

WiFiNetMon: Interference Measurement in Long Distance WiFi Mesh Networks

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Master of Technology

by
Akhilesh Bhadauria



to the
Department of Computer Science & Engineering

Indian Institute of Technology, Kanpur

June 2007

Abstract

Over the past few years long distance WiFi links have emerged as a viable option to provide connectivity to rural regions. Its cost effectiveness compared to wired networks has been particularly attractive. However, networks consisting of such links has been a topic of research only recently. As a result, although several long distance WiFi deployments exist, little has been done in the area of network monitoring and management for such networks. WiFiNetMon is an initiative taken at IIT Kanpur to deal with this dimension of long distance WiFi links. This thesis forms an integral part of this initiative.

In this thesis we focus on measurement of interference experienced by the links of a long distance WiFi network. We measure (a) the link signal strength for each link in the network, (b) the interference experienced by each link in the operational network, (c) the interference experienced by a link if all other links were running on the same channel (same channel interference), and (d) the interference experienced by a link if all other links were running on a non-overlapping adjacent channel (adjacent channel interference).

We handle the problem of the *interference range* of a node being larger than its *transmission range* by clever use of transmit power and transmit rate during measurements. The tool developed as a part of the thesis is able to deal with temporary unavailability of some links of the network. It performs the above measurements with minimal overheads, rendering the parts of the network non-functional for less than 1 second per experiment performed. Our solution incurs an estimated bandwidth overhead for measuring interference of only about 3% of the available throughput, for the Ashwini network, which is a deployment of 21 nodes in West Godavari district, Andhra Pradesh, India. The measurements obtained through the tool are intended to act as primary information to study the health of a network and also form the source of information for a network management tool.

Acknowledgments

I would like to thank Dr. Bhaskaran Raman for his guidance and support during this thesis work. I am grateful to him for the encouragement and new ideas I received from him during different phases of this thesis. I will always remember him for providing a friendly and fun filled environment at work.

I would like to thank my lab mates Arkaprava, Dattatraya, Dheeraj, Hari, Naveen, Jugash and Zahir for tremendous environment in the lab. I am indebted to my lab mates for the technical help and discussions I had with them. It was not possible to spend complete day at lab without the fun I had with my lab mates. The frequent coffee breaks with brownie parties are important contributors to this thesis work.

I would like to thank IIT Kanpur for the wonderful environment it provides to the students. I will always remember this place for wonderful residents of IIT Kanpur. I must take this opportunity to thank my batch mates at CSE Dept. I will never forget Y5 Mtech CSE, for the wonderful people I stayed with for two years. I would specially like to thank Deepak, Puneet, Rahul, Ravi, Renjith and Sumit for their support, they have been a great company for me during the stay at IIT Kanpur.

I dedicate this thesis to my parents and family. Without their support and love this work would have not been possible. I thank them from the bottom of my heart.

Contents

1	Introduction	5
2	Problem Overview	9
2.1	Background and Motivation	9
2.1.1	Long distance WiFi networks	9
2.1.2	Point to multi point links	10
2.2	Problem Statement	12
2.3	Challenges Involved	15
3	Related Work	18
3.1	Interference Measurement	18
3.2	Wireless Network Monitoring and Management	19
4	WiFiNetMon Interference Measurement and Monitoring Architecture	21
4.1	Terminology	21
4.2	WiFiNetMon Overview	22
4.3	Overview of the Architecture	23
4.4	Collecting Link Packet Details	28
4.5	Measuring Interference in Operational Network	29
4.5.1	Challenges in collecting interference values	31
4.5.2	Possible approaches for collecting interference values	31
4.5.3	Master mode operation in WDS setting	34
4.6	Measuring Same/Adjacent Channel Interference	36
4.6.1	All at the same time approach	36
4.6.2	One at a time approach	37

4.6.3	Interference measurement at the granularity of group	40
4.7	Issues Involved in Overall Solution	40
4.7.1	Getting network topology	40
4.7.2	Inserting missing interference values	41
4.7.3	Detecting link misalignment	41
4.7.4	False link misalignment call	42
4.7.5	Identifying zero interference	42
4.8	Implementation Details	43
5	Evaluation	44
5.1	Emulation for WiFiNetMon	44
5.2	Channel Switching Overhead	46
5.3	CPU Utilization and Memory Overhead	48
5.4	Network Overhead	50
5.5	Time Taken to Create Same/Adjacent Channel Interference Matrices	51
6	Conclusion and Future Work	53
	Bibliography	55

List of Tables

3.1	Network monitoring and management in wireless networks	20
4.1	Example of trace summary information	30
5.1	Packet lost on link during channel switching	47

