Radio Resource Management in Dense WiFi Networks

by

Shahjehan Hakim

B.E., Computer and Information Systems,

NED University of Eng. and Technology, Pakistan, 2005

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Masters of Science

Interdisciplinary Telecommunications Program

2010

This thesis entitled:
Radio Resource Management in Dense WiFi Networks
written by Shahjehan Hakim
has been approved for the Interdisciplinary Telecommunications Program

Timothy X Brown

Dr. Kenneth Baker

Dr. Thomas Schwengler

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Hakim, Shahjehan (M.S., Telecommunications)

Radio Resource Management in Dense WiFi Networks

Thesis directed by Prof. Timothy X Brown

The use of WiFi in commercial and home networking is increasing at a very fast pace. This increase in popularity especially in unmanaged WiFi networks lead to increased interference among the devices and reduced goodput for the consumers. Most of the domestic users living in apartment buildings use their WiFi access points at default channel settings and transmit at highest power. Multiple devices being on the same channel and transmitting at highest power degrade the cumulative goodput of the network.

This thesis seeks to determine if in an unmanaged high density WiFi network, the cumulative goodput of a network can be increased by changing some RF parameters. The parameters we choose to tune are data rate, transmit power and carrier sensing. To answer this question a controlled test bed was designed which would provide repeatable and reliable results.

Finally, for each physical scnerio tesed we recommend a combination of data rate, transmit power and carrier sensing which would provide maximum cumulative goodput.

Dedication

To my parents, brothers and friends...

Acknowledgements

I would like to acknowledge and thank Prof. Timothy X Brown for his constant technical guidance and support throughout this research work. I would especially like to mention the motivation he provided for automating the experimental setup. I would take this opportunity to thank him for all the resources he provided for performing the work in a professional setting. I would also like to thank Dr. Kenneth Baker and Dr. Thomas Schwengler for being on my thesis committee. Their wide experience in wireless industry and research helped in polishing this document. I would like to acknowledge the support of Sumitomo Electric Co. for initiating this research, especially Hide Miyoshi. Finally, I would like to thank my colleagues Martin Heusse, Daniel Henkel, Siddharth Maru, Vamshidhar Dantu and Sears Merritt for being there to help me keep the research work focused and optimizing solutions for best performance.

Contents

Tables

Table

Figures

Figure

Chapter 1

Introduction

Example of Local Interference

Malibu, California: Jeff is awakened up by his automated house controller which opens the curtains at exactly 8:00am. After enjoying the beautiful pacific morning through his glass windows for a while, he picks up the remote and a turns on his new HDTV. Flipping through the channels; received by a satellite device and distributed in his house by a wireless set top box to his TV, computer, and phone, he stopped at his favorite music channel. He loved the crisp sound of his wireless surround speakers, and appreciated the advice of his engineer friend to go cableless in his house. His phone went off, anwering it from his bluetooth, he fired up the coffee maker and mircowave with the same remote and walked to the kitchen appreciating technology once again. Chatter entered into his phone, and WiFi got slower as he got closer to the microwave, and his bluetooth device failed. This was the third time it happened this week, at the same exact same spot and situation.

Example of Global Interference

Boulder, Colorado: Julie is a student at the University of Colorado, Boulder. She just moved from a townhouse in the suburbs to an apartment complex very close to campus. Even though she doesn't have a great monthly income, she does love gadgets and owns an iPod, a laptop, and a smartphone. To save on her monthly phone bill, she has a subscription for unlimited calling in North America using Skype, which she can use from all of her devices over her home WiFi connection. It gives her equal mobility at WiFi spots where she spends most of her time (her home and school). It saves her a lot of money on her phone bill. This strategy used to work just fine in the townhouse, but now her Skype calls often fail. After consulting with one of her engineering friends, she discovered that the problem was intereference from her multiple neighbours and their wireless devices.

Example of Unmanaged Environment

San Francisco, California: David is a professional software developer. He flew in for a day to witness the launch of a new operating system. The conference hall was filled with more than 500 people, each carrying at least one WiFi device. He opened his laptop, launched WiFi, and experienced what he already expected. First, it took him a while to get onto the network. Second, it was just as good as having no Internet access at all. He fondly recalled the quality WiFi service provided at an esteemed computer professionals conference. He took out his 3G AirCard to access the Internet as usual, and thanked his company for providing him one.

Example of Managed Environment

Chicago, Illinois: Sarah works as a wireless network administrator for a multinational company. Her company acquired a local call center to support their needs. She was assigned to design the wireless network for the new three floor building. She developed a design after mulitple discussions with their current vendor and estimating the needs of the new facility. Because of time and financial constraints, she decided to go all wireless. One of the major considerations was the very high density on all three floors of the call center, which was different than their current sparsely populated single floor campus. Sarah made sure that there were enough access points (APs) on each floor, and did implement a non-overlapping channel strategy. Moreover, the centralized controller of all APs could dynamically adjust power if they interfere with each other. A few weeks after turning on the system she started getting complaints about sporadic network disconnections, lower throughput, and lower voice quality.

This thesis addresses the effect of changing RF parameters like Physical Carrier Sensing, Data Rate, Transmit Power, RTS/CTS at individual nodes and pairs, and studies the impact on individual and cumulative throughput of the nodes and pairs. Furthermore, we also analyze if it is possible to maximize the cumulative throughput in an environment, by using a node or pair which automatically adjusts these RF parameters.

1.1 Research Question

WiFi is a term coined by the Wi-Fi Alliance[2] for devices that conform to IEEE 802.11 standards[18]. These devices use radio waves in the 2.4GHz or 5GHz unlicensed spectrum and can form a high-speed Wireless Local Area Network (WLAN). The products bearing the WiFi certification are ensured to work smoothly in a WiFi network. This does not imply that other non WiFi certified products will not interoperate. It is that these devices were not tested by the WiFi Alliance in a lab with other WiFi products. The number of WiFi deployments are increasing at an exponential rate and the number of wireless networks will exceed the number of its wired rivals sooner or later. As Glenn writes, users will be able to enjoy gigabit per second speed on their wireless devices in a couple of years, making wireless the most efficient, easy, and reliable means of communication [8]. Wide deployment of WiFi in public places and using WiFi as a marketing strategy for attracting customers is quite common nowadays.

For in home networks, wireless is considered the best option for ease of installation, affordability and mobility. Wireless networks, though assumed simple to deploy, are very complex in nature and can be affected by numerous factors. These deployments when performed in a business environment are preceeded with a site survey and then careful deployment of the network customized to the physcial environment. A home user doesn't have the expertise or resources to perform these site surveys and locate the best place for their access point. Numerous studies have been performed for the characterization of home wireless networks, their challenges, and designs to optimize the throughput of a home wireless network [14]. Similarly, public wireless hotspots can run into the same problems of having less frequent monitoring and management.

This popularity and increased use of WiFi for personal use can be observed by scanning for the number of WiFi networks in your surroundings. A quick scan in student housing at CU shows 20+ networks. The number of these networks increases as we move to much denser environments, such as multi-story apartment buildings.

In a dense WiFi network this study seeks to understand how different WiFi parameters can affect overall throughput in a network. This throughput is a combination of all the clients participating in the network and is not distributed equally among clients as generally assumed. As discussed in [12], the reasons for decreased throughput can be co-channel interference or collisions, and can be addressed by tuning Physical Carrier Sensing (PCS) thresholds, transmit power, and contention window size.

First, this work studies how the throughput available to each user in a dense WiFi network is affected by selecting an appropriate combination of some RF parameters. These dense networks can occur in either managed environments such as within an enterprise, or in unmanaged environments as in an apartment building. We focus on the unmanaged environment as in a dense deployment of unmanaged home WiFi Networks. Considering that inexperienced users often operate with default configurations, the access points might be on the same channel and hence share the same collision domain. Each user may see low thoroughput as they interfere with each other. We conducted realtime experiments to understand the behavior of WiFi in dense networks. We explored configurations of the parameters which will optimize throughput in such networks. Second, we ask if it is possible for a node to adjust its RF parameters in a default environment to get a higher throughput. Also of interest were the affect on the throughput of its neighbours.

We can change a number of parameters in WiFi including Beacon Intervals, DTIM period, Turbo mode, Bit-rate selection algorithm, ACK timeout, CTS timeout, Slot time, Antenna Diversity, Preambles, Fragmentation threshold, RTS/CTS tranmission, and many more using MadWiFi [15] or other available drivers. We want to identify a combination of the following parameters which can lead to an optimized throughput for each user.

(1) Channel rate

(2) Transmit power level

(3) Physical carrier sensing

(4) Virtual carrier sensing

These are described in detail in Section 1.4.

1.2 Methodology

We use a top to bottom approach to solve this problem. The first challenge was to create a reliable testbed on which all the test cases could be run. Second, was to execute the experiments to study the behaviour of WiFi in IEEE 802.11 b/g with numerous possible combinations of transmit powers, channel rate and carrier sensing. Third, we conducted experiments to find if tuning these parameters in the presence of an external network will help increase the cumulative throughput. Finally, was to gather information and display it in a visual form to study the behavior of WiFi. We describe these efforts and conclude with possible future work on this project and discussions on current results.

1.3 Relevant Work

1.3.1 WiFi Testbeds

The section describes the relevant work done WiFi test beds. In [6], the authors emphasise the heterogeneity and limitations in the hardware implementation of the MAC layer in WiFi cards. They tested different cards on the basis of window size used during contention, EIFS used, and transmission rate reduction to test a propriety implemention of the ARP algorithm which could provide potential unfairness. Their results show that the MAC implementation from different vendors is not consistent with the IEEE 802.11 specifications, and conclude that an unfairness exists in commerical cards due to this variation in implementation rather than on other factors such as propagation conditions, laptops employed, and traffic generating tools used. Considering this, we have used the same laptop, traffic generator, operating system, and WiFi cards throughout our experiments.

In [16], a controlled wireless ad-hoc network testbed is designed for the experiments which can provide results much closer to reality than simulations. They have tried to make the testbed to be controlled, repeatable, similar to the RF environment, independent of the wireless technology being tested, easy to design, and easy to change the layout. Moreover, the testbed should be affordable, manageable, and support a quick testing of devices, and protocols. We have addressed the hardware, software, and obervational limiatations of this testbed in our design and analysis.

1.3.2 Optimizing Throughput

Bruno et al. in [11], address the problem of throughput reduction due to global interference among the access points (APs) of heterogenous networks and nodes. They make APs adapt frequencies in order to minimize the effect of the interference. Second, the nodes can choose the access point they connect to in order to get their fair share in an unmanaged high density WiFi network. The authors present a fully distributed self-configuring algorithm using Gibbs sampling [5] to address these issues. Channels are selected to reduce global interference and access points are selected to reduce the minimum potential delay of the user over an extended period of time. However, in a managed dense WiFi network, a centralized controller monitors the interference and the load on the network, and tunes the parameters of the APs for optimized throughput. This implementation of the algorithm won't affect any MAC layer protocol, and can be implemented easily by a firmware upgrade. Bruno et al. proved this algorithm analytically to yield optimal bandwidth sharing globally, and demonstrated it with comprehensive simulation results.

Bejerano et al. in [4], addresses the fairness and load balancing in WiFi networks using association control. To solve the problem of every user connecting to the AP with the highest RSSI, and to balance out the users to idle and less loaded APs; this paper presents an approach called **max-min fairness**. Assuming that the load on an AP is inversely proportional to the effective bit rate, and allowing the nodes to associate with multiple APs simultaneously; the fractional load balancing solution is found. Then, using the Shmoys and Tardos rounding method [17] it is extended for the unweighted greedy, and weighted but bounded-demand users. This was accomplished by using 2-approximation and 3-approximation algorithms simultanesouly, to connect to a single AP.

This approach not only provides equal bandwidth to each user network-wide, regardless of location but also provides time fairness among them.

Akella et al. in [1], studied the affect of interference in unmanaged dense WiFi networks on the client throughput. Accordingly, the degradation in throughput in these chaotic dense WiFi deployments can be as large as a factor of 3. The authors collected extensive data from several cities to prove their point and then proposed a self-managing network as the solution. They have developed an alorigthm called Power-controlled Estimated Rate Fallback (PERF), which uses the lowest transmit power for a link to keep up the highest data rate with the AP. The results show that a significant increase in throughput can be observed by using the PERF algorithm in the scenarios of multiple AP-Client pairs in close vicinity.

Ergin et al. in [7], found an interesting affect of AP density on WLAN performance, especially on TCP flows. It is presumed to be true that the factor affecting the throughput in an unmanaged dense WiFi network is the number of active clients. However, the study shows that it is instead affected by the number of interfering AP in the network. After rigorous experimentation using more than 100 nodes, and upto 4 AP in realtime using the ORBIT testbed [13], the results give a peculiar insight into the system. First, with an equal number of clients the throughput of four APs compared to a single one turned out to be half. Second, focusing on the TCP concentrated flows, the number of concurrently backlogged stations in the network is twice the number of active APs. These results effectively demonstrate the problems associated with chaotic dense WiFi networks. Our work focuses on the UDP traffic flows and its performance optimization.

Judd et al. in [10], developed a Channel Aware Rate Adaptation Algorithm (CHARM) to increase the throughput in a mobile wireless environment by adapting the transmission rate. The rate is adapted currently by a trial-and-error method. The rate is usually increased until the nodes start to observe significant packet loss, at which point the transmission rate is reduced. The CHARM algorithm works by, monitoring signal strength through the WiFi card and also takes advantage of the channel reciprocity to get channel information without the RTS/CTS overhead. To avoid stale channel information a time-aware signal prediction technique is utilized. The SINR threshold is calibrated automatically to avoid false information of transmit power, receiver noise, unreported interference, and multipath effects. This algorithm has been implemented using the Madwifi [15] driver, and has been tested in a controlled testbed environment, as well as real-time setting with stationary and dynamic nodes. The results show this algorithm yields better results in all scenarios compared to the traditional probe-based techniques used for rate adaptation.

1.4 Key Concepts

1.4.1 Channel Rates

The channel rate, which is also called the Data Rate, is the physical rate at which the connection is established between the nodes. In IEEE 802.11b the rates can go upto 11Mbps, whereas IEEE 802.11 a/g can support upto 54Mbps. In this thesis we will be evaluating the IEEE 802.11b and IEEE 802.11g standard by setting the rates to 1, 6 Mbps (lowest rates of IEEE 802.11b and IEEE 802.11g), 12, 36 Mbps (medium rates for IEEE 802.11g), and finally 11, 54 Mbps (highest rates for IEEE 802.11 b and IEEE 802.11g).

Table 1.1: Data Rate Specifications for IEEE 802.11b

Data Rate (Mbps)	Code Length	Modulation	Symbol Rate	Bits/Symbol
1	11 (Barker Seq)	BPSK	1 Msps	1
2	11 (Barker Seq)	QPSK	1 Msps	2
5.5	8 (CCK)	QPSK	1.375 Msps	4
11	8 (CCK)	QPSK	1.375 Msps	8

These rates give a useful overview of the whole spectrum from 1 to 54 Mbps. The rates 11 and 12 Mbps give a comparision of how the two different modulations affect the throughput we can obtain at almost equal data rates. The comprehensive details of the different rates used in IEEE 802.11b/g and their modulation and encoding scheme can be seen in Table 1.1 and Table 1.2 respectively.

Table 1.2: Date Rate Specifications for IEEE 802.11g

Data Rate(Mbps)	Modulation	Coding Rate	Coded bits per subcarrier	Coded bits per OFDM symbol	Data bits per OFDM symbol
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16-QAM	1/2	4	192	96
36	16-QAM	3/4	4	192	144
48	16-QAM	2/3	6	288	192
54	64-QAM	3/4	6	288	216

1.4.2 Transmit Power Level

Transmit power is the power which the WiFi card uses to transmit data, control and management packets on the wireless medium. This is an essential parameter of any wireless network, as the coverage area can be controlled, and efficient WiFi networks can be desgined. We propose that using lower transmit power can increase the overall throughput of the network, as with reduced transmit power the interference to the neighboring nodes is decreased which enhances the throughput.

We are using MadWiFi drivers to gain precise control over the IEEE 802.11 cards we are using. The transmit power of the card can be changed from 0dBm to 20dBm, in steps of 1dBm. In our experiments, we studied the behavior of the nodes with High (20dBm), Medium (10dBm) and Low(1dBm) transmit powers.

1.4.3 Carrier Sensing

Physical Carrier Sensing (PCS) is a function of the media access control (MAC) protocol of the IEEE 802.11 protocol which reduces collision in wireless networks. The physical channel is sensed by the transmitter before each transmission for any traffic. The transmitter will not send any data if it detects any signal on its particular channel, or strong signal on adjacent channels.

In this study we want to evaluate if it is really necessary to use the Physical Carrier Sensing (PCS) before each transmission. PCS causes the transmitter to defer whenever it hears a nearby co-channel transmitter. In a dense WiFi network there can be many nearby transmitters. However, the data can be successfully received even in the presence of interference as long as the Signal to Noise and Interference ratio (SNIR) is greater than a minimum threshold. The SNIR is the ratio of the transmitted signal to the co-channel and adjacent channel noise and interference. The minimum SNIR threshold depends on the channel rate and generally increases with increasing rate. Lower rates can tolerate lower SNIR and are thus are more robust to interference. With PCS, the transmitter sits idle while the tranmission could have been successfully made. Turning off PCS can avoid these idle periods.

Virtual Carrier Sensing (VCS) is a mechanism used to alleviate the so-called hidden node problem in wireless networks. Every packet contains a field in its header which specifies how long the transmitter and receiver will occupy the channel for their packet exchange. For instance, in a DATA/ACK exchange if a node waiting to transmit can't hear the ACK being transmitted, it can know that it is being transmitted from the Network Allocation Vector (NAV) in the DATA packet. VCS can optionally be facilitated with the transmission of two frames: Request to Send (RTS) and Clear to Send (CTS). Whenever a node wants to transmits it sends a RTS frame to the AP requesting for a time slot to use the channel. Once the request is granted by the AP, it broadcasts a CTS frame informing all nodes that the channel will be busy for the time indicated in the NAV.

In a dense WiFi network, using lower transmit power can decrease the PCS and VCS range. This reduced range allows the transmitter to communicate with nearby receivers without being interfered by relatively far-off interfering transmitters, as long as the SNIR is acceptable. Therefore, we expect that reducing the transmit power or turning off PCS could have a positive affect on the throughput of the network. If the nodes can automatically judge the level of transmit power to use, and when to use PCS, the aggregate throughput in dense WiFi networks can be increased.

The PCS of the wireless card was controlled by applying a patch [3] to the madwifi driver. This patch can be activated during runtime, so we don't need to reset the hardware each time. The PCS is disabled by reducing the slot times, EIFS (Extended Interframe Spacing), and SIFS (Short Interframe Spacing) to very small values. Therefore, the wireless card keeps transmitting data without listening to other transmissons in its carrier sensing range.

1.5 Thesis Document Roadmap

This document is organized into seven chapters. We start with defining the research question, methodology, relevant work and key concepts in Chapter 1. We then discuss the different test beds we designed, and how they compare to each other. We also analyze the advantages and disadvantages of each test bed in Chapter 2. Chapter 3 iterates in detail the hardware used during the experiments, the experimental setup, and a description of the scenarios which will be used in our experiments. Chapter 4 talks about the methodology we used to perform the experiments, script for automation, graphs and how they relate in answering our research question. Chapter 5 and Chapter 6 include different graphs developed for each test case, their analyses and discussion. Finally, the document ends with conclusion and future work in Chapter 7.

Chapter 2

Testbed Designs

2.1 Methodology

We explain the methodology we used to achieve the research objective in Section 1.4. First, we provide a description of the general testbed. Second, we explain different testbeds we have used to conduct the experiments. Third, we discuss the advantages and deficiency of each testbed. Finally, we compare the results we obtained from the different test beds. This study led to our final test bed design in Chapter 3.

There are several characteristics of a good testbed which we try to achieve in designing the testbed for our experiements.

- Repeated experiments should give similar results.
- Unknown parameters in the testbed are at a minimum.
- Interference to the testbed is well understood and minimized.
- Input and output to the testbed is controlled.
- Control of all the nodes in the testbed is centralized.

2.1.1 General Setup

The basic setup consists of four nodes A, B, C, and D. The nodes A and B form one communicating pair and C and D form another. Each pair represents a unique network. The Pair of A and

B will be called Pair-1, and the pair of C and D will be called Pair-2 in the rest of the document.

Pair-1 and Pair-2 are setup with vairous geometries for placing A, B, C and D in a shared collision domain to evaluate the throughput and interference. First, we have a case in which each network uses the medium alone (so no interference from the other network). Second, both networks share the medium. The throughput of both the cases are compared to evaluate the interference and loss of throughput caused by it.

In the following three sections, we will be discussing the different testbeds we used to address this experiment. We start with the description of choosing a particular testbed, its advantages and then the problems associated with it. The testbeds we used evolved during the course of the experimentation as we discovered problems associated with the testbeds being used.

The configuration of the nodes throughout the experiments was as follows unless stated otherwise:

- Mode : Ad-Hoc
- Rate : [1, 6, 11, 12, 36, 54] Mbps
- TxPower : [20, 10, 1] dBm
- Frequency : 2.412 GHz (Channel 1)
- RTS/CTS : Off
- Laptop : Lenovo R61e
- WiFi Card : Netgate PCMCIA adaptor with Hirose MS-147 antenna connector (CB9-GP)
- PCS : Both On and Off
- ESSID and IP addresses: The ESSID and IP subnet for A and B were different than the ESSID and IP subnet for C and D.
- Traffic: UDP traffic was generated using Internet Protocol Measuring Tool (IPMT) [9]. A sent traffic to B and C sent traffic to D.

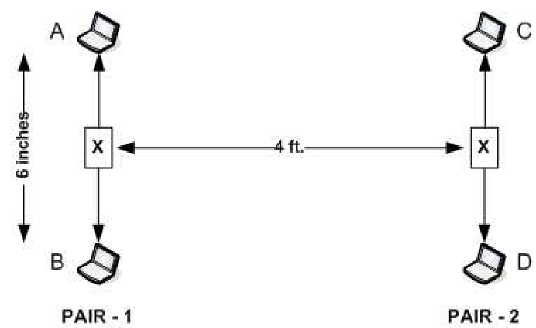

Figure 2.1: Testbed on a Tabletop

- Traffic Rate: Traffic was always sent at the maximum rate of the link. The IPMT tool checks the data rate set on the WiFi card and sends out the traffic at its full capacity.
- Performance Metric: sum of A to B and C to D throughputs.

2.1.2 Testbed on Tabletops

This is the first experiment performed to evaluate the basis of our core concept. As seen in Figure 2.1 we had all the nodes placed on tabletops. Pair-1 included two laptops connected to each other in ad-hoc mode, one acting as the Transmitter (traffic source) while the other as Receiver (traffic destination). The A to B and C to D distances were 15 cm. The distance between pairs was 1.2m. All distances were measured from interface card to interface card.

A second table top setup using cables is shown in Figure 2.2(a) and Figure 2.2(b) as per the block diagram shown in Figure 2.1. The attenuation on the cable was varied as per values of X and Y in Table 2.1, where Y is the attenuation between the pairs.

As both the networks were operating on the same frequency, they were sharing a collision domain. When the PCS was being performed it was observed that the two networks were dividing the channel between them, and the throughput obtained by each network was between 300kbps and 350kbps. It was also observed that the throughputs were higher at lower transmit powers, instead of when the cards where at their maximum power.

In the second part of the experiment, the PCS of Node A (TX) was turned OFF. This lead to the increase in the throughput of Pair-1 to 800kbps–900kbps on a 1Mbps channel. During this time period, the throughput of Pair-2 reduced very quickly and finally dropped down to zero.

Third, the PCS of Node C(TX) was turned OFF. Now, both the networks are transmitting data without considering the other network's transmitter. Simultaneous transmissions causes co-channel interference that could degrade throughput of both networks. On the contrary, without PCS the throughput of both the networks improved. We observed throughput of between 800kbps and 900kbps on each network, which is between 1.6 Mbps to 1.8Mbps collectively on a 1Mbps channel. The higher throughput on both networks can be explained by the low attenuation between the nodes, as compared to the higher attenuation between networks. As a result, the signal was much stronger than the interference.

These observations apply to the wired and wireless table top setups.

2.1.2.1 Advantages

This experimental setup had the following advantages:

- Simple to setup. The setup as shown in Figure 2.2(a) and Figure 2.2(b) requires no equipment beyond the laptops and WiFi Cards. The laptops are placed on the table with the external antennas of the WiFi cards connected in the fashion as shown in Figure 2.1.

- Serves as an initial test of the theory to show whether carrier sensing, and transmit power have any affect on the aggregate throughput of the network. Also, it can support or reject the proposed concept without going into much detailed and time consuming experimental testbeds.

- Closest to an actual dense WiFi environment. As there are no specialized RF isolations developed for this experiment, this testbed is similar to the practical environment. However, the dense WiFi network can include many more nodes interacting with each other.

2.1.2.2 Deficiencies

This experimental setup had the following deficiencies:

- Internal and external antennas: Our tests show that the internal antenna is functional even after an external antenna is connected which in these cases provided multiple paths between the nodes, which makes the cabled observations unreliable.

- Uncontrolled environment: As we see in Figure 2.2(a) and Figure 2.2(b), there is no control over the path taken by the transmitted/received data. Unknown factors affecting the throughput which we didn't take into consideration. There are many environmental variables which need to be considered: co-channel and adjacent channel interferences, interference from other RF devices operating in the lab, etc.

(a) (b)

Figure 2.2: Tabletop Testbed

2.1.3 Testbed with Cabinets

After doing experiments on the tabletop and to address the problems associated with it, we decided to do the experiments using Cabinets as housings for the Nodes. We know that environmental inteference can affect the results. Therefore, we wanted to provide better isolation to the nodes so the results are more repeatable. The setup is shown in Figure 2.3:

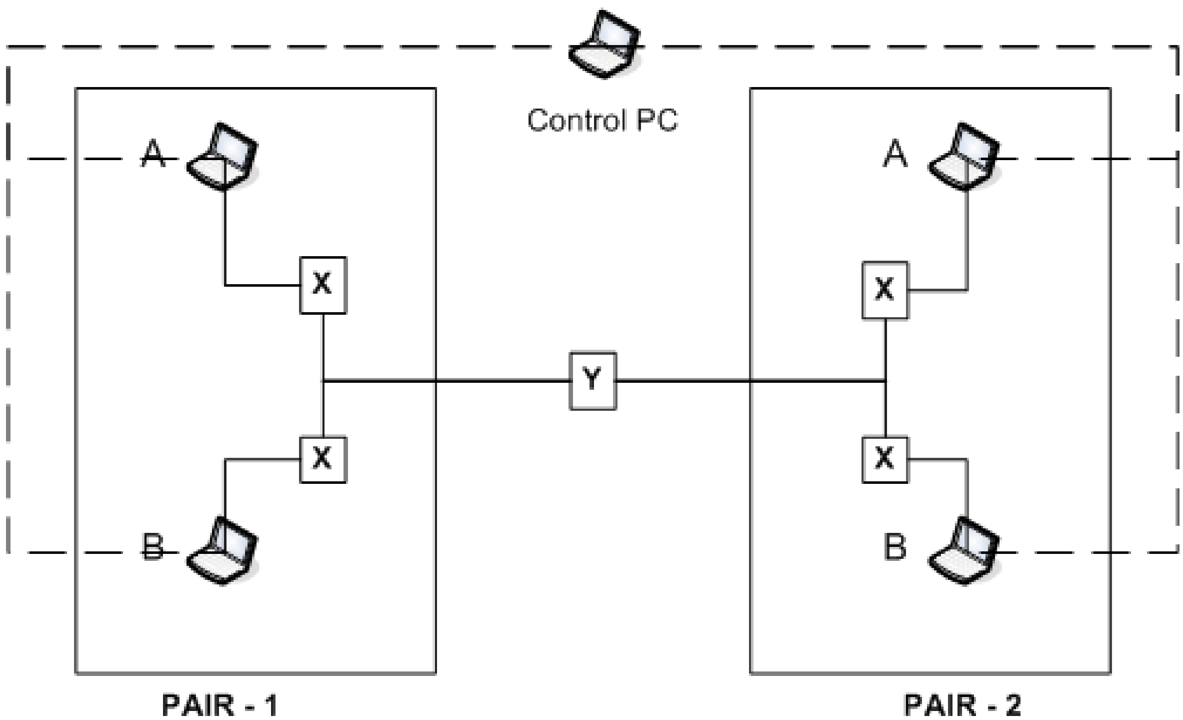

Figure 2.3: Testbed designed using metal cabinets to isolate networks

- The solid line boxes shown are actually metal cabinets in two different rooms almost 50 feet (15m) apart. These cabinets were used to serve multiple purposes. First, to reduce the co-channel interference and other RF noise in the environment. Second, to make sure that no communication was possible between two pairs over the air.

- All the laptops were connected through coaxial cables. The lighter solid lines in Figure 2.3 represent the RF connections. This connection was made using BNC cables, T-connectors and the External Antenna connectors to the PCMCIA Wi-Fi cards. We are using a coaxial cable approach to reduce the interference from the environment to a minimum level.

- The Wi-Fi cards were also covered with Carbon fiber cloth to reduce the leakage of RF signal from the card. It was observed that even after connecting the external antenna, the internal antenna continued to transmit. Without Carbon Fiber on the card, the radiation from the card made the whole cabinet a radiator. The carbon fiber overcomes this problem.

Pair-1 is in one cabinet and Pair-2 is in the other cabinet. We made sure that no connection could be made between the laptops in different cabinets. This required that the wireless card was wrapped with carbon fiber and the metal cabinet closed. The boxes with X and Y you see in Figure 2.3, are the attenuators on the links. We used different values of X and Y to represent the distance between nodes in the cabinets and also the distance between the pairs. All four laptops are connected to a switch and a fifth laptop, called the control PC that is also connected to the switch. All four nodes are controlled from the Control PC; this was done to make sure that the environment in the cabinet remains stable. We observed in our initial experiments that opening up the cabinets for management purposes led to unstable results, many minutes after closing the doors. The dotted lines are the Ethernet connections.

2.1.3.1 Advantages

This experimental setup had the following advantages:

(a) (b)

Figure 2.4: Cabinet Testbed 1

- Nodes are better isolated when we keep them in the cabinets and close the doors. As you can see in Figure 2.4(a), the cabinets provide more isolation than the nodes being on a tabletop.

- The cables we used in this case were double shielded coax cables, along with BNC connectors which provide a wired path to the RF signals. Also the attenuators as in Figure 2.4(b) could control the attenuation of the RF path.

- RSSI measurements were performed using a spectrum analyzer sitting inside the cabinet.

2.1.3.2 Deficiencies

This testbed had the following deficiencies:

- This setup still has many uncontrolled factors which might be affecting the measurements including not enough isolation, reflections inside the cabinets, coax cables do not support WiFi frequencies well etc.

- As the cabinets were not desgined for this purpose, it was not possible to properly ground them.

- As the cabinets shown in Figure 2.4(a) are metal, so they can start radiating themselves. When the boxes are closed for the experiments. We noticed that communication could still be performed within the nodes without the cable. The reasons can be that the isolation from the cabinets is not enough, or the cabinets themselves start behaving like a radiator when closed.

- Signal strength as received by the WiFi card are affected by the attenuation introduced but still the throughputs from A to B were persistently good. This showed that the path was not controlled by the physical attenuation introduced by us, and had other mediums of communications.

- The paths from Transmitter to Receiver is not known for sure (i.e. whether over the air or through the cable). Some data packet exchanges were still observed in cases where both of the cabinets were not closed carefully.

2.1.4 Testbeds with RF Isolation Boxes

The unstable and unpredictable nature of the observations in the previous two environments led us to think about a more controlled testbed in which the parameters affecting our experiments are more well known. After researching we settled on RF Isolation boxes claming to provide at least 60dB of attenuation from inside to outside the box (JRE Test: JRE-2218 Shielded Test Enclosure). The organization of the isolation box testbed can be seen in Figure 2.5

- Each laptop was put into its own RF Isolation Box, whereas the structure of the network remains the same as seen in the block diagram of Figure 2.3. It can also be seen in Figure 2.6 and Figure 2.7.

- As can be seen in the Figure 2.5, attenuation within the pairs is 2X, while the attenuation between the pairs is 2X+Y.

- The laptops were placed inside the box, and an ethernet connection was made from each laptop to a switch connected to the control PC. This was done to provide remote access to all machines and ensure isolation.

