these tests will do the performance analysis in context of network specific metrics.

The *fourth* step is to perform the same experiments with a second AP device in order to do a better analysis of the affect of AP metrics on the performance.

3.2 Performance Measurement

The results are then compared on a performance metric that is measured to analyze the results. The goal of all the experiments is to do the performance analysis and comparison study. The primary metric chosen in this thesis for the performance measurement of TCP over WLAN is the throughput. *Throughput* is defined as the amount of data sent and received by a user in a specified time of transaction. It is measured in bits per second (bps), bytes per second (Bps) and frames per second (fps).

In the experiments, the header bytes associated with the TCP and IP header are not taken into consideration. This type of throughput calculation is also termed as goodput [5].

The throughput calculation is based on the number of TCP socket connections present in the experiment. The number of end point pairs in the IxChariot application script represents the number of socket connections that individually contribute to the throughput calculation as shown below.

$$\text{Average Throughput (n)} = \sum_{i=1}^{n} \frac{bytes(i)}{time}$$

where n represents the number of endpoint pairs in the test and i is the i^{th} endpoint pair.

In order to provide validity of the data recorded in the experiments conducted, 95% confidence level has been considered and calculated as explained.

$$\text{Standard Deviation} = \frac{\sqrt{n\sum x^2 - (\sum x)^2}}{n(n-1)}$$

where n is the number of samples and x is the value of the throughput.

$$\text{Confidence Interval} = \frac{1.96 * \sigma}{\sqrt{n}}$$

Constant 1.96 is the value in normal distribution table that corresponds to 95% confidence interval.

The above confidence interval gives the assurance of the value to be within the interval average ± confidence interval. The experiments are run for 12<= n <=60 in order to obtain the result in the specified 95% confidence interval range.

3.3 Overview of Testing Tools

The testing tools required to carry out the experiments are as follows:

- Ixia Chariot (IxChariot) Console

This is a socket application, which provides the ability to emulate real world TCP traffic using the TCP/IP stacks of the operating system (OS) and calculates the performance characteristics; i.e., throughput of TCP traffic on wired as well as wireless medium. IxChariot console provides information to the performance endpoints installed on the STAs about the type of traffic to be exchanged between STAs and the network protocol use. Only a single IxChariot console is required to control the various

performance endpoints on different STAs. After the completion of a test, it collects the results and presents the result to the user in a graphical window. For the experiments, the IxChariot 6.10 console is used. This console software supports a maximum of fifty endpoint pairs to run simultaneously on a STA.

- Ixia Performance Endpoints

The Ixia performance endpoints are lightweight software agents used by IxChariot [10]. They are installed on the STAs, between which traffic is exchanged and monitors network transactions. After the completion of a test, each endpoint collects the result and forwards it the IxChariot console for analysis and reporting. The experiments are run on 6.20 Windows service version.

- Ixia Application Scripts

These scripts are used by performance endpoints to emulate different types of traffic flow required between the STAs. They can be customized based on application type and network environment. These scripts make the same functional calls and load on the underlying network stack as the real applications. These scripts can emulate a range of applications from a simple FTP application to a complicated voice and video streaming transaction.

The experiments are run using *"throughput.scr"* application script to emulate file transactions running on TCP protocol. This script is configured based on the parameters as described in Table 3. All other parameters are set to default. These scripts are

configured to obtain a sufficient number of timing records to get the relative precision of the transactions performed by different endpoint pairs.

# Of Socket Connections	File Size (in bytes)	# Of Timing Records	Transactions/ Timing Record	Run Duration (in minutes)	Run Option Performance Testing
2	100000	1600	16	5	Enabled
4	100000	1600	8	5	Enabled
8	100000	1600	4	5	Enabled
16	100000	1600	2	5	Enabled
32	100000	1600	1	5	Enabled

Table 3. Configuration of IxChariot Script.

- AnueSystem GEL2 Network Emulator

  This device is used as a LAN emulator to provide impairment for Layer1 and Layer2+ testing. It has the ability to filter selective or focused impairments based on the user choice. Some of the main impairments it can provide are latency, jitter, packet loss, re-ordering of packets, duplication of packets, bit error insertion and CRC error. The experiments are run on GEL2 1.78.00 version.

- Azimuth W series

This is a WLAN Test platform to test 802.11 compliant devices in complete RF isolation. It consists of a chassis with station modules and a mini-test head, which are interconnected by RF cables. The Azimuth Director software runs on a computer that provides the user interface for all the station modules and configuration. The version of the Director used is A 4.2.0.87.

- Rohde & Schwarz SMU 200A

This is the Vector Signal Generator and is equipped with digital modulation that controls the power and frequency offset in real time. This signal generator can be used to vary the SNR in the 802.11 network and works in the frequency range of 100 kHz to 6 GHz.

- WinIQSIM™

This is the simulation software for Rohde & Schwarz SMU 200A [13]. This is used to generate complex single carrier or multi carrier waveforms by modulating the in-phase (I) and quadrature (Q) signals. We used this WinIQSIM™ to generate OFDM waveforms and the version used was 4.30 that runs on Windows 32 bit machines.

- Access Point (AP)

This device is the Wireless router that provides a wired interface to the wired station and a wireless interface to the station on the wireless side. All the communication between the stations on the wired network and on the wireless network takes place through the access point. The experiments are performed using two high-end enterprise AP's of two different vendors.

- Ethereal

A network analyzer that is used to analyze traffic on the wired networks [11]. The version used is 0.10.9.

- Nifty Sniffer Interface (NSI)

A software [12] that facilitates post-capturing analysis of the 802.11 frames by reading the files that are captured using another hardware/software solutions such as Airopeek and Atheros DK. The version used is 2.4.8.

3.4 Application & Environment

The application used to measure the performance of TCP is IxChariot's *throughput* script of size 10000 bytes. This script simulates the core file transactions and is configured to run for duration of five minutes. The environment in which the experiments are performed consists of the following important parameters as shown in Table 4.

802.11 WLAN	IEEE 802.11b (2.4 GHz band)
AP Channel	6 (2437 MHz)
AP Security	Disabled
TcpMaxDataRetransmissions	500
Number of Active Wireless STA's	1
Operating System Ixia Enpoint1	Windows 2000 Server 5.00.2195
Operating System Ixia Endpoint2	Windows XP professional Version 2002

Table 4. Environment Variables.

TcpMaxDataRetransmissions is a window registry TCP/IP parameter that has a default value of 5 in Windows XP and Windows 2000 operating system (OS). The valid range of this parameter is 0 – 4294967295 (decimal) [16]. The need to change this registry value on Ixia endpoints OS's is because under heavy stress and interference, if the TCP sender does not get an acknowledgement back for a given time, then there is a possibility for the OS to leave the threads in TCP_receive state, thereby not freeing them for future TCP connections. This would lead to TCP connection timeout and the test script would stop in between and display error. Therefore, increasing the TCP data retransmission gives the ability to obtain the result of the experiments under network impairment conditions.

3.5 Test Setup

Fig. 16 describes the experimental setup that is used to carry out the baseline and AP configurable parameters experiments. The AP is placed in the RF chamber sitting at the top in the Azimuth W- Series, i.e., mini-test head, which is connected by RF cables to the station modules (STMs). These STMs are basically laptops, which run on a Windows XP operating system and are used to place the client wireless cards. The attenuation level between the AP and the client STA for the experiments is chosen as 48 dB because it was at this power level, the receiver was able to receive the frames at its best. This is also the minimum level of attenuation provided by Azimuth W-series and anything above 48 dB was resulting in a decrease in signal strength between the transmitter and receiver. Fig. 17 shows the experiment setup for impairment experiments with a LAN emulator in it.

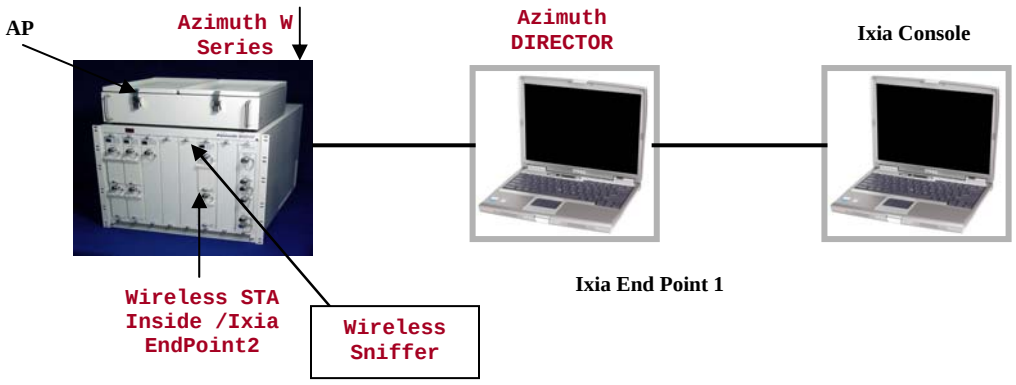

Figure 16. Baseline Experiment Setup.