List of Figures

2.1	A long distance WiFi link	9
2.2	The Digital Gangetic Plains testbed	10
2.3	Example of point to multipoint setup	11
2.4	The Ashwini network topology, as of "July 2006"	11
2.5	Interference scenario	13
2.6	Spectral mask for DSSS (<i>courtesy: www.wi-fiplanet.com/tutorials/article.php/990101</i>) 14	
2.7	Spectral mask for OFDM (<i>courtesy: www.wi-fiplanet.com/tutorials/article.php/990101</i>) 15	
2.8	Example of adjacent channel interference	15
2.9	Interference ranges are longer than transmission ranges	16
4.1	WiFiNetMon overview	23
4.2	Overview of proposed system	24
4.3	Wireless distribution system between two access points	35
4.4	Network moving to channel 1 from normal operation	36
4.5	Node A moving to channel 1 from normal operation	38
4.6	Node A moving to channel 6 and 11	39
4.7	Node A transmitting at normal and maximum transmit power	41
5.1	Setup for emulation of 16 node deployment	45
5.2	Setup for channel switching experiment	47
5.3	CPU utilization at 0 millisecond inter packet gap	49
5.4	CPU utilization at 10 millisecond inter packet gap	50

Chapter 1

Introduction

Communication technology has drastically changed the world in the last few decades. In spite of this huge growth, developing regions still lag behind developed regions in availability of Internet technologies. In developing regions most of the population lives in rural areas. Proliferation of Internet technologies has been very small in these regions. The main reason for this is that the cost to profit gap of wired network deployment is too high in rural settings. This is mainly because rural areas are not densely populated as compared to urban areas. This creates a big digital divide between urban and rural areas.

Long distance WiFi networks is an initiative with the objective of bridging this digital divide. Long distance links based on 802.11 technology are used to provide wireless communication. These links can be as long as few tens of kilometers [1]. There are several network deployments around the world based on such long distance links [1, 2].

Network monitoring and management is important aspect in any operational network. This is more important in long distance WiFi networks because of rural settings of the network. There has been active research in network monitoring and management in wireless networks but most of it is for 802.11 based infrastructure networks. WiFiNetMon is the project started at IIT Kanpur for looking into network monitoring and management aspect in long distance WiFi networks. Our contribution to WiFiNetMon is in measuring the interference each link faces in the network. We have designed a system for link monitoring and interference measurement in long distance WiFi networks.

Interference among wireless links degrade performance drastically. This has been studied by several researchers. The work in [3, 4] have studied such degradation specifically in long distance wireless networks. Many studies have proposed solutions to improve the link

performance in the presence of interference [6, 12, 8]. But most of them assume that interference information is somehow available without mentioning how to measure it.

Interference in WiFi-based wireless networks can be classified as

- WiFi : this is the interference because of transmissions made from a WiFi device.

This can be further classified as

- internal : this is interference from a WiFi device belonging to the same network.
 - external : this is interference from a WiFi device that do not belong to the same network.
- non WiFi : this is the interference because of electromagnetic transmissions made from non WiFi sources like microwave ovens and cordless phones.

In this thesis work we focus on interference from WiFi sources only. The non WiFi sources are not known to be a major performance degrader. In [5], the authors observed that in presence of external non-WiFi noise, the noise floor linearly increases with increasing noise. In [4], researchers have tried to find effect of non WiFi interference on long distance links in both urban and rural settings. They have not found any significant non WiFi interference in any of the two settings. They observed that the noise floor remains mostly constant with only minor 1-2 dB variation.

Our work is aimed at measuring the interference scenario in a long distance WiFi network. The interference in the network can be decreased by using optimal transmit power and channel assignment. But deciding these optimal values require estimation of amount of interference currently present in the network. There is work on channel assignment and transmit power assignment in presence of interference [8, 9, 10, 11], but they do not talk about how to measure interference in the first place.

This is a challenging problem for several reasons. The actual modeling of interference based on path loss models is difficult because this requires considering many hardware and environment specific factors. The interference ranges are much longer than transmission ranges this makes it challenging to correctly measure interference. Interference measurement in an operational network require links to change from designated channels to other channels, this should be done efficiently so as to cause minimum disturbance to the operational

network. The interference scenario changes with change in network topology so this is not a one time task. Hence, it is important to do it efficiently.

Our architecture for interference measurement and monitoring is a client-server based model. In our architecture we run a light weight daemon on all the nodes of the network, this daemon is called *client module*. All the client modules send the interference and link related information to a central location. This central location has a *server module* running to collect the information sent by the clients. This architecture suits the resource poor client nodes of the network. In addition, this architecture gives a centralized view of the network, which is needed for controlled interference measurement experiments.