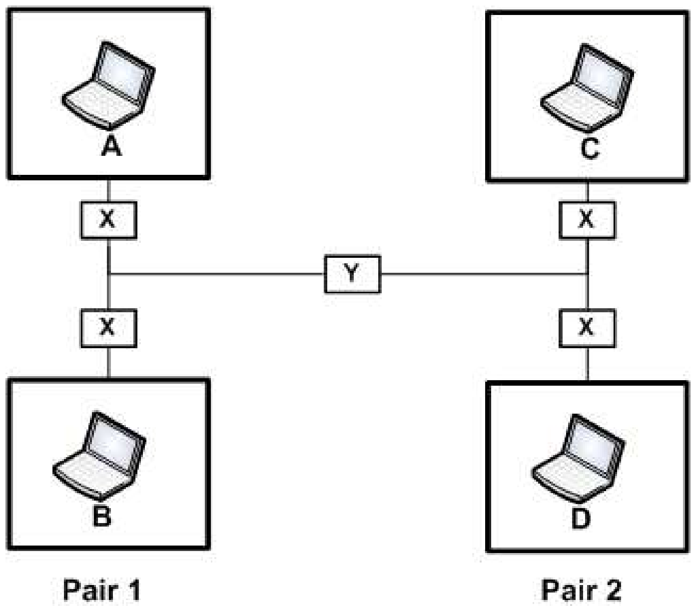

Figure 2.5: Testbed designed with RF Isolation box for each laptop keeping the connection same as Figure 2.3

- We used the external antenna of the WiFi Cards, and connected it to the SMA output of box.

- The RF signal from the SMA output of the box, now could be merged, attenuated, or monitored as needed.

- Connections were made as shown in the Figure 2.5. It was ensured that SMA type cables, connectors, and attenuators are used which have lower loss than BNC at 2.4GHz.

- A portable spectrum analyzer was used to measure the signal strength at a Resolution Bandwidth (RBW) of 10kHz. The signals were captured on the Spectrum Analyzer, which were closely analyzed after downloading to a computer.

(a) (b)

Figure 2.6: RFI Box Testbed 1

2.1.4.1 Advantages

This experimental setup has the following advantages:

- This setup provides the most isolation from the environment in our testbeds. As shown in Figure 2.6(a) the isolation was measured using a spectrum analyzer inside the box, and we observed an isolation of 40dB from inside to outside of the box.

(a) (b)

Figure 2.7: RFI Box Testbed 2

- The RF environment inside the box is kept constant by not opening the box at anytime once the experiment is started. Also, remote access to the box ensures realtime monitoring and control over the experiment through the connectors shown in Figure 2.6(b).
- Attenuation on any of the six paths between the nodes can be controlled with a precision of 1dB using fixed and variable attenuators of SMA type. The distance between the nodes, and the pairs can be controlled precisely and emulated using the attenuators on the paths.
- The SMA connectors and cables used provide much more accurate results compared to BNC used in the previous testbeds at the frequencies used in WiFi networks. It is perhaps because of the less leakage in these cables and connectors.

2.1.4.2 Deficiencies

This experimental setup has the following deficiencies:

- The power strips, ethernet and SMA connectors can also carry signals and interference in and out of the box compromising isolation.
- The cables, attenuators, and three-way splitters can have reflections. Different cables and attenuators might not be exactly equivalent creating asymmetric paths as opposed to our assumptions.

2.1.5 Testbed Comparision

The three testbeds were compared in terms of isolation within the pairs (Node A and Node B), as well as between the pairs (Node A and Node C).

Table 2.1 compares the three testbeds we have discussed: tabletops, cabinets, and RFI boxes. The table shows two experiments performed A to B and then A to C. The location of these nodes is the same as shown in the schematics in their respective sections.

We will briefly describe the measurement procedure. When the data is measured from A to B (within the pair), Y is set to 100dB to disconnect the two pairs. Second, when data is measured from A to C (across the pair), X is set to 0dB so the only attenuation affecting the path is Y. This exercise measures the pathloss between the nodes, and shows how isolated and controlled attenuation can be provided on our testbeds. The attenuations used for X and Y are Low(0 dB), Medium (50 dB), High(100 dB) and then "Disconnected" shows no cable connection at all.

The three sets, namely table top, cabinets and RFI Boxes evaluate the experimental testbeds for the signal of the received packets once attenuation is introduced on the cable and then corresponding throughput values are observed. This throughput shows real attenuation on the link followed by the signal and shows how much it corresponds to the explicitly introduced attenuation. The goal is to see if the paths followed by the signal is the same as we assumed.

First, we observed that in the Table Top case, the change in physical attenuation on the cable affects the RSSI measured by the spectrum analyzer at cable output but doesn't affect the throughput of the WiFi card. The technical specifications of the WiFi card specifies the internal antenna disconnects once external connections are made, which didn't prove valid in our experiments. This suggests that the signal is following a path not controlled by us and there is no isolation between the nodes.

Second, the Cabinet case shows relative change in the RSSI and throughput of received signal with changing physical attenuation. Between cabinets the signal strength and throughput decreases as expected. However, within the cabinets, the performance is similar to the tabletop experiment. We also observed that the measurements vary over a wide range and are very unstable. This unstability can be due to several factors like impedence mismatch in cables, attenuators, T-connectors, unstable RF environment, human interaction, and many more. We conclude that this testbed, while it gave some control over the testbed, is not very stable and precisely controlled when it comes to performing repeatable, reproducible, reliable experiments.

Finally, in the RFI Box case the RSSI measurements show a precise relative change in the signal strength when physical attenuations are changed. We observed that the values were very stable, and reacted correspondingly to even 1dB changes in attenuation. Though not recorded, it was observed that the throughput within pairs and between pairs both decreased with increasing attenuation as expected. This testbed can be considered the most reliable testbed among the three testbeds we explored in our experiments.

Table 2.1: Testbed Comparision

		TABLE TOP (dBm)		**CABINETS**		**RFI BOX**
A to B	**Y = 100 dB**	**Rx Signal (dBm)**	**Goodput**	**Rx Signal(dBm)**	**Goodput**	**Rx Signal (dBm)**
Strongest	X = 0 dB	**-20 to -30**	859Kbps	**-30 to -80**	859Kbps	-17
Low	2X = 50 dB	**-70 to -80**	850 Kbps	**-68 to -76**	750 Kbps	-69
Lowest	2X = 100 dB	**Less than -85**	850Kbps	**-81 to -86**	800Kbps	-91
Isolation	**X Disconnected**	**No Isolation**		**Unstable, No Isolation**		-74
A to C	**X = 0 dB**					
Strongest	Y = 0 dB	**-30 to -60**	800-850Kbps	**-40 to -45**	800-860Kbps	-22
Low	Y = 50 dB	**-80 to -85**	800-850Kbps	**-70 to -80**	11-250 Kbps	-73
Lowest	Y = 100 dB	**Less than -85**	800-850Kbps	**-80 to -85**	0Kbps	-95
Isolation	**X Disconnected**	**No Isolation**		**Uncontrolled Isolation**		-73

The lessons learned during these experiments helped us to understand the best practices in the test beds. We incorporated these lessons learned into our own test bed used for our experiments.

Chapter 3

Experiments Performed

3.1 Hardware Used

In the previous section we discussed different testbeds which could be used to do the experiments. Each testbed has its own limitations and advantages. We will be using the testbed with RFI boxes, which according to our tests, is the most reliable and stable. Based on our experience, we modified the RFI box setup to provide us with more control over the setup. The components used in the physical setup are as follows:

(1) Four RF Isolation (RFI) boxes.

(2) Five laptops (Lenovo R61e): Four nodes, and one remote monitoring and control.

(3) Four Atheros chipset WiFi Cards (Netgate CB9-GP).

(4) Four 1 to 3 way RF splitters - SMA type.

(5) External antenna connector for WiFi Cards.

(6) Two ethernet switches.

(7) RF Cables to provide a controlled RF environment (SMA Type).

(8) Fixed and Variable SMA RF attenuators (S.M. Electronics: SA3550S, SA6-XX).

3.2 Exerimental Setup

The physical connections were made as per Figure 3.1. An explanation of the physical setup follows:

3.2.1 Connections

- Each of black rectangular box represent an RFI box.
- The RFI box is provided with power strips on the inside to power laptops which enable us to run experiments independent of the battery life.
- There are vents with active fans to avoid high temperatures inside the box.
- Each laptop is connected to the inside of the serial port using Ethernet to serial connectors provided by the box manufacterer.
- The external antenna on the WiFi card is connected to the input of 1-to-3 way splitter. This splitter distributes the signal in three equal paths which transports it to one desired path, and two interference paths for the other pair's transmitter and receiver.
- All the laptops are connected through a switch. They have the same subnet so that they can be controlled from the Monitoring PC (which was shown in Figure 2.3). This helps in controlling and monitoring all the nodes from one centralized location.
- There are six physical paths among all four nodes, and attenuation on each path is controlled with a physical attenuation. Our evaluation of the testbed proves that this is the only path of tranmission between the nodes so the observations are minimally affected by any other parameter.

3.2.2 Attenuator Combinations

- We have six different attenuators in this testbed, each of which can be controlled from 0 to 100dB with a precision of 1dB. The setup has 1 trillion different combinations of

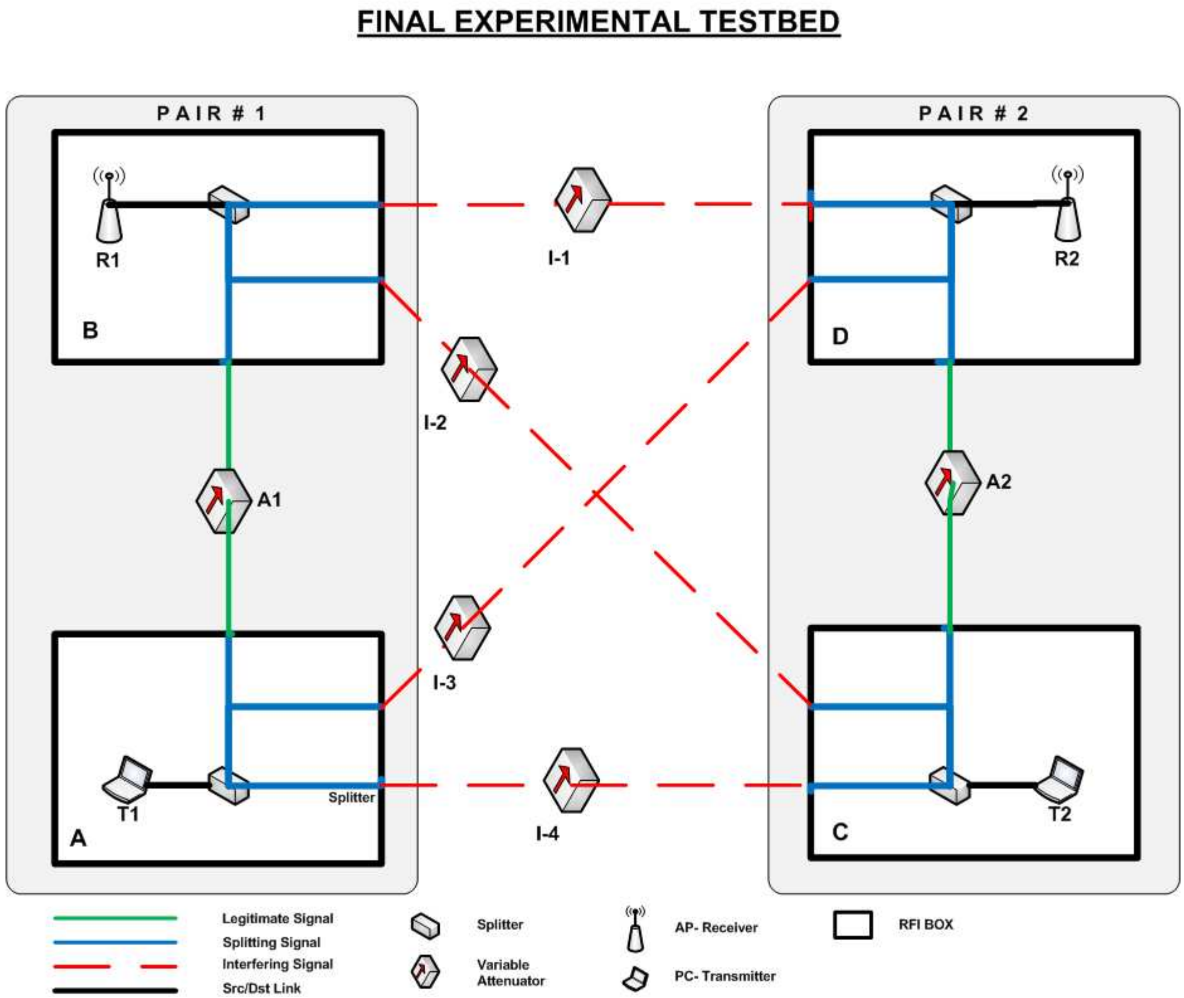

Figure 3.1: The final experimental testbed using RFI boxes.

attenuation.

- Instead of going through all of the possible combinations, we decided to have a symmetric attenuation set. These sets are defined in Table 3.1

- We have three symmetric attenuation pairs in this testbed which are A1-A2, I1-I4, and I2-I3. The attenuation of these three pairs are always kept equal to simplify the experiment. Note that the attenuation is emulating the distance between the nodes, which fixes the distance between symmetric pairs to be same.

- For each network attenuation value shown in row-1 of Table 3.1 we have four values of between network attenuation (row-2). The cross network attenuation values (row-3) are calculated geometrically from P1 and P2.

- We are using 40dB as the lower value of attenuation. The reason is that at less than 28dB attenuation, our tests show that the front-end of the WiFi card is overloaded and throughput is negatively affected. We use 40dB of attenuation to be sure that the front-end of the WiFi card is not over-loaded.

Table 3.1: Symmetric Attenuator Pairs

S.No.	**Attenuator Pair**	**Attenuation Used (dB)**	**Description**
1	A1 & A2	40, 80	*Near and Far cases*
2	I-1 & I-4	40, 80	*Geometric Progression*
3	I-2 and I-3	50, 80, 86	*Geometrically Calculated*

3.3 Characteristics

After rigorous analysis, we developed a testbed which provides not only precise control over the testbed but also provides repeatable results. The characteristics of this testbed are as follows:

(1) Environmental interference is minimized in the box.

(2) Except for the defined paths, a node within an RF Isolation box is not able to communicate with the other three nodes as the minimum isolation through the box is 40dB. So, the cumulative isolation between any two nodes is at least 80dB, in addition to the propagation losses which are at least 35dB.

(3) All four nodes are remotely controlled by a Monitor node, which are all connected using an Ethernet feed-through into the RFI box. This remote controlling of the nodes without opening the RFI boxes ensures that the environment within the RFI box remains very stable.

(4) The RF signal from the WiFi card is passed through a one-to-three way splitter which connect to different SMA ports inside the RFI box. This is shown in the Figure 3.1 with solid blue lines. The signal to each box is sent though a different cable to make sure that the attenuation on each path could be controlled very precisely. The three-way splitter divided the signal on any give path by three (5dB attenuation).

(5) As shown in the figure solid green lines are the desired signals, and dashed red lines are the interference.

(6) The attenuation in the diagram emulates the distance between nodes. So, lowering all attenuations can bring the nodes very close and vice versa. This would give us a precise control over the accurate test scenarios we want to create and study.

3.4 Test Cases Used

The details of the attenuation values are shown in Table 3.2. First, both pairs are close. Also, the TX and RX within each pair are also close. Even though the desired signal is strongest, the interference in this case is also the highest. Second, the TX and TX within each pair remain close but the interfering pair is moved further. In this case, the desired signals are stronger than the interfering signals. Third, the TX and RX within each pair are moved apart and the interfering pair is brought close. In this case, the desired signals are weak compare to the interfering signal. In the last case, both pairs are moved far apart. This emulates a scenario in which both the desired and interfering signals are weak.

Table 3.2: Four selected cases for current experiments with attenuation, A in dB and equivalent distance, D in meters for each of the 6 paths

	Attenuator Combinations											
C.No	**A1**		**A2**		**I-1**		**I-2**		**I-3**		**I-4**	
	A(dB)	D(m)	A(dB)	D(m)	A(dB)	D(m)	A(dB)	D(m)	A(dB)	D(m)	A(dB)	D(m)
1	40	20	40	20	40	20	50	25	50	25	40	20
2					80	40	80	40	80	40	80	40
3	80	40	80	40	40	20	80	40	80	40	40	20
4					80	40	86	43	86	43	80	40

3.5 Cable loss

The loss of each cable used in the experiment was calculated separetly. The data collected is shown in Table 3.4. We used two brands and four different lengths of cables. We connected each cable between a Signal Generator (SG) and Signal Analyzer (SA). The TX frequency was 2.412GHz. The difference between the sent and received signal determines cable loss. For each cable, five different signals (from 5dBm to -30dBm) were sent and then cable loss was calculated by averaging the loss at different signal strengths. Also, we can see the total loss due to cables and connectors between two nodes in Table 3.3.

Table 3.3: This is the total loss between nodes in dB for the 6 cabled paths

Node-1	**Node-2**	**Loss (dB)**
A	B	4
A	C	5.8
A	D	6.2
B	C	6.2
B	D	5.8
C	D	4

Table 3.4: Shows the loss for different cables used in the experiments at 2.412GHz

Cable Brand	**Length (ft)**	**Iteration**	**SG (dBm)**	**SA (dBm)**	**Difference**	**Avg Loss (dB)**	**Loss/ft (dB)**
LMR-0240	*1*	1	5	4.84	0.16	*0.20*	*0.2*
		2	0	-0.13	0.13		
		3	-10	-10.29	0.29		
		4	-20	-20.10	0.10		
		5	-30	-30.30	0.30		
	3	1	5	4.78	0.22	*0.30*	*0.1*
		2	0	-0.19	0.19		
		3	-10	-10.47	0.47		
		4	-20	-20.29	0.29		
		5	-30	-30.34	0.34		
	6	1	5	4.41	0.59	*0.63*	*0.1*
		2	0	-0.56	0.56		
		3	-10	-10.63	0.63		
		4	-20	-20.65	0.65		
		5	-30	-30.70	0.70		
PE300-120	*12*	1	5	2.24	2.76	*2.48*	*0.2*
		2	0	-2.53	2.53		
		3	-10	-12.36	2.36		
		4	-20	-22.32	2.32		
		5	-30	-32.44	2.44		

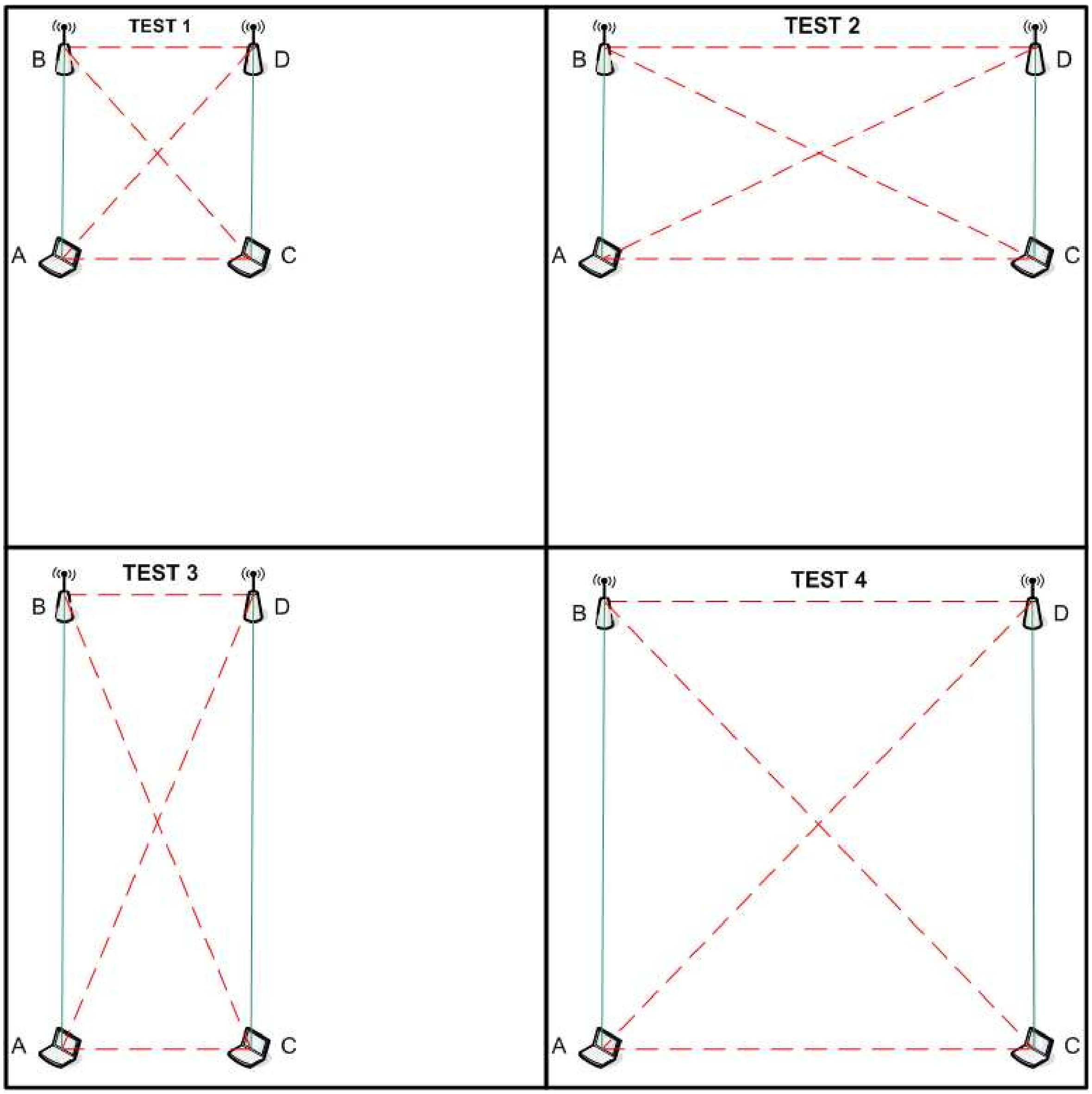

Figure 3.2: Four selected cases for current experiments

Chapter 4

Approach and Methodology

4.1 Methodology

Figure 3.1 shows the nodes, connections, and attenuation placement during each test. Also, Figure 3.2 shows the four standard cases we have used in three different scenarios. These three scenarios are as follows:

(1) In Scenario-1, the nodes are place as shown in 3.1 and both pairs go through the same combination of data rate, transmit power, and carrier sensing simultanously.

(2) In Scenario-2, the position of transmitter and receiver of Pair-2 in Figure 3.1 is swapped. The rest of the setup remains the same and both pairs go through the same combination of data rate, transmit power, and carrier sensing simultanously.

(3) In Scenario-3, we use the physical placement of Scenario-1. The difference is Pair-2 is fixed with a certain combination of data rate, transmit power, and carrier sensing. Then, Pair-1 goes through all the combinations of data rate, transmit power, and carrier sensing on both of its nodes.

Scenario-1 emulates an environment in which both pairs can be tuned by us to obtain the highest cumulative goodput. Scenario-2, verifies the results from Scenario-1, and provides two more cases in which the interference to the nodes are changed. This helps in understanding the interaction among nodes and change in goodput in these scenarios. Scenario-3, emulates an external network with fixed RF parameters, where we try to tune our parameters (Data Rate, Transmit Power and Carrier Sensing) to optimize cumulative goodput.

For this section, the four physical experimental test cases we have chosen to observe the behavior of WiFi in differnet scenarios. In each of the four tests performed, a standard procedure was followed as per the pseudocode shown below. In each of the four test cases, the physical attenuation has to be set manually, and the rest of the process is automated. The automated script used the algorithm in the pseudocode below.

PSEUDOCODE:

```
Select Physical Attenuation Set Manually
   For ( N = 1 to 4 ) // For four non-correlated sets of data
      For all Rates in the RateArray
      do SetRate while(rate not verified) // keep trying until set
          For all TxPowers in PowerArray
          do SetPower while(power not verified) // keep trying until set
                  For all Carrier Sensing combinations in CSArray
                  SetCarrierSensing
                  Run Measurement Test
                  Capture all data
                  Compute Average
```

The pseduocode shows an overview of the automated experimental testbed, which controls, verifies, performs, and logs the data in a defined format. We have three parameters data rate, transmit power, and carrier sensing which are changed throughout the experiment. First, the data rates and transmit power are provided into RateArray and PowerArray. Second, the rate is iterated in a loop by the script. The script keeps verifying, and setting the rate until it is set to the desired value. Third, the transmit power is set within a loop in the rate script. The transmit power level is also monitored until it is set to the desired value. Finally, the carrier sensing is set and the testbed is configured to start exhanging UDP traffic and measure the link throughput. The UDP traffic generator is started on both the transmitters, and the traffic is received at the corresponding receiver. The script goes through all the combinations of rate, power, and carrier sensing which in our testcase for this thesis are:

- Date Rate(Mbps): 1, 6, 11, 12, 36, 54
- Trasmit Power (dBm): 20, 10, 1
- Carrier Sensing: Both On and Off for each pair (4 combinations)

Thus, we have 72 (6x3x4) different testcases. Each testcase is run for 100 seconds, which gives us 100 one second throughput measurements, the average of which is calculated. This process is repeated for all 72 cases, which takes around 2 hours to compelete. To make sure that built-in shifts in the clocks of transmitter and receiver don't lead them out of synchronization, a three-way handshake was performed after every two cases. This whole 72-case process is repeated four times, which gives us four different sets of data which are not correlated in time. This provides the data with robustness to any unusual behavior in time. Finally, a mean is evaluated from the four means calculated from the four non-correlated data sets.

4.2 Tables and Graphs

This section briefly explains how the data is gathered and displayed. We have three tables and four graphs for each of the four cases of physical attenuation set. The following information is represented using these tables and graphs:

- The Table "Aggregate Throughput" shows the sum of the throughputs of both networks at that instance. As both of the networks share the same channel and data rate, the theoritical limit to this throughput is the data rate being used. This data is represented with the first two graphs, one of which shows actual data and the other uses logarithmic scale to enhance the details at lower rates.

- The Table “Normalized Throughput” shows the aggregate througput normalizd by the throughput of an isolated network without any interferences (a best case scenario). The third graphs represents this table, and clearly shows if an approach gives extra capacity for a particular case. This graph ranges from 0–200%, where values greater than 100% identify cases in which extra capacity can be achieved using a particular combination of parameters. The isolated network throughput is shown in Table 4.1 and Table 4.2 below.

Table 4.1: Goodput for isolated Pair-1 with 40dB between TX and RX

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.61	0.74	0.61	0.74	0.73	0.70	0.73	0.70	0.70	0.72	0.70	0.72
6		4.10	4.08	4.10	4.08	3.99	4.02	3.99	4.02	3.93	4.23	3.93	4.23
11		6.25	6.62	6.25	6.62	5.90	6.14	5.90	6.14	5.92	6.57	5.92	6.57
12		7.62	7.78	7.62	7.78	7.28	7.70	7.28	7.70	7.42	8.32	7.42	8.32
36		18.31	19.31	18.31	19.31	18.04	18.92	18.04	18.92	18.78	20.04	18.78	20.04
54		25.40	26.33	25.40	26.33	24.21	27.07	24.21	27.07	25.21	27.51	25.21	27.51

Table 4.2: Goodput for Isolated Pair-1 with 80dB between TX and RX

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.57	0.71	0.57	0.71	0.70	0.69	0.70	0.69	0.62	0.60	0.62	0.60
6		4.13	4.04	4.13	4.04	3.86	3.41	3.86	3.41	3.29	0.77	3.29	0.77
11		6.19	6.41	6.19	6.41	5.91	5.97	5.91	5.97	2.36	2.54	2.36	2.54
12		7.50	7.80	7.50	7.80	7.13	6.88	7.13	6.88	4.85	1.00	4.85	1.00
36		18.01	19.10	18.01	19.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
54		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

- The Table “Fractional Throughput” shows the fraction of total throughput that is from the first network. This gives a clear understanding of fairness in each case, and if more bandwidth is consumed by one of the networks. A value of 100% means Network-1 is getting all the throughput. A value of 0% means Network-2 is getting all the throughput. A value of 50% means the throughput is equally shared.

- In all the tables aggregate throughput is shown in Mbps. The fractional and normalized throughputs are in percentages. The normalized and fractional throughputs can go upto 200% and 100% respectively.

- Each network, Network-1 or Network-2 in each table can have values of 0 or 1. These values show the presence (0) or absence (1) of carrier sensing with the pair. Within a network, the carrier sensing is set for both transmitter and receiver of that pair.

- The data from each of the physical attenuation cases is represented using four different graphs to get a good understanding of the different aspects of the results.

- We have three different styles of lines used in the graphs to represent the transmit power. The different styles used are solid (txpower=20dBm), dashed (txpower=10dBm), and dotted (txpower=1dBm).

- The colors represent different combinations of carrier sensing used with the pairs. Green line shows that both of the pairs are doing CS. Red shows that both pairs are not doing CS. Dark blue represents the case in which only Pair-1 is CS. Last, light blue represents when only Pair-2 is CS. It is to be noted that both nodes of a pair have always the same CS state.

- Each data rate in the graph has twelve data points plotted (3 power levels by 4 carrier sensing combinations). Each of the data point is the sum of the means of the throughputs of Pair-1 (P1) and Pair-2 (P2).

- For the first two graphs showing aggregate throughput in each test, the mean value for each pair is calculated using the four mean values calculated from the four non-correlated experiments performed for 100 seconds each.

- The second and third graphs show the normalized and fractional throughput as compared to the throughput of pair 1 in isolation as discussed earlier in this section.

Chapter 5

Results and Discussion for Hypothesis-1

5.1 Hypothesis-1

This chapter discusses the results obtained in the scenario when both the pairs are in our control. This scenario represents when both pairs can communicate with each other. We assume that both pairs are using the same parameters of data rate, and transmit power simultanously. However, the carrier sensing on both the pairs can be different. This chapter analyzes Scenario-1 and Scenario-2, as described in Chapter 4. The description of the tests related to each scenario is as follows:

- Scenario-1: Test1a, Test2a, Test3a, Test4a
- Scenario-2: Test1b, Test2b, Test3b, Test4b (Transmitter/Receiver of P2 swapped)

5.2 Test 1a: All Nodes Close Together

In this test, all nodes are close together. We can consider this as the worst case in terms of interference between the nodes. So, we expect that turning off carrier sensing wouldn't be of much help. We have the following observations from the graphs shown in Figure 5.1 and Figure 5.2:

- Disabling carrier sensing (CS) on both pairs only slightly helps at 1Mbps with the lowest transmit power (1dBm). The reason is the frames are most robust at this rate, and interference is also least at 1dBm.

- It is very clear that at higher rates, greater than 11Mbps (which are less robust), turning off CS on both pairs yields much worse throughput.

- Using CS on both pairs, provides consistant results at different power levels. It clearly shows how the two pairs (all four nodes) co-ordinate with each other to share the collision domain equally.

- Turning off CS at P1 helps improve the overall throughput all rates except 1Mbps. It is interesting that only at 54Mbps, medium transmit power has the highest throughput. However, in all other cases, the difference between throughputs from different power levels is minimal.

- Turning off CS only at P1 and only at P2 do yield the same results. This shows asymmetry in the setup.

- The data shows that we can get around a 10% of increase in cumulative throughput by not using carrier sensing on P1 or P2.

We have the following observations for normalized, and fractional throughput graphs shown in Figure 5.3, and Figure 5.4:

- In Figure 5.3, we see that bandwidth is equally divided between P1 and P2 when both do CS. It is interesting to note that the behavior is independent of the rate and transmit power of the node.

- It is very clear that P1 is the dominant network here when it comes to disablig CS. It has most of the throughput in every case.

- When both P1 and P2 are not using CS, we see that higher transmit power works better at lower rates and vice versa.

- It is unusual to see that P2 usually get less than 10% of the throughput, even when not doing CS regardless of the transmit power used.

- In Figure 5.4, we see that even CS does get more than 100% of the throughput and at all different rates.

- At 1Mbps, no CS on both pairs yields 200% of throughput at lowest transmit power. This shows that the mutual interference does increase with the transmit power of the nodes. However, at higher data rates it doesn't work well as they are less robust and more prone to interference.

- We see that P1 or P2 not doing CS always achieves more than 110% throughput from the network.