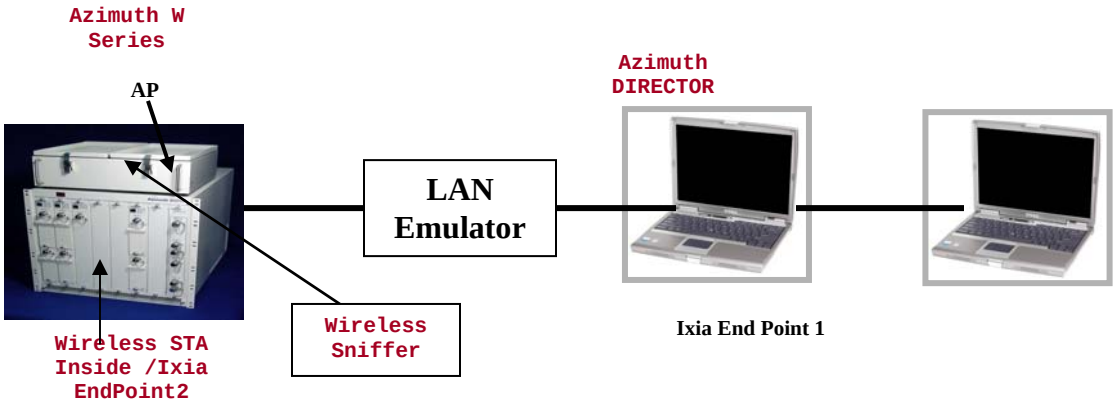

Figure 17. Impairment Experiment Setup.

3.6 Script Automation Process

The experiments are automated through the use of *tcl* language and SNMP package. The tcl scripts are made to run for different set of experiments by changing the *802dot11MIB* values of the fragmentation and RTS threshold automatically. The flow chart in Fig. 18 explains the process.

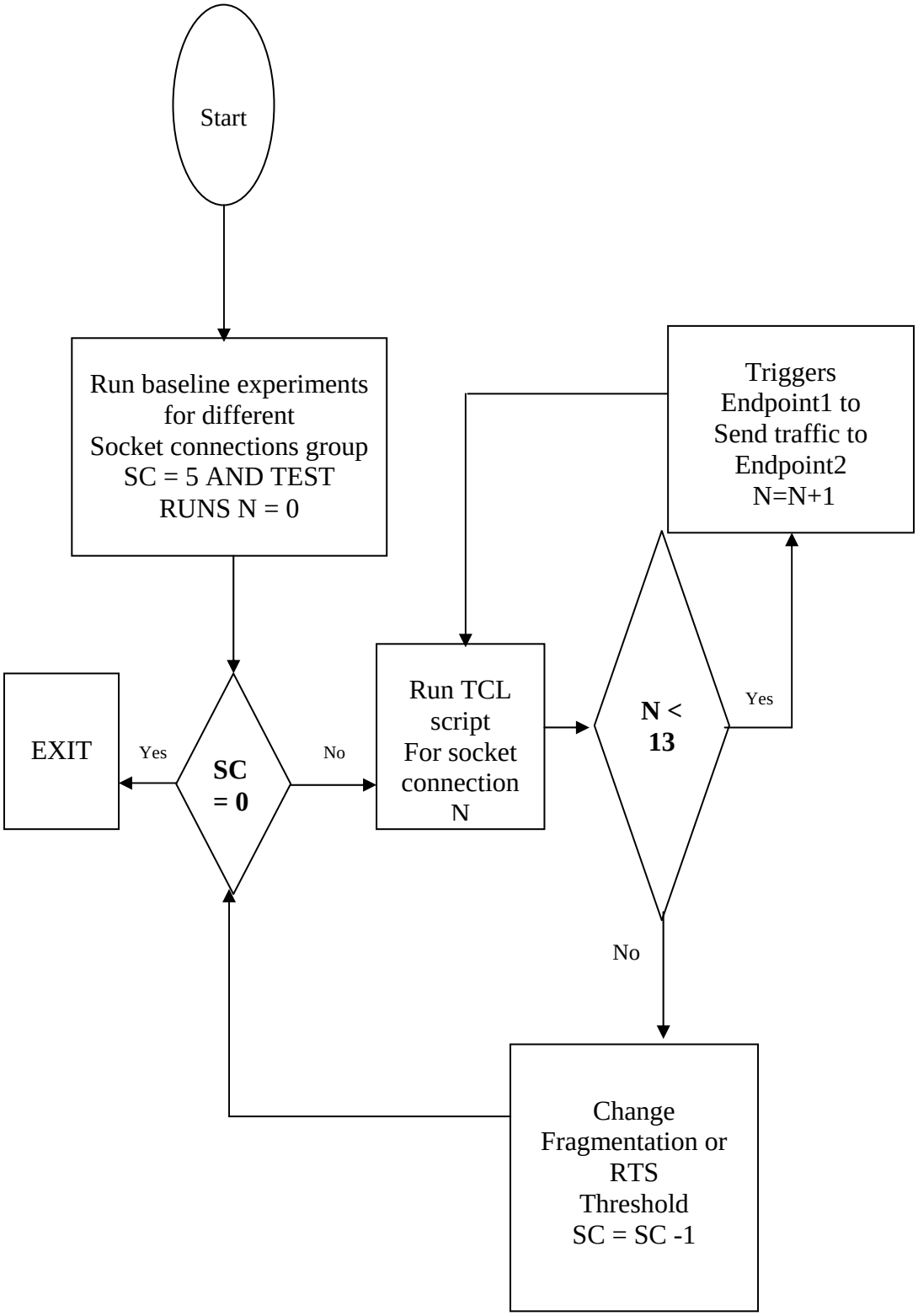

Figure 18. Flow Chart of Baseline Experiments.

3.7 Experiment Start Up Procedure

- Install the Ixia Chariot Console on one of the laptops and performance endpoints on two different laptops. It is recommended to install console and endpoints on separate computers because a laptop that runs too many applications may affect the performance of the experiments. Fig. 16 shows the Ixia console loaded on a separate computer. Another computer, which has Azimuth Director load into it, can be used to install the two performance end points. Azimuth Director provides remote desktops for all the STMs present in the Azimuth W-Series chassis, which in fact are separate laptops present in a single chassis.

- Install the performance endpoint on the Windows Server itself, which had Azimuth Director install, and another endpoint on the remote desktop that corresponds to the STM chosen for the experiment.

- Place the wireless card in the wireless cardholder inside the STM chosen.

- Place the AP in the RF chamber at the top of the Azimuth chassis and connect the AP and the STM through an SMA cable.

- Prepare the IxChariot scripts for socket connections equivalent to 2, 4, 8, 16, and 32 as described in Table 3.

- Make sure that the console and the performance end points should be able to ping each other and also to the AP and the wireless STA.

- Enable the SNMP service on the AP side if available.

- Perform a dry run, obtain a trace and check if there is any unnecessary packets loss. Adjust the programmable attenuation level between the AP and the STA on the Azimuth Directory until the trace shows no undesirable packets loss on rerun. The attenuation chosen in this experiment is 48 dB between the STA and AP.

- Run the programmed *tcl* script for different test cases.

- Repeat each experiment for at least twelve times.

- Observe the throughput and calculate the average throughput for the number of times the test performed.

- Analyze the throughput with a 95% confidence interval and perform additional trials if data not in confidence interval.

- Show the results in graphical form.

- Repeat the same steps for impairment experiments and change the impairment parameters on the AnueSystems GEL2 and Rohde & Schwarz devices accordingly.

Chapter 4

RESULTS AND CONCLUSIONS

This chapter contains a table of experiments and the results and conclusions based on the methodology discussed in Chapter 3.

4.1 Outline of Experiments

This section contains a summary of the experiments that are done as shown in Table 5.

4.2 Baseline Experiments

This section contains the result of the baseline experiments for two AP devices. The results obtained from these experiment serve as the reference graphs for the remaining impairment experiments. Baseline experiments involve configuration on the APs only and with no intentional impairment in the network. If a configuration metric on an AP is *disabled* or *off,* it implies that the metric is set to its maximum default value. For example, fragmentation *threshold off* means that its value is set to the maximum default and will fragment frames to 1536 bytes on an 802.11 link. Similarly, *RTS threshold disable* implies that no RTS-CTS mechanism occurs on the 802.11 links.