We measure interference in terms of signal strength. We use the RSSI (received signal strength indication) value the wireless card reports to the driver for each received packet. To measure interference we use packets transmitted at lowest transmit rates, this increases the possibility of detecting the interference.

In this thesis work we design, implement, and evaluate a system to :

- monitor the link signal strength for each link.
- measure the interference experienced by each link in the operational network.
- measure the interference experienced by a link if all other links were running on the same channel (same channel interference).
- measure interference experienced by a link if all other links were running on a non-overlapping adjacent channel (adjacent channel interference).

We monitor the link signal strength by looking at the RSSI value reported for link packets. This monitoring helps in detecting link misalignment and diagnosing performance anomaly in the links. We measure the interference in operational network by looking at the RSSI values reported for interfering packets during normal operation of the network. The estimation of interference is important in deciding on the optimal transmit powers for nodes in the network. We measure the same channel and adjacent channel interference by performing controlled experiments. The same channel and adjacent channel information is useful in deciding the optimal channel assignment for the network.

The rest of this thesis is organized as follows. First, we present the problem overview in Chapter 2. We then discuss the related work in WiFi network monitoring and interference

measurement in Chapter 3. Next, we present the architecture for interference measurement and monitoring in Chapter 4. Subsequently in Chapter 5 we evaluate our system. Finally, Chapter 6 discusses future work and concludes the thesis.

Chapter 2

Problem Overview

In this thesis work we have designed a system to measure interference in long distance WiFi mesh networks. In addition to measuring interference our system monitors all the links passively. We start this chapter by discussing the background and motivation in Section 2.1. Subsequently, we define the problem statement in Section 2.2. Finally, in Section 2.3 we discuss the challenges involved in this work.

2.1 Background and Motivation

2.1.1 Long distance WiFi networks

These networks comprise of long range wireless links using 802.11 technology. A typical long distance link is depicted in the following Figure 2.1. These long distance links are established using high gain directional antennas.

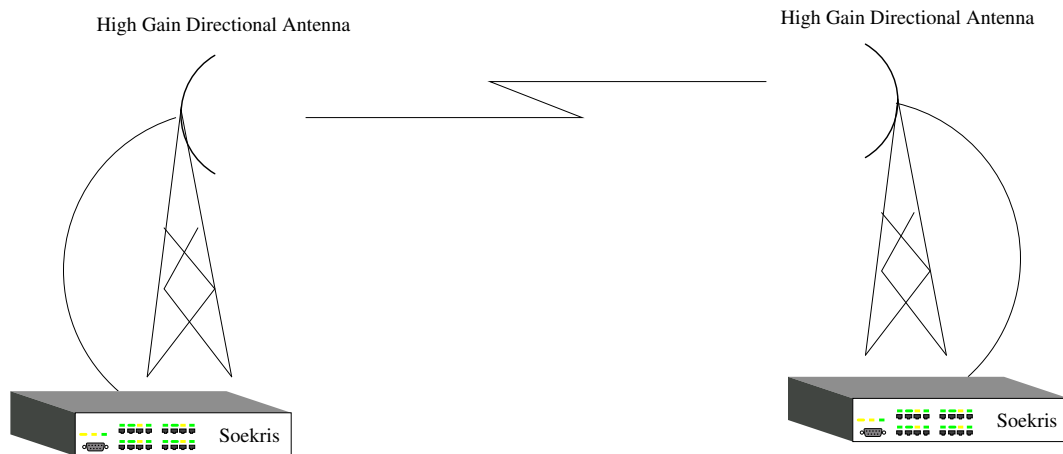


Figure 2.1: A long distance WiFi link

These links can be as long as several tens of kilometers. The idea behind long distance WiFi networks is to use such long distance wireless links to provide Internet connectivity in rural villages. In these networks there is a central location that has a wired Internet connectivity. Long distance links connects various surrounding villages to central location through a multi-hop mesh network. Figure 2.2 shows the long distance WiFi test bed at IIT Kanpur.