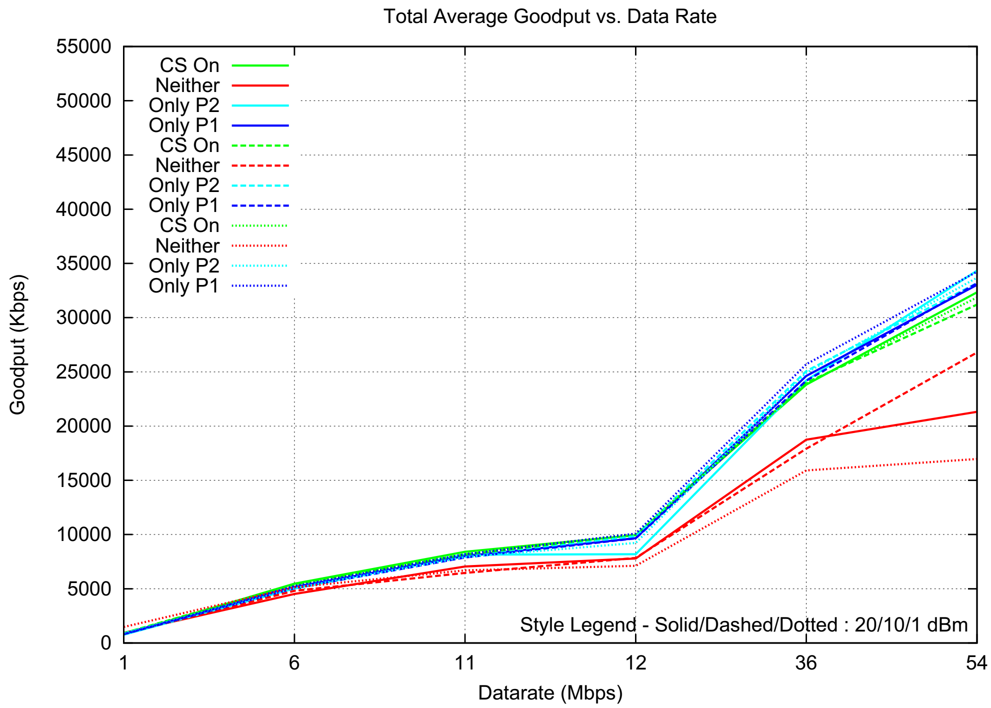

Figure 5.1: Test 1a: All nodes close together

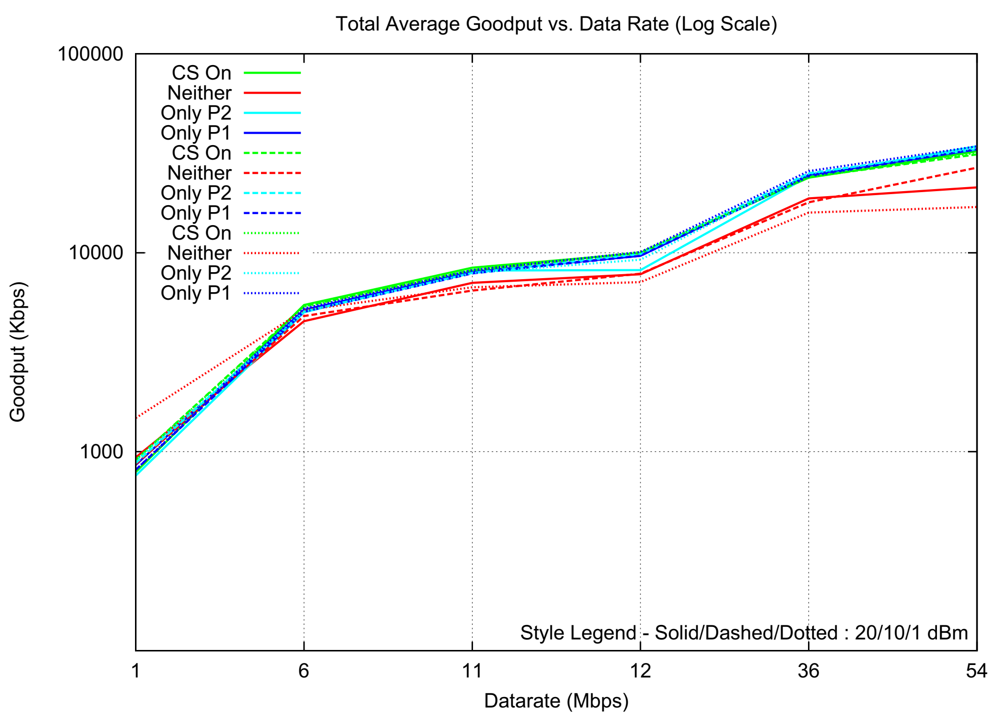

Figure 5.2: Test 1a: All nodes close together

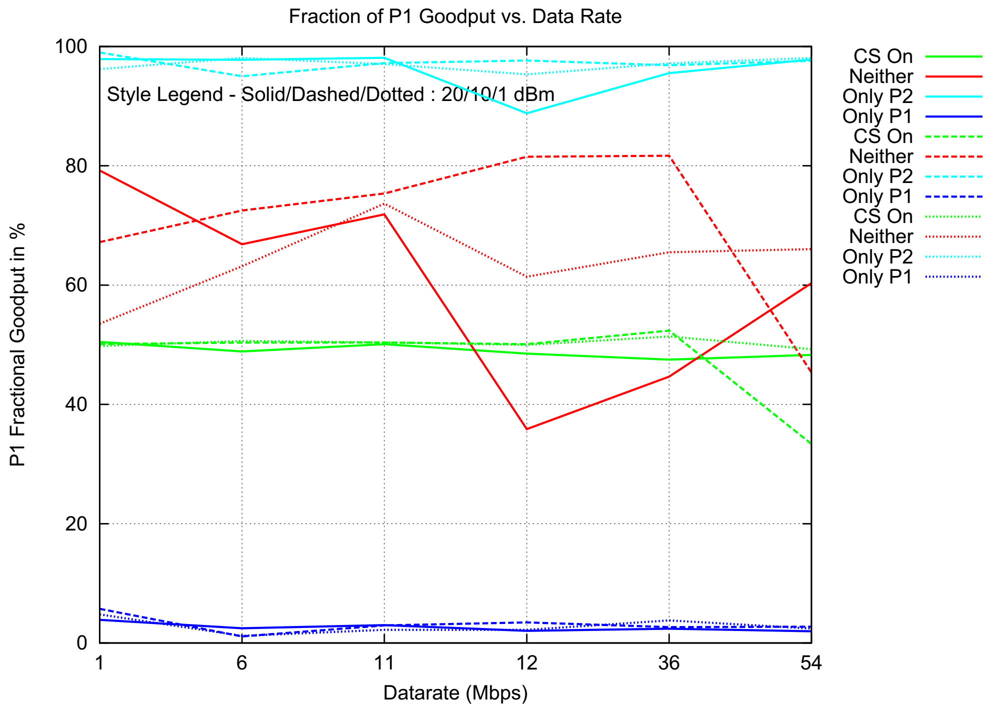

Figure 5.3: Test 1a: All nodes close together

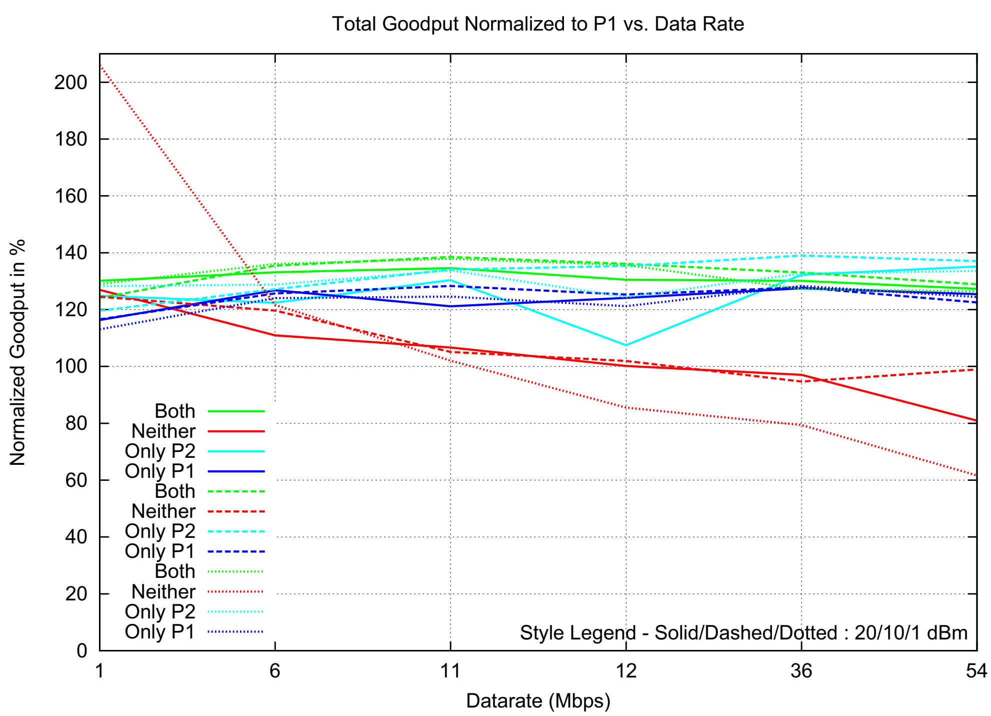

Figure 5.4: Test 1a: All nodes close together

5.3 Test 2a: Nodes Close, Interference Far

In Test2, the nodes are close together and the interferences are far. We can consider this as the best case because the SNIR at the receviers is expected to be highest in this case. We have the following observations from the graphs shown in Figure 5.5 and Figure 5.6:

- In Figure 5.5, we see that there is very minute increase in throughput at lower rates (less than 36Mbps) with disabling CS. Ideally we would assume that lower rate should be more robust and be able to yield more throughput in no CS cases, which is not the case.
- At 36Mbps, turning off the CS on both P1 and P2 does help to increase the cumulative throughput by 30%. However, at 54Mbps, the throughput can be 100% by not using CS.
- It is also observed that turning off CS at one of the pairs P1 or P2 also helps in squeezing extra throughput, but only at medium or high transmit powers. It is because weak signal are again more prone to interference and thus are not able sustain 54Mbps of data rate.

We have the following observations for normalized, and fractional throughput graphs shown in Figure 5.7 and Figure 5.8:

- In Figure 5.7, we observe that the channel is equally shared between both pairs in case of CS or no CS. It is to be noted that even without CS, both networks can share all of the channel effectively.
- We see that at lowest (1Mbps) and at high rates (36Mbps and 54Mbps) P1 has most of the channel. We have observed that asymmetry in the previous case too.
- We observe that at lower transmit powers and no CS, P1 will use most of the channel, but P2 will not do the same in the inverse scenario. Again asymmetry in the behavior of P1.
- It is noticed that at all rates, the highest transmit power (20dBm) better supports fairness between the networks.

- In Figure 5.8, in almost all of the cases we see throughput values are greater than 100%.

- At every case higher transmit power yield better throughput except at 54Mbps, where lower transmit powers get better throughput. The reason is that as we reduce the transmit power, interference among the nodes is also decreased.

- It is clear that no CS always yields much better throughputs (always more than 160%) compared to the CS cases. The difference is more visible at the data rates of 36Mbps, and 54Mbps.

- The results we observed while performing experiments with table tops were reproduced here. We could see throughput increasing to 200% in some cases.

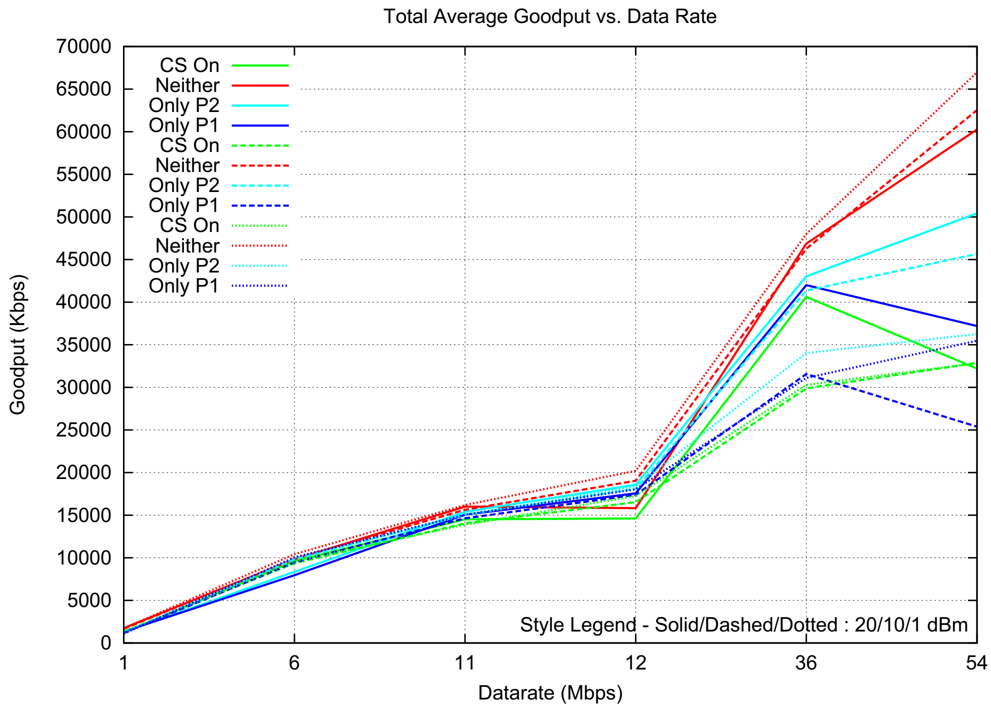

Figure 5.5: Test 2a: Nodes Close, Interference Far

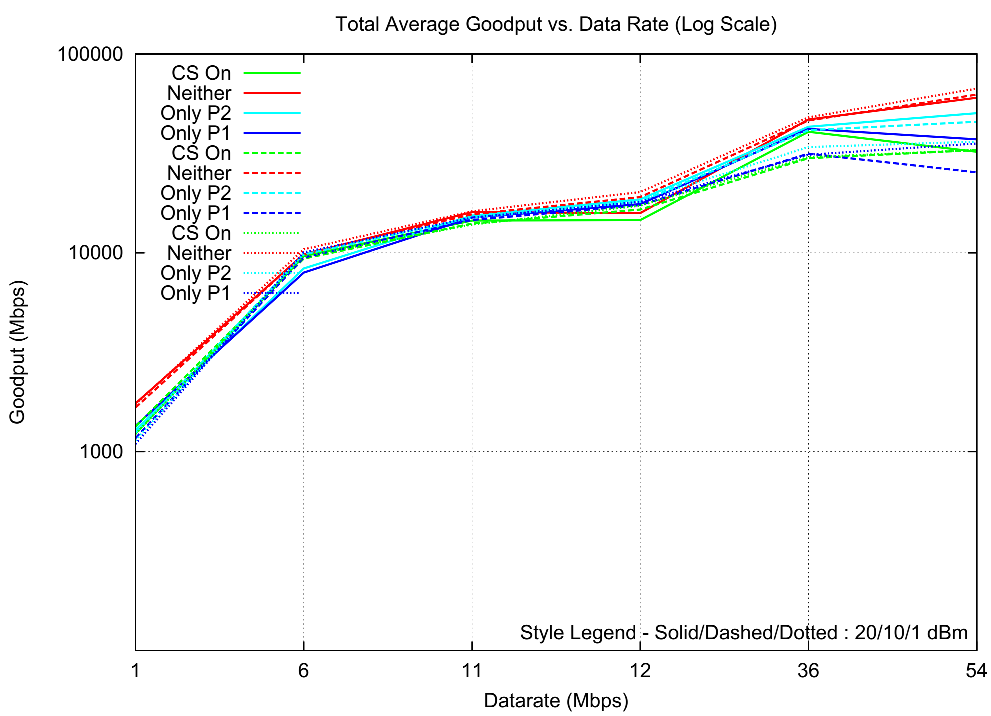

Figure 5.6: Test 2a: Nodes Close, Interference Far

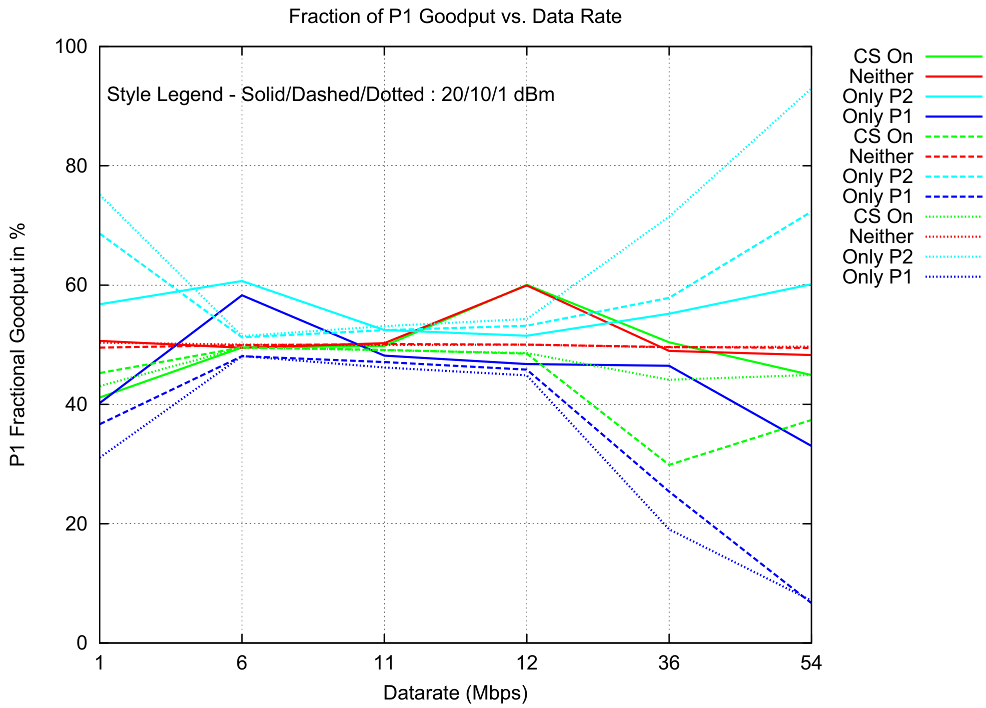

Figure 5.7: Test 2a: Nodes Close, Interference Far

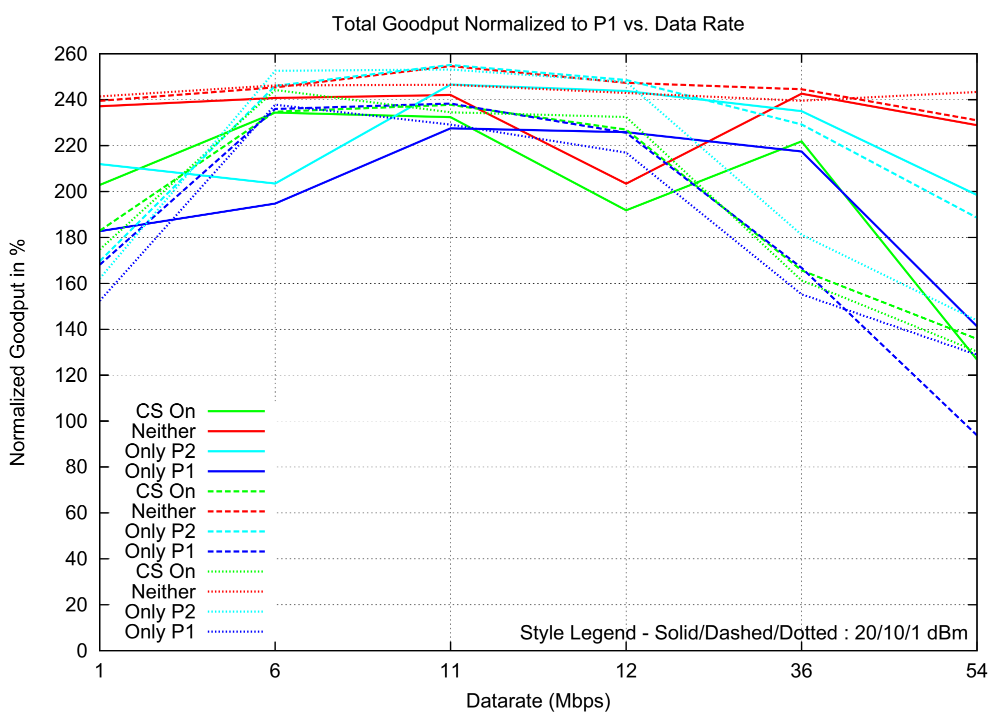

Figure 5.8: Test 2a: Nodes Close, Interference Far

5.4 Test 3a: Nodes Far, Interference Close

In Test3, the nodes are far away and the interferences are close. This case has the lowest SINR, as the interference is 40dB stronger than the signal. We have the following observations from the graphs shown in Figure 5.9 and Figure 5.10:

- In Figure 5.9, we see that disabling CS on the pairs doesn't help at all. It is because the pairs are so close to each other that interference among the pairs is too high.
- In the case of no CS, the higher transmit powers are easier to be received by the reciver despite the very high interference.
- Lower rates (1Mbps, 6Mbps, 11Mbps, 12Mbps) can still operate at mid and low transmit power levels.
- The default setting of CS, seems like the best option in this scenario as it yield almost equal throughput if compared to modified (only P1, or only P2 CS) cases.
- We see that at 36Mbps only transmit power 20dBm works reasonably. However, at 54Mbps (least roboust from interference) almost every combination fails because of insufficient SNR.

We have the following observations for normalized, and fractional throughput graphs shown in Figure 5.11 and Figure 5.12:

- In Figure 5.11, we see that CS yield equal sharing of bandwidth between P1 and P2 at lower rates and mid (10dBm) to high (20dBm) transmit powers.
- The data essentially is all over the place, and shows the instability in the environment when the interference is high.
- At high rate (36Mbps) and highest transmit power (20dBm) we see that we do get throughput but the bandwith is not equally shared.

- The asymmetric behavior of P1 is still noticed. It takes away all the bandwidth when not doing CS, while P2 can't do the same.
- In Figure 5.12, disabling CS on P1 and transmitting at 20dBm does help in getting throughputs greater 100%. Even though it is a selfish behavior we do get the highest cumulative throughput in this case.

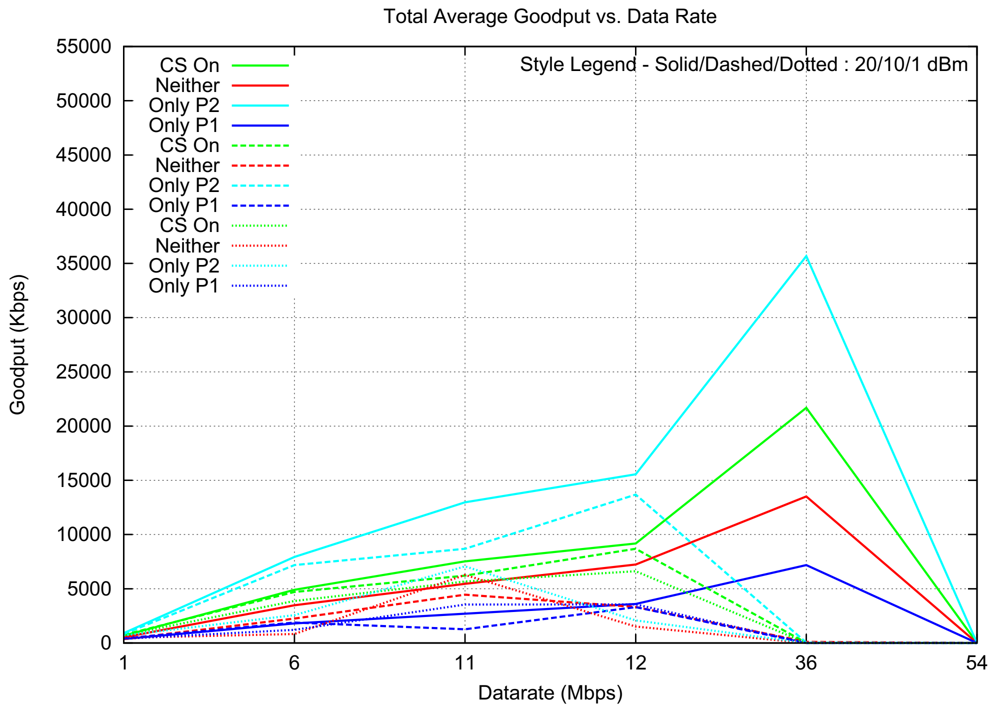

Figure 5.9: Test 3a: Nodes Far, Interference Close

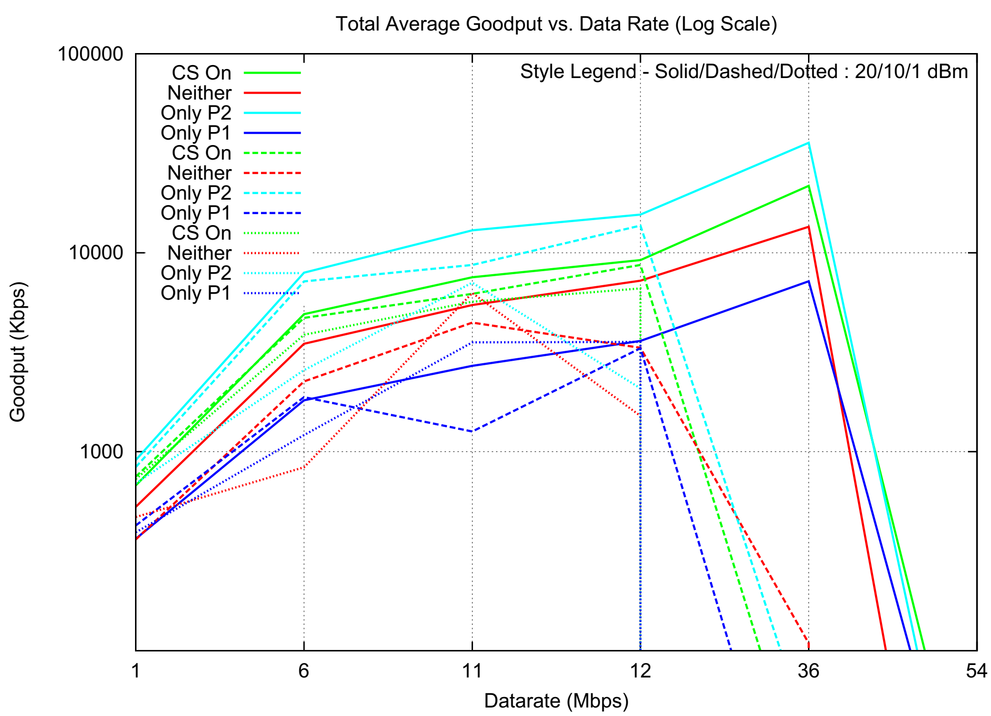

Figure 5.10: Test 3a: Nodes Far, Interference Close

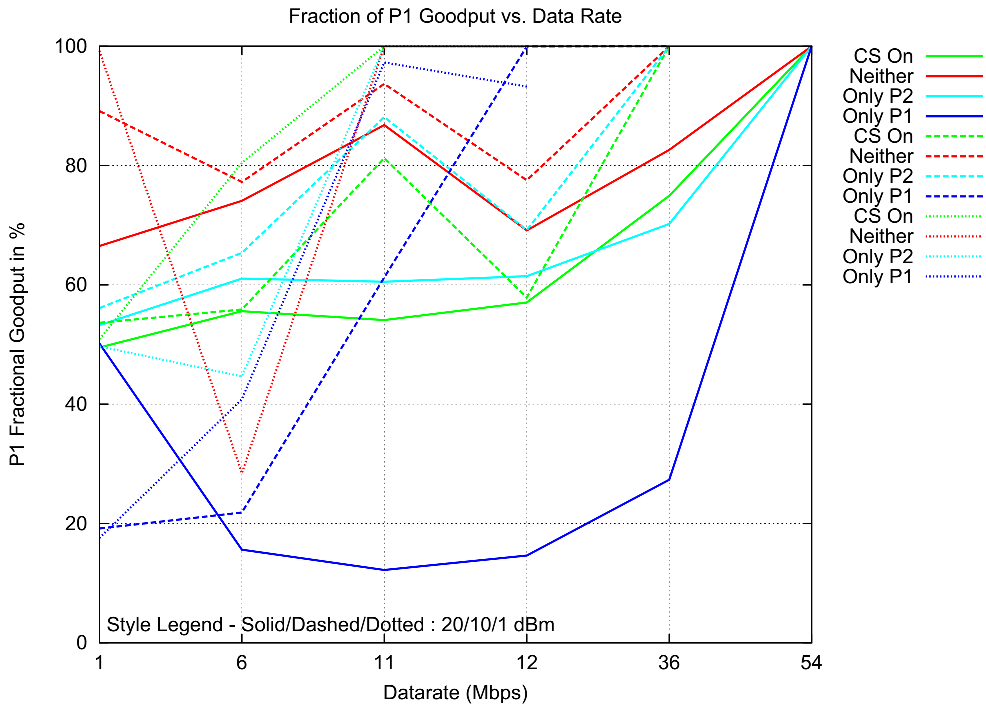

Figure 5.11: Test 3a: Nodes Far, Interference Close

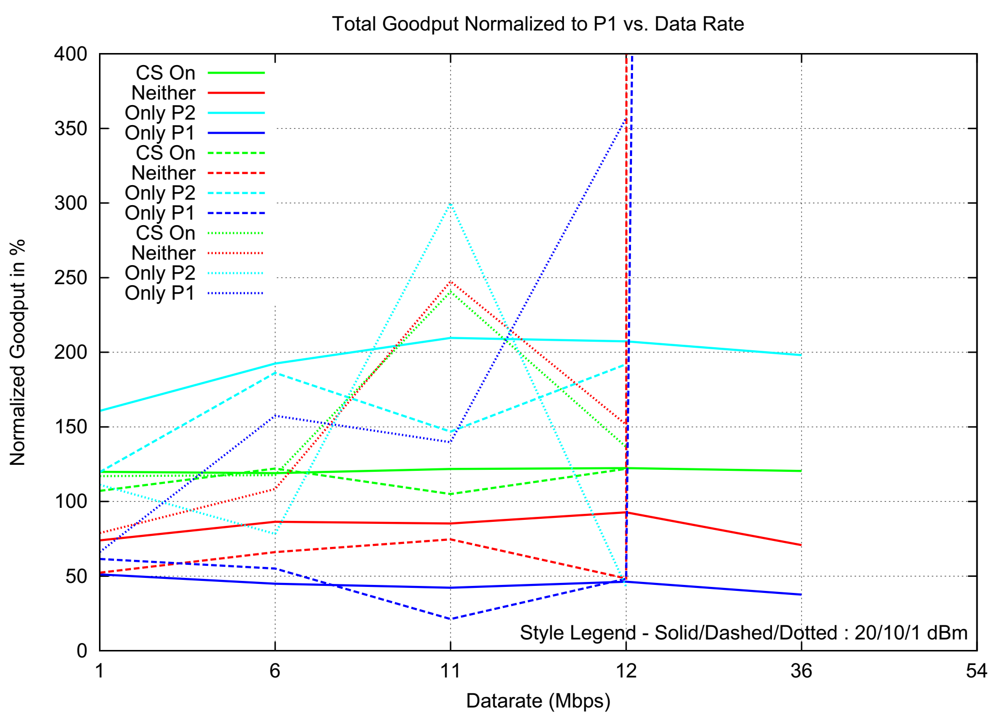

Figure 5.12: Test 3a: Nodes Far, Interference Close

5.5 Test 4a: Nodes Far, Interference Far

In Test4, the nodes and interference both are far away. In this case, we are pushing the limits of IEEE 802.11b and IEEE 802.11g protocol by introducing very high attenuations between the nodes. We observe how weak signals cope with weak interference. We have the following observations from the graphs shown in Figure 5.13 and Figure 5.14:

- In Figure 5.13, we see that at low rates (1Mbps, 6Mbps, 11Mbps, 12Mbps) most of the cases yield almost equal throughput levels, except when lowest transmit power (1dBm) is used. This implies that turning off the CS is not useful in this case.

- At 54Mbps, we can only achieve a low throughput at highest transmit power (20dBm). This rate is not robust enough for such high attenuation even if we disable CS.

- At 36Mbps, we see that disabling CS does not provide higher throughput level than the default case. However, disabling CS on either P1 or P2 does help in increasing the cumulative throughput.

- In Figure 5.14, we see that 1dBm stops communicating at 12Mbps, 10dBm gets very low throughput at 36Mbps, and 20dBm yields very low throughput levels at 54Mbps.

We have the following observations for normalized, and fractional throughput graphs shown in Figure 5.15 and Figure 5.16:

- Figure 5.15, shows that in almost all the cases P1 does not share the channel with P2. It is because of the very high attenuations between the pairs, which provides both networks a separate collision domain.