Experiment No	Experiment Name
1	Baseline Throughput
2	Vendor A – Fragmentation
3	Vendor B - Fragmentation
4	Vendor A – Fragmentation & RTS On
5	Vendor B - Fragmentation & RTS On
6	Vendor A – Effect of Fragmentation Threshold on Throughput
7	Vendor B – Effect of Fragmentation Threshold on Throughput
8	Different RTS Threshold Values
9	Effect of TCP Duplicate Acknowledgements
10	Effect of TCP Data Segments
11	Effect of TCP Acknowledgement Drop
12	Effect of TCP Data Drop
13	Effect of Latency
14	Effect of Bit Error Rate
15	Effect of Variable Signal Strength
16	Effect of Co-channel Interference
17	Effect of CCI and ACI on WLAN
18	Effect of Noise Transmission length

Table 5. Outline of Experiments.

4.2.1 Fragmentation & RTS Off

This experiment is expected to show maximum performance in terms of throughput because of the recommended default configuration on each AP and there are no intentional impairments in the network.

Fig. 19 shows that the throughput decreases as the number of socket connections increase. This is appropriate because a higher number of TCP socket connections get opened up on both sender and receiver side and all sessions try to compete for the medium to transfer data, thereby increasing the overall overhead and sharing the bandwidth.

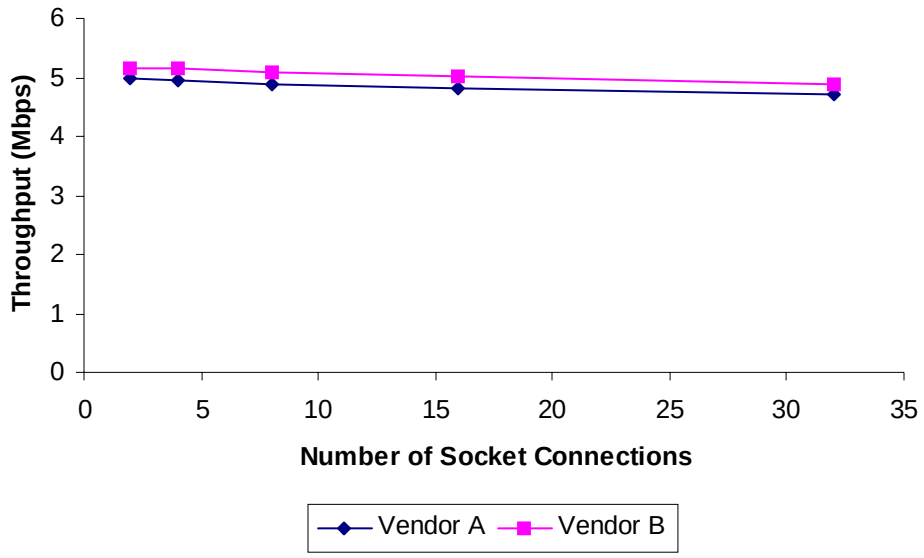

Figure 19. Baseline Throughput.

4.2.2 Fragmentation Threshold On – RTS Off

Fig. 20 shows that the throughput increases with the increase in the value of fragmentation threshold value and decrease with the number of socket connections. This helps to understand that a large *fragmentation threshold* value gives better results than a smaller *fragmentation threshold* value. The overhead involved in the transmission of each

smaller fragment with a PLCP header, signal information and data payload lowers the bandwidth utilization significantly. For every fragment constructed, there is an extra preamble and PLCP encapsulation along with MAC header and CRC which together attributes to the lower bandwidth utilization.

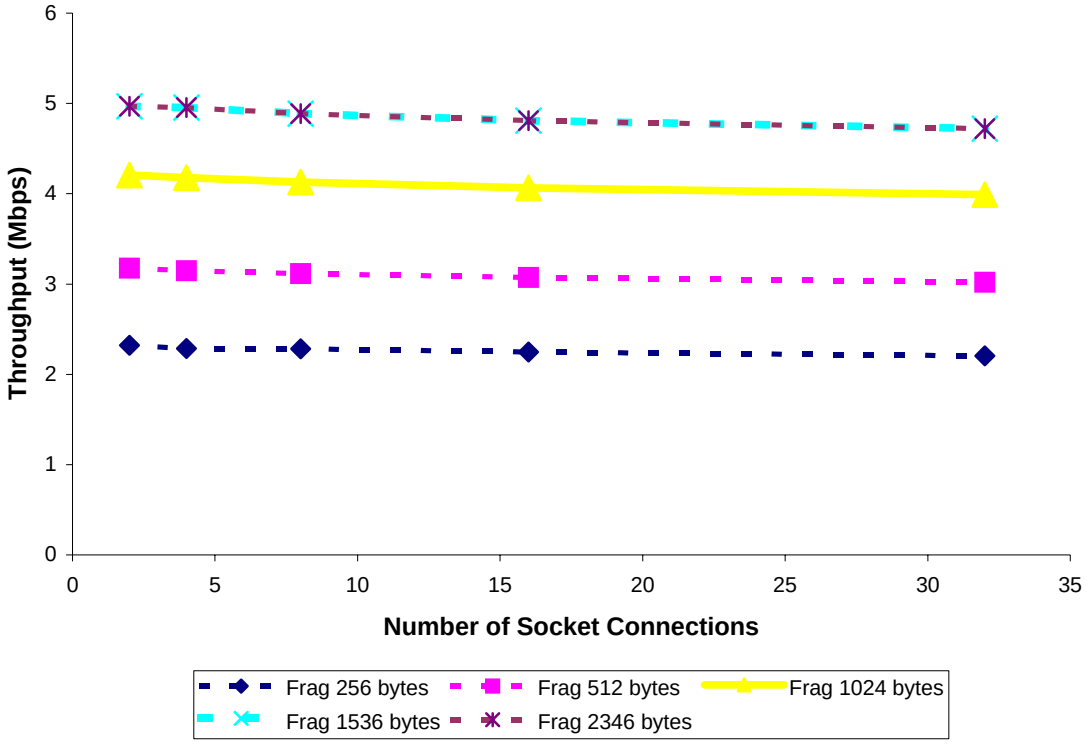

Figure 20. Vendor A – Socket Connections with Fragmentation & RTS Off.

The throughput observed for vendor B also has the same trend as Vendor A in Fig. 21. The throughput decreases with smaller fragment sizes and with the increase in the number of socket connections.

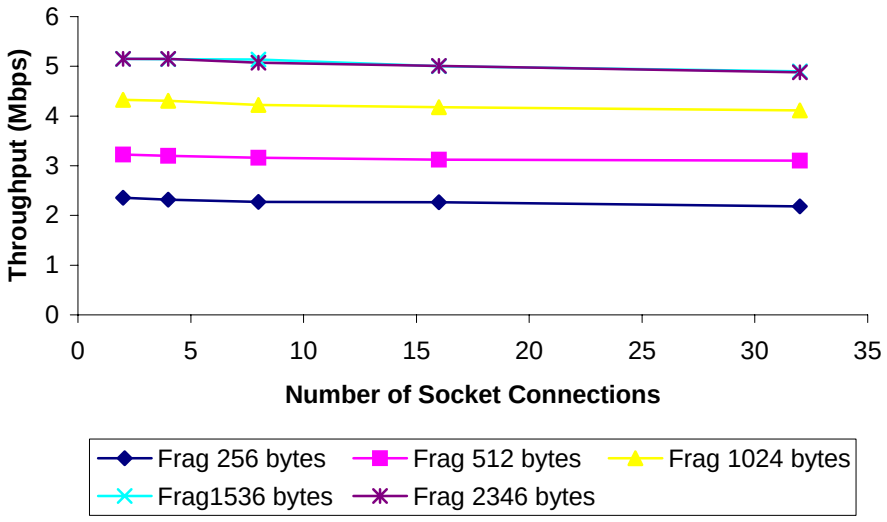

Figure 21. Vendor B – Socket Connections with Fragmentation & RTS Off.

4.2.3 Fragmentation On & RTS Threshold On

Fig. 22 shows results consistent with Fig. 21. The throughput increases with the increase in the *fragmentation threshold* value and decreases with the increase in the number of socket connections. The overall throughput for each socket connection and each fragment value decreases with the usage of RTS-CTS mechanism. This is because of the transmission of extra RTS-CTS exchanges before the actual data exchanges between the STA's.