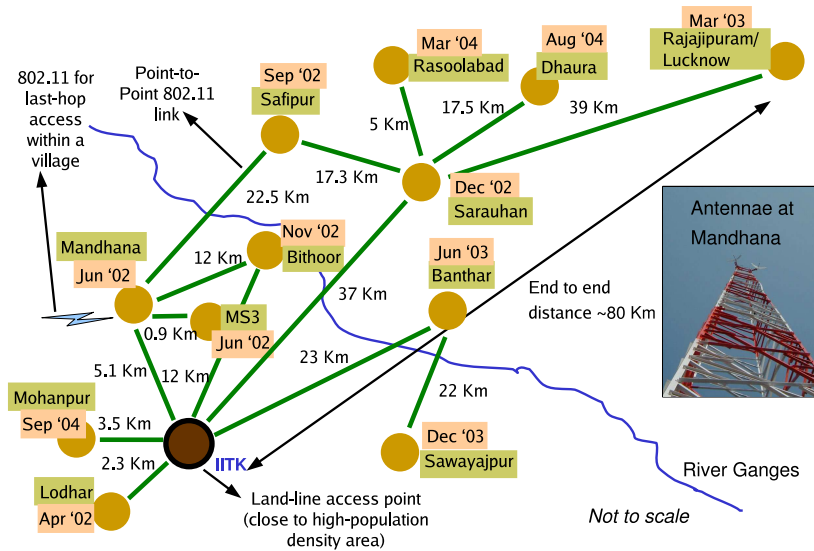


Figure 2.2: The Digital Gangetic Plains testbed

2.1.2 Point to multi point links

In addition to point to point links, point to multi point links are also used in some long distance WiFi deployments [1]. Figure 2.3 shows a point to multi point (P2MP) link. In P2MP link there is a single radio establishing link with multiple other radios. Communication with multiple radios is done in a time shared manner. The antenna used at the common point is generally a sector type with half beam width between 30 to 120 degrees.

The Figure 2.4 shows the topology of the active links in Ashwini deployment, as of Aug 2006. The Ashwini project is a network deployment effort by the Byrraju Foundation, to provide broadband access and services to a collection of villages in the West Godavari district of Andhra Pradesh, India. Currently a variety of interactive video-based applications such as distance-education, tele-medicine, etc. are being run on the network.

Network monitoring and management is an important aspect for any working network.

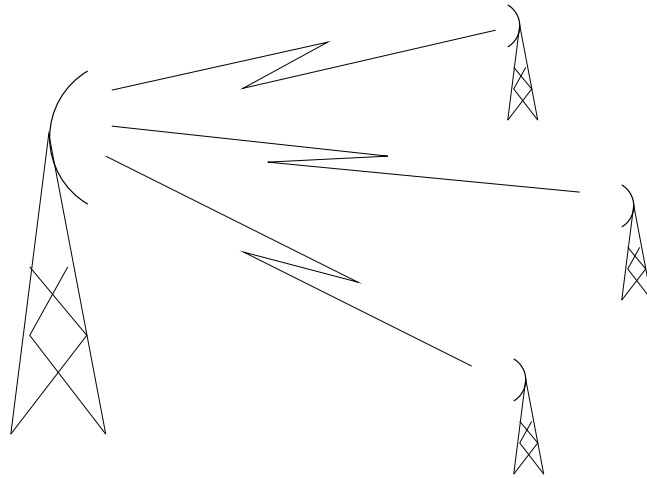


Figure 2.3: Example of point to multipoint setup

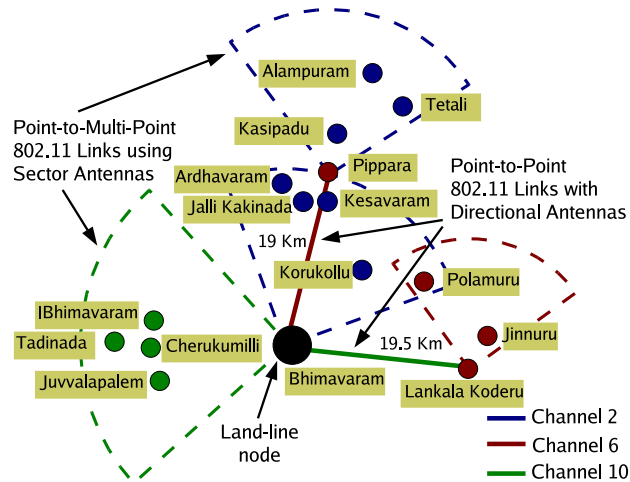


Figure 2.4: The Ashwini network topology, as of "July 2006"

Such issues are still in the research phase for enterprise WiFi deployments. Little work is done on these issues in the long distance WiFi scenario. There is insufficient understanding on fault diagnosis in the long distance WiFi setting. Monitoring and management is specially important in long distance setting for following reasons:

- Poor availability of trained personnel in rural settings.
- Site locations in rural scenarios are relatively inaccessible.
- Its costly to visit sites every time, to diagnose a fault.

WiFiNetMon is such a project aimed at building a network monitoring and management

tool for long distance WiFi network. We will be talking about WiFiNetMon in Section 4.2. In this thesis work we have worked on interference measurement aspect of WiFiNetMon.