- We see that turning off CS at P1 and using the complete channel is a good option. Because interference to the other network is least due to very high attenuation between the pairs.

- In Figure 5.16, show that disabling CS on both pairs doesn't help significantly in any of the cases.

- The throughput is less than 100% in all of the cases expect when we disable the CS on P1. This is the case which constantly gets more than 110% throughput except at 54Mbps.

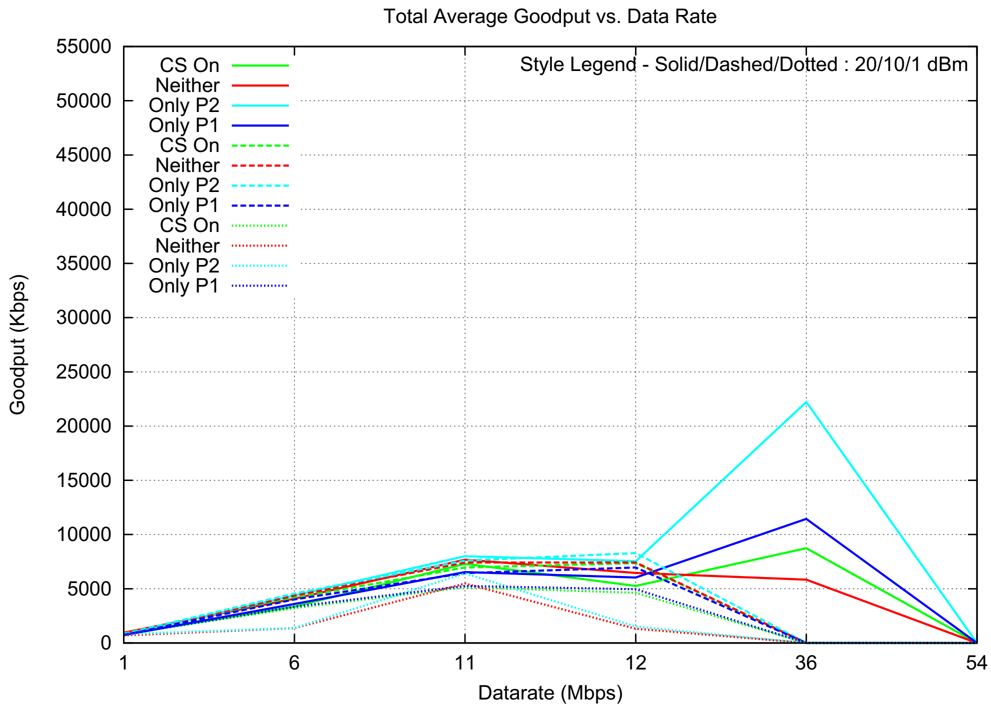

Figure 5.13: Test 4a: Nodes Far, Interference Far

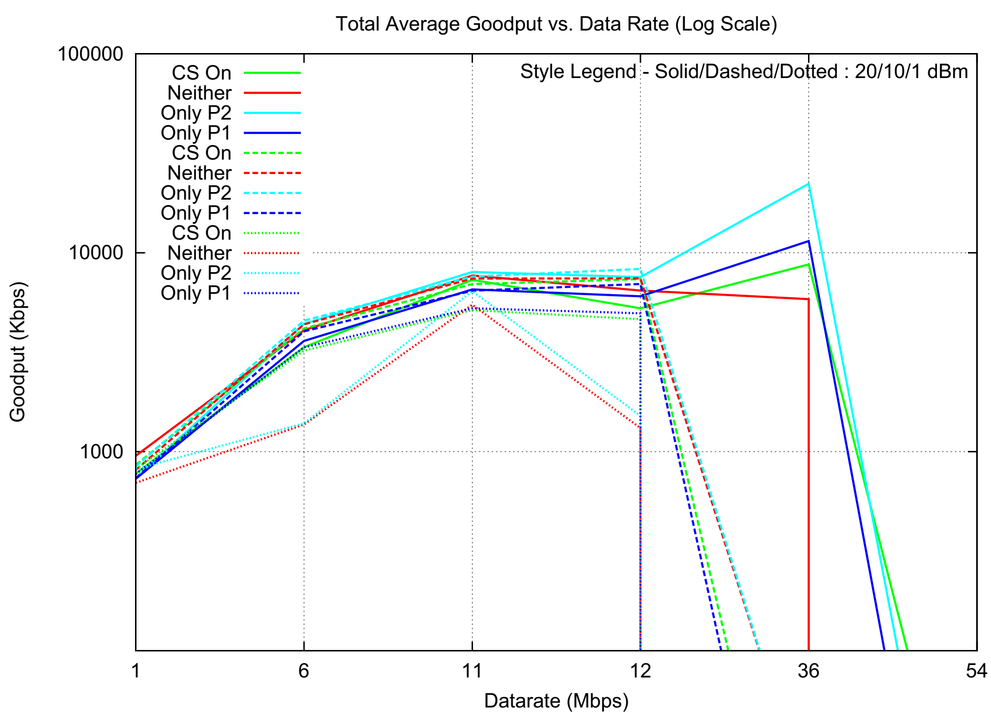

Figure 5.14: Test 4a: Nodes Far, Interference Far

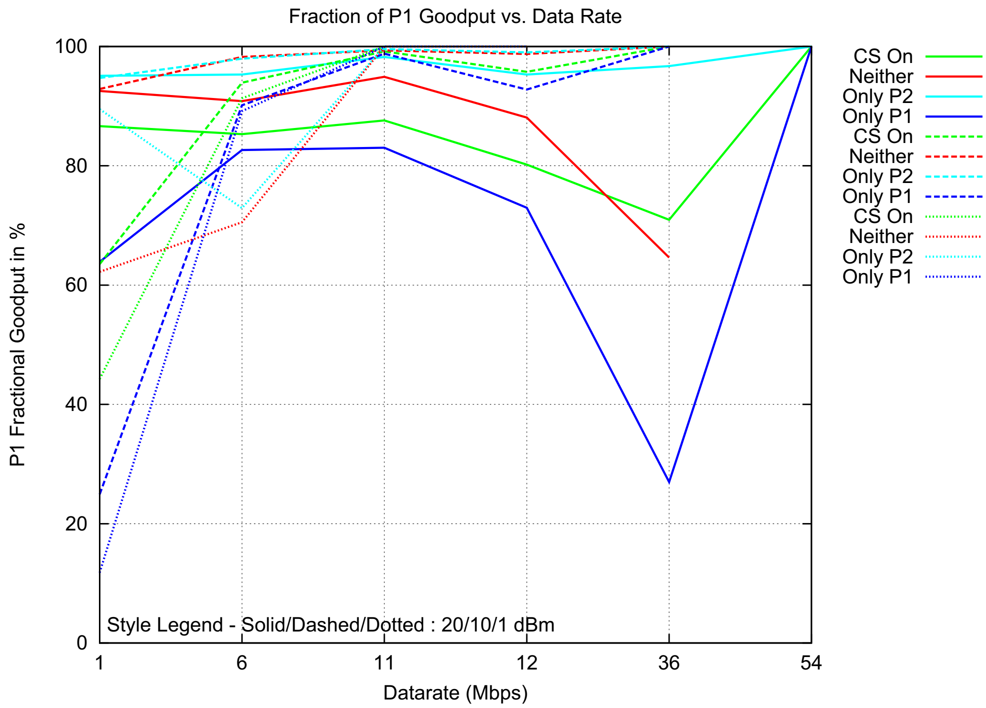

Figure 5.15: Test 4a: Nodes Far, Interference Far

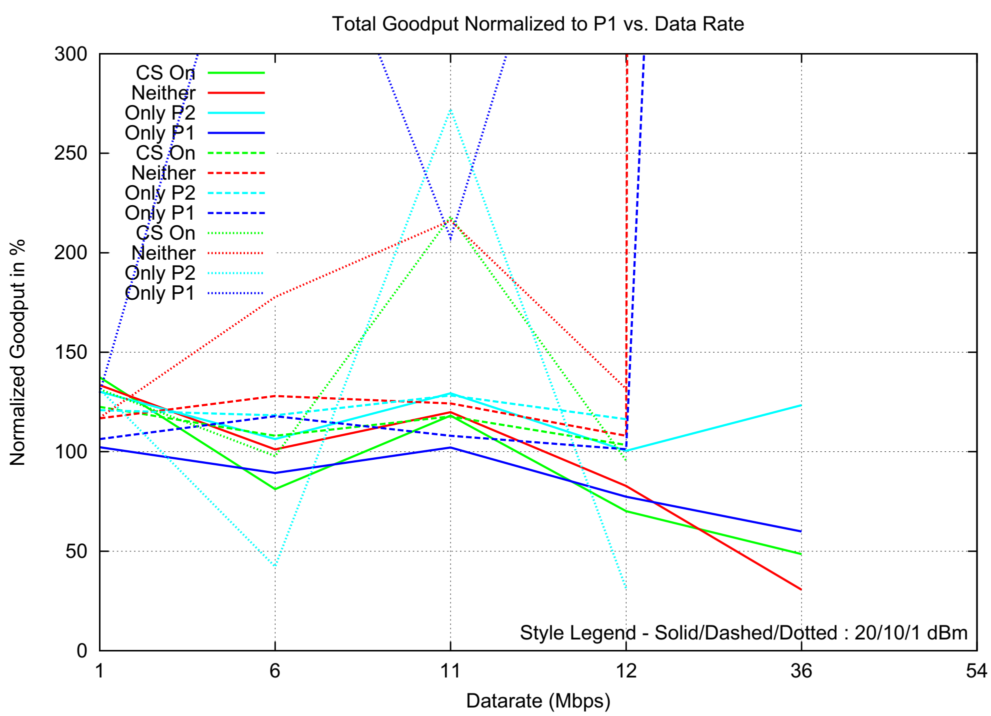

Figure 5.16: Test 4a: Nodes Far, Interference Far

5.6 Test 1b: All Nodes Close Together

This section discusses the scenario in which Pair-2 as seen in Figure 3.1 swaps its transmitter and receiver. This Case-B 5.18, though looks similar to Case-A 5.17 gives us a very different test bed. As we see in Case-A, the transmitters interfere more with each other and less with the receivers. However, in Case-B the transmitters interferes with the receivers. As transmitters send data most of the time in our experiments, and receivers just send acknowledgements the interference among the nodes are different.

In this tests all nodes are close together. As the transmitter and receiver of P2 are swapped, we expect both the receivers to have more interference. We have the following observations from the graphs shown in Figure 5.19, Figure 5.20, Figure 5.21 and Figure 5.22:

- We see that the throughput values in this test are relatively lower compared to Test1a.
- The cases with no CS on both pairs, get a litte lower throughput at higher rates. This is because of the increased time of interference at the receiver.
- Even using the lowest transmit power doesn't increase the throughput much in this case. This is because the transmitter are now contantly affecting the receiver of the pair.
- It is interesting to see that unlike Test1a, using carrier sensing with lowest transmit power doesn't share channel equally among the pairs.
- We again see the trend of P1 consuming all of the bandwidth when not using carrier sensing.
- When CS is disabled on both pairs. We see that higher power does help in sharing the bandwidth between the pairs.
- It is interesting that at 1Mbps, P2 gets most of the bandwidth and at 54Mbps, P1 gets more bandwidth.
- Even though the results are not exactly the same as Test1a, we see that more than 100% throughput can be consistently achieved of the system when CS is disabled on P1.

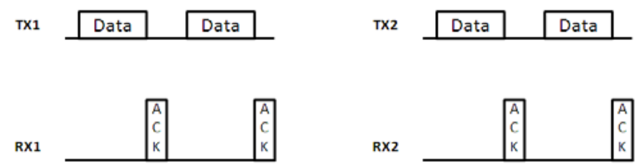

Figure 5.17: CASE A: Typical transmission of Data and Acks with no CS

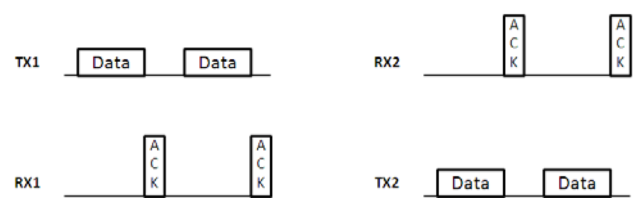

Figure 5.18: CASE B: Typical transmission of Data and Acks with no CS

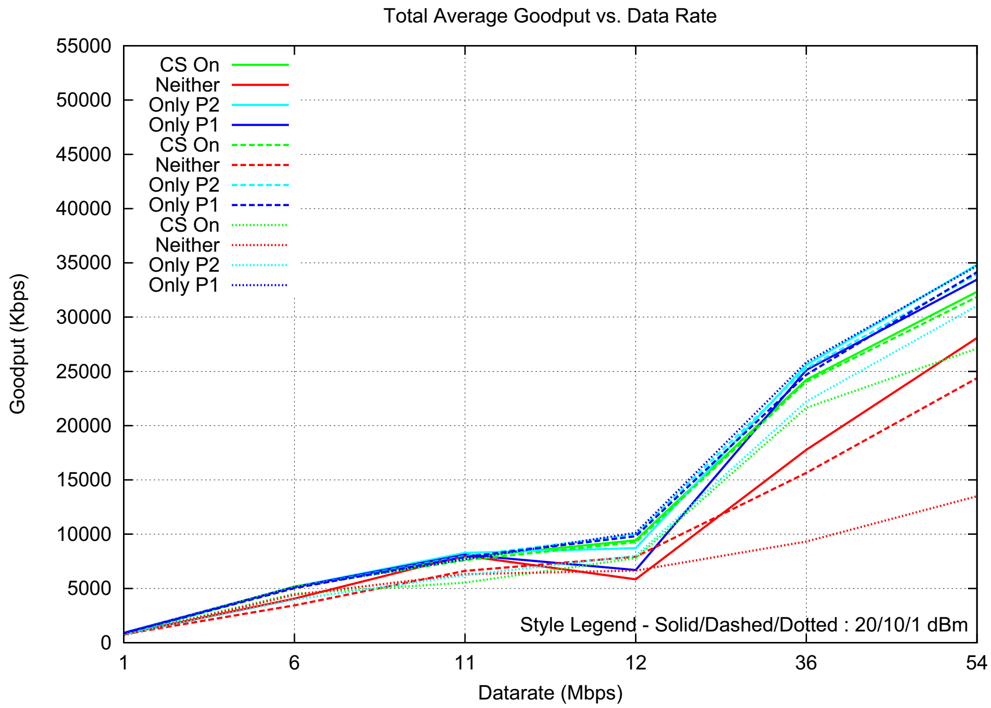

Figure 5.19: Test 1b: All nodes close together (P1 nodes swapped)

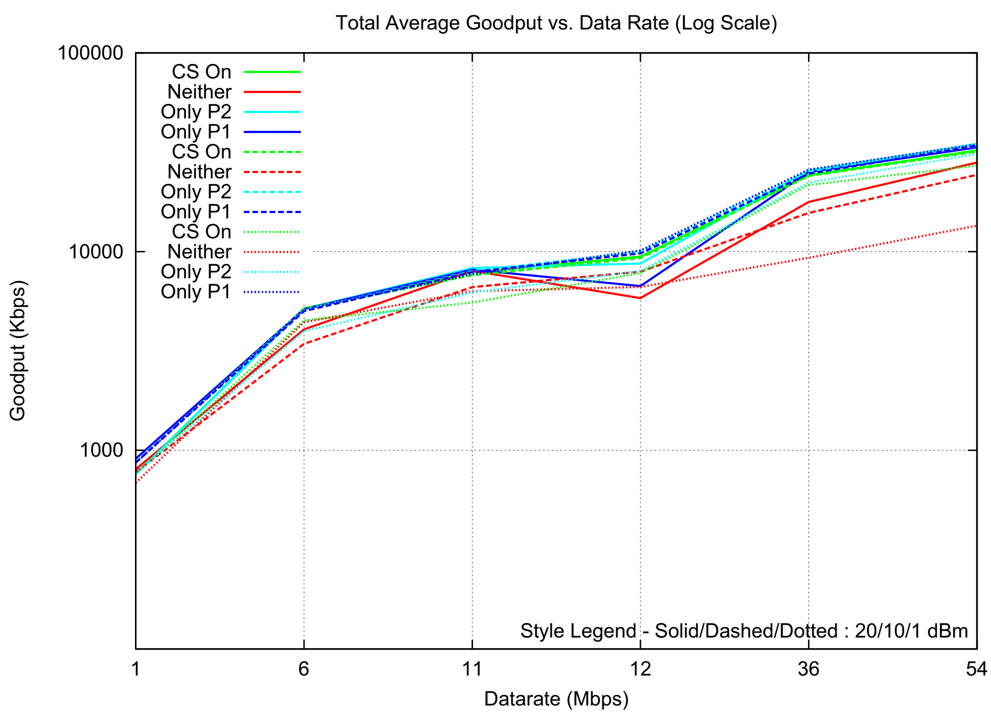

Figure 5.20: Test 1b: All nodes close together (P1 nodes swapped)

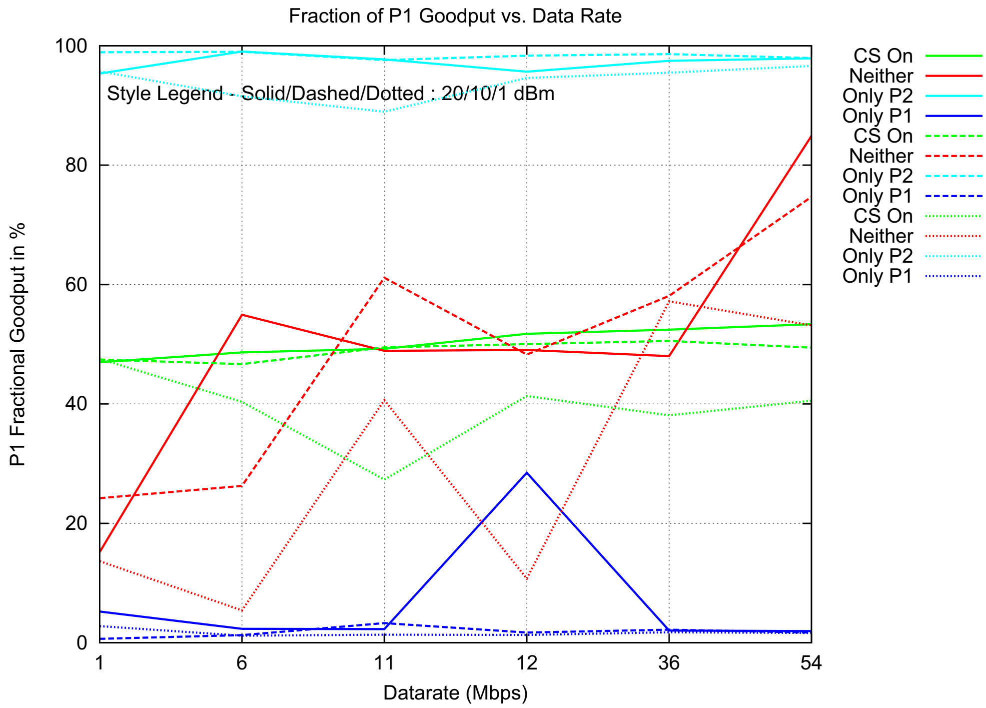

Figure 5.21: Test 1b: All nodes close together (P1 nodes swapped)

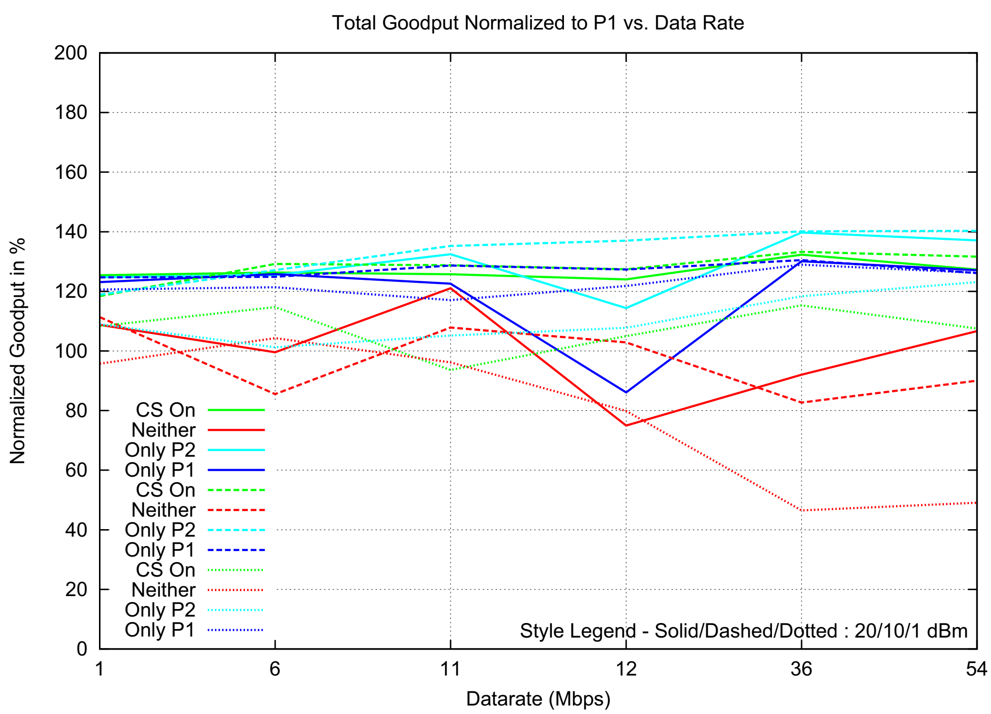

Figure 5.22: Test 1b: All nodes close together (P1 nodes swapped)

5.7 Test 2b: Nodes Close, Interference Far

In Test2b, the nodes are close together and the interferences are far. We can consider this as the best case because the SNIR at the receviers is expected to be greatest in this case. Even though we have swapped the transmitter and reciever position at P2, the attenuations remain the same on each of the paths. This suggests that essentially the setup is the same, so we expect results similar to Test2a. We have the following observations from the graphs shown in Figure 5.23, Figure 5.24, Figure 5.25 and Figure 5.26:

- The results are very similar to Test2a (reference-section-test2a here).
- The lowest transmit power when there is no CS on P1, does get higher throughput at 54Mbps. However, in Test2a the highest power gets highest throughput.
- As seen in Test2a, in most of the cases we are able to squeeze more than 100% throughput out of the network.
- We see the unusual behavior when CS is disabled at P1 again in this case. The throughput in some cases is even more than 200%.
- The fractional throughput gained from P1, shows almost same results as Test2a. With the exception of medium transmit power (10dBm)with no CS at P1 performing better than the low transmit power (1dBm).

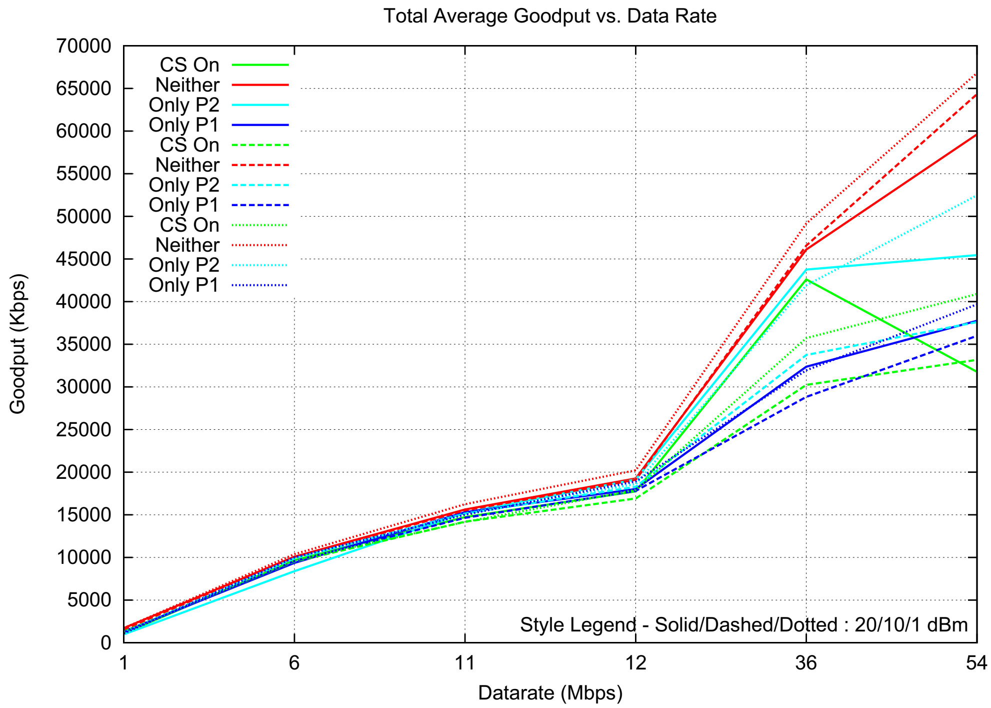

Figure 5.23: Test 2b: Nodes Close, Interference Far (P1 nodes swapped)

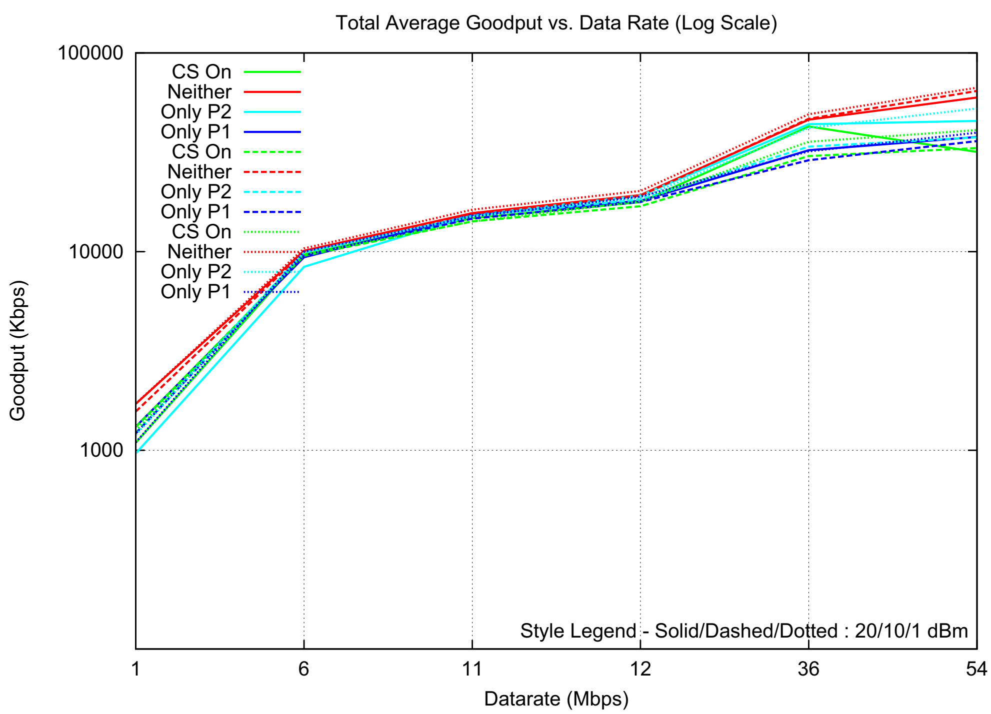

Figure 5.24: Test 2b: Nodes Close, Interference Far (P1 nodes swapped)

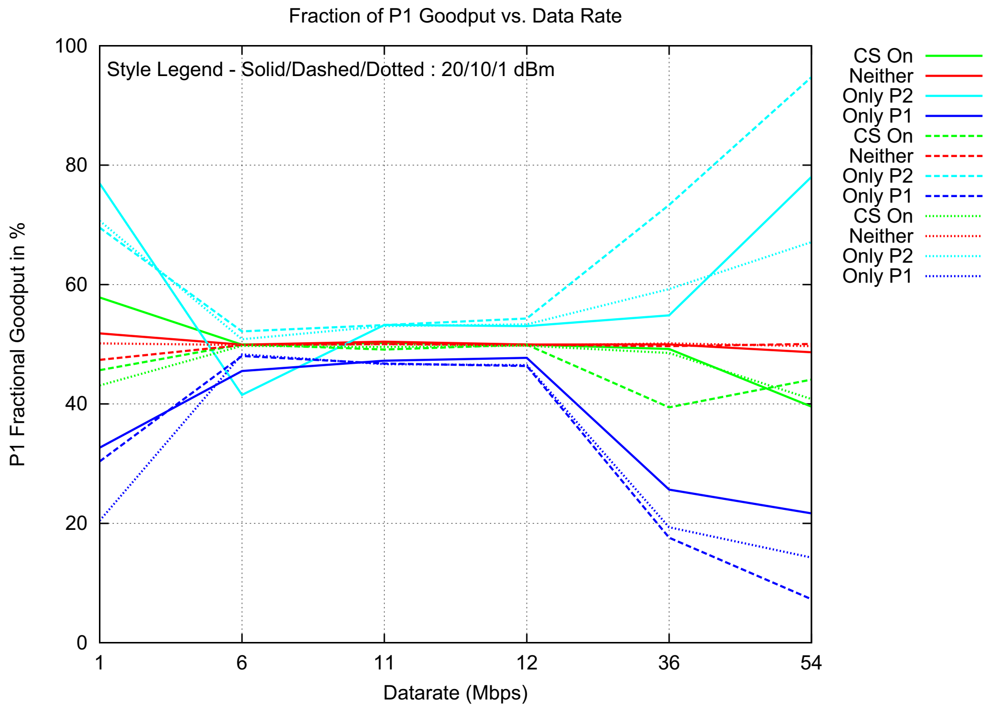

Figure 5.25: Test 2b: Nodes Close, Interference Far (P1 nodes swapped)

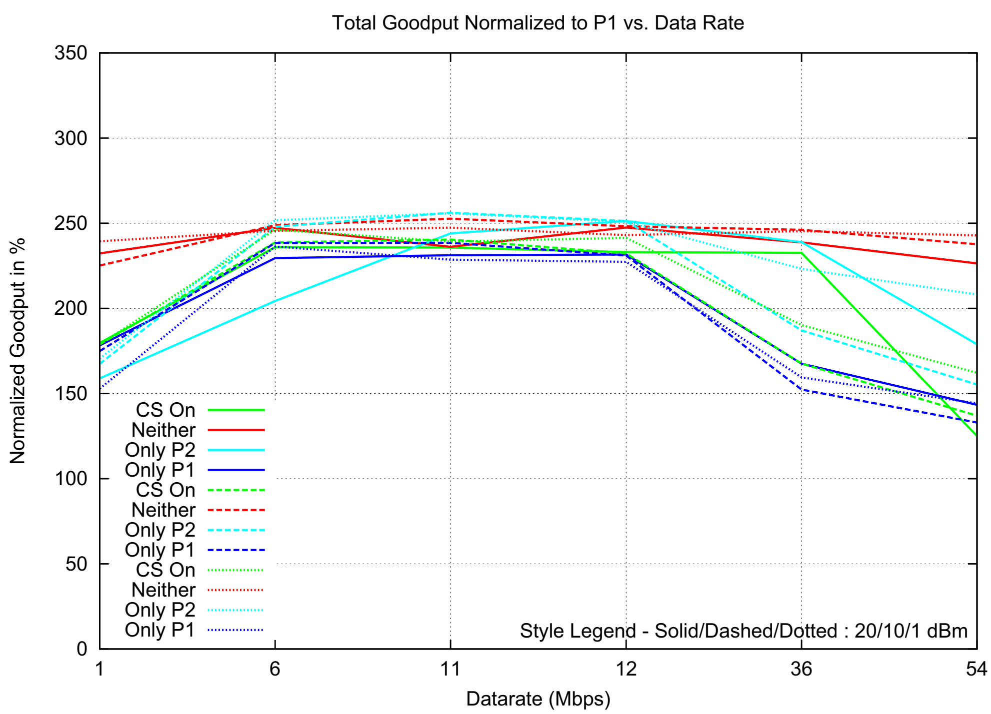

Figure 5.26: Test 2b: Nodes Close, Interference Far (P1 nodes swapped)

5.8 Test 3b: Nodes Far, Interference Close

In Test3b, the nodes are far away and the interferences are close. This case has the lowest SINR, as the interference to R1 is 40dB stronger than the signal. It is a lot more consistent. For each receiver, now the transmitter of the other pair is close and constantly transmitting. In this case the interference is much worse than Test3a, and we expect a lot less throughput values. We have the following observations from the graphs shown in Figure 5.27, Figure 5.28, Figure 5.29 and Figure 5.30:

- As we see in Figure 5.27, the throughput is much lower in this case.
- In Test 3a, only at 36Mbps, can decent throughput be achieved from the network. However, this is not the case in Test3b. It is expected as now the interference from both the transmitters is much stronger to the receivers of the other pair.
- In Figure 5.28, we see the same trend of high data rates only supported by higher transmit powers.
- In Figure 5.29, the values are erratic. However, in Test3b, we did observe P1 using up all of the channel when not performing CS especially at higher transmit powers.
- Figure 5.30, shows that Test3b yields a lower throughput if compared with Test3a. We do observe that no CS, does work better than CS in Test3b compared to Test3a. It is due to the interference being strong and closer, and no CS becomes successful in these scenarios.
- We see that no CS at P1, keeps the throughput to less than 100%, where as in Test3a we could almost always get throughput more than 100%.