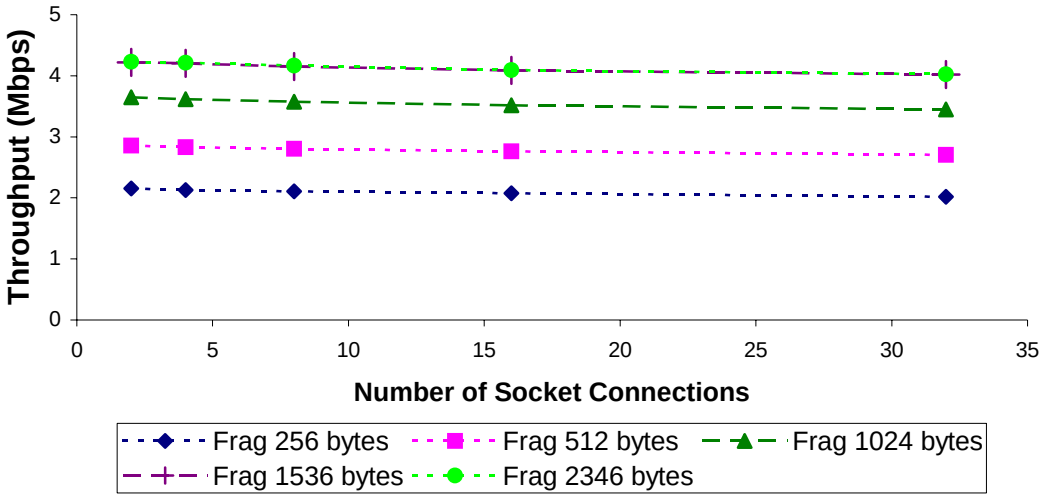

Figure 22. Vendor A – Socket Connections with Fragmentation & RTS On.

The performance for Vendor B in Fig. 23 decrease with the number of socket connections and with the lower fragmentation size. The throughput decreases with the increase in the number of socket connections and decrease of fragmentation size.

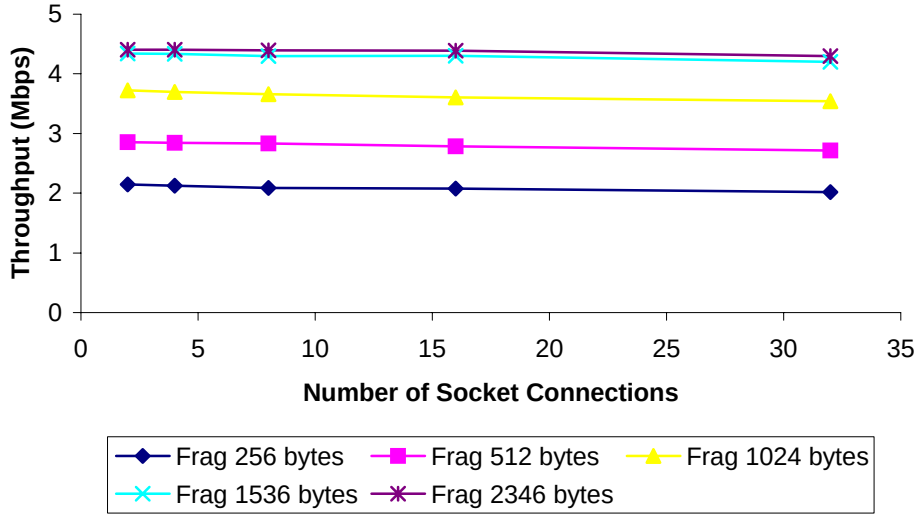

Figure 23. Vendor B - Socket Connections with Fragmentation & RTS On.

4.2.4 Effect of Fragmentation Threshold

Fig. 24 and Fig. 25 for Vendor A & B show that throughput decreases with the use of RTS-CTS mechanism, which is attributed to the overhead involved in the transmission of extra frames with no data payload in it. The throughput jumps to a higher value at 1536 bytes and then remains constant because 1536 is the maximum 802.11 fragment size for TCP Maximum Segment Size (MSS), which is 1460 bytes. Any value beyond 1536 bytes results in the transmission of 1536 bytes 802.11 packets only and hence fragmentation threshold is ineffective beyond 1536 bytes.

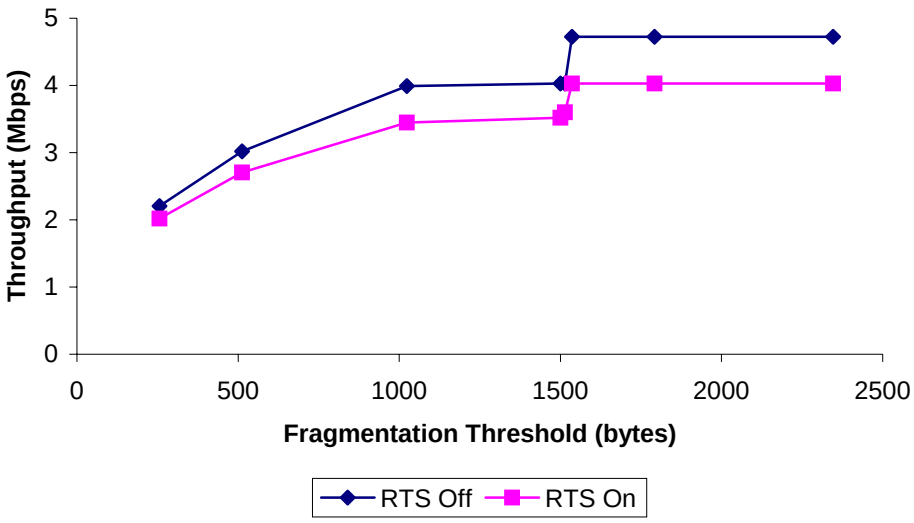

Figure 24. Vendor A - Effect of Fragmentation Threshold.

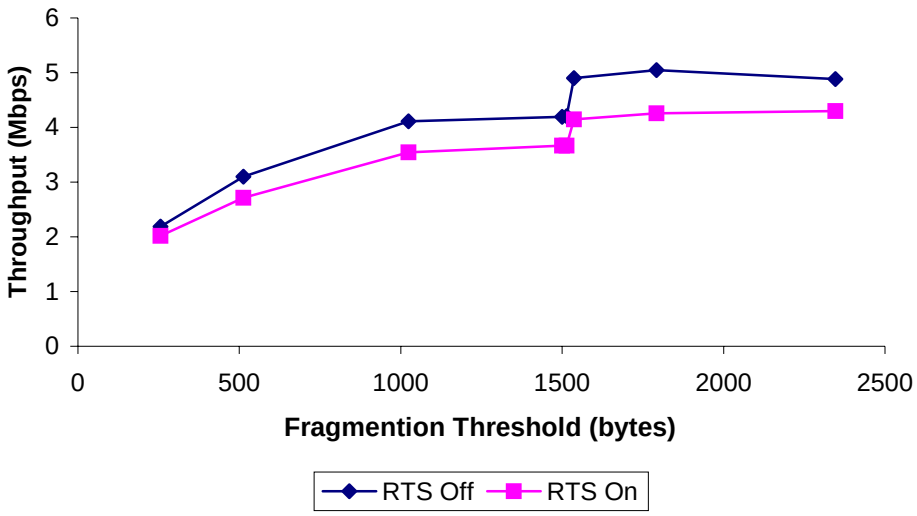

Figure 25. Vendor B - Effect of Fragmentation Threshold.

4.2.5 Effect of RTS On – Fragmentation Off

Fig. 26 initially shows a constant throughput value for different *RTS threshold* values. The throughput jumps at a value of 1536 bytes. It remains constant for values of *RTS threshold* beyond that. This is because RTS-CTS mechanism does not trigger when value of an MSDU, i.e., 802.11 packet is equal to or less than the *RTS threshold* value.

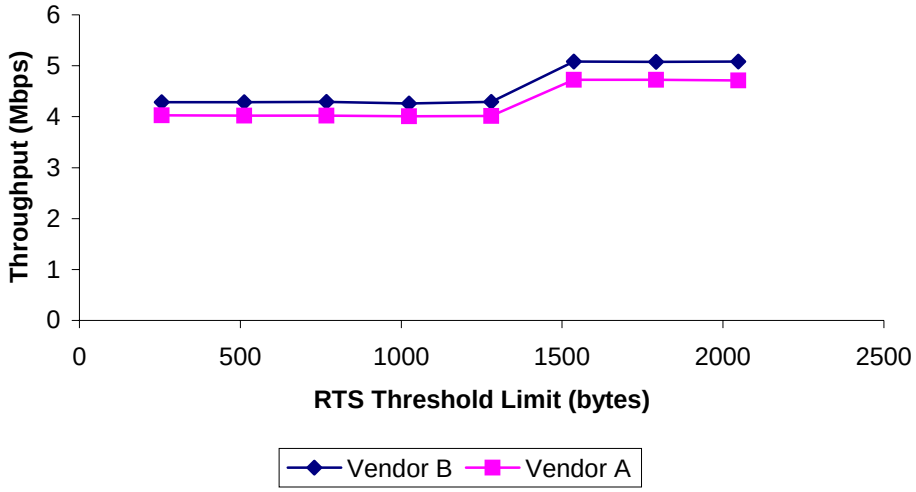

Figure 26. Effect of Different RTS Threshold Values.