2.2 Problem Statement

Performance degradation because of interference is well known among researchers. Many studies have built up solutions to improve the link performance in the presence of interference [6, 12, 8]. But yet very little work has been done in direction of estimating interference. The recent publications [4, 3] shows performance degradation in long distance because of interference. The impact of interference is important in these networks as many of these networks are being used for supporting real time applications [1]. Interference is more harmful in networks providing real time services as links will be in use continuously. These active links will be interfering continuously with other operational links in the vicinity.

Our motivation for measuring interference in long distance settings is to manage interference well. The estimation of interference can help in reducing the interference. Interference is dependent on the transmit power at which links are operating as well as the channel assignment which is being chosen for the network. These two parameters can be set to optimal values for managing or decreasing interference. But finding optimal settings for transmit power and channel assignment requires the estimation of amount of interference at the first place. Recent (July 2006) performance measurement experiments on the long distance network deployment at project Ashwini [1] has shown poor performance results. On investigation of wireless parameters both power assignment and channel assignment were found to be non optimal. This is further motivation for us to explore interference measurement in depth.

Power assignment

A wireless link needs a received signal strength above a threshold SINR (signal to interference noise ratio) value to deliver packets with minimal disruption. This threshold SINR value is dependent on the data rate at which the packet is being transmitted. Performance studies on long distance network [3] have validated this threshold in long distance networks.

Figure 2.5 shows three radio's of the network. There is a communication link between Node A and Node B. Node C is interfering with Node B. Consider P as the received signal

strength at B, I as the combined interference from all other nodes in the network and SIR as the threshold signal to interference ratio required. We are ignoring noise here, hence using SIR in place of SINR. For packets on link between A and B to be received properly following condition should be satisfied :

$$P - I \geq SIR_{reqd}$$

Received signal strength P is dependent on the transmit power of the radio at the other end of the link. P at Node B will depend on transmit power of the Node A. Optimal transmit power at Node A will be transmit power at which above equation is just met. If a link is operating at above optimum transmit power its interference to other other radios will be more as compared to interference it will cause at optimum transmit power.

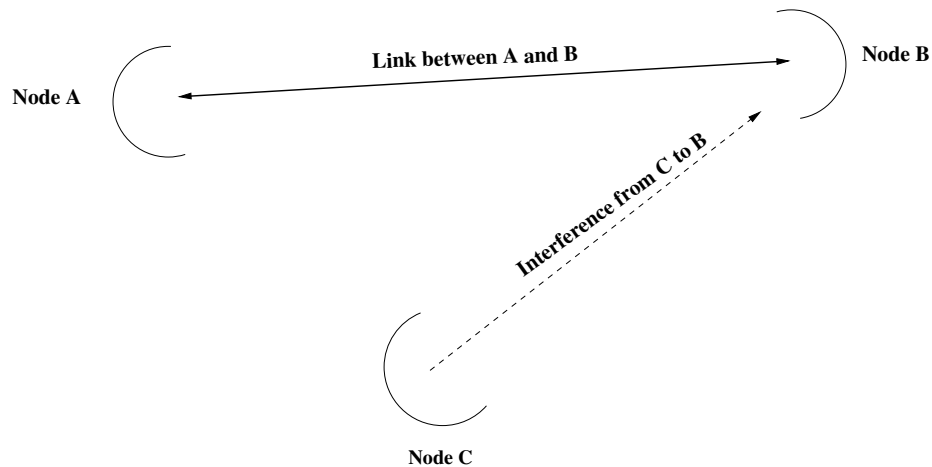


Figure 2.5: Interference scenario

As discussed above, the optimal transmit power is dependent on the total interference other links are causing. Suppose we have a network in which all links are operating at maximum transmit power. If we have knowledge of interference each link faces we can find links which are operating above optimal transmit power. By lowering transmit power for these links to an optimum value we can reduce overall interference at other links. This way overall interference scenario can be improved. Measurement of interference in an operating network can be used to devise an algorithm for power assignments which will improve overall interference scenario in the network.

Channel assignment

By channel allocation we mean to assign each link a channel (frequency) at which radios

of the link will operate. Channel allocation is close to the graph coloring problem of assigning colors. Channel allocation needs to assign channel such that links close to each other are assigned different channels, so that they interfere less. Channel allocation is essentially with an aim to assign channel, so that links face less interference. If we know the amount of interference each link faces because of other links, then we can potentially decide on a better channel assignment. There is previous work in the literature, on interference aware channel assignment [9]. But this work assumes that interference information is some how already available.

The work in [9] considers only the same channel interference values. We believe more optimal channel assignment can be obtained if adjacent channel interference is also measured. Figure 2.6 and Figure 2.7 shows the transmit spectral mask for DSSS and spread spectrum OFDM encoding specified by the IEEE 802.11b and 802.11g specification respectively. These figures have been taken from [13]. OFDM encoding has about 30 dB loss for frequency 22 MHz away from central frequency.