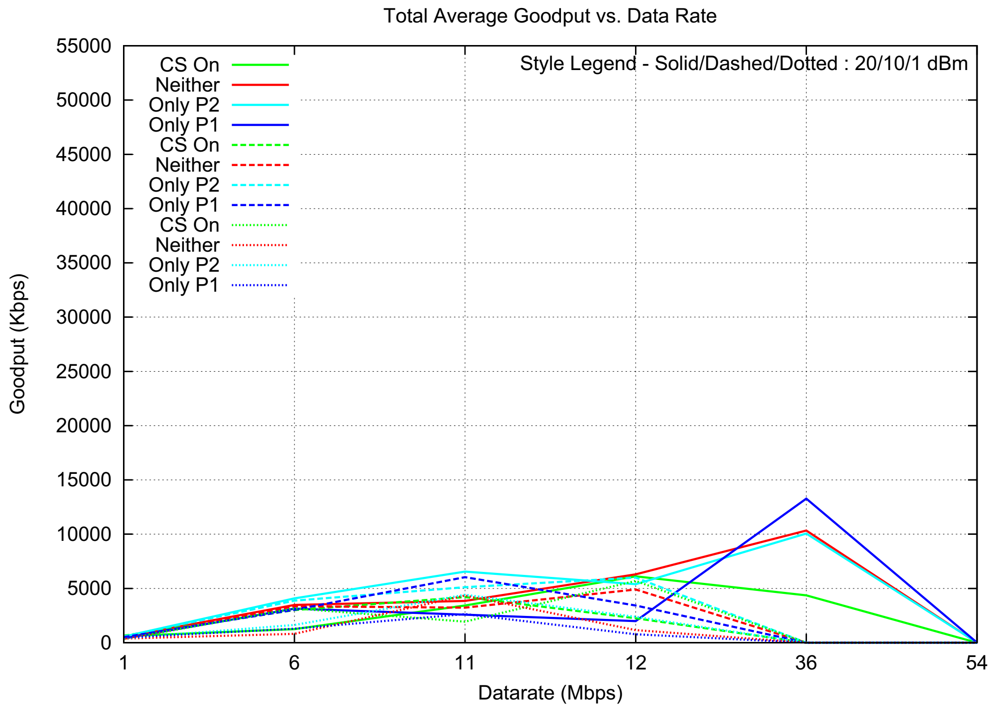

Figure 5.27: Test 3b: Nodes Far, Interference Close (P1 nodes swapped)

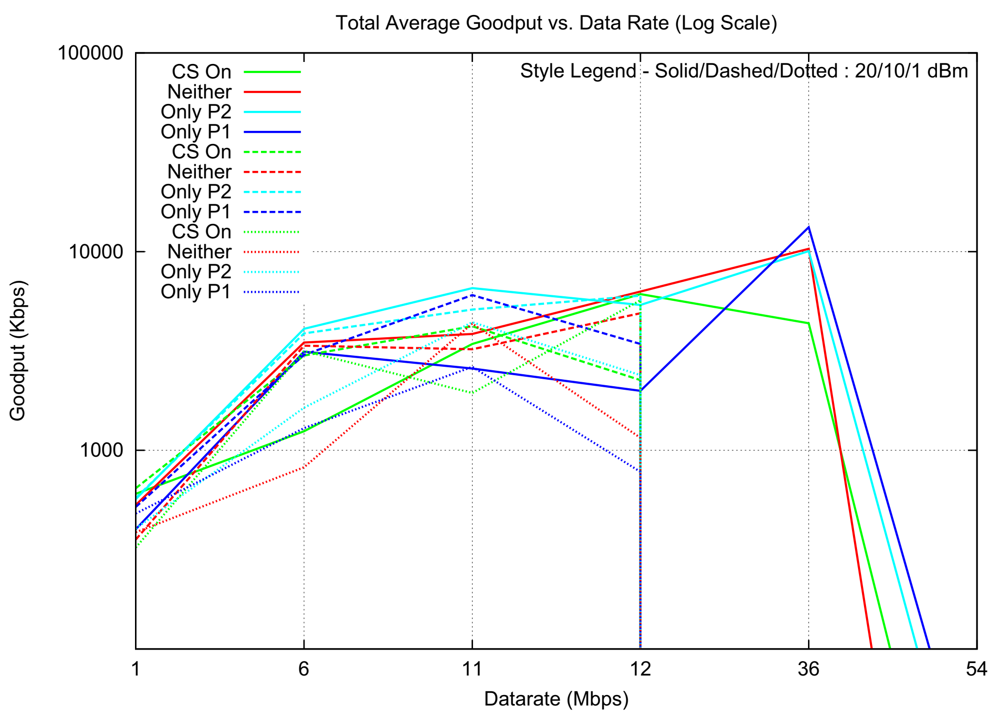

Figure 5.28: Test 3b: Nodes Far, Interference Close (P1 nodes swapped)

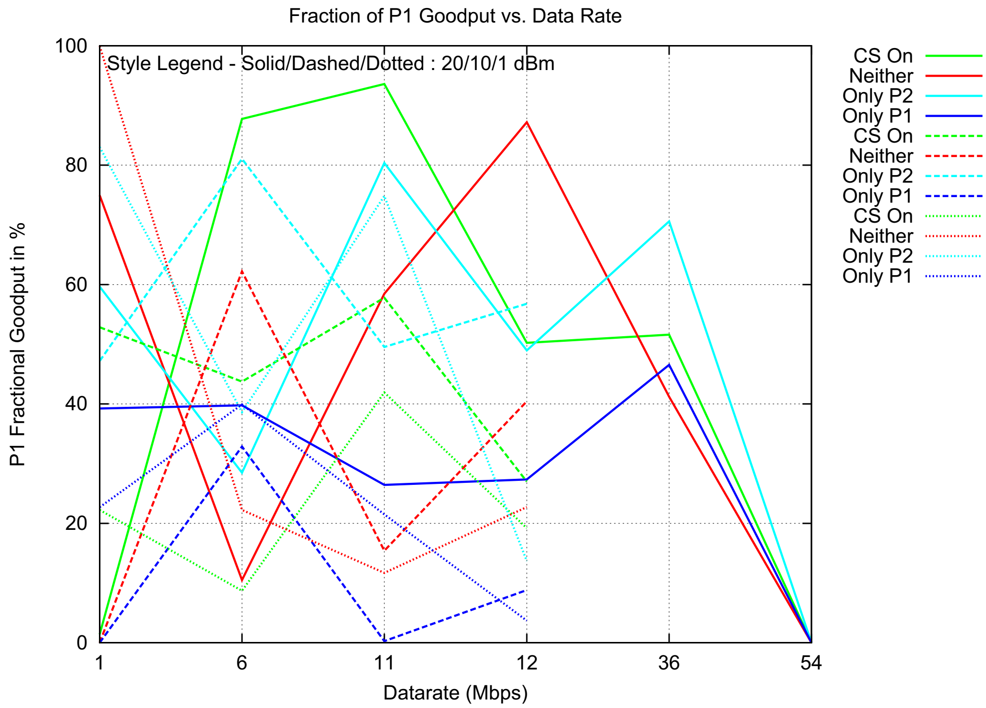

Figure 5.29: Test 3b: Nodes Far, Interference Close (P1 nodes swapped)

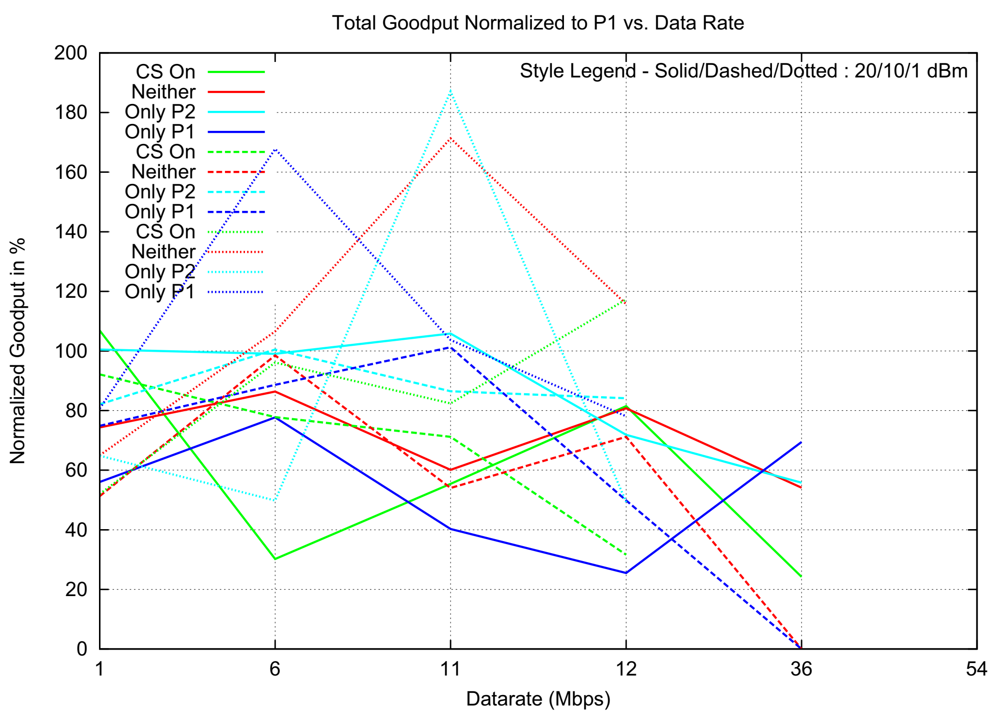

Figure 5.30: Test 3b: Nodes Far, Interference Close (P1 nodes swapped)

5.9 Test 4b: Nodes Far, Interference Far

In Test4b, the nodes and interferences are both far away. This case is very similar to Test4a in terms of the interference to the nodes. As the attenuations are the same on the paths, we consider the results to be the same as Test4a. We have the following observations from the graphs shown in Figure 5.31, Figure 5.32, Figure 5.33 and Figure 5.34

- The results obtained in Figure 5.31 are very similar to the results from Test4a as expected. However, this is a low throughput case with very high attenuations, there is a slight increase in throughput at 36Mbps in Test4b.
- Figure 5.32 shows the same trend as Test4a. Higher transmit powers support higher data rates.
- When transmit power of 20dBm and 10dBm are used, Figure (ref-test12-normalized) shows 80% throughput values at 1Mbps and 6Mbps, and lower throughput values for higher rates. However, Test4a yields consistent 80% throughput up to 12Mbps of data rate.
- Turning off carrier sensing at P1, helps at only 11Mbps to yield more than 100% throughput. However, in Test4a we got more than 100% throughput most of the time.
- Figure 5.33 shows that P1 is not able to get most of the bandwidth as we saw in Test4a.

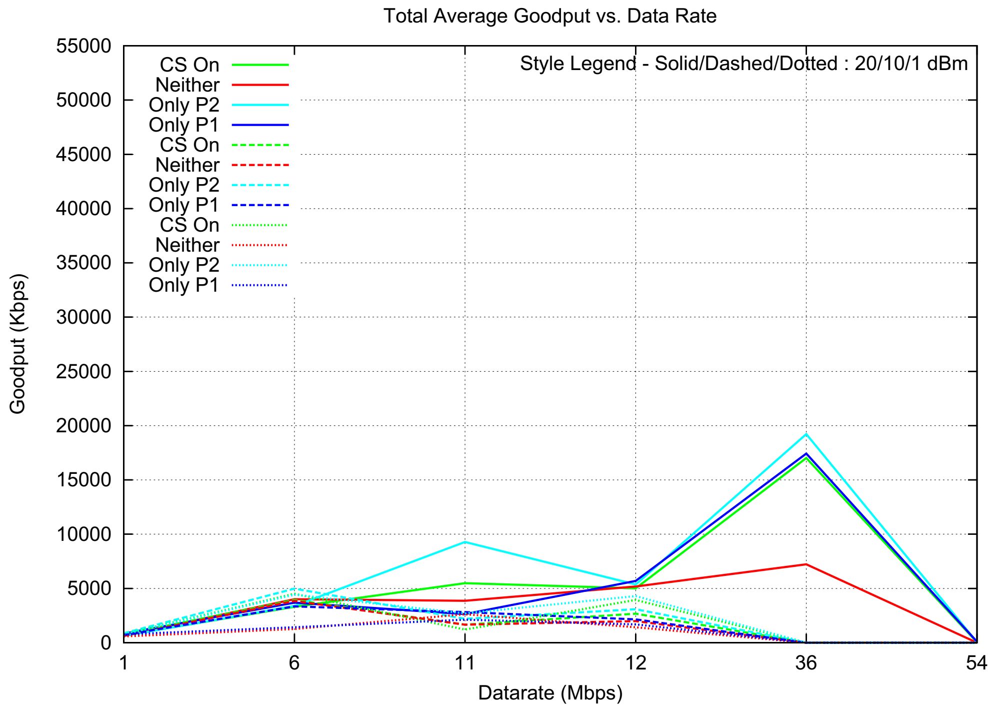

Figure 5.31: Test 4b: Nodes Far, Interference Far (P1 nodes swapped)

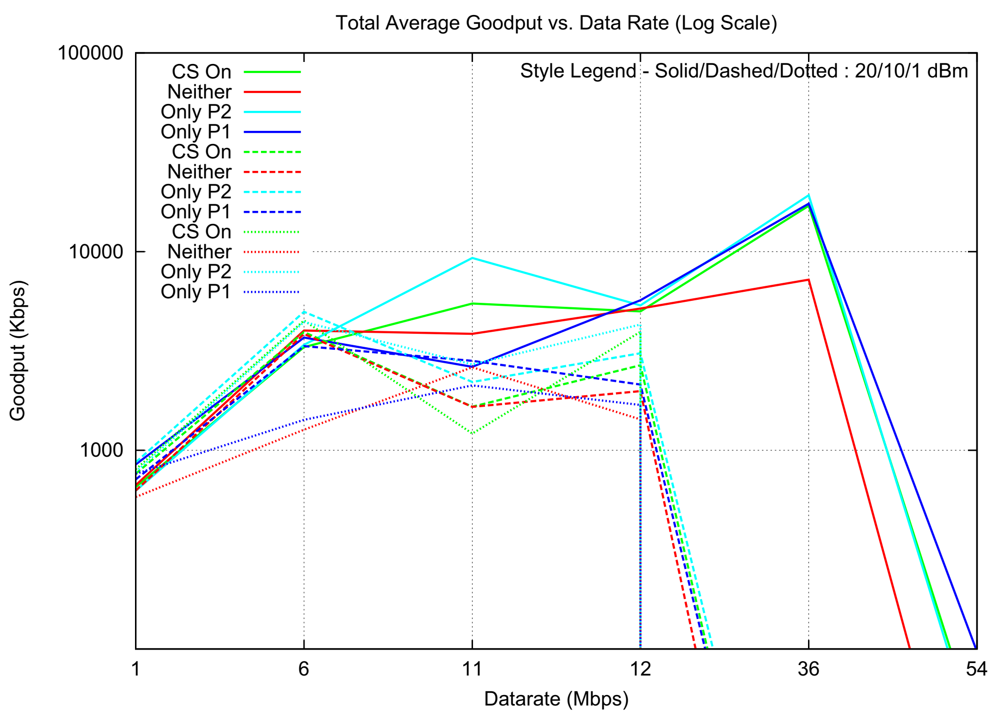

Figure 5.32: Test 4b: Nodes Far, Interference Far (P1 nodes swapped)

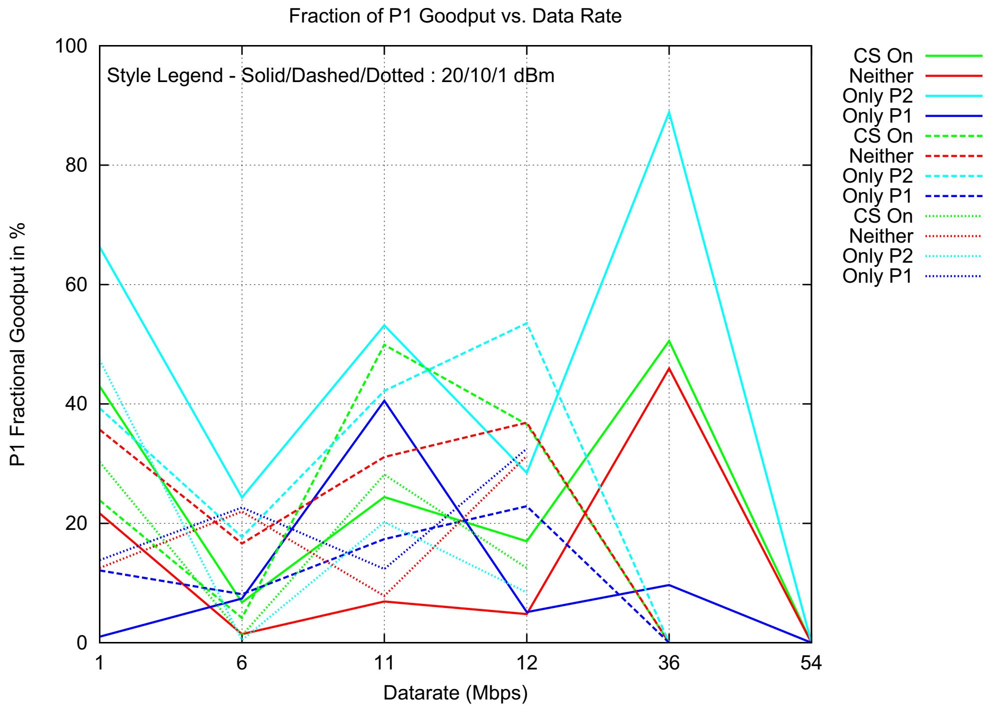

Figure 5.33: Test 4b: Nodes Far, Interference Far (P1 nodes swapped)

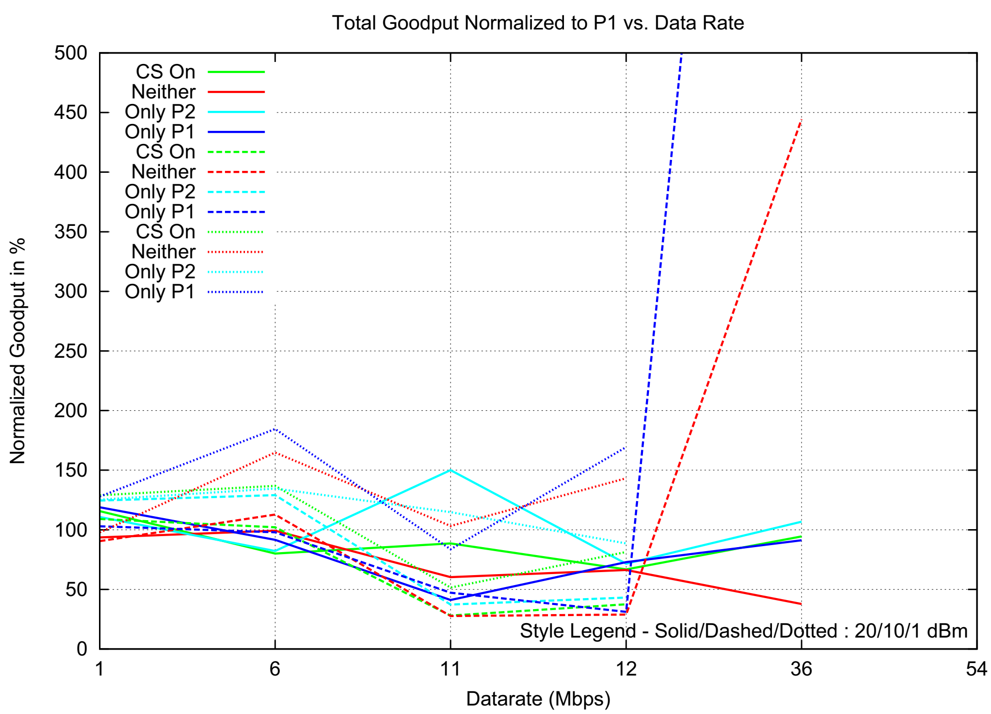

Figure 5.34: Test 4b: Nodes Far, Interference Far (P1 nodes swapped)

5.10 Deviation in Results

The deviation in results was calculated to evaluate the reliability and repeatability of the experiments. For each of the four iterations performed a mean value was calculated. Then the mean and standard deviation (SD) of these four means was calcuated. Finally the SD was normalized by the mean value to analyze the deviation in the SD.

The data shows that the results are consistent. Most of the values are less than 3%. The deviation in the results is high in case of Test-3a (23%), Test-3b (29%), this deviation is expected as the interference is close compare to the desired signal. In Test-4b the deviation is 14%. This is the case in which all of the nodes are far away, and the desired and interfering signals are both weak. There are outliers in the experiments for which we need to do a more critical analysis of data for better understanding.

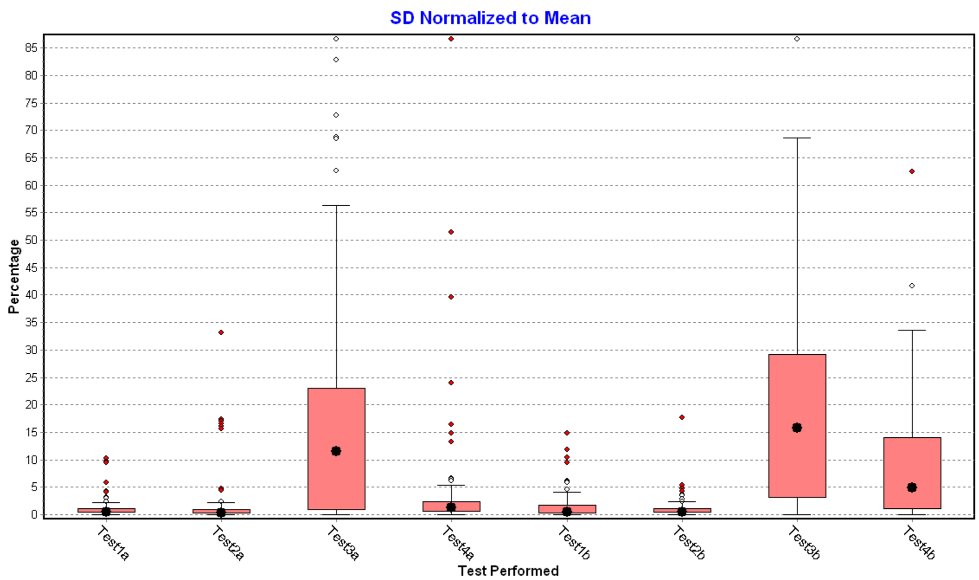

Figure 5.35: Shows the deviation in the SD test performed normalized to Mean

Chapter 6

Results and Discussion for Hypothesis-2

6.1 Hypothesis-2

This chapter considers the situation in which our target pair is in a hostile situaiton. We consider that one pair is fixed at a particular data rate, transmit power, and carrier sensing. This pair, P2, is run through all the combinations in each of the four physical setups, when only P2 was operating. The combination which yields the highest throughput in each of four physical setups was selected for that particular case. While P2 is fixed at one combination for each test, P1 goes through all the 36 combinations of data rate, transmit power, and carrier sensing.

We then analyze and identify the combinations which yield the highest cumulative throughput for a scenario. For the four test scenarios, we found two combinations for P2.

- For Test1 and Test2: Data Rate 54Mbps, Transmit Power 20dBm, CS Enabled
- For Test3 and Test4: Date Rate 36Mbps, Transmit Power 20dBm, CS Enabled

6.2 Test 1: All Nodes Close Together

In this tests all nodes are close together. We can consider this as the worst case in terms of interference between the nodes. So, we expect that turning off carrier sensing wouldn't be of much help. We have the following observations from the graphs shown in Figure 6.1, Figure 6.2, Figure 6.3 and Figure 6.4:

- Figure 6.1 shows that no CS does not help in achieving higher cumulative throughput. Also, it does not matter what level of transmit power is used. It is because the nodes are so close that the interference is too high when CS is disabled.

- At lower rates of P1, CS does help a lot as the packets are longer and interfere more with the shorter data packets (36Mbps) of P2.

- We see that at higher rates of 36Mbps and 54Mbps, the channel is shared equally between both pairs.

- Figure 6.3 shows that with no CS P1 does overtake all of the channel, but doesn't provide the highest cumulative throughput.

- Figure 6.4 shows that when CS is disabled we can never get more than 100%. This is because in this case the throughput is limited by the pair with lesser rate, bringing down the cumulative throughput.

- In Figure 6.4, the high throughput values are at lower rates are mostly due to P2 set at 36Mbps. When both P1 and P2 are at high rates (36Mbps and 54Mbps) the throughput is not more than 100%.

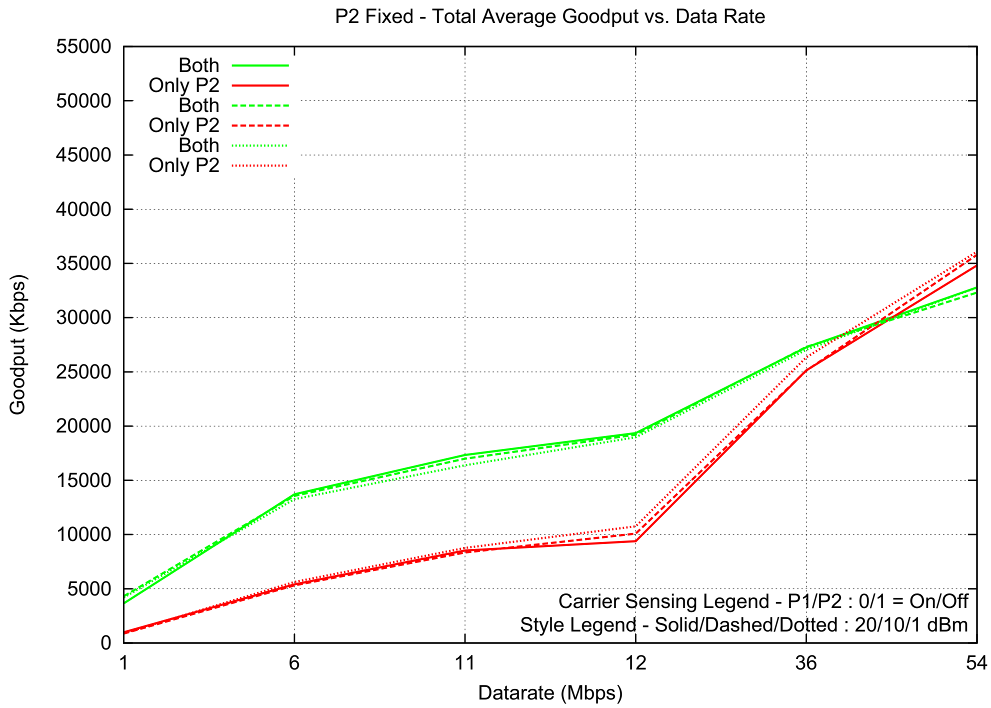

Figure 6.1: Test 1: All nodes close together

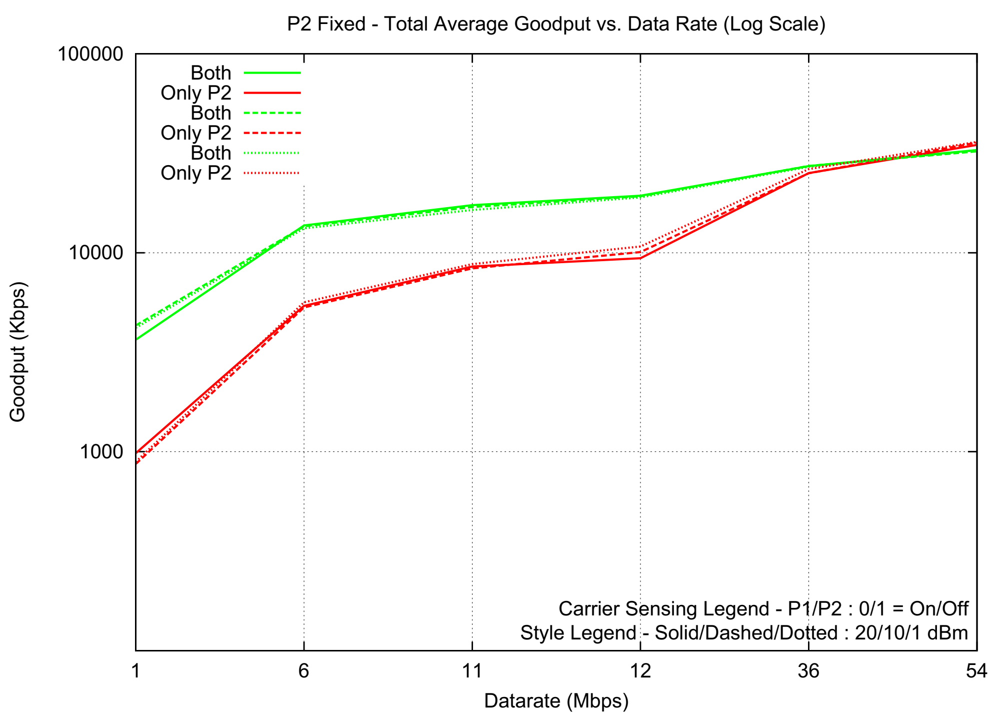

Figure 6.2: Test 1: All nodes close together

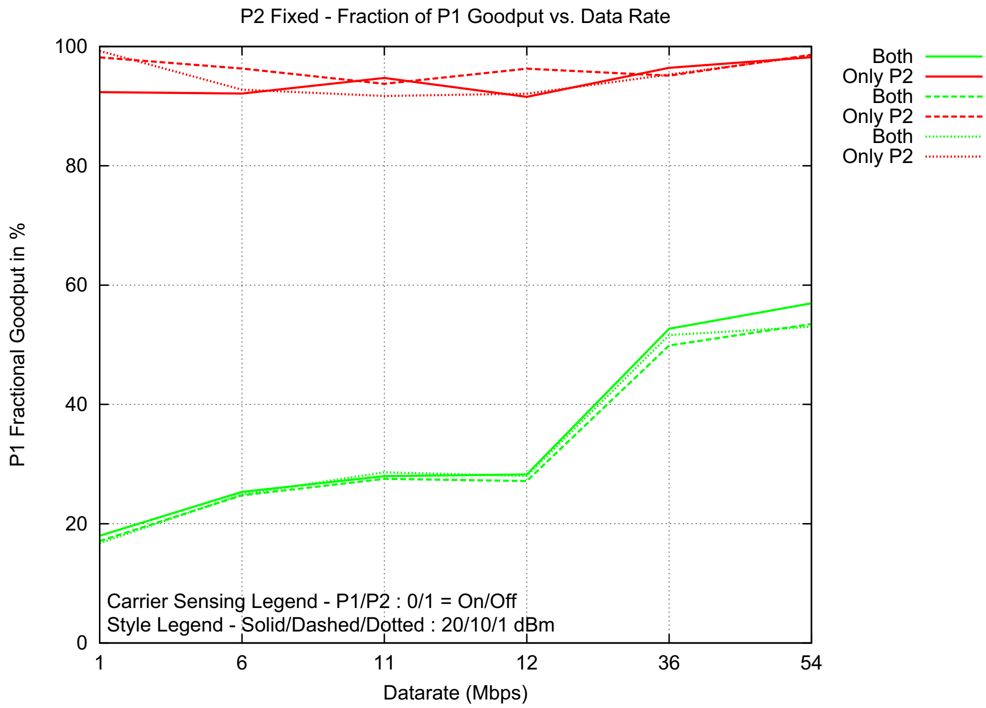

Figure 6.3: Test 1: All nodes close together

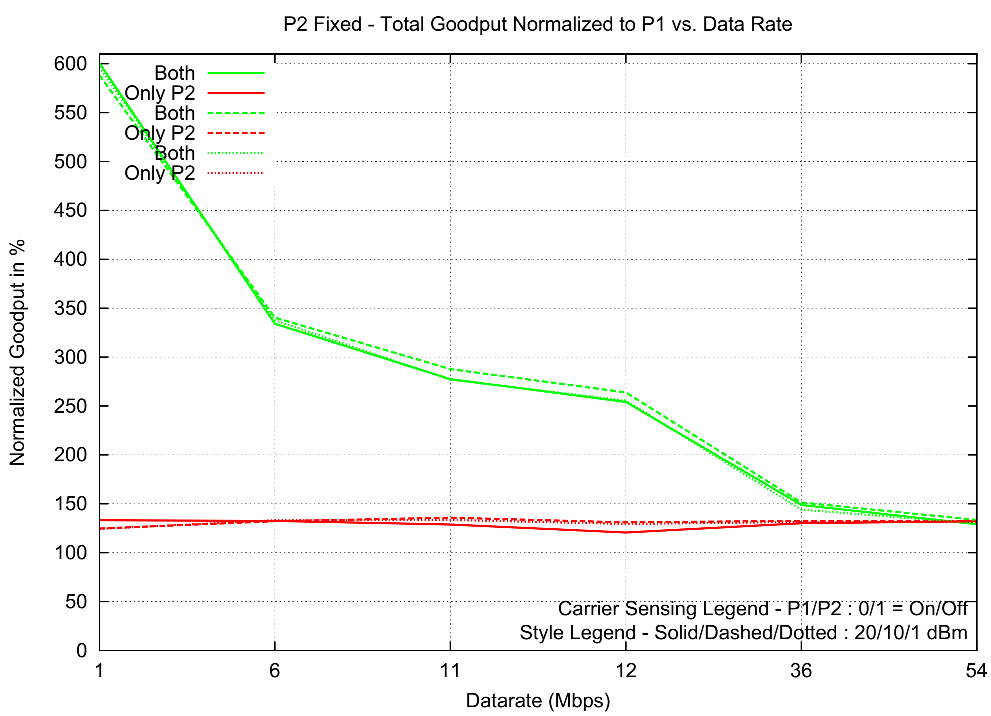

Figure 6.4: Test 1: All nodes close together

6.3 Test 2: Nodes Close, Interference Far

In this case we have the nodes close and interference far. This can be considered as the best case as the interference is 40dB weaker than the desired signal. We expect that disabling carrier sensing should help in this particular case. We have the following observations from the graphs shown in Figure 6.5, Figure 6.6, Figure 6.7 and Figure 6.8:

- In Figure 6.5 we see that at rates less than 12Mbps or less using 1dBm with no CS can yield the highest throughput.
- At 36Mbps and 54Mbps both pairs do better as both the pairs can share the medium equally, and the difference between rates is not large.
- In Figure 6.7 we see that the channel is overtaken by P2 at lower rates (12Mbps or less) and by P1 at higher rates (36Mbps and 54Mbps) when CS is disabled. The same trend is found in the cases with CS enabled.