4.3 Impairment Experiments

This is the second type of experiments that are based on the methodology discussed in Chapter 3. The results show the affect of various types of impairments in the network on TCP performance. These experiments are carried out for maximum fragment size on an 802.11 link for a TCP MSS, i.e., 1460 bytes unless otherwise stated differently in the experiments. The reason to demonstrate the experiments with maximum 802.11 fragment size is because experiments conducted with smaller fragment sizes showed proportional decrease in their performance and hence are not displayed. The experiments in this section are performed on 32 IxChariot endpoint pairs in order to emulate sufficient number of different TCP connections in the real world.

4.3.1 Duplicate TCP Acknowledgements

This experiment is conducted by duplicating number of TCP Acks in different probabilities. In one experiment, a single Ack is duplicated either once or twice (1 - 2) in order to avoid fast retransmission mechanism of TCP. In another experiment, a single Ack is duplicated atleast once and atmost four (1 - 4) to trigger fast retransmissions.

Fig. 27 shows that the presence of DupAcks more than two decreases the performance of the network significantly. This is because of the fast retransmission mechanism and the transmission of duplicated TCP data segments in the network. The performance degradation of the network is proportional to the increase in the probability of DupAcks. This is because of the proportional increase in the number of fast retransmissions. A higher percent probability of the DupAcks received results in an increased number of data retransmissions. The useful data in the network decrease, i.e., the total bandwidth utilization is less in the presence of more data retransmissions.

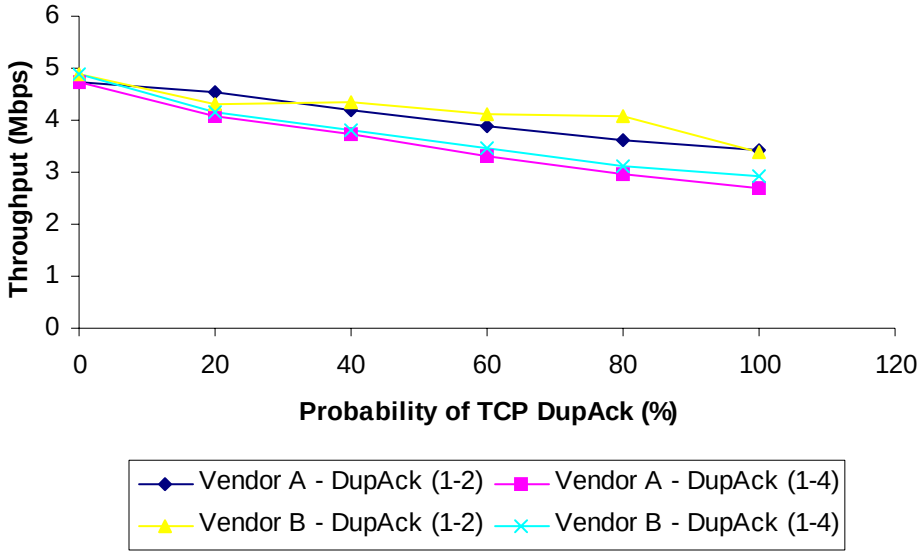

Figure 27. Effect of Duplicate TCP Acknowledgements.

4.3.2 Duplicate TCP Data Segments

The aim of this experiment is to analyze the throughput degradation in the presence of duplicated TCP data segments in different percentages in the wired network.

Fig. 28 shows a decrease in the performance in the network and the throughput decreases with the increase in the probability of the TCP data segments for both Vendor A and B. This is because of the increased number of retransmissions of the data segments in the network. This retransmission consumed a significant part of the bandwidth and resulted in a longer wait for the TCP source to get the Acks for its original data segments. The TCP sender was found to retransmit the original data packets, the Acks for which were delayed because of the bandwidth utilization by the duplicated TCP data packets. The performance drops significantly when the percentage of duplicate TCP segments reaches fifty. This is attributed to the reduced congestion window maintained by the TCP and not being able to get the acknowledgement back for its data packets. This lead to long delays due to which the endpoint pairs on the STA were unable to receive TCP responses. The result shows that presence of more than forty percent data retransmission results in severe performance degradation of TCP over 802.11 link.

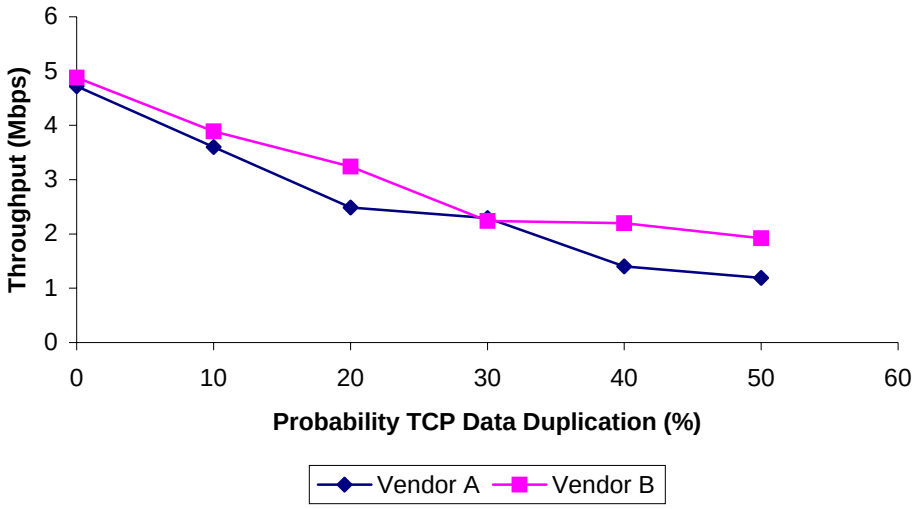

Figure 28. Effect of Duplicate TCP Data Segments.

4.3.3 TCP Acknowledgement Drop

Both wireless networks and wired networks experience the loss of TCP acknowledgement packets. If the TCP retransmission timeout occurs and an Ack is not received, then TCP source retransmit the TCP data segment again to be delivered. The experiment is done by dropping the TCP Acks in different percentages on the wired side. Fig. 29 shows a that performance decreases relatively slowly with the increase in the percentage of Ack drop. This is because TCP uses cumulative acknowledgements and the TCP receiver transmits Acks for the TCP data segments it received in a TCP burst. When an Ack was dropped, the TCP source received the Ack with the next higher sequence number. This resulted in less reduced retransmission of TCP data segment for every Ack loss and hence relatively low performance degradation. When the percentage of Acks

drop was increased further from 16%, the problem of TCP connection time out started on the network. One or more of the TCP socket connections were unable to establish TCP connections with the port on the receiver side because of the increased chances of dropping of the connection establishment frames. The result shows that the performance degradation of TCP is relatively low for small percentage of TCP Ack drop.

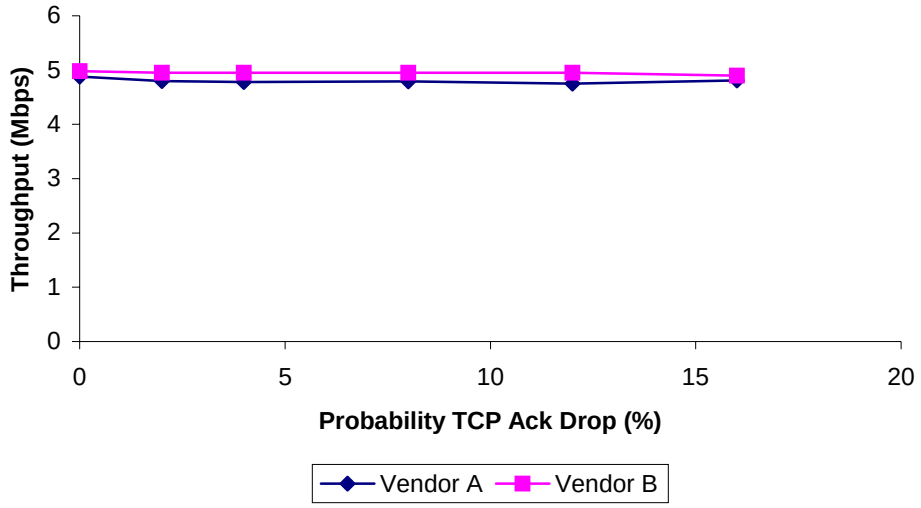

Figure 29. Effect of TCP Acknowledgement Drop.