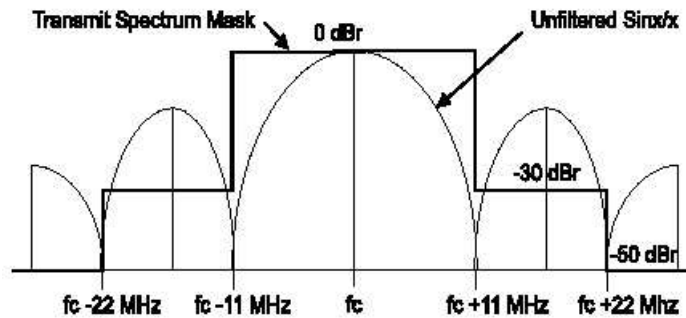


Figure 2.6: Spectral mask for DSSS (courtesy: www.wifiplanet.com/tutorials/article.php/990101)

To validate the possibility of interference from adjacent channel consider the scenario shown in Figure 2.8.

Suppose node C is transmitting at 50mW. The path loss from node C to node B will be about 113 dBm according to Frii's pathloss equation.

$$pathloss = 32.5 + 20\log_{10}(2400) + 20\log_{10}(5)$$

We assume antenna gain to be 24 dBi for both nodes B and C and that C is in the direction of maximum gain to B. The interference value at node B because of node C can be as high as 78 dBm for a 802.11g radio.

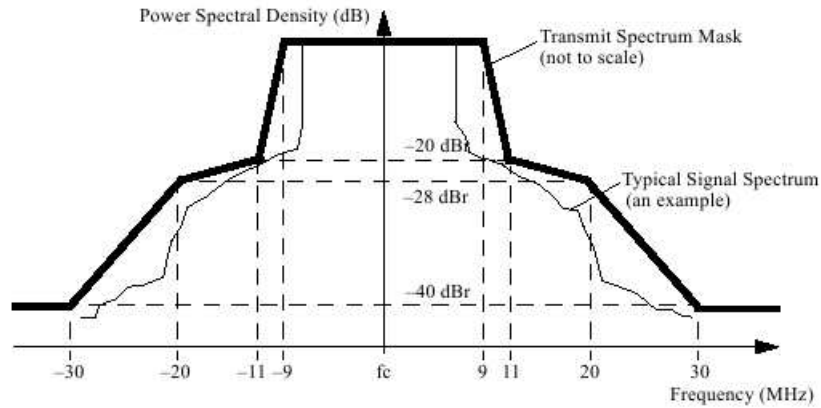


Figure 2.7: Spectral mask for OFDM (courtesy: www.wi-fiplanet.com/tutorials/article.php/990101)

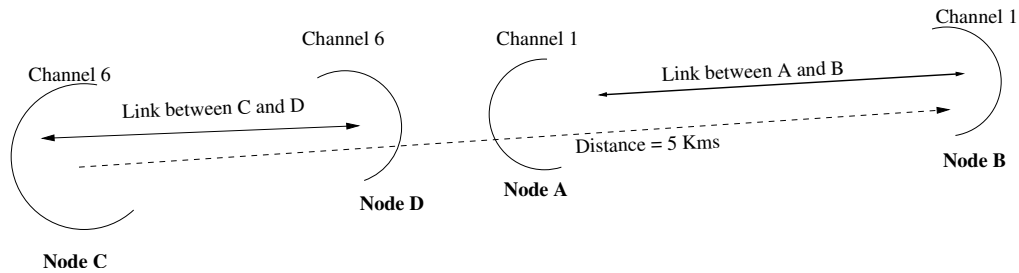


Figure 2.8: Example of adjacent channel interference

Problem statement

To do passive monitoring and to measure the interference scenario for all the links in a long distance WiFi mesh network. Metric to be used for quantifying interference will be the signal strength. Interference will be measured in terms of signal strength of the interfering signal. Network monitoring will be with respect to signal strength, noise floor on per packet basis for the packets transmitted over the link.

2.3 Challenges Involved

Most of the research related to interference management assumes that information about interference is readily available. There are several aspects to interference measurement in long distance setting that makes it challenging.

- Interference ranges are normally larger than transmission ranges. Figure 2.9 explains this scenario clearly. Node C is not in transmission range of node A so node C is

not able to receive packets from node A. Because of difference in transmission and interference ranges node C is not able to detect interference from node A. In a mesh network with large number of nodes this undetected interference summed together can lead to performance disruption.