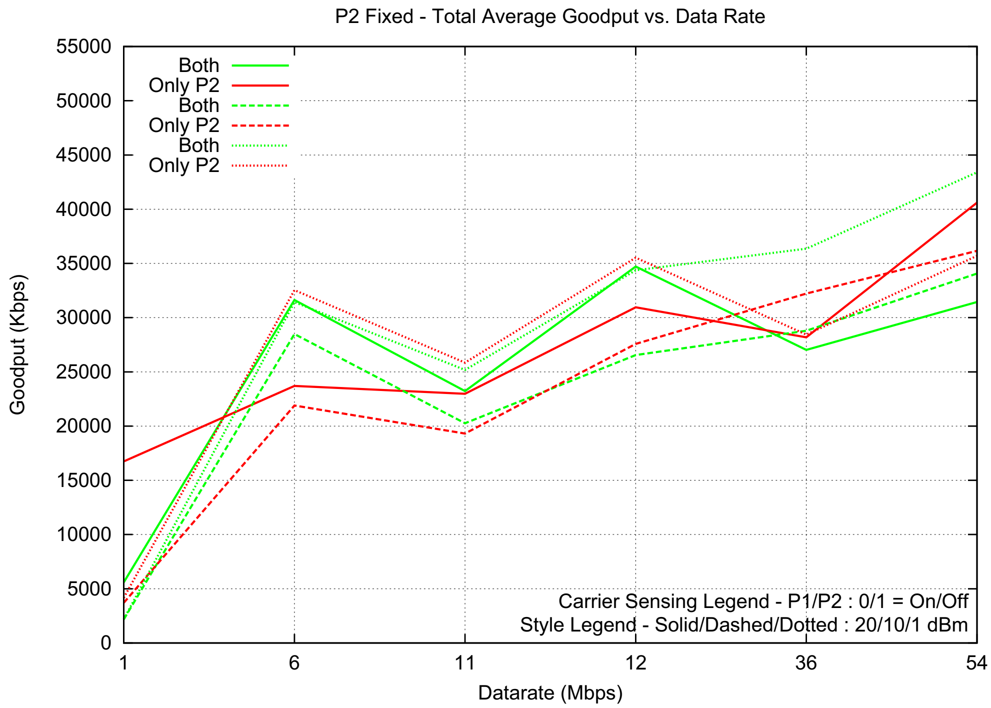

Figure 6.5: Test 2: Nodes Close, Interference Far

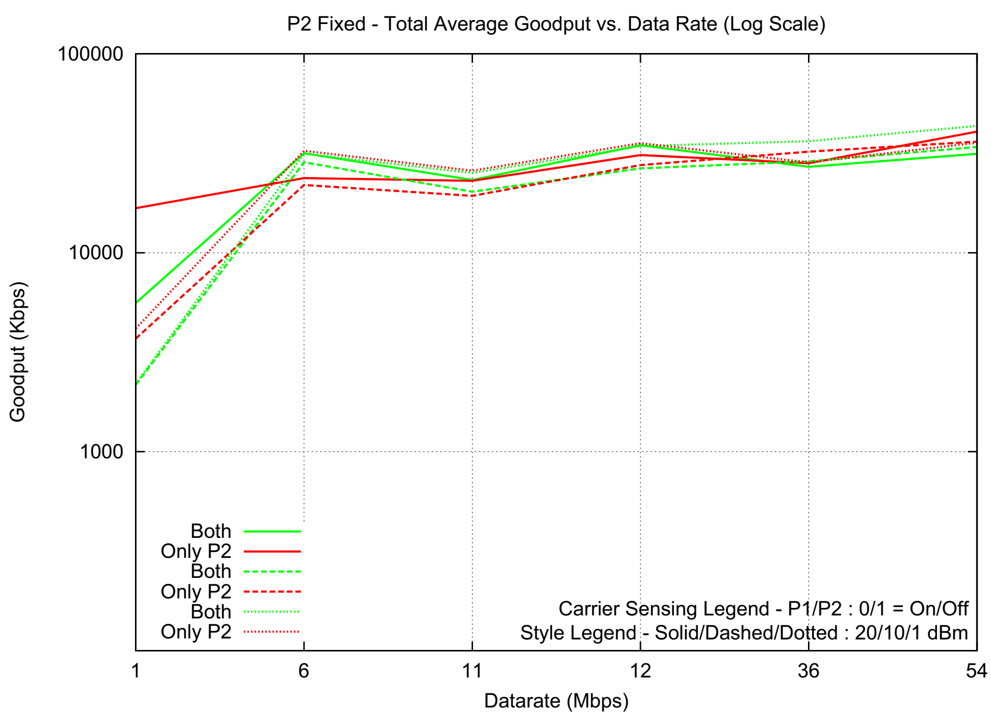

Figure 6.6: Test 2: Nodes Close, Interference Far

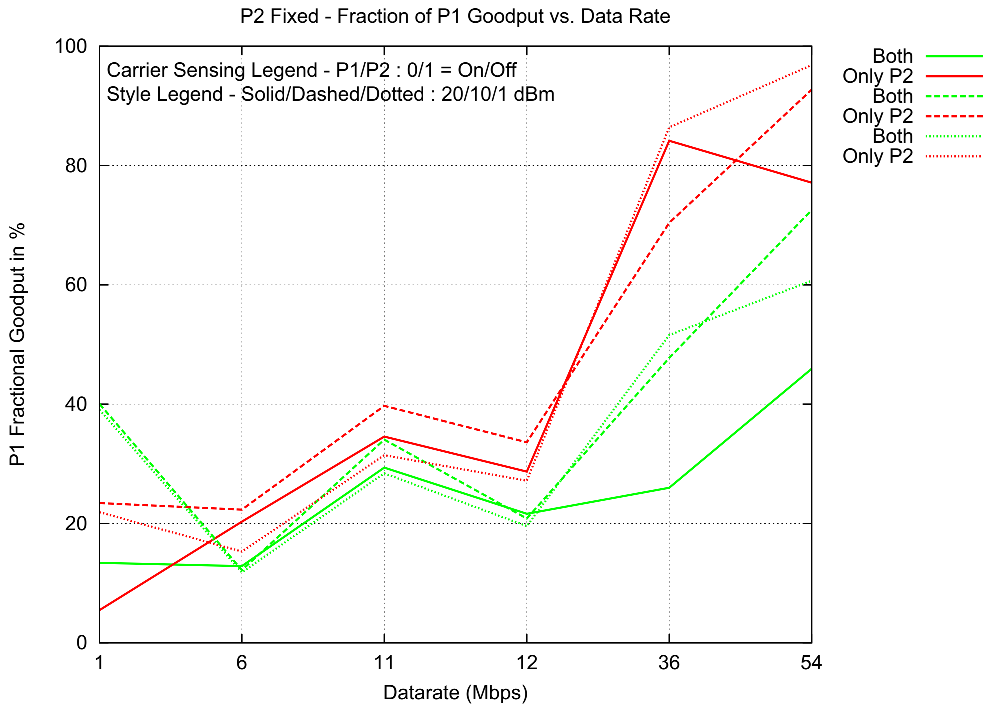

Figure 6.7: Test 2: Nodes Close, Interference Far

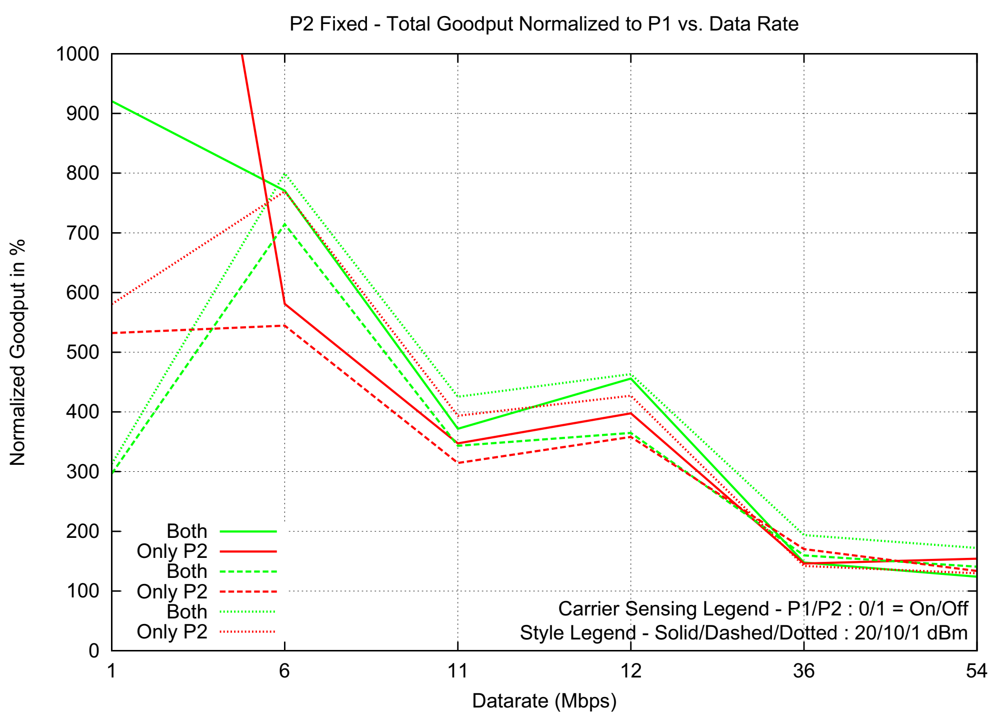

Figure 6.8: Test 2: Nodes Close, Interference Far

6.4 Test 3: Nodes Far, Interference Close

In this case we have the nodes are far and the interference is close. This case can be considered as the worst case as the interference is stronger than the desired signal. We expect that disabling carrier sensing should not help in this particular case as it will increase the number of collisions. We have the following observations from the graphs shown in Figure 6.9, Figure 6.10, Figure 6.11 and Figure 6.12:

- We see in Figure 6.9 and Figure 6.10 that CS sensing helps in every case despite the transmit power used.
- It is expected to see that at 54Mbps, no CS with highest transmit power performs worst. As the interference is highest in this particular case.

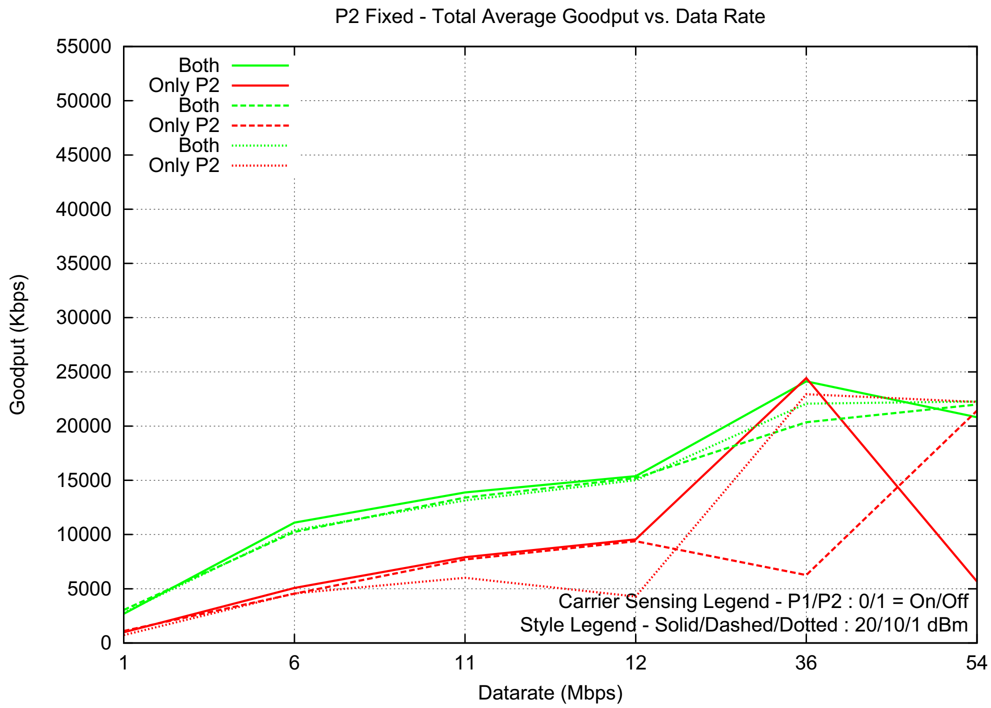

Figure 6.9: Test 3: Nodes Far, Interference Close

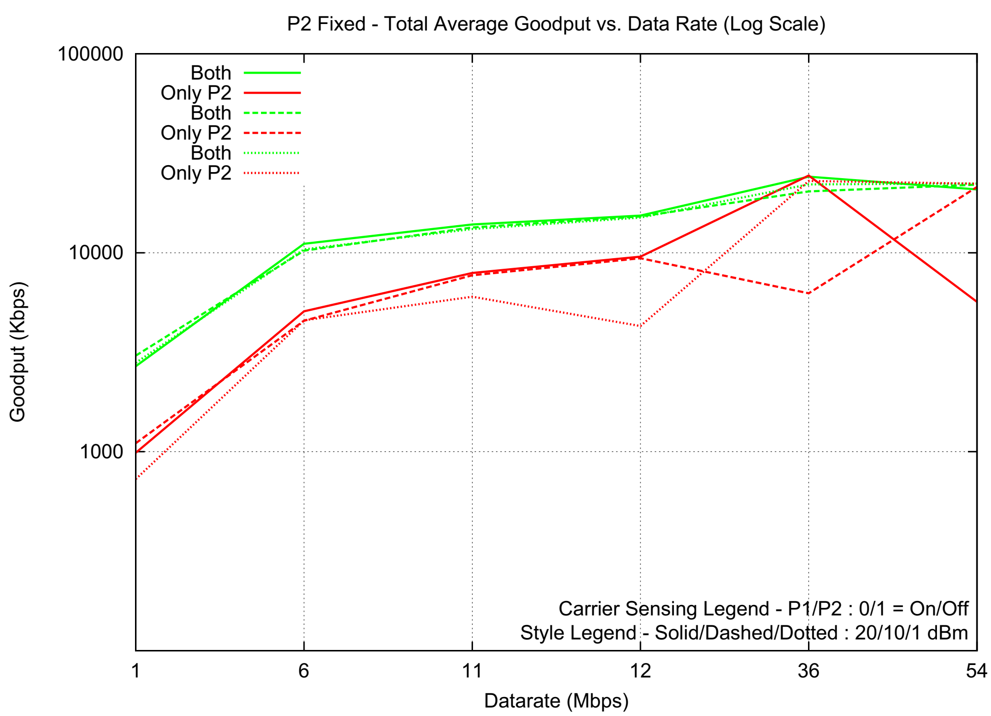

Figure 6.10: Test 3: Nodes Far, Interference Close

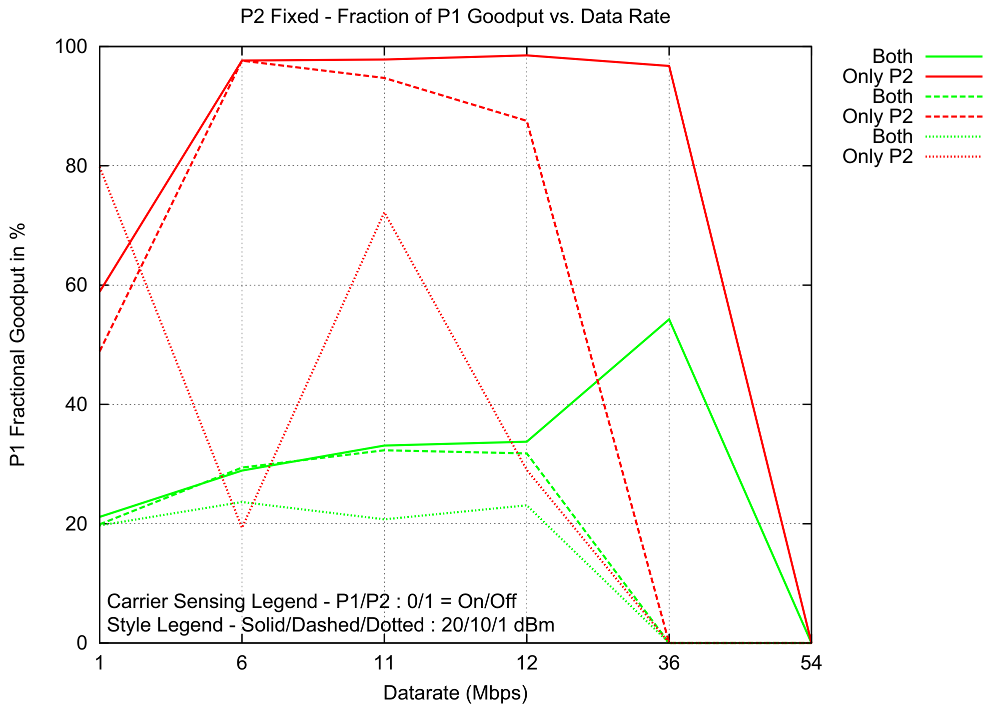

Figure 6.11: Test 3: Nodes Far, Interference Close

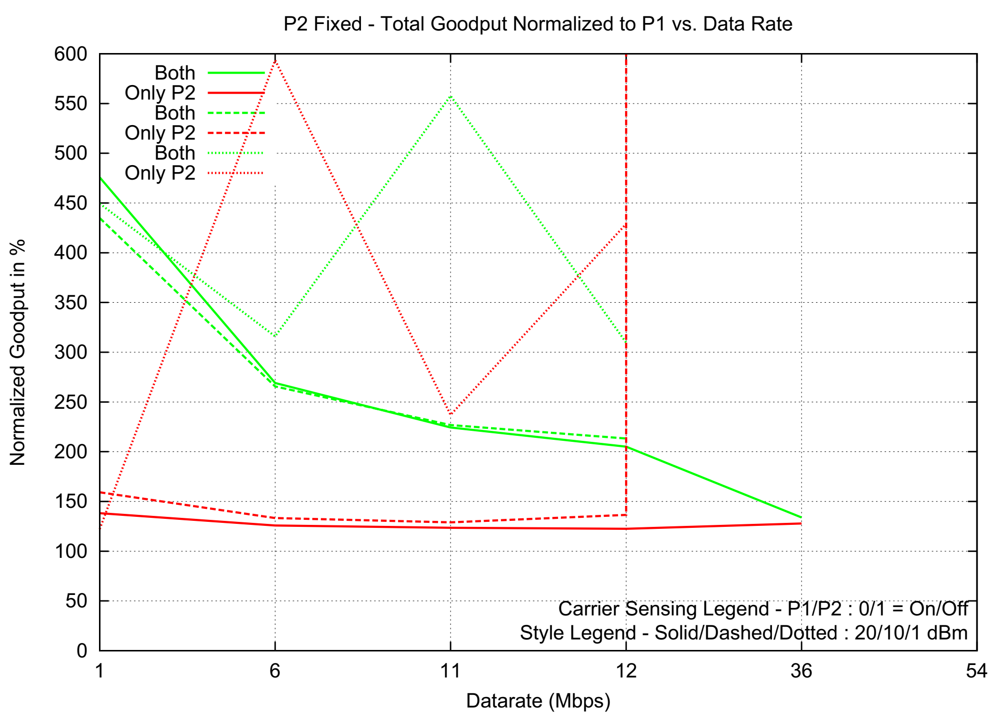

Figure 6.12: Test 3: Nodes Far, Interference Close

6.5 Test 4: Nodes Far, Interference Far

In this case we have the nodes and interferences are far away. Even though the desired signals are weak, the interference is weak too. We assume that using highest transmit power without CS should help increase the cumulative throughput. We have the following observations from the graphs shown in Figure 6.13, Figure 6.14, Figure 6.15 and Figure 6.16:

- Figure 6.13 shows that using highest transmit power yields the least throughput. The higher transmit power increases the interferences and eventually leads to lower throughput.
- Figure 6.15 shows that P2 overtakes the whole channel when P1 is not using the highest transmit power. This is expected as P2 is always using the highest transmit power.
- The channel is shared chaotically between the pairs when highest transmit power is used.

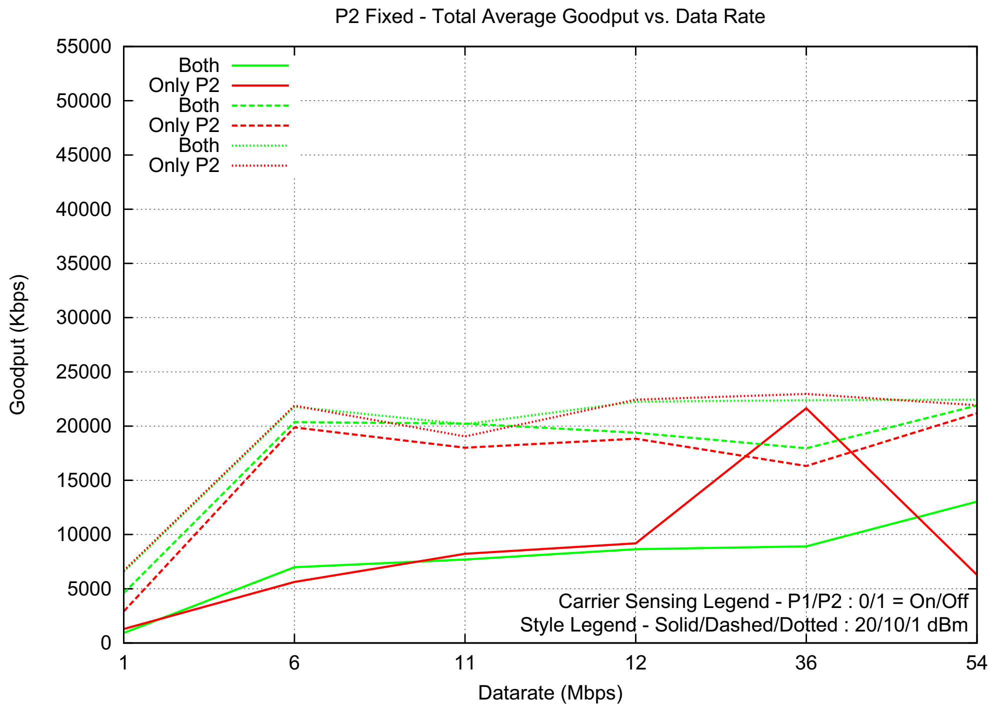

Figure 6.13: Test 4: Nodes Far, Interference Far

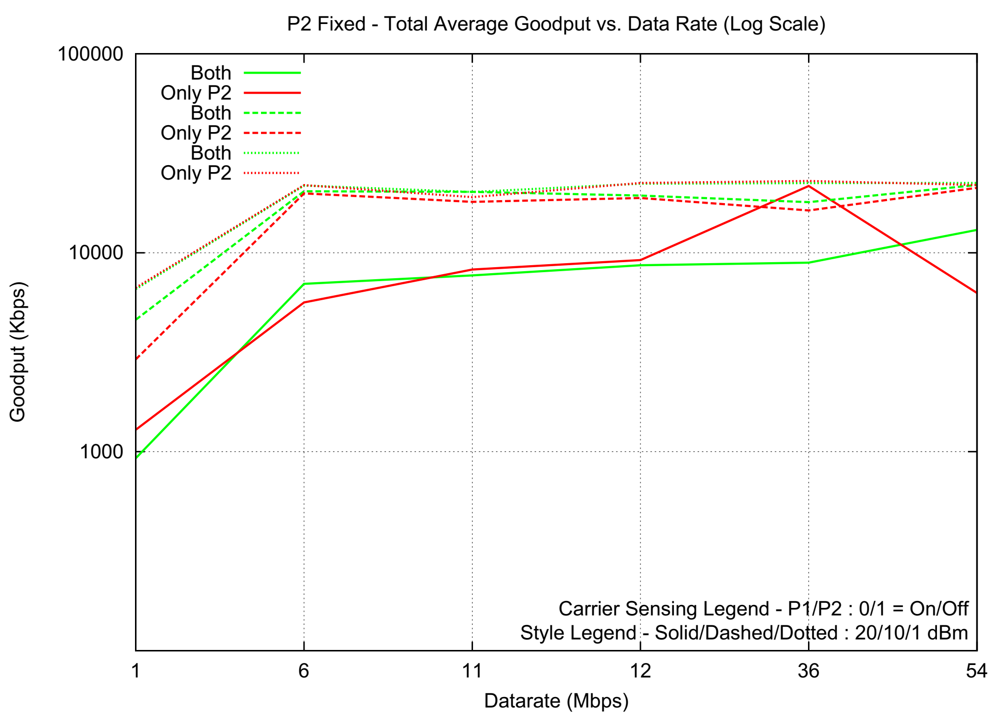

Figure 6.14: Test 4: Nodes Far, Interference Far

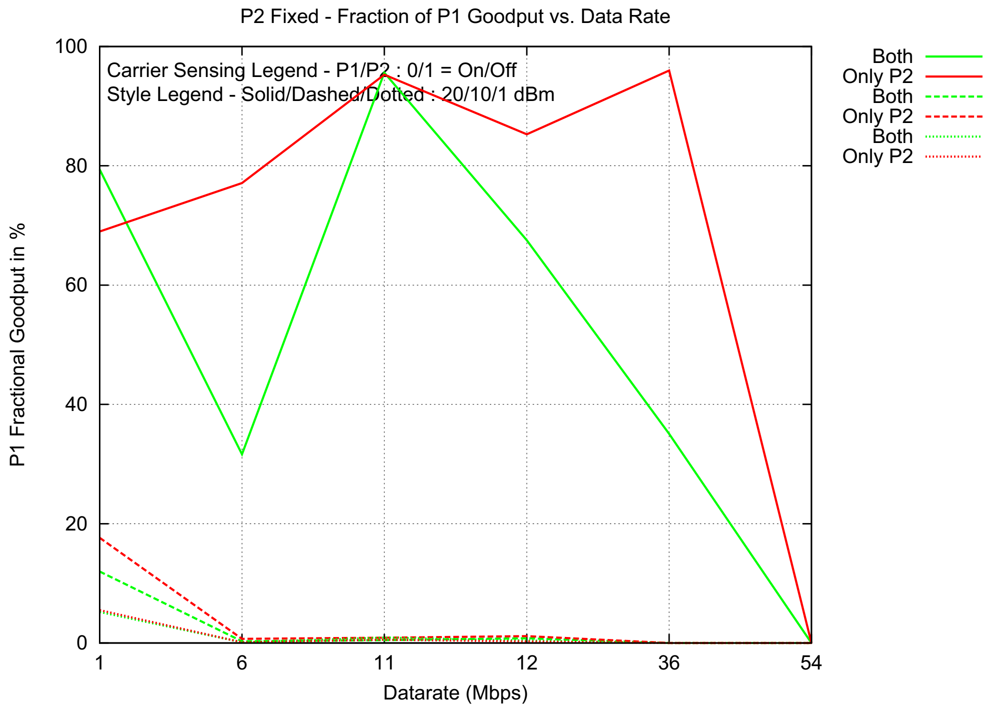

Figure 6.15: Test 4: Nodes Far, Interference Far

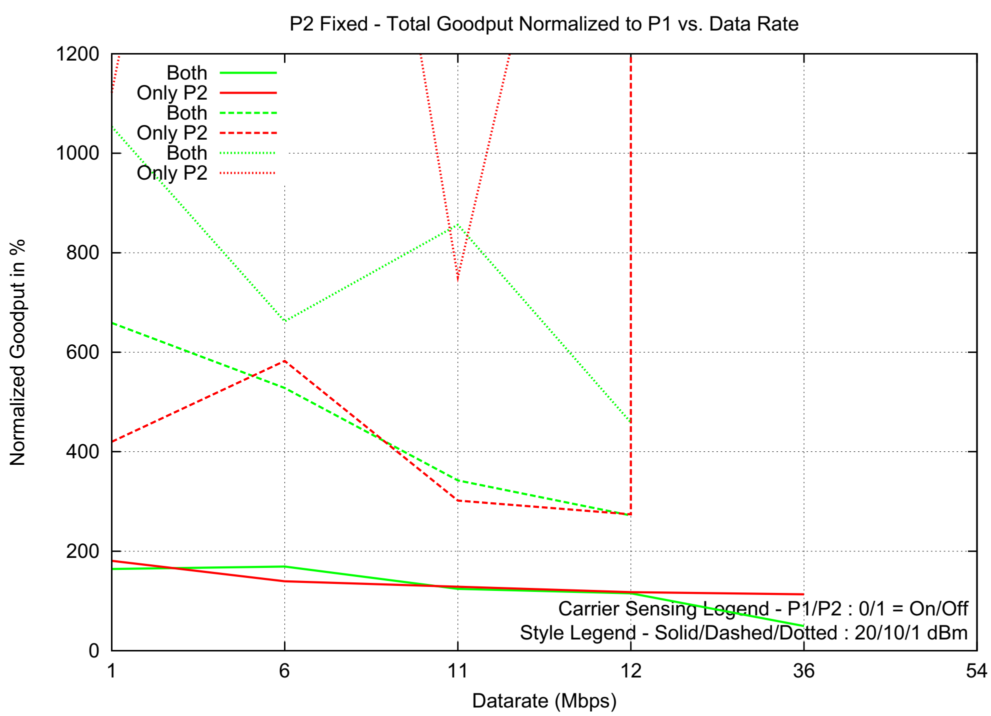

Figure 6.16: Test 4: Nodes Far, Interference Far

6.6 Deviation in Results

The deviation in results was calculated to evaluate the reliability and repeatability of the experiments. For each of the four iterations performed a mean value was calculated. Then the mean and standard deviation (SD) of these four means was calcuated. Finally the SD was normalized by the mean value to analyze the deviation in the SD.

The data shows that the results are consistent. Most of the values are less than 4%. There are outliers in the experiments for which we need to do a more critical analysis of data for better understanding.