4.3.4 TCP Data Segment Lost

This experiment analyses the performance due to the different percentages of TCP data segment loss on the wired side. Fig. 30 shows the performance degradation with the increase in TCP data drop percentage. This is because the TCP receiver received TCP data segments out of order and sent DupAcks immediately. The TCP source initiated fast retransmission when it received three or more DupAcks consecutively. The retransmitted

data packets were also dropped many a times which resulted in further retransmissions of the data packets This lead to a high number of retransmitted traffic in the network. The test was not being analyzed beyond 16% data drop because of the connection timeout by the TCP.

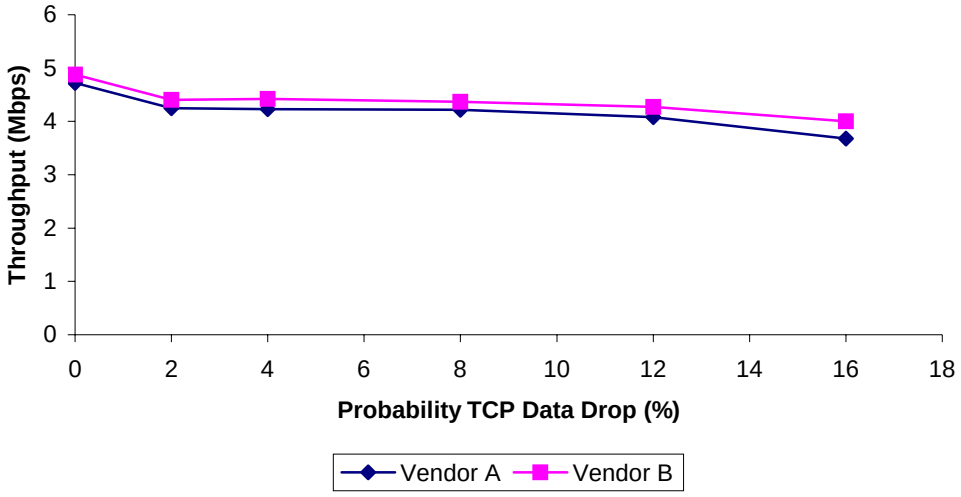

Figure 30. Effect of TCP Data Drop.

4.3.5 Latency

This experiment considers the affect of latency on the performance of TCP. Fig. 31 shows that performance decreases with the increase in latency in the network. This is caused by the delay it takes for the TCP end-points to exchange data-acknowledgement sequence. Throughput is inversely proportional to the latency in the network and adds extra time for the sender and receiver to exchange TCP segments. The reason to choose a maximum value of 500 ms is because it was the maximum possible configuration available on LAN emulator.

The throughput decreased with the increase in latency because of longer time delays between TCP end-to end connections.

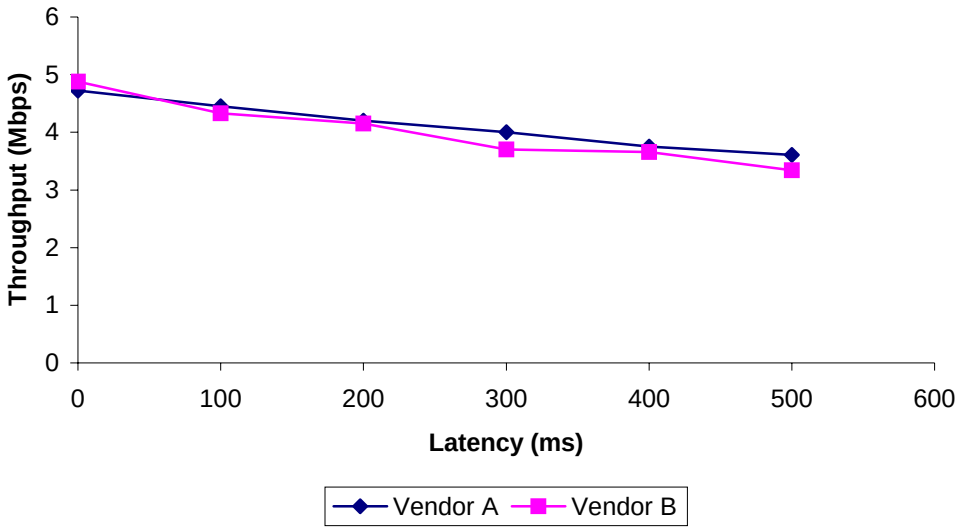

Figure 31. Effect of Latency.

4.3.6 Bit Error Rate

This experiment is performed to analyze the effect of bit errors on the performance of TCP. The bit error is injected in different proportions in the TCP data frames as shown in Fig. 32. The corrupted frames are received by the wireless STA and are discarded. The TCP source transmit the data frame again after it receives atleast three consecutive DupAcks or retransmission timeout.

The experiment was able to run for BER values of 10^{-6}, 10^{-5} and 10^{-4} only. The results could not be recorded for higher BER because the end point pairs were running forever

and never stopped. The result obtained from this experiment is incomplete unless the effect for BER for values more than 10^{-4} is recorded.

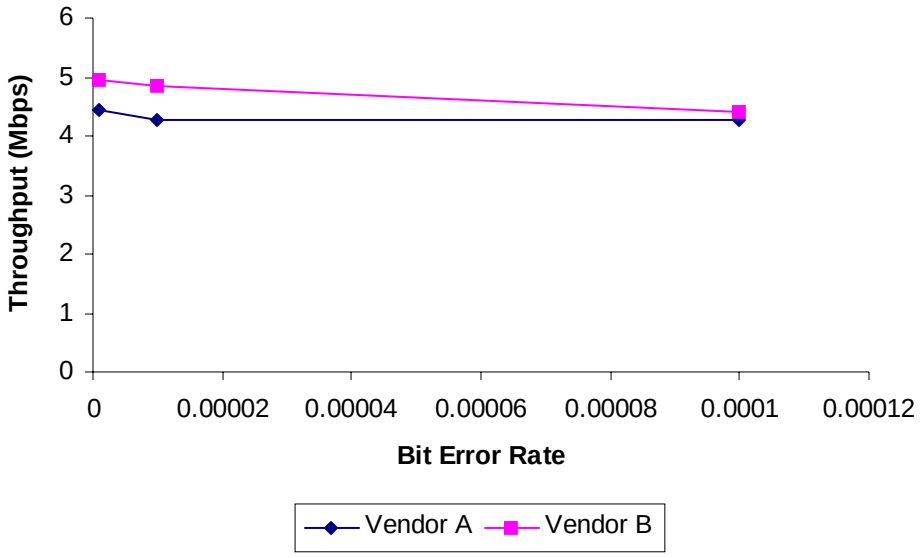

Figure 32. Effect of Bit Error Rate.

4.3.7 Reorder TCP Packets

This experiment analyses the affect of reordering of TCP data packets on the performance of TCP. The reordering is performed in a range of 1 - 7, i.e., the TCP data and Ack packets are delayed by as much as one to seven packets with a value chosen between *one* and *seven* in a uniform distribution manner. The run traces analyzed after the experiments did not observe reordering of packets more than 3 or 4. The amount of retransmission for different reorder probabilities was observed to be very much similar. This did not result in the correct interpretation of the results and hence could not be analyzed.

4.3.8 Variable Signal Strength

This experiment is performed by varying the power level between the transmitter and the receiver with the help of digital attenuator that is controlled by a software module in Azimuth system. This is achieved by amplifying and diminishing the attenuation between the STA and the AP at different times. The experiment starts with an initial attenuation of 48 dB between the STA and the AP because it is the default minimum attenuation provided by the Azimuth system and provides optimal connection between the AP and the STA with no packet loss.

This experiment is performed on different 802.11 fragment sizes in order to see the effectiveness of small fragments in the presence of varying power levels. Fig. 33 shows that larger fragment sizes give better throughput than smaller fragment sizes. This is because the 802.11 PHY adapts to a lower transmission rate in low SNR in order to transmit the frames successfully. The 802.11 PHY switched to lower transmission rates for small fragments more often than for large fragments with the changing SNR.