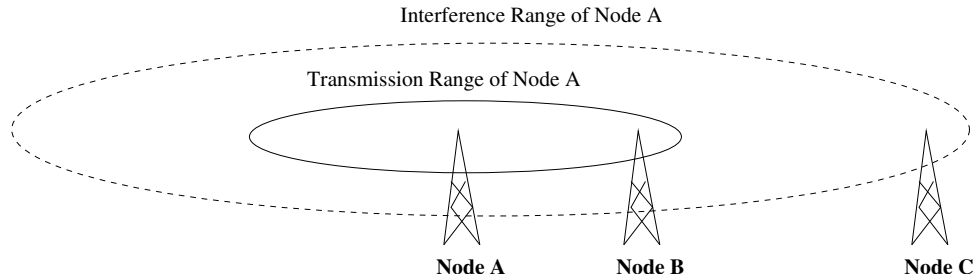


Figure 2.9: Interference ranges are longer than transmission ranges

- Interference measurement should be as passive as possible. In an operating network, interference measurement should not interfere with normal operation of the network. Any extra traffic introduced should be kept to a minimum and should not decrease the network throughput to a noticeable extent.
- Availability of trained professional in rural areas is relatively poor. Our solution of interference measurement should not require continuous input from the network administrator. It should be as automated as possible.
- Measurement of same channel interference needs all the links working on same channel. This constraint implies that measuring same-channel interference will affect the normal working of the network. In a normal operating network links will be operating in one of the three non overlapping channels. Requiring the entire network to move to same channel will essentially make the network unavailable for the duration for which the interference measurement is done.
- Long distance WiFi networks are targeted for rural connectivity. Links that are operating in rural settings have power constraints. Its quite regular for the links to go down because of non availability of power. Any assumption of whole network being up at the same time will not work in these settings. This makes measuring interference for a particular radio from all other radios of the network difficult.

- Wireless links have the possibility of link misalignment. Work done in [4] has pointed that link misalignment is quite common in long distance networks. Any link misalignment requires the interference measurements to be updated to reflect the correct interference scenario in the network. In addition to misalignment networks normally grow in phases. New links are added with time. This again implies the need for detecting the changed interference scenario.
- We want to implement interference measuring solution for real networking platform. These networking platform have low CPU speed, memory and flash space and, we have to work within these constraints in our design.

Chapter 3

Related Work

We discuss related work with respect to interference measurement in Section 3.1. Subsequently in Section 3.2 we talk about related work in network monitoring and management in WiFi based wireless networks.

3.1 Interference Measurement

Performance degradation in wireless networks because of interference is long known among the researchers. There are several works [14, 15] on the impact of interference on the performance of a wireless networks. These work consider the impact of interference on the overall capacity of the multi-hop wireless networks. They assume that the information about the interference is available somehow but they do not talk about how to measure it.

The works in [16, 3] talk about performance characterization in long distance WiFi links. They point to performance degradation because of interference specifically in long distance Wifi networks. These works also do not consider the problem of estimating the interference in long distance Wifi networks.

There are work on interference aware channel assignment [8, 9]. These works consider the interference scenario in the network while deciding on channel assignment. They assume that information about interference is some how available, without explaining how to measure it. The works in [10, 11] look into problem related to assigning transmit powers in a wireless mesh network. They estimate the interference values on the basis of Frii's path loss model. We believe that accurately modeling signal propagation is difficult as we have to consider many environment and hardware specific factors like antenna type, antenna orientation etc.

The work done in [17] is the one most closely related to our problem of interference measurement. They measure interference among links in a static, IEEE 802.11, multi-hop wireless network. They propose a methodology that can predict pairwise interference among n^2 possible links using $O(n^2)$ measurements. Their solution does not apply to our scenario due to the following reasons.

- Their solution assumes the entire network to be on the same channel. This is not applicable to our setting. We want to measure the interference scenario in an operational network also, where links might be operating in different channels.
- They propose performing $O(n^2)$ measurements to estimate interference, this will make the network non operational for the duration for which these measurements are performed. This overhead is too high for interference measurement.
- The metric they propose is in terms of link interference ratio which is, ratio of aggregate throughput of the links when they are active simultaneously, to their aggregate throughput when they active individually. We are interested in using the estimated interference values for optimal power assignment. For this, the estimation of interference in terms of the power of the interfering signal is more appropriate.