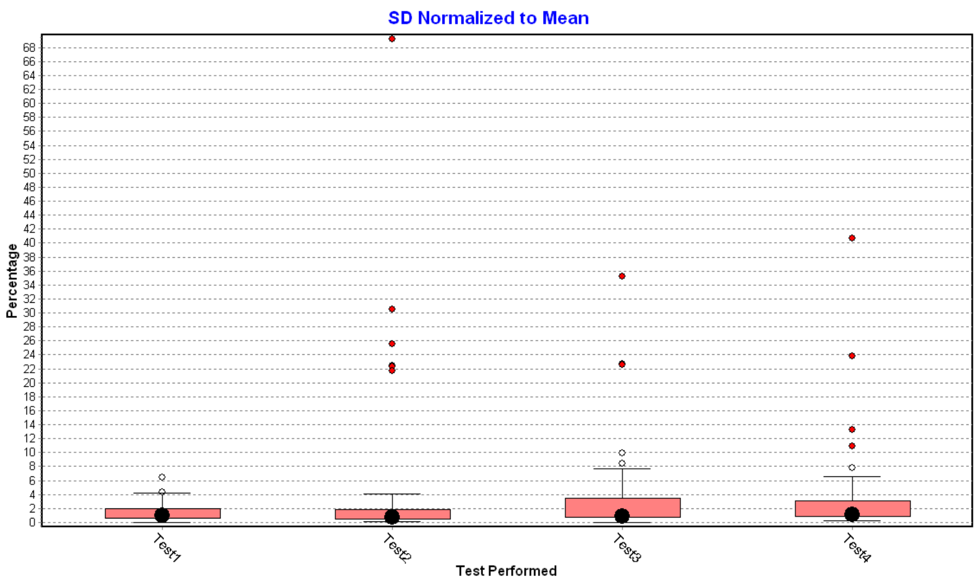

Figure 6.17: Shows the deviation in the SD test performed normalized to Mean

Chapter 7

Conclusion

We conclude that reducing the transmit power and disabling the carrier sensing can increase the cumulative goodput in a dense WiFi network. Previous work considers the effect of changing the transmit power and carrier sensing separately on the goodput. However, this thesis presents a controlled approach in performing the experiments, while changing multiple parameters and physical scenarios.

Four different physical scenarios were developed using RF Isolation boxes. During this process we realized that creating a reliable test bed which provides repeatable results is a great challenge. The experiments were performed with using automated scripts, which multiple locks to synchronize the nodes used in the experiments. The data was collected, processed and plotted using scripts to reduce human error as much as possible.

A summary of the combinations which provide the best good put in the tested scenarios is shown in the Table 7.1. In Test-1, we see that transmitting at higher power is better for Hypothesis-I (H-1) and Hypothesis-II (H-2). In Hypothesis-III (H-3) when the other pair has fixed data rate and transmit power it is best to use no CS and lowest transmit power. In Test-2, when the pairs are far apart we see that using lowest transmit power with no CS yields best cumulative goodput. The exception again is H-3 when it is better to do CS when the other pair is transmitting at highest transmit power. In Test-3, as the pairs are close it is best to transmit at highest power and only one of the pairs doing CS. Both pairs doing CS help in H-3 to yield better cumulative goodput. In Test-4, as the nodes are all far it is clearly the best to use maximum transmit power and a lower

Table 7.1: This table shows the combination which yields best goodput in each scenario.

	Test 1 (Nodes Close, Interference Close)	Test 2 (Nodes Close, Interference Far)	Test 3 (Nodes Far, Interference Close)	Test 4 (Nodes Far, Interference Far)
Hypothesis – I	54 Mbps, 10 dBm, Only P2 CS	54 Mbps, 1 dBm, No CS	36 Mbps, 20 dBm, Only P2 CS	36 Mbps, 20 dBm, Only P2 CS
Hypothesis – II (P2 Swapped)	54 Mbps, 20 dBm, Only P2 CS	54 Mbps, 1 dBm, No CS	36 Mbps, 20 dBm, Only P1 CS	36 Mbps, 20 dBm, Only P2 CS
Hypothesis – III (P2 Fixed)	54 Mbps, 1 dBm, No CS	54 Mbps, 1 dBm, CS	36 Mbps, 20 dBm, CS	36 Bps, 1 dBm, CS and No CS

rate (36Mbps) to yield higher cumulative goodput.

In conclusion, when nodes are close it helps to use low transmit power and high data rate for better goodput without CS. If the nodes are far using higher transmit power with lower data rate works better with either or both of the pairs not doing CS.

The test bed measurements can be extended in several directions. First, extensive testing of the networks with more than two pairs can be performed to figure out their interaction among each other. However, the complexity for the number of interconnecting cables grows quadratically in the number of pairs. For a large number of pairs the approach will need to be modified. Second, currently we have considered only four physical scenarios in our experiments. As noted before, the testbed can simulate 1 trillion different physical scenarios. Third, more combinations of data rate, transmit power and carrier sensing can be used. Finally several anomolies were noted in the data including Pair-1 outperforming Pair-2, first iteration of the tests was more erractic than the others, we saw normalized values of more than 200% which needs to be investigated. Further testing should be able to provide more insights on their source.

Bibliography

[1] Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. Self-Management in Chaotic Wireless Deployments. Wirel. Netw., 13(6):737–755, December 2007.

[2] WiFi Alliance. WiFi Alliance, November 2010. http://www.wi-fi.org/.

[3] Eric Anderson. Clear-channel assessment suppression, November 2010. https://systems.cs.colorado.edu/projects/carp/wiki/WikiStart.

[4] Yigal Bejerano, Seung-Jae Han, and Li E. Li. Fairness and Load Balancing in Wireless LANs Using Association Control. IEEE/ACM Transactions on Networking, 15(3):560–573, June 2007.

[5] Pierre Bremaud. Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues. 1st edition edition, 1999.

[6] A. Di Stefano, G. Terrazzino, L. Scalia, I. Tinnirello, G. Bianchi, and C. Giaconia. An Experimental Testbed and Methodology for Characterizing IEEE 802.11 Network Cards. In WOWMOM '06: Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, pages 513–518, Washington, DC, USA, 2006. IEEE Computer Society.

[7] Mesut A. Ergin, Kishore Ramachandran, and Marco Gruteser. Understanding the Effect of Access Point density on Wireless LAN Performance. In Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, MobiCom '07, pages 350–353, New York, NY, USA, 2007. ACM.

[8] Glenn Fleishman. The future of WiFi: Gigabit Speeds and Beyond. Ars Techinica Business Guide, December 7, 2009. http://arstechnica.com/business/guides/2009/12/wifi-looks-to-1-gigabit-horizon.ars.

[9] Martin Heusse. IPMT Measurement Tool, November 2010. http://ipmt.ligforge.imag.fr/.

[10] Glenn Judd, Xiaohui Wang, and Peter Steenkiste. Efficient Channel-Aware Rate Adaptation in Dynamic Environments. In Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services, MobiSys '08, pages 118–131, New York, NY, USA, 2008. ACM.

[11] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot. Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks. INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, pages 1451–1459, 2007.

[12] Hui Ma, Jing Zhu, Sumit Roy, and Soo Y. Shin. Joint Transmit Power and Physical Carrier Sensing Adaptation Based on Loss Differentiation for High Density IEEE 802.11 WLAN. Comput. Netw., 52(9):1703–1720, 2008.

[13] ORBIT. The orbit project, November 2010. http://www.orbit-lab.org/.

[14] K. Papagiannaki, M. Yarvis, and W. S. Conner. Experimental Characterization of Home Wireless Networks and Design Implications. INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pages 1–13, 2006.

[15] The MadWiFi Project. The MadWiFi Project, November 2010. http://madwifi-project.org/.

[16] S. Sanghani, T. X. Brown, S. Bhandare, and S. Doshi. EWANT: The Emulated Wireless Ad Hoc Network Testbed. pages 1844–1849, 2003.

[17] David B. Shmoys and Éva Tardos. An Approximation Algorithm for the Generalized Assignment Problem. Math. Program., 62:461–474, December 1993.

[18] IEEE Standards. IEEE 802.11: The Working Group for WLAN Standards, November 2010. http://www.ieee802.org/11/.

Appendix A

Initializing the WiFi Card

```
#!/bin/sh
#first arg: wlanmode
#second arg: essid
#third arg : IP address

ath_str=`ifconfig | grep ath`
ath_num=`expr "$ath_str" : '\(ath.\)'`
wlanconfig $ath_num destroy
wlanconfig $ath_num create wlandev wifi0 wlanmode $1
ifconfig $ath_num down
sleep 1
iwconfig $ath_num freq 2.412G
sleep 1
iwconfig $ath_num rate 1Mbps
sleep 1
iwconfig $ath_num essid "$2"
sleep 1
ifconfig $ath_num up $3
iwconfig $ath_num
```

Appendix B

Setup Rate of WiFi Card

```
#!/bin/sh
#first arg: rate
#second arg: logFile Name

doneFlag=0
while [ $doneFlag -eq 0 ]
do
ath_str=`ifconfig | grep ath`
ath_num=`expr "$ath_str" : '\(ath.\)'`
ifconfig $ath_num down
sleep 1
iwconfig $ath_num rate $1Mbps
sleep 1
ifconfig $ath_num up

testString=$(iwconfig $ath_num | grep Rate)
testRate=$(echo $testString | cut -d " " -f2 | cut -d "=" -f2)
if [ $testRate == $1 ]
then
        doneFlag=1
        echo "Rate setting is successful. Rate has been set to $1" >> "$2"
else
        echo "Error in setting Rate. Rate setting will be reattempted" >> "$2"
fi
sleep 1
done
```

Appendix C

Setup Power of WiFi Card

```
#!/bin/sh
#first arg: power
#second arg: logFile Name

doneFlag=0
while [ $doneFlag -eq 0 ]
do
ath_str=`ifconfig | grep ath`
ath_num=`expr "$ath_str" : '\(ath.\)'`
ifconfig $ath_num down
sleep 1
iwconfig $ath_num txpower $1
sleep 1
ifconfig $ath_num up

powerString=$(iwconfig $ath_num | grep Tx-Power)
testPower=$(echo $powerString | cut -d " " -f4 | cut -d "=" -f2)
if [ $testPower = $1 ]
then
        doneFlag=1
        echo "Power set successfully. Power has been set to $1" >> "$2"
else
        echo "Error in setting the power. Power set will be reattempted" >> "$2"
fi
sleep 1
done
```

Appendix D

Gather Information of WiFi Card

```
#!/bin/sh

ath_str=`ifconfig | grep ath`
ath_num=`expr "$ath_str" : '\(ath.\)'`

iwconfig $ath_num | grep ESSID | cut -d : -f2|cut -d "␣" -f1>>../files/inputFiles/headInfo
iwconfig $ath_num | grep Mode | cut -d : -f2|cut -d "␣" -f1>>../files/inputFiles/headInfo
iwconfig $ath_num | grep Frequency | cut -d : -f3|cut -d "␣" -f1>>../files/inputFiles/headInfo
iwconfig $ath_num | grep Rate | cut -d "=" -f2|cut -d "␣" -f1>>../files/inputFiles/headInfo
iwconfig $ath_num | grep Tx-Power | cut -d "=" -f3|cut -d "␣" -f1>>../files/inputFiles/headInfo
iwconfig $ath_num | grep Signal | cut -d "=" -f3|cut -d "␣" -f1>>../files/inputFiles/headInfo
iwconfig $ath_num | grep Noise | cut -d "=" -f4|cut -d "␣" -f1>>../files/inputFiles/headInfo
cat /proc/sys/dev/wifi0/disable_cca >> ../files/inputFiles/headInfo
```

Appendix E

Script at Receiving Node

```
#!/bin/sh

rm scpReleaseT
fileName="FileCase_$4"
filePath="$PWD"
#filePath="/root/WorkingDirectory/TrafficTesting/Research_Repository"

# Obtain ip address for correct receiver from file
ipaddress=$(cat ../files/inputFiles/tranIdFile | grep ""$7":" | cut -d ":" -f2)
# scp path to that ip address
scpCommand="root@$ipaddress:$filePath/."

if [ $5 -eq 0 ]
then # run selected tests from the file.
        while read line
        do
                rate=$(echo $line | cut -d "," -f1)
                power=$(echo $line | cut -d "," -f2)
                carsen=$(echo $line | cut -d "," -f3)
                flag=$(echo $line | cut -d "," -f4)

if [ $flag -eq 1 ]
then
        # Set rates
        ./script1.sh $rates ../files/logFiles/logFile_$fileName
        sleep 1
        # Set power levels
        ./script2.sh $power ../files/logFiles/logFile_$fileName
        sleep 2
        # Enable carrier sensing
        echo $carsen > /proc/sys/dev/wifi0/disable_cca
        sleep 1
        ./collectInfo
        sleep 1
        # Compute start time stamp and write to output and log files
        startTime=`date +%Y%m%d%H%M%S`
        echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/outputFiles/$fileName
```

```
        echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/logFiles/logFile_$fileName
        iwconfig >> ../files/logFiles/logFile_$fileName

        # Executing the measurement tool
        ../udptarget -p $1 -w $4 -f $6

        # Compute end time stamp and write to output and log files
        endTime=`date +%Y%m%d%H%M%S`
        echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/outputFiles/$fileName
        echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/logFiles/logFile_$fileName
        timeDiff=`expr $endTime - $startTime`
        echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/outputFiles/$fileName
        echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/logFiles/logFile_$fileName
        sleep 5 # This sleep determines # packets sent
fi
done < "../files/inputFiles/optionsFile"

else # run form loop. i.e. run everything

        ./script.sh adhoc $2 $3

        rates=( 1 6 11 12 36 54 )
        power=( 20 10 1 )

        numrates=`expr ${#rates[*]} - 1`
        numpower=`expr ${#power[*]} - 1`

for (( i = 0; i <= $numrates; i++ ))
do
  # Setting the rates
        ./script1.sh ${rates[$i]} ../files/logFiles/logFile_$fileName
        sleep 1
        for (( j = 0; j <= $numpower; j++ ))
do
        # Setting the power levels
        ./script2.sh ${power[$j]} ../files/logFiles/logFile_$fileName
        sleep 2
        # Enable carrier sensing
        echo 0 > /proc/sys/dev/wifi0/disable_cca
        sleep 1
        ./collectInfo
        sleep 1

        # Compute the start time stamp and write to output and log files
        startTime=`date +%Y%m%d%H%M%S`
        echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/outputFiles/$fileName
        echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/logFiles/logFile_$fileName
        iwconfig >> ../files/logFiles/logFile_$fileName # LogFile
        scp scpRelease $scpCommand

        # Execute the testing tool
```

```
../udptarget -p $1 -w $4 -f $6

# Compute the end time stamp and write to output and long files
endTime=`date +%Y%m%d%H%M%S`
echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/outputFiles/$fileName
echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/logFiles/logFile_$fileName
timeDiff=`expr $endTime - $startTime`
echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/outputFiles/$fileName
echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/logFiles/logFile_$fileName

# Disable carrier sensing
echo 2 > /proc/sys/dev/wifi0/disable_cca
sleep 1
./collectInfo
sleep 1

# Compute the start time stamp and write to output and log files
startTime=`date +%Y%m%d%H%M%S`
echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/outputFiles/$fileName
echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/logFiles/logFile_$fileName
iwconfig >> ../files/logFiles/logFile_$fileName # LogFile
scp scpRelease $scpCommand

# Execute the testing tool
../udptarget -p $1 -w $4 -f $6

# Compute the end time stamp and write to output and log files
endTime=`date +%Y%m%d%H%M%S`
echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/outputFiles//$fileName
echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/logFiles/logFile_$fileName
timeDiff=`expr $endTime - $startTime`
echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/outputFiles/$fileName
echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/logFiles/logFile_$fileName

# Two way synchronization handshake between transmitter and receiver
relFlagT=0;
scp scpReleaseR $scpCommand
    # Check for scpRelease received from the transmitter
      while [ $relFlagT -eq 0 ] # Control stays until scpRelease found
        do
        if [ -f scpReleaseT ]
        then
          echo "Removed_scpReleaseT"
          rm scpReleaseT
          relFlagT=1
        fi
        done

done
done
fi
```

Appendix F

Script at Transmitting Node

```
#!/bin/sh

if [ -f scpReleaseR ]
then
  rm scpReleaseR
fi


filePath="$PWD"

# Obtain ip address for correct receiver from file
ipaddress=$(cat ../files/inputFiles/tranIdFile | grep ""$8":" | cut -d ":" -f2)
echo $ipaddress
# scp path to that ip address
scpCommand="root@$ipaddress:$filePath/." # Need to pull from a file
echo $scpCommand

relFlag=0
fileName="FileCase_$7"
if [ $6 -eq 0 ]
then # Run from file i.e. run selected tests

  while read line
  do
    #echo $line
    rate=$(echo $line | cut -d "," -f1)
    power=$(echo $line | cut -d "," -f2)
    carsen=$(echo $line | cut -d "," -f3)
    flag=$(echo $line | cut -d "," -f4)

    if [ $flag -eq 1 ]
    then
            # Adjust rates
      ./script1.sh $rates ../files/logFiles/logFile_$fileName
      sleep 1
            # Adjust power
      ./script2.sh $power ../files/logFiles/logFile_$fileName
```

```
      sleep 4
    # Enable carrier sensing
      echo $carsen  > /proc/sys/dev/wifi0/disable_cca
    #sleep 1

    startTime=`date +%Y%m%d%H%M%S`
    echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/logFiles/logFile_$fileName
    iwconfig >> ../files/logFiles/logFile_$fileName

    # Execute traffic measurement tool
    ../udpmt -p $1 $2 -d $5

    endTime=`date +%Y%m%d%H%M%S`
    echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/logFiles/logFile_$fileName
    timeDiff=`expr $endTime - $startTime`
    echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/logFiles/logFile_$fileName

                sleep 15 # This sleep determines # packets sent
    fi
done < optionsFile


else # Run from loop i.e. run everything

        ./script.sh adhoc $3 $4

        rates=( 1 6 11 12 36 54 )
        power=( 20 10 1 )
        numrates=`expr ${#rates[*]} - 1`
        numpower=`expr ${#power[*]} - 1`

        for (( i = 0; i <= $numrates; i++ ))
        do
                # Adjust rates
                ./script1.sh ${rates[$i]} ../files/logFiles/logFile_$fileName
                sleep 1
                for (( j = 0; j <= $numpower; j++ ))
                do
                        # Adjust power
                        ./script2.sh ${power[$j]} ../files/logFiles/logFile_$fileName
                        sleep 4
                        # Enable carrier sensing
                        echo 0 > /proc/sys/dev/wifi0/disable_cca
                        # Check for scpRelease
                        while [ $relFlag -eq 0 ] # Control stays inside until scpRelease found
                        do
                                if [ -f scpRelease ]
                                then
                                        echo "Removed scpRelease"
                                        rm scpRelease
                                        relFlag=1
```

```
                        fi
                done

# This piece of code helps us synchronize with the other transmitter
echo "Waiting_for_file"
while read line
do
  ipaddress=$(echo $line | cut -d "_" -f1)
  myipaddress=$(ifconfig eth0 | grep Bcast | cut -d ":" -f2 | cut -d "_" -f1)
        if [ "$ipaddress" = "$myipaddress" ]
        then
                scpOpt=$(echo $line | cut -d "_" -f2)
                scpFile="SR$scpOpt"
                echo $scpFile
  else
         t_address=$ipaddress
                rOpt=$(echo $line | cut -d "_" -f2)
                rFile="SR$rOpt"
                echo $t_address
        fi
                done < ../files/inputFiles/transSyncFile
                scpCmd="root@$t_address:$filePath/."
                scp $scpFile $scpCmd
                rFlag=0
                while [ $rFlag -eq 0 ] # Control stays inside while until $rFile file is found
                do
                        if [ -f $rFile ]
                        then
                                echo "Removed_$rFile"
                                rm $rFile
                                rFlag=1
                        fi
                done

                sleep 1

startTime=`date +%Y%m%d%H%M%S`
echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/logFiles/logFile_$fileName
iwconfig >> ../files/logFiles/logFile_$fileName

echo "Starting_udpmt"
# Executing traffic measurement tool
../udpmt -p $1 $2 -d $5

endTime=`date +%Y%m%d%H%M%S`
echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/logFiles/logFile_$fileName
timeDiff=`expr $endTime - $startTime`
echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/logFiles/logFile_$fileName

relFlag=0
```

```
# Disable carrier sensing
echo 2 > /proc/sys/dev/wifi0/disable_cca
# Check for scpRelease
while [ $relFlag -eq 0 ] # Control stays until scpRelease file found
        do
                if [ -f scpRelease ]
                then
                        echo "Removed scpRelease"
                        rm scpRelease
                        relFlag=1
                fi
        done
        sleep 1

# This piece of code helps us synchronize with the other transmitter
echo "Waiting for file"
while read line
        do
    ipaddress=$(echo $line | cut -d " " -f1)
    myipaddress=$(ifconfig eth0 | grep Bcast | cut -d ":" -f2 | cut -d " " -f1)
            if [ "$ipaddress" = "$myipaddress" ]
                    then
                        scpOpt=$(echo $line | cut -d " " -f2)
                                        scpFile="SR$scpOpt"
                                        echo $scpFile
                         else
                                        t_address=$ipaddress
                                        rOpt=$(echo $line | cut -d " " -f2)
                                        rFile="SR$rOpt"
                                        echo $t_address
                         fi
        done < ../files/inputFiles/transSyncFile
        scpCmd="root@$t_address:$filePath/."
        scp $scpFile $scpCmd
        rFlag=0
        while [ $rFlag -eq 0 ] # Control stays inside while until $rFile file is found
        do
                if [ -f $rFile ]
                then
                        echo "Removed $rFile"
                        rm $rFile
                        rFlag=1
                fi
        done

        sleep 1

  startTime=`date +%Y%m%d%H%M%S`
  echo "<STARTTIME>$startTime</STARTTIME>" >> ../files/logFiles/logFile_$fileName
  iwconfig >> ../files/logFiles/logFile_$fileName
        echo "Starting udpmt"
```

```
                        # Executing traffic measurement tool
                        ../udpmt -p $1 $2 -d $5

endTime=`date +%Y%m%d%H%M%S`
echo "<ENDTIME>$endTime</ENDTIME>" >> ../files/logFiles/logFile_$fileName
timeDiff=`expr $endTime - $startTime`
echo "<TIMEDIFF>$timeDiff</TIMEDIFF>" >> ../files/logFiles/logFile_$fileName


# Two way synchronization handshake between transmitter and receiver
relFlagR=0;
scp scpReleaseT $scpCommand
# Check for scpRelease received from the transmitter
while [ $relFlagR -eq 0 ] # Control stays until scpRelease found
  do
     if [ -f scpReleaseR ]
     then
        echo "Removed_scpReleaseR"
        rm scpReleaseR
        relFlagR=1
     fi
  done

done
done
fi
```

Appendix G

Wrapper Script for Receiver Script

```
#!/bin/sh
if [ $1 -eq 1 ]
then
        ./recvTest1 5000 hakim1 20.0.0.2 rx1Test$2run1a 1 100 1
        sleep 60
        ./recvTest1 5000 hakim1 20.0.0.2 rx1Test$2run2a 1 100 1
        sleep 60
        ./recvTest1 5000 hakim1 20.0.0.2 rx1Test$2run3a 1 100 1
        sleep 60
        ./recvTest1 5000 hakim1 20.0.0.2 rx1Test$2run4a 1 100 1
else
        ./recvTest1 5000 hakim2 20.0.0.4 rx2Test$2run1a 1 100 2
        sleep 60
        ./recvTest1 5000 hakim2 20.0.0.4 rx2Test$2run2a 1 100 2
        sleep 60
        ./recvTest1 5000 hakim2 20.0.0.4 rx2Test$2run3a 1 100 2
        sleep 60
        ./recvTest1 5000 hakim2 20.0.0.4 rx2Test$2run4a 1 100 2
fi
```

Appendix H

Wrapper Script for Transmitter Script

```
#!/bin/sh
if [ $1 -eq 1 ]
then
        ./transTest1 5000 20.0.0.2 hakim1 20.0.0.1 100 1 tx1Test$2run1a 3
        sleep 60
        ./transTest1 5000 20.0.0.2 hakim1 20.0.0.1 100 1 tx1Test$2run2a 3
        sleep 60
        ./transTest1 5000 20.0.0.2 hakim1 20.0.0.1 100 1 tx1Test$2run3a 3
        sleep 60
        ./transTest1 5000 20.0.0.2 hakim1 20.0.0.1 100 1 tx1Test$2run4a 3
else
        ./transTest1 5000 20.0.0.4 hakim2 20.0.0.2 100 1 tx2Test$2run1a 4
        sleep 60
        ./transTest1 5000 20.0.0.4 hakim2 20.0.0.2 100 1 tx2Test$2run2a 4
        sleep 60
        ./transTest1 5000 20.0.0.4 hakim2 20.0.0.2 100 1 tx2Test$2run3a 4
        sleep 60
        ./transTest1 5000 20.0.0.4 hakim2 20.0.0.2 100 1 tx2Test$2run4a 4
fi
```

Appendix I

Processing Raw Data for Averages

```python
#!/usr/bin/env python

import numpy as np
import matplotlib.pyplot as mpl

from math import *
from pylab import *

# ---------------------- #
# Variable Initializations #
# ---------------------- #
flag = 0
data = zeros( [72,120] )
count, counter = 0, 0

# ---------------------------- #
# Function for computing averages #
# ---------------------------- #
def computeAverages(fileName,outFileID):
        # -------------- #
        # Reading the file #
        # -------------- #
        global flag, data, count, counter
        data = zeros( [72,120] )
        count = 0 # Counts the number of lines
        counter = 0 # Counts the number of data points for each test
        print "Computing Average"

        f = open(fileName,"r") # Open the xml data file for reading
        for line in f.readlines(): # Read each line
            line = line.strip()  # Strip out leading and trailing spaces
            if line == " ": # If line is empty, ignore it
                continue
            else: # If line is not empty do this
                dataline = line.split(" ") # Break line on the basis of spaces
                if line == "</DATA>": # When the data end tag is read, reset flag
```

```
                flag = 0
                count = count + 1
        if flag == 1: # When the flag is set, read the data
                data[count][counter] = float(dataline[-1])
                counter = counter + 1 # Increment the data point
        if line == "<DATA>":
                flag = 1 # Set flag on encountering begin data tag
                counter = 0 # Set data counter to zero
f.close() # Close file
# ------------------------------------ #
# Carrying out computations on the data #
# ------------------------------------ #
dataAverages = zeros( [72,6] )
for ind1 in range(0,len(data)): #
        tmpdata = data[ind1,:]
        dataAverages[ind1][0] = np.average(data[ind1,:]) # Mean
        dataAverages[ind1][1] = np.std(data[ind1,:]) # Variance
        dataAverages[ind1][2] = tmpdata.min() # Minimum
        dataAverages[ind1][3] = tmpdata.max() # Maximum
        dataAverages[ind1][4] = np.median(abs(data[ind1,:] - np.median(tmpdata)))
        dataAverages[ind1][5] = np.median(tmpdata) # Median

if outFileID == 1:
   fout = open('rx1a_Test1_avgFileOut1.txt',"w")
elif outFileID == 2:
   fout = open('rx1a_Test1_avgFileOut2.txt',"w")
elif outFileID == 3:
   fout = open('rx1a_Test1_avgFileOut3.txt',"w")
elif outFileID == 4:
   fout = open('rx1a_Test1_avgFileOut4.txt',"w")


# Writing the statistics to a file.
for ind1 in range(0,len(dataAverages)):
        tmpline = ''
        tmpline += str(dataAverages[ind1,0])
        tmpline += ','
        tmpline += str(dataAverages[ind1,1])
        tmpline += ','
        tmpline += str(dataAverages[ind1,2])
        tmpline += ','
        tmpline += str(dataAverages[ind1,3])
        tmpline += ','
        tmpline += str(dataAverages[ind1,4])
        tmpline += ',' # A1
        tmpline += str(dataAverages[ind1,5]) # A2


        tmpline = str(tmpline)

        fout.writelines(str(tmpline))
```

```
            fout.writelines("\n") # Newline after each line of 6 statistics
        fout.close() # Close file
        return data # Return all data read to the main program

# MAIN PROGRAM
if __name__ == "__main__":
    # Call compute averages for each of the four files
    fileName = "FileCase_rx1Test1run1a"
    data1 = computeAverages(fileName,1)
    fileName = "FileCase_rx1Test1run2a"
    data2 = computeAverages(fileName,2)
    fileName = "FileCase_rx1Test1run3a"
    data3 = computeAverages(fileName,3)
    fileName = "FileCase_rx1Test1run4a"
    data4 = computeAverages(fileName,4)

    # Forming a single array containing data from all files
    dataArray = zeros( [72,480] )
    for ix1 in range(0,72):
        for ix2 in range(0,120):
            dataArray[ix1,ix2] = data1[ix1,ix2]
            dataArray[ix1,ix2+120] = data2[ix1,ix2]
            dataArray[ix1,ix2+240] = data3[ix1,ix2]
            dataArray[ix1,ix2+360] = data4[ix1,ix2]

    # Computing statistics exactly as in case of individual files
    dataAverages = zeros( [72,6] )
    for ind1 in range(0,len(dataArray)):
        tmpdata = dataArray[ind1,:]
        dataAverages[ind1][0] = np.average(dataArray[ind1,:]) # Mean
        dataAverages[ind1][1] = np.std(dataArray[ind1,:]) # Variance
        dataAverages[ind1][2] = tmpdata.min() # Minimum
        dataAverages[ind1][3] = tmpdata.max() # Maximum
        dataAverages[ind1][4] = np.median(abs(data[ind1,:] - np.median(tmpdata)))
        dataAverages[ind1][5] = np.median(tmpdata) # Median

    # Writing the statistics to a separate file
    fout = open('rx1a_Test1_avgFileOutCombined.txt',"w")
    for ind1 in range(0,len(dataAverages)):
         tmpline = ''
         tmpline += str(dataAverages[ind1,0])
         tmpline += ','
         tmpline += str(dataAverages[ind1,1])
         tmpline += ','
         tmpline += str(dataAverages[ind1,2])
         tmpline += ','
         tmpline += str(dataAverages[ind1,3])
         tmpline += ','
         tmpline += str(dataAverages[ind1,4])
         tmpline += ','
         tmpline += str(dataAverages[ind1,5])
```

```
        tmpline = str(tmpline)

        fout.writelines(str(tmpline))
        fout.writelines("\n")
fout.close() # Close file

# Writing the data to a separate file
foutdata = open('rx1a_Test1_DataFileOutCombined.txt',"w")
for ind1 in range(0,len(dataArray)):
    tmpline = ''
    for ind2 in range(0,len(dataArray[0])):
        tmpline += str(dataArray[ind1,ind2])
        tmpline += ','

    tmpline = str(tmpline)
    foutdata.writelines(str(tmpline))
    foutdata.writelines("\n")
foutdata.close() # Close file

print "Averages Computed"

# End program
```

Appendix J

Parsing Average Files for Cumulative Goodput

```
#!/bin/bash

# Gets the Top 36 rows of each of file in which we are interested

head -36 rx1a_Test1_avgFileOutCombined.txt | awk -F, '{print $1}' >> temp1.txt
head -36 rx1b_Test1_avgFileOutCombined.txt | awk -F, '{print $1}' >> temp1.txt
head -36 rx2a_Test1_avgFileOutCombined.txt | awk -F, '{print $1}' >> temp2.txt
head -36 rx2b_Test1_avgFileOutCombined.txt | awk -F, '{print $1}' >> temp2.txt

# Combines the data in Order from Case a and Case b in one File
paste temp1.txt temp2.txt >> temp3.txt

# Generates the file with throughput of P1, P2, and P1+P2 in Col 1,2,3 respectively.
awk '{print $1, $2, $1+$2}' temp3.txt >> Test1-P1P2_Averages.txt

# Generates the file with throughput of P1, P2, and P1+P2 in Col 1,2,3 respectively.

awk '{print $1+$2}' temp3.txt >> temp5.txt        # Gets P1+P2 column
awk '{print $1}' P1-Isolated.txt >> temp6.txt        # Gets First column of P1-Isolated
paste temp5.txt temp6.txt >> temp7.txt     # P1+P2, P1(isolated) in Col 1 and 2 in a file

awk '{print $1, $2, $1/$2*100}' temp7.txt >> Test1-P1P2_Normalized_P1.txt

# Generates the file with throughput of P1, P2, and P1+P2 in Col 1,2,3 respectively.
awk '{print $1, $2, $1/($1+$2)*100}' temp3.txt >> Test1-P1P2_Fractional_P1.txt

rm temp*
```