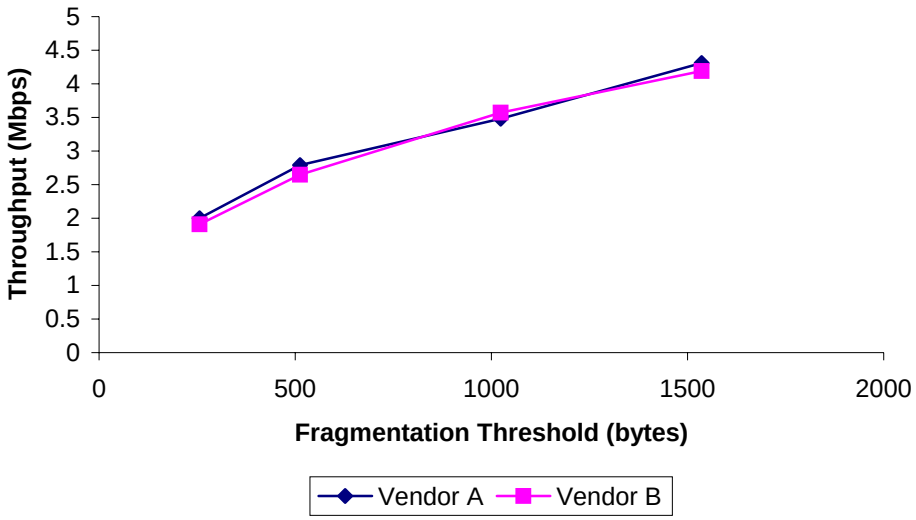

Figure 33. Effect of Fragmentation Threshold in Variable Signal Strength.

4.3.9 Co-Channel Interference

This experiment analyzes performance in the presence of co-channel interference on 802.11b network. The interference has been produced through Rohde and Schwarz Vector signal generator. This device is configured to transmit orthogonal frequency-division multiplexing (OFDM) waveforms in 2.437 GHz frequency band. An OFDM frame consists of a PLCP preamble, followed by a signal field and the data payload. A preamble is used for synchronization and the signal field contains information about the rate and length of the data payload. The preamble consists of 12 symbols and the signal field is of 24 bits. The data payload considered is zero bytes in order to keep the minimum transmission time of an OFDM frame in the air. The transmission times of a PLCP preamble, signal and zero data payload are 16 μsec, 4 μsec and 4 μsec [6]. It takes additional 6 μsec signal extension time in OFDM frames. Therefore, it takes a total of 30

μsec for an OFDM waveform to transmit in the air. The magnitude of the power signal is maintained at –20 dBm because this much strength was found to be adequate to corrupt the 802.11 frames of the AP and the wireless STA. The experiment is designed to consider interference at different interval of frequency of OFDM frames. In this experiment, the frame transmission frequency was varied while keeping the carrier frequency, i.e., 2.437 GHz constant.

Fig. 34 shows that the performance decreases with the increase in frequency of impairments due to co-channel interference on the 802.11b link. Smaller fragments have a greater negative impact on the performance degradation and the throughput drops substantially at around a rate of 1908 Hz of noise injection. The reason for this is that the AP deauthenticated the STA for fragment size 256 bytes time and again. The AP switched to a lower transmission rate of 1 Mbps to retransmit the data frames. The TCP socket pairs were not able to receive the acknowledgements back from TCP host. The loss of 802.11 data frames increased with the increase in the frequency of the OFDM noise. For each retried fragment, the AP had to contend for the channel again and perform random backoff procedure and wait until the channel was clear. The time taken by the AP to retry all the fragments was significantly higher than to retry larger fragments. The larger fragments were transmitted most of the time with higher data rates and therefore the performance obtained with bigger frames was better.

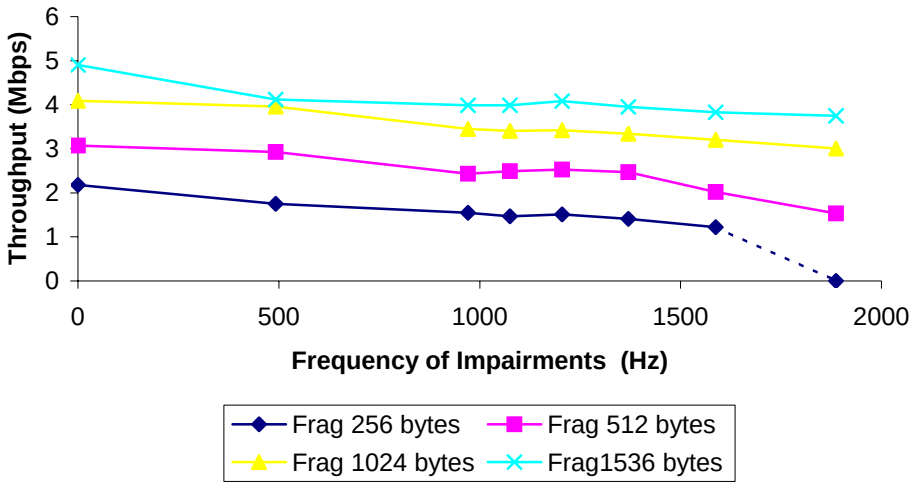

Figure 34. Vendor A - Effect of Co-Channel Interference.

4.3.10 Adjacent-Channel Interference

This experiment is done to analyze the affect of adjacent channel interference on TCP performance. The Rohde and Schwarz device is enabled to transmit OFDM frames in the 2.432 GHz frequency band to interfere with the encapsulated TCP packets in 802.11 frames in the 2.437 GHz band. The magnitude of the power level is maintained the same as for co-channel interference, i.e., –20 dBm. This experiment also compares the effect of *ACI* and *CCI*.

Fig. 35 shows the performance of TCP in the presence of CCI and ACI. The performance degradation due to CCI is much more than ACI that is because of more noise level in the same frequency band. The performance degrades in ACI and CCI with the increase in frequency of the noise because of higher interference in the 2.437 GHz band.

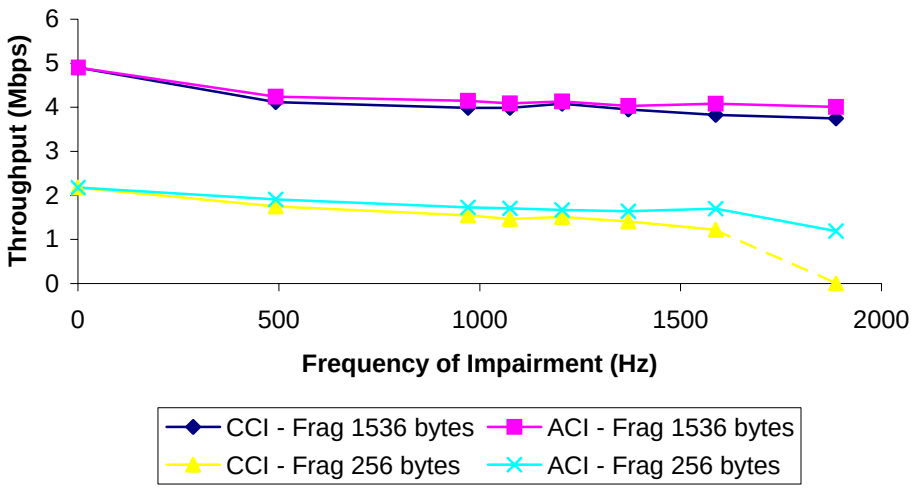

Figure 35. Vendor A – Effect of Co-Channel and Adjacent-Channel Interference.

4.3.11 Varying Co-Channel Interference

This experiment analyzes CCI on 2.4 GHz frequency band by varying transmission times, i.e., increased data payload of OFDM frames in the 802.11 link. This experiment is conducted through Rohde and Schwarz SMU. The power level of the OFDM frames is maintained the same, i.e., –20 dBm.

Fig. 36 shows that the performance degrades significantly in smaller 802.11 fragment sizes. This is because larger fragment sizes were transmitted at 11 Mbps and were then switched back to 5.5 Mbps and 2 Mbps rate at few times. The AP switched to transmission rate of 2 Mbps and then 1 Mbps rapidly for retried 802.11 smaller data segments. The AP took a longer time to retransmit the smaller fragments and the SIFS interval between the acknowledgement and the next fragment was significantly high. The

802.11 link with reduced data rate of 1Mbps became the bottleneck of the network and TCP socket connections did not get back the TCP acknowledgements. This resulted in the deterioration of the network for 256 bytes fragments and 512 bytes fragments.