Appendix K

Generating Goodput Graph

```
set title "Total Avg Throughput - Data Rate Vs Goodput"
set xlabel "Datarate (Mbps)"
set ylabel "Goodput (Kbps)"

set ytics 5000
#set grid mytics

set grid nopolar
set grid xtics nomxtics ytics nomytics noztics nomztics
set grid nox2tics nomx2tics noy2tics nomy2tics nocbtics nomcbtics
set grid layerdefault

#labels at Top left
#set label "Carrier Sensing Legend - P1/P2 : 0/1 = On/Off" at graph 0.01, 0.96
#set label "Style Legend - Solid/Dashed/Dotted : 20/10/1 dBm" at graph 0.01, 0.92

#labels at Bottom Right
set label "Carrier Sensing Legend - P1/P2 : 0/1 = On/Off" at graph 0.99, 0.07 right
set label "Style Legend - Solid/Dashed/Dotted : 20/10/1 dBm" at graph 0.99, 0.03 right

set samples 16
#set key outside right top
set key left top

set xtics ("1"1,"6"2,"11"3,"12"4,"36"5,"54"6);
set yrange [0:55000]


LW=3
FileName="Final-Test1-CumAvg.txt"

set style line 1  lw LW lt 1 lc 2
set style line 2  lw LW lt 1 lc 1
set style line 3  lw LW lt 1 lc 5
set style line 4  lw LW lt 1 lc 3

set style line 5  lw LW lt 2 lc 2
```

```
set style line 6  lw LW lt 2 lc 1
set style line 7  lw LW lt 2 lc 5
set style line 8  lw LW lt 2 lc 3

set style line 9  lw LW lt 4 lc 2
set style line 10  lw LW lt 4 lc 1
set style line 11  lw LW lt 4 lc 5
set style line 12  lw LW lt 4 lc 3

plot\
 FileName using 1:2 w l ls 1 title "Both",
 FileName using 1:3 w l ls 2 title "Neither",
 FileName using 1:4 w l ls 3 title "Only_P2",
 FileName using 1:5 w l ls 4 title "Only_P1",
 FileName using 1:6 w l ls 5 title "Both",
 FileName using 1:7 w l ls 6 title "Neither",
 FileName using 1:8 w l ls 7 title "Only_P2",
 FileName using 1:9 w l ls 8 title "Only_P1",
 FileName using 1:10 w l ls 9 title "Both",
 FileName using 1:11 w l ls 10 title "Neither",
 FileName using 1:12 w l ls 11 title "Only_P2",
 FileName using 1:13 w l ls 12 title "Only_P1";
set term post eps
set output "Test1-normal.eps"
replot
```

Appendix L

Generating Goodput Graph in Log Scale

```
set title "Total Avg Throughput - Data Rate Vs Goodput (Log Scale)"
set xlabel "Datarate (Mbps)"
set ylabel "Goodput (Kbps)"

set ytics (1000, 10000, 100000)
#set grid mytics

unset logscale; set logscale y

set grid nopolar
set grid xtics nomxtics ytics nomytics noztics nomztics
set grid nox2tics nomx2tics noy2tics nomy2tics nocbtics nomcbtics
set grid layerdefault

#labels at Top left
#set label "Carrier Sensing Legend - P1/P2 : 0/1 = On/Off" at graph 0.01, 0.96
#set label "Style Legend - Solid/Dashed/Dotted : 20/10/1 dBm" at graph 0.01, 0.92

#labels at Bottom Right
set label "Carrier Sensing Legend - P1/P2 : 0/1 = On/Off" at graph 0.99, 0.07 right
set label "Style Legend - Solid/Dashed/Dotted : 20/10/1 dBm" at graph 0.99, 0.03 right

set samples 16
#set key outside right top
set key left top

set xtics ("1"1,"6"2,"11"3,"12"4,"36"5,"54"6);
set yrange [100:100000]


LW=3
FileName="Final-Test1-CumAvg.txt"

set style line 1  lw LW lt 1 lc 2
set style line 2  lw LW lt 1 lc 1
set style line 3  lw LW lt 1 lc 5
set style line 4  lw LW lt 1 lc 3
```

```
set style line 5  lw LW lt 2 lc 2
set style line 6  lw LW lt 2 lc 1
set style line 7  lw LW lt 2 lc 5
set style line 8  lw LW lt 2 lc 3

set style line 9  lw LW lt 4 lc 2
set style line 10  lw LW lt 4 lc 1
set style line 11  lw LW lt 4 lc 5
set style line 12  lw LW lt 4 lc 3

plot\
 FileName using 1:2 w l ls 1 title "Both",
 FileName using 1:3 w l ls 2 title "Neither",
 FileName using 1:4 w l ls 3 title "Only_P2",
 FileName using 1:5 w l ls 4 title "Only_P1",
 FileName using 1:6 w l ls 5 title "Both",
 FileName using 1:7 w l ls 6 title "Neither",
 FileName using 1:8 w l ls 7 title "Only_P2",
 FileName using 1:9 w l ls 8 title "Only_P1",
 FileName using 1:10 w l ls 9 title "Both",
 FileName using 1:11 w l ls 10 title "Neither",
 FileName using 1:12 w l ls 11 title "Only_P2",
 FileName using 1:13 w l ls 12 title "Only_P1";
set term post eps
set output "Test1-logscale.eps"
replot
```

Appendix M

Generating Normalized Goodput Graph

```
set title "Total Avg TP Normalized to P1 - Data Rate Vs Goodput"
set xlabel "Datarate (Mbps)"
set ylabel "Goodput in %"

set ytics 20
#set grid mytics

set grid nopolar
set grid xtics nomxtics ytics nomytics noztics nomztics
set grid nox2tics nomx2tics noy2tics nomy2tics nocbtics nomcbtics
set grid layerdefault

#labels at Top left
#set label "Carrier Sensing Legend - P1/P2 : 0/1 = On/Off" at graph 0.01, 0.96
#set label "Style Legend - Solid/Dashed/Dotted : 20/10/1 dBm" at graph 0.01, 0.92

#labels at Bottom Right
set label "Carrier Sensing Legend - P1/P2 : 0/1 = On/Off" at graph 0.99, 0.07 right
set label "Style Legend - Solid/Dashed/Dotted : 20/10/1 dBm" at graph 0.99, 0.03 right

set samples 16
#set key outside right top
set key left top

set xtics ("1"1,"6"2,"11"3,"12"4,"36"5,"54"6);
set yrange [0:200]


LW=3
FileName="Final-Test1-CumAvg-Normalized-P1.txt"

set style line 1  lw LW lt 1 lc 2
set style line 2  lw LW lt 1 lc 1
set style line 3  lw LW lt 1 lc 5
set style line 4  lw LW lt 1 lc 3

set style line 5  lw LW lt 2 lc 2
```

```
set style line 6  lw LW lt 2 lc 1
set style line 7  lw LW lt 2 lc 5
set style line 8  lw LW lt 2 lc 3

set style line 9  lw LW lt 4 lc 2
set style line 10  lw LW lt 4 lc 1
set style line 11  lw LW lt 4 lc 5
set style line 12  lw LW lt 4 lc 3

plot\
 FileName using 1:2 w l ls 1 title "Both",
FileName using 1:3 w l ls 2 title "Neither",
FileName using 1:4 w l ls 3 title "Only_P2",
FileName using 1:5 w l ls 4 title "Only_P1", F
ileName using 1:6 w l ls 5 title "Both",
FileName using 1:7 w l ls 6 title "Neither",
FileName using 1:8 w l ls 7 title "Only_P2",
FileName using 1:9 w l ls 8 title "Only_P1",
FileName using 1:10 w l ls 9 title "Both",
FileName using 1:11 w l ls 10 title "Neither",
FileName using 1:12 w l ls 11 title "Only_P2",
FileName using 1:13 w l ls 12 title "Only_P1";
set term post eps
set output "Test1-normalized-p1.eps"
replot
```

Appendix N

Generating Fractional Goodput Graph

```
set title "Total Avg TP Fraction of P1 - Data Rate Vs Goodput"
set xlabel "Datarate (Mbps)"
set ylabel "Goodput in %"

set ytics 20
set grid nopolar
set grid xtics nomxtics ytics nomytics noztics nomztics
set grid nox2tics nomx2tics noy2tics nomy2tics nocbtics nomcbtics
set grid layerdefault

#labels at Top left
set label "Carrier Sensing Legend - P1/P2 : 0/1 = On/Off" at graph 0.01, 0.96
set label "Style Legend - Solid/Dashed/Dotted : 20/10/1 dBm" at graph 0.01, 0.92


set samples 16
set key outside right top

set xtics ("1"1,"6"2,"11"3,"12"4,"36"5,"54"6);
set yrange [0:100]


LW=3
FileName="Final-Test1-CumAvg-Fractional-P1.txt"

set style line 1  lw LW lt 1 lc 2
set style line 2  lw LW lt 1 lc 1
set style line 3  lw LW lt 1 lc 5
set style line 4  lw LW lt 1 lc 3

set style line 5  lw LW lt 2 lc 2
set style line 6  lw LW lt 2 lc 1
set style line 7  lw LW lt 2 lc 5
set style line 8  lw LW lt 2 lc 3

set style line 9  lw LW lt 4 lc 2
set style line 10  lw LW lt 4 lc 1
```

```
set style line 11  lw LW lt 4 lc 5
set style line 12  lw LW lt 4 lc 3

plot\
 FileName using 1:2 w l ls 1 title "Both",
FileName using 1:3 w l ls 2 title "Neither",
FileName using 1:4 w l ls 3 title "Only_P2",
FileName using 1:5 w l ls 4 title "Only_P1",
FileName using 1:6 w l ls 5 title "Both",
FileName using 1:7 w l ls 6 title "Neither",
FileName using 1:8 w l ls 7 title "Only_P2",
FileName using 1:9 w l ls 8 title "Only_P1",
FileName using 1:10 w l ls 9 title "Both",
FileName using 1:11 w l ls 10 title "Neither",
FileName using 1:12 w l ls 11 title "Only_P2",
FileName using 1:13 w l ls 12 title "Only_P1";
set term post eps
set output "Test1-fractional-p1.eps"
replot
```

Appendix O

Raw Data for Test1a

Table O.1: Test1a: Average Cumulative Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.8	0.9	0.8	0.9	0.9	0.9	0.9	0.8	0.9	1.5	0.9	0.8
6		5.5	4.5	5.0	5.2	5.4	4.8	5.1	5.1	5.3	5.1	5.1	5.2
11		8.4	7.1	8.1	8.0	8.2	6.5	7.9	7.9	8.2	6.7	7.9	8.2
12		9.9	7.8	8.2	9.7	9.9	7.9	9.8	9.6	10.1	7.1	9.2	10.1
36		23.8	18.7	24.2	24.6	24.0	17.9	25.1	24.2	23.9	15.9	24.9	25.7
54		32.3	21.3	34.3	33.0	31.2	26.8	33.2	33.2	31.9	17.0	33.7	34.2

Table O.2: Test1a: Fraction of P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.05	0.08	0.1	0	0.05	0.07	0.1	0.01	0.05	0.05	0.1	0
6		0.05	0.07	0.1	0	0.05	0.07	0.1	0	0.05	0.06	0.1	0
11		0.05	0.07	0.1	0	0.05	0.08	0.1	0	0.05	0.07	0.1	0
12		0.05	0.04	0.09	0	0.05	0.08	0.1	0	0.05	0.06	0.1	0
36		0.05	0.04	0.1	0	0.05	0.08	0.1	0	0.05	0.07	0.1	0
54		0.05	0.06	0.1	0	0.03	0.05	0.1	0	0.05	0.07	0.1	0

Table O.3: Test1a: Normalized to P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.13	0.13	0.12	0.12	0.12	0.12	0.12	0.12	0.13	0.21	0.13	0.11
6		0.13	0.11	0.12	0.13	0.14	0.12	0.13	0.13	0.14	0.12	0.13	0.12
11		0.13	0.11	0.13	0.12	0.14	0.11	0.13	0.13	0.14	0.1	0.13	0.12
12		0.13	0.1	0.11	0.12	0.14	0.1	0.14	0.13	0.14	0.09	0.12	0.12
36		0.13	0.1	0.13	0.13	0.13	0.09	0.14	0.13	0.13	0.08	0.13	0.13
54		0.13	0.08	0.14	0.13	0.13	0.1	0.14	0.12	0.13	0.06	0.13	0.12

Appendix P

Raw Data for Test2a

Table P.1: Test2a: Average Cumulative Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		1.23	1.75	1.29	1.35	1.34	1.67	1.24	1.17	1.23	1.73	1.14	1.09
6		9.62	9.82	8.35	7.95	9.36	9.86	9.81	9.48	9.6	10.41	9.93	10.06
11		14.52	16.01	15.41	15.05	14.04	15.64	15.06	14.64	13.88	16.19	14.98	15.05
12		14.61	15.83	18.57	17.57	16.52	19.05	18.09	17.38	17.25	20.2	18.38	18.04
36		40.63	46.86	43.03	41.99	29.88	46.27	41.36	31.56	30.29	48	34.02	31.1
54		32.2	60.27	50.42	37.2	32.86	62.53	45.65	25.38	32.85	66.96	36.26	35.46

Table P.2: Test2a: Fraction of P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.04	0.05	0.06	0.04	0.05	0.05	0.07	0.04	0.04	0.05	0.08	0.03
6		0.05	0.05	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
11		0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
12		0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.04
36		0.05	0.05	0.06	0.05	0.03	0.05	0.06	0.03	0.04	0.05	0.07	0.02
54		0.04	0.05	0.06	0.03	0.04	0.05	0.07	0.01	0.04	0.05	0.09	0.01

Table P.3: Test2a: Normalized to P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.2	0.24	0.21	0.18	0.18	0.24	0.17	0.17	0.17	0.24	0.16	0.15
6		0.23	0.24	0.2	0.19	0.23	0.25	0.25	0.24	0.24	0.25	0.25	0.24
11		0.23	0.24	0.25	0.23	0.24	0.25	0.26	0.24	0.23	0.25	0.25	0.23
12		0.19	0.2	0.24	0.23	0.23	0.25	0.25	0.23	0.23	0.24	0.25	0.22
36		0.22	0.24	0.23	0.22	0.17	0.24	0.23	0.17	0.16	0.24	0.18	0.16
54		0.13	0.23	0.2	0.14	0.14	0.23	0.19	0.09	0.13	0.24	0.14	0.13

Appendix Q

Raw Data for Test3a

Table Q.1: Test3a: Average Cumulative Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.68	0.53	0.91	0.37	0.75	0.36	0.84	0.43	0.73	0.47	0.69	0.39
6		4.91	3.49	7.94	1.81	4.71	2.26	7.18	1.88	3.87	0.84	2.57	1.21
11		7.53	5.46	12.97	2.7	6.21	4.45	8.68	1.26	5.67	6.28	7.07	3.54
12		9.18	7.24	15.55	3.6	8.69	3.33	13.69	3.31	6.62	1.51	2.08	3.56
36		21.68	13.51	35.68	7.18	0.02	0.11	0.04	0.01	0	0	0	0
54		0.01	0	0	0.01	0	0	0	0	0	0	0	0

Table Q.2: Test3a: Fraction of P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.05	0.07	0.05	0.05	0.05	0.09	0.06	0.02	0.05	0.1	0.05	0.02
6		0.06	0.07	0.06	0.02	0.06	0.08	0.07	0.02	0.08	0.03	0.04	0.04
11		0.05	0.09	0.06	0.01	0.08	0.09	0.09	0.06	0.1	0.1	0.1	0.1
12		0.06	0.07	0.06	0.01	0.06	0.08	0.07	0.1	0.1	0.1	0.1	0.09
36		0.07	0.08	0.07	0.03	0.1	0.1	0.1	0.1	0	0	0	0
54		0.1	0.1	0.1	0.1	0	0	0	0	0	0	0	0

Table Q.3: Test3a: Normalized to P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.12	0.07	0.16	0.05	0.11	0.05	0.12	0.06	0.12	0.08	0.11	0.07
6		0.12	0.09	0.19	0.04	0.12	0.07	0.19	0.06	0.12	0.11	0.08	0.16
11		0.12	0.09	0.21	0.04	0.11	0.07	0.15	0.02	0.24	0.25	0.3	0.14
12		0.12	0.09	0.21	0.05	0.12	0.05	0.19	0.05	0.14	0.15	0.04	0.36
36		0.12	0.07	0.2	0.04	0	221.69	0	10.22	0	0	0	0
54		0	0	0	0	0	0	0	0	0	0	0	0

Appendix R

Raw Data for Test4a

Table R.1: Test4a: Average Cumulative Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.78	0.95	0.74	0.73	0.86	0.81	0.85	0.74	0.82	0.7	0.82	0.77
6		3.35	4.08	4.39	3.6	4.17	4.37	4.56	4.02	3.22	1.37	1.39	3.34
11		7.31	7.68	8	6.54	6.96	7.42	7.59	6.45	5.14	5.48	6.42	5.26
12		5.26	6.46	7.54	6.04	7.37	7.43	8.3	6.96	4.63	1.31	1.51	4.96
36		8.74	5.84	22.23	11.45	0	0.02	0.02	0	0	0	0	0
54		0	0	0	0	0	0	0	0	0	0	0	0

Table R.2: Test4a: Fraction of P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.09	0.09	0.1	0.06	0.06	0.09	0.09	0.02	0.04	0.06	0.09	0.01
6		0.09	0.09	0.1	0.08	0.09	0.1	0.1	0.09	0.09	0.07	0.07	0.09
11		0.09	0.09	0.1	0.08	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
12		0.08	0.09	0.1	0.07	0.1	0.1	0.1	0.09	0.1	0.1	0.1	0.1
36		0.07	0.06	0.1	0.03	0.1	0.1	0.1	0.1	0	0	0	0
54		0.1	0	0.1	0.1	0	0	0	0	0	0	0	0

Table R.3: Test4a: Normalized to P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.14	0.13	0.13	0.1	0.12	0.12	0.12	0.11	0.13	0.12	0.13	0.13
6		0.08	0.1	0.11	0.09	0.11	0.13	0.12	0.12	0.1	0.18	0.04	0.43
11		0.12	0.12	0.13	0.1	0.12	0.12	0.13	0.11	0.22	0.22	0.27	0.21
12		0.07	0.08	0.1	0.08	0.1	0.11	0.12	0.1	0.1	0.13	0.03	0.5
36		0.05	0.03	0.12	0.06	0	34.09	0	2.04	0	0	0	0
54		0	0	0	0	0	0	0	0	0	0	0	0

Appendix S

Raw Data for Test1b

Table S.1: Test1b: Average Cumulative Goodput

		Goodput (Mbps)											
TX Power		20dBm				10dBm				1dBm			
CS Status	P1	0	1	0	1	0	1	0	1	0	1	0	1
	P2	0	1	1	0	0	1	1	0	0	1	1	0
Rate (Mbps)	CASE	1	2	3	4	5	6	7	8	9	10	11	12
1		0.76	0.8	0.76	0.91	0.87	0.77	0.87	0.87	0.76	0.69	0.76	0.86
6		5.18	4.06	5.15	5.13	5.15	3.44	5.07	5.02	4.51	4.41	3.98	5.13
11		7.85	8.01	8.28	8.11	7.6	6.62	7.98	7.9	5.54	6.32	6.23	7.69
12		9.44	5.84	8.71	6.7	9.27	7.92	9.97	9.81	7.79	6.64	8	10.13
36		24.23	17.78	25.59	25.12	24.04	15.65	25.28	24.69	21.65	9.31	22.23	25.84
54		32.33	28.09	34.83	33.46	31.86	24.37	33.96	34.14	27.11	13.5	31.04	34.73

Table S.2: Test1b: Fraction of P1 Goodput

		Goodput (Mbps)											
TX Power		20dBm				10dBm				1dBm			
CS Status	P1	0	1	0	1	0	1	0	1	0	1	0	1
	P2	0	1	1	0	0	1	1	0	0	1	1	0
Rate (Mbps)	CASE	1	2	3	4	5	6	7	8	9	10	11	12
1		0.05	0.02	0.1	0.01	0.05	0.02	0.1	0	0.05	0.01	0.1	0
6		0.05	0.05	0.1	0	0.05	0.03	0.1	0	0.04	0.01	0.09	0
11		0.05	0.05	0.1	0	0.05	0.06	0.1	0	0.03	0.04	0.09	0
12		0.05	0.05	0.1	0.03	0.05	0.05	0.1	0	0.04	0.01	0.09	0
36		0.05	0.05	0.1	0	0.05	0.06	0.1	0	0.04	0.06	0.1	0
54		0.05	0.08	0.1	0	0.05	0.07	0.1	0	0.04	0.05	0.1	0

Table S.3: Test1b: Normalized to P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.13	0.11	0.12	0.12	0.12	0.11	0.12	0.12	0.11	0.1	0.11	0.12
6		0.13	0.1	0.13	0.13	0.13	0.09	0.13	0.12	0.11	0.1	0.1	0.12
11		0.13	0.12	0.13	0.12	0.13	0.11	0.14	0.13	0.09	0.1	0.11	0.12
12		0.12	0.07	0.11	0.09	0.13	0.1	0.14	0.13	0.1	0.08	0.11	0.12
36		0.13	0.09	0.14	0.13	0.13	0.08	0.14	0.13	0.12	0.05	0.12	0.13
54		0.13	0.11	0.14	0.13	0.13	0.09	0.14	0.13	0.11	0.05	0.12	0.13

Appendix T

Raw Data for Test2b

Table T.1: Test2b: Average Cumulative Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		1.09	1.71	0.97	1.32	1.31	1.57	1.23	1.22	1.26	1.72	1.2	1.1
6		9.68	10.09	8.39	9.37	9.52	10	9.88	9.58	9.68	10.39	9.89	9.99
11		14.73	15.62	15.25	15.3	14.2	15.52	15.13	14.65	14.16	16.25	15.15	15.02
12		17.75	19.26	19.14	18.02	16.91	19.12	18.29	17.79	17.91	20.21	18.61	18.9
36		42.59	46.12	43.75	32.36	30.24	46.58	33.75	28.83	35.73	49.16	41.92	31.95
54		31.76	59.61	45.45	37.75	33.17	64.34	37.6	35.98	40.89	66.82	52.47	39.68

Table T.2: Test2b: Fraction of P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.06	0.05	0.08	0.03	0.05	0.05	0.07	0.03	0.04	0.05	0.07	0.02
6		0.05	0.05	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
11		0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
12		0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
36		0.05	0.05	0.05	0.03	0.04	0.05	0.07	0.02	0.05	0.05	0.06	0.02
54		0.04	0.05	0.08	0.02	0.04	0.05	0.09	0.01	0.04	0.05	0.07	0.01

Table T.3: Test2b: Normalized to P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.18	0.23	0.16	0.18	0.18	0.23	0.17	0.17	0.18	0.24	0.17	0.15
6		0.24	0.25	0.2	0.23	0.24	0.25	0.25	0.24	0.25	0.25	0.25	0.24
11		0.24	0.24	0.24	0.23	0.24	0.25	0.26	0.24	0.24	0.25	0.26	0.23
12		0.23	0.25	0.25	0.23	0.23	0.25	0.25	0.23	0.24	0.24	0.25	0.23
36		0.23	0.24	0.24	0.17	0.17	0.25	0.19	0.15	0.19	0.25	0.22	0.16
54		0.13	0.23	0.18	0.14	0.14	0.24	0.16	0.13	0.16	0.24	0.21	0.14

Appendix U

Raw Data for Test3b

Table U.1: Test3b: Average Cumulative Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.6	0.53	0.57	0.4	0.64	0.36	0.57	0.52	0.32	0.39	0.4	0.48
6		1.25	3.49	4.09	3.14	3	3.36	3.88	3.03	3.16	0.82	1.64	1.29
11		3.43	3.85	6.55	2.58	4.21	3.23	5.11	6.04	1.94	4.35	4.41	2.63
12		6.12	6.3	5.39	1.99	2.25	4.9	6	3.43	5.69	1.16	2.39	0.78
36		4.36	10.34	10.05	13.27	0	0	0	0	0	0	0	0
54		0	0	0.01	0.01	0	0	0	0	0	0	0	0

Table U.2: Test3b: Fraction of P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0	0.07	0.06	0.04	0.05	0	0.05	0	0.02	0.1	0.08	0.02
6		0.09	0.01	0.03	0.04	0.04	0.06	0.08	0.03	0.01	0.02	0.04	0.04
11		0.09	0.06	0.08	0.03	0.06	0.02	0.05	0	0.04	0.01	0.07	0.02
12		0.05	0.09	0.05	0.03	0.03	0.04	0.06	0.01	0.02	0.02	0.01	0
36		0.05	0.04	0.07	0.05	0	0	0	0	0	0	0	0
54		0	0	0	0	0	0	0	0	0	0	0	0

Table U.3: Test3b: Normalized to P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.11	0.07	0.1	0.06	0.09	0.05	0.08	0.07	0.05	0.06	0.06	0.08
6		0.03	0.09	0.1	0.08	0.08	0.1	0.1	0.09	0.1	0.11	0.05	0.17
11		0.06	0.06	0.11	0.04	0.07	0.05	0.09	0.1	0.08	0.17	0.19	0.1
12		0.08	0.08	0.07	0.03	0.03	0.07	0.08	0.05	0.12	0.12	0.05	0.08
36		0.02	0.05	0.06	0.07	0	0	0	0	0	0	0	0
54		0	0	0	0	0	0	0	0	0	0	0	0

Appendix V

Raw Data for Test4b

Table V.1: Test4b: Average Cumulative Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.65	0.67	0.63	0.85	0.76	0.63	0.87	0.71	0.8	0.58	0.78	0.76
6		3.3	4.01	3.39	3.69	3.94	3.85	4.98	3.34	4.5	1.27	4.43	1.42
11		5.48	3.86	9.29	2.63	1.66	1.66	2.21	2.82	1.21	2.62	2.71	2.12
12		5	5.17	5.37	5.69	2.68	1.98	3.08	2.14	3.95	1.43	4.29	1.69
36		17.02	7.22	19.23	17.43	0	0	0	0	0	0	0	0
54		0.04	0.01	0.03	0.1	0	0	0	0	0	0	0	0

Table V.2: Test4b: Fraction of P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.04	0.02	0.07	0	0.02	0.04	0.04	0.01	0.03	0.01	0.05	0.01
6		0.01	0	0.02	0.01	0	0.02	0.02	0.01	0	0.02	0	0.02
11		0.02	0.01	0.05	0.04	0.05	0.03	0.04	0.02	0.03	0.01	0.02	0.01
12		0.02	0	0.03	0.01	0.04	0.04	0.05	0.02	0.01	0.03	0.01	0.03
36		0.05	0.05	0.09	0.01	0	0	0	0	0	0	0	0
54		0	0	0	0	0	0	0	0	0	0	0	0

Table V.3: Test4b: Normalized to P1 Goodput

		Goodput (Mbps)											
TX Power		**20dBm**				**10dBm**				**1dBm**			
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		0.12	0.09	0.11	0.12	0.11	0.09	0.12	0.1	0.13	0.1	0.13	0.13
6		0.08	0.1	0.08	0.09	0.1	0.11	0.13	0.1	0.14	0.16	0.13	0.18
11		0.09	0.06	0.15	0.04	0.03	0.03	0.04	0.05	0.05	0.1	0.11	0.08
12		0.07	0.07	0.07	0.07	0.04	0.03	0.04	0.03	0.08	0.14	0.09	0.17
36		0.09	0.04	0.11	0.09	0	0.44	0	1.51	0	0	0	0
54		0	0	0	0	0	0	0	0	0	0	0	0

Appendix W

Raw Data for Test1

Table W.1: Test1: Average Cumulative Goodput

		Goodput (Mbps)					
		20dBm		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		3.65	0.98	4.31	0.87	4.19	0.89
6		13.71	5.4	13.57	5.31	13.27	5.63
11		17.33	8.53	16.99	8.35	16.39	8.77
12		19.35	9.39	19.21	10.08	18.97	10.76
36		27.26	25.14	27.3	25.11	27.04	26.34
54		32.79	34.82	32.31	35.79	32.77	36.08

Table W.2: Test1: Fraction of P1 Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.02	0.09	0.02	0.1	0.02	0.1
6		0.03	0.09	0.02	0.1	0.02	0.09
11		0.03	0.09	0.03	0.09	0.03	0.09
12		0.03	0.09	0.03	0.1	0.03	0.09
36		0.05	0.1	0.05	0.1	0.05	0.1
54		0.06	0.1	0.05	0.1	0.05	0.1

Table W.3: Test1: Normalized to P1 Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.6	0.13	0.59	0.12	0.6	0.12
6		0.33	0.13	0.34	0.13	0.34	0.13
11		0.28	0.13	0.29	0.14	0.28	0.13
12		0.25	0.12	0.26	0.13	0.26	0.13
36		0.15	0.13	0.15	0.13	0.14	0.13
54		0.13	0.13	0.13	0.13	0.13	0.13

Appendix X

Raw Data for Test2

Table X.1: Test2: Average Cumulative Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		5.59	16.75	2.17	3.7	2.19	4.16
6		31.62	23.71	28.49	21.89	31.42	32.54
11		23.23	22.97	20.26	19.32	25.19	25.86
12		34.72	30.95	26.55	27.58	34.37	35.53
36		27.03	28.18	28.79	32.22	36.35	28.44
54		31.45	40.61	34.06	36.17	43.41	35.71

Table X.2: Test2: Fraction of P1 Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.01	0.01	0.04	0.02	0.04	0.02
6		0.01	0.02	0.01	0.02	0.01	0.02
11		0.03	0.03	0.03	0.04	0.03	0.03
12		0.02	0.03	0.02	0.03	0.02	0.03
36		0.03	0.08	0.05	0.07	0.05	0.09
54		0.05	0.08	0.07	0.09	0.06	0.1

Table X.3: Test2: Normalized to P1 Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.92	2.27	0.3	0.53	0.31	0.58
6		0.77	0.58	0.71	0.54	0.8	0.77
11		0.37	0.35	0.34	0.31	0.43	0.39
12		0.46	0.4	0.36	0.36	0.46	0.43
36		0.15	0.15	0.16	0.17	0.19	0.14
54		0.12	0.15	0.14	0.13	0.17	0.13

Appendix Y

Raw Data for Test3

Table Y.1: Test3: Average Cumulative Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		2.69	0.99	3.04	1.1	2.8	0.73
6		11.1	5.08	10.24	4.55	10.4	4.57
11		13.88	7.91	13.41	7.71	13.15	6.01
12		15.38	9.56	15.21	9.39	15.02	4.28
36		24.12	24.43	20.35	6.26	22.07	22.94
54		20.81	5.66	21.99	21.42	22.27	22.22

Table Y.2: Test3: Fraction of P1 Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.02	0.06	0.02	0.05	0.02	0.08
6		0.03	0.1	0.03	0.1	0.02	0.02
11		0.03	0.1	0.03	0.09	0.02	0.07
12		0.03	0.1	0.03	0.09	0.02	0.03
36		0.05	0.1	0	0	0	0
54		0	0	0	0	0	0

Table Y.3: Test3: Normalized to P1 Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.48	0.14	0.43	0.16	0.45	0.12
6		0.27	0.13	0.27	0.13	0.32	0.59
11		0.22	0.12	0.23	0.13	0.56	0.24
12		0.21	0.12	0.21	0.14	0.31	0.43
36		0.13	0.13	0	0	0	0
54		0	0	0	0	0	0

Appendix Z

Raw Data for Test4

Table Z.1: Test4: Average Cumulative Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.93	1.29	4.61	2.91	6.56	6.68
6		6.98	5.63	20.37	19.88	21.78	21.88
11		7.69	8.23	20.24	18.01	20.19	19.04
12		8.65	9.18	19.39	18.84	22.25	22.43
36		8.92	21.64	17.95	16.32	22.39	22.96
54		13.04	6.27	21.9	21.18	22.43	21.92

Table Z.2: Test4: Fraction of P1 Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.08	0.07	0.01	0.02	0.01	0.01
6		0.03	0.08	0	0	0	0
11		0.1	0.1	0	0	0	0
12		0.07	0.09	0	0	0	0
36		0.04	0.1	0	0	0	0
54		0	0	0	0	0	0

Table Z.3: Test4: Normalized to P1 Goodput

		Goodput (Mbps)					
TX Power		**20dBm**		**10dBm**		**1dBm**	
CS Status	**P1**	**0**	**1**	**0**	**1**	**0**	**1**
	P2	**0**	**1**	**0**	**1**	**0**	**1**
Rate (Mbps)	**CASE**	**1**	**2**	**3**	**4**	**5**	**6**
1		0.16	0.18	0.66	0.42	1.05	1.12
6		0.17	0.14	0.53	0.58	0.66	2.84
11		0.12	0.13	0.34	0.3	0.86	0.75
12		0.12	0.12	0.27	0.27	0.46	2.25
36		0.05	0.11	0	0	0	0
54		0	0	0	0	0	0