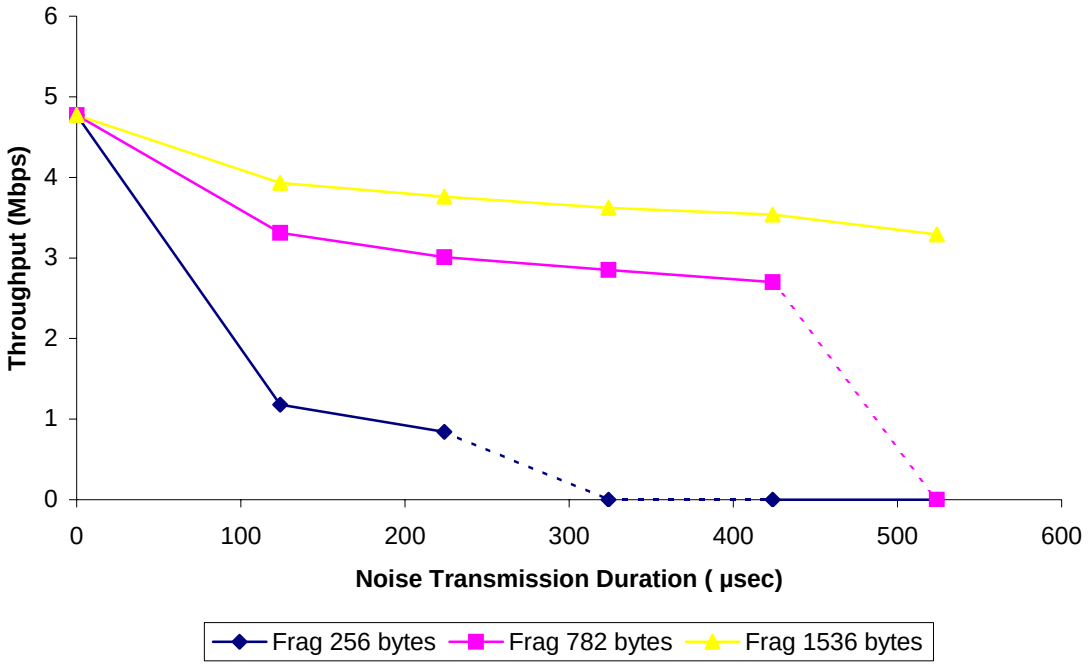

Figure 36. Vendor A - Effect of Increase in Noise Transmission Duration.

Chapter 5

SUMMARY AND FUTURE WORK

5.1 Summary

This work provides an extensive set of experiments to determine the performance of TCP over 802.11 links in various network conditions and configurations. The performance of TCP was found to vary in different experiment setups. The performance cost associated with the metrics on the Access Point was analyzed and compared. Each experiment is unique and gives insight into the performance cost of TCP associated with 802.11 links.

The results obtained from the experiments also found that fragmentation threshold can affect network performance. Fragmentation threshold should be set to the highest possible value in case of the network environments considered in the experiments.

The results show specific trends regarding TCP traffic on WLANs with different network conditions. Through an examination of deterioration parameters, the Network administrator may be able to match the numbers with those in this study to “predict the performance”.

5.2 Future Work

The present work can be expanded to understand some of the more complex scenarios of TCP over 802.11 WLANs as discussed below.

5.2.1 Multiple Wireless Stations

The experiments performed in this work have only one wireless station associated to the access point. One future experiment could test multiple stations in the same BSS, sharing the available bandwidth with the same impairment parameters as in this thesis.

5.2.2 Bi-Directional TCP Data Traffic

This work considers TCP data traffic only from the wired station to the wireless station. A future experiment could test a bi-directional flow of TCP data traffic and bi-directional TCP acknowledgments. This would investigate the efficiency of piggybacking where acknowledgements of received packets are transmitted as part of the data segments. Many TCP exchanges include both data and acknowledgements, these results would be useful to the researchers.

5.2.3 Wireless as First Hop

The same experiments in this work can be done by considering 802.11 link as the first hop in the network topology. How much does the performance gets affected in such a case would be a good topic to analyze. How do the retransmission time out period changes and the bandwidth utilized would be worth noticing.

5.2.4 Quality of Service Access Point

Performance of TCP over WLAN with a Quality of Service Access Point (QAP). The performance analysis of TCP with higher priority traffic such as voice or video. This wood be a good study to do the analysis of TCP behavior in the presence Quality of Service (QoS) traffic.

5.2.5 Noise Using Complimentary Code Key Frames

The current work can be extended by injecting noise on the wireless side using *complimentary code key* frames. This would prevent the wireless STA as well as the AP to not transmit any traffic when the noise was present on the wireless medium. The probability of corruption of packets decreases in this case but the STA's will need to wait for more time because of CSMA/CA mechanism.

5.2.6 Reference Guide

The results obtained from the experiments will help the network administrator or system user to predict the behavior of TCP over 802.11 links in different impairment conditions. A future work item could be to create a system of optimal configurations and a reference guide for network administrators to optimize their network configurations with existing impairments in their networks.

BIBLIOGRAPHY

[1] C. Parsa, JJ Garcia-Luna-Aceves, "TULIP: A link-level protocol for improving TCP over wireless links," in *Proc.* IEEE Wireless Communications and Networking Conference, Volume 5, Issue 1, Pages: 57 – 71, March 2000.

[2] M. Chinta and A. Helal, "ILC-TCP: An Interlayer Collaboration Protocol for TCP, Performance Improvement in Mobile and Wireless Environments," at Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL, 2003.

[3] H. Balakrishnan, V.N. Padmanabhan, S Seshan, R.H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," in IEEE/ACM Transactions on Networking, Vol. 5, No. 6, Pages: 756 – 769, December 1997.

[4] H. Balakrishnan, S. Seshan, E. Amir and R.H. Katz, "Improving TCP/IP Performance over Wireless Networks," in *Proc.* 1st ACM Int'l Conf. on Mobile Computing and Networking (Mobicom), Pages: 2 – 11, November 1995.

[5] B.Hickman, D. Newman, S. Tadjudin, T. Martin, "RFC 3511, Benchmarking Methodology for Firewall Performance," April 2003.

[6] IEEE Wireless LAN Edition, IEEE Std 802.11-1999, Reaffirmation 2003.

[7] J. Geier, "Wireless LANs, Implementing High Performance IEEE 802.11 Networks," (*second edition),* Sams Publishing, Indianapolis, Indiana, 2002.

[8] A.B. Tanebaum, "Computer Networks," (*fourth edition*), Prentice Hall, Upper Saddle River, NJ, 2003.

[9] Information Science Institute, "Transmission Control Protocol," RFC 0793, September 1981,http://www.faqs.org/rfcs/rfc793.html.

[10] Ixia Support, "Endpoint library," http://www.ixiacom.com/support/endpoint_library.

[11] Ethereal, "The world's most popular network protocol analyzer," http://www.ethereal.com.

[12] InterOperability Laboratory, “Network Sniffer Interface Test Tool,” http://www.iol.unh.edu/consortiums/wireless/tools/nsi/nsi.php.

[13] Rohde & Schwarz, “Vector Signal Generator R&S SMU 200A,” http://www.rohde−schwarz.com/test_and_measurement/signal_generation/smu200a.html.

[14] G. Fairhurst, L. Wood, “RFC 3366, Advice to link designers on link Automatic Repeat reQuest (ARQ),” August 2002.

[15] M. Allman, V.Paxson, W.Stevens, “RFC 2581, TCP Congestion Control,” April 1999.

[16] Microsoft Technet, “Microsoft Windows Server TechCenter,” http://technet2.microsoft.com/WindowsServer/en/Library/7dac9001-3e55-4e9c-b0fa-52841ece2fdd1033.mspx?mfr=true

[17] F. Filali, “ Link-Layer Fragmentation and Retransmission Impact on TCP Performance in 802.11-based Networks,” in The 7th IFIP International Conference on Mobile and Wireless Communications Networks, Marrakech, Morocco, 19 – 21 September, 2005.

DEFINITIONS

ACI: Adjacent Channel Interference

AP: Access Point

ARQ: Automatic Repeat Request

BER: Bit Error Rate

BSS: Basic Service Set

CCI: Co-channel Interference

CS: Carrier Sense

CSMA/CA: Carrie Sense Multiple Access/ Collision Avoidance

CRC: Cyclic Redundancy Check

CTS: Clear to send

DCF: Distributed Coordination Function

DIFS: DCF Inter Frame Space

DoD: Department of Defense

DS: Distributed System

ESS: Extended Service Set

FCFS: First come first serve

FCS: Frame check sequence

FTP: File Transfer Protocol

HDR: Header

HTTP: Hyper Text Transfer Protocol

I: In-phase

IBSS: Independent Basic Service Set

IEEE: Institute of Electrical and Electronics Engineer

IP: Internet Protocol

MAC: Media access control

MIB: Management Information Base

MPDU: MAC Protocol Data Unit

MSDU: MAC Service Data Unit

MSS: Maximum Segment Size

MTU: Maximum Transmission Unit

NIC: Network Interface card

OFDM: Orthogonal frequency-division multiplexing

OSI: Open System Interconnection

PHY: Physical

PLCP: Physical Layer Convergence Procedure

Q: Quadrature

QoS: Quality of Service

RF: Radio Frequency

RTP: Real Time Transport Protocol

RTS: Request to send

SIFS: Short Inter Frame Space

SMTP: Simple Mail Transfer Protocol

SNMP: Simple Network Management Protocol

SNR: Signal to Noise Ratio

STA: Station

TCP: Transmission Control Protocol

WAN: Wide Area Network

WDS: Wireless Distribution System

WLAN: Wireless Local Area Network