3.2 Wireless Network Monitoring and Management

There is large body of work on wireless network monitoring and management. The work in [5, 18] is for network monitoring and management in infrastructure networks. The former talks about fine-grained detection algorithms that are capable of distinguishing between root causes of wireless anomalies at the depth of the physical layer. The latter presents architecture for detecting and diagnosing faults in IEEE 802.11 infrastructure wireless networks. Their work primarily focuses on fault diagnosis of disconnected clients, detecting unauthorized access point and analyzing performance problems in the network. The work in [19] presents, a scalable framework for wireless network monitoring. Their work relies on a distributed set of agents within the network to monitor network devices and store the collected information at data repositories. Their work does not include any diagnostic feature. [20] present a diagnostic system that employs trace-driven simulations to detect faults and perform root cause analysis. WiFiNetMon is focused on network monitoring and

management in long distance WiFi networks. Table 3.1 summarizes these related work. In this table we summarize the architecture and important diagnostic features of the referred work. We have summarized our WiFiNetMon also in the last row of the table. The contribution of our work is a system for measuring and monitoring interference in long distance WiFi networks. The knowledge of interference in the system will help in better diagnosis and management of the system.

Title	Target Network	Architecture	Diagnostic feature
MOJO (Univ. of Colorado) [5]	Infrastructure networks	Distributed sniffers	Hidden terminal, capture effect, noise signal variation
VISUM (Univ. of Santa Barbara) [19]	Generic wireless networks	Distributed sniffers	None specified
Trouble shooting Wireless Mesh Networks (MSR) [20]	Mesh networks	Integrates network simulator into the management system	Packet dropping, detecting MAC misbehavior, external noise sources
Architecture Techniques for Diagnosing Faults (MSR) [18]	Infrastructure networks	Monitoring modules on clients and access points	Unauthorized access points, low throughput, high latency, connectivity problems
WiFiNetMon (IIT Kanpur)	Long distance 802.11 WiFi mesh networks	Light weight daemon on link nodes and a server module	Link misalignment, noise variation, poor SINR, packet dropping, internal and external interference

Table 3.1: Network monitoring and management in wireless networks

Chapter 4

WiFiNetMon Interference Measurement and Monitoring Architecture

In this chapter we describe the proposed architecture for interference measurement and link monitoring. We start with section 4.1 introducing some terminology that is used in this chapter. The architecture of WiFiNetMon which is a solution designed for network monitoring and management is described in Section 4.2. The complete overview of our proposed architecture for interference management and monitoring, is described in Section 4.3. The Section 4.4 talks about the issues related to collecting per packet low level details like signal and noise strength. Section 4.5 discuss possible design approaches for collecting interference information. Methodology related to measuring same channel and adjacent channel interference is described in Section 4.6. In Section 4.7 we present some issues related to overall monitoring architecture. Finally, the Section 4.8 gives the implementation details.

4.1 Terminology

In this section we introduce some terminology we have used in describing the architecture.

- **Nodes:** by node we refer to an network device with isolated radio in the network. If multiple radios are operating in same physical location we will consider them as separate nodes.
- **Link:** by link we mean the physical channel (medium) established for exchanging the information between two radios.

- Link packets: these are the packets for which the sender and receiver radio are part of some link. These include the data packets and beacon packets sent over a link.
- Interfering packets: these are the packets for which sender and receiver radio are not part of any link.
- Group: nodes belonging to same point to multi point set are considered to be part of a group.

4.2 WiFiNetMon Overview

WiFiNetMon is aimed at building a tool for diagnosing performance problems and faults in long distance WiFi Networks. The Figure 4.1 shows the objectives to be achieved by WiFiNetMon along with methods to be followed for achieving it. Problems to be addressed by WiFiNetMon are detecting performance anomalies in the operational network, to diagnose the causes of performance anomaly and to correct performance problems in the network. Support required to achieve these objectives is in terms of collecting wireless configuration and network topology information, measuring signal strength and noise level for each link, detecting interference from same and adjacent channels and ability to perform on demand experiments. In this thesis work we have looked at issues related to interference measurement and passive link monitoring. By passive monitoring we mean to monitor signal strength, noise floors on per packet basis for the packets transmitted over the link. Link monitoring is used to detect link misalignments, it also works as a trigger to update interference scenario. By interference measurement we mean measuring the following:

- **Interference in operating network:** This is the interference each radio of the network faces in a operational network. This interference may be because of WiFi radios belonging to same network or those belonging to other networks. The nodes may be operating on any of the channels available. Measurement of interference in operating network will be useful in deciding on optimal transmit power at which radios should be working. This is already discussed in Section 2.2.
- **Maximum power same channel interference:** This is the interference each radio faces from other radios operating on same channel and belonging to same network.

The interfering radios operate at the maximum possible transmit power. Same channel interference values are useful in deciding channel assignment. Interference values can work as heuristic for deciding the channels to be allocated to interfering nodes.

- **Maximum power adjacent channel interference:** This is the interference a radio faces from other radios operating on adjacent non overlapping channel. The interfering radios operate at maximum possible transmit power. As discussed in Section 2.2, adjacent channel interference values could also be high. So for deciding channel assignment this adjacent channel interference values should also be considered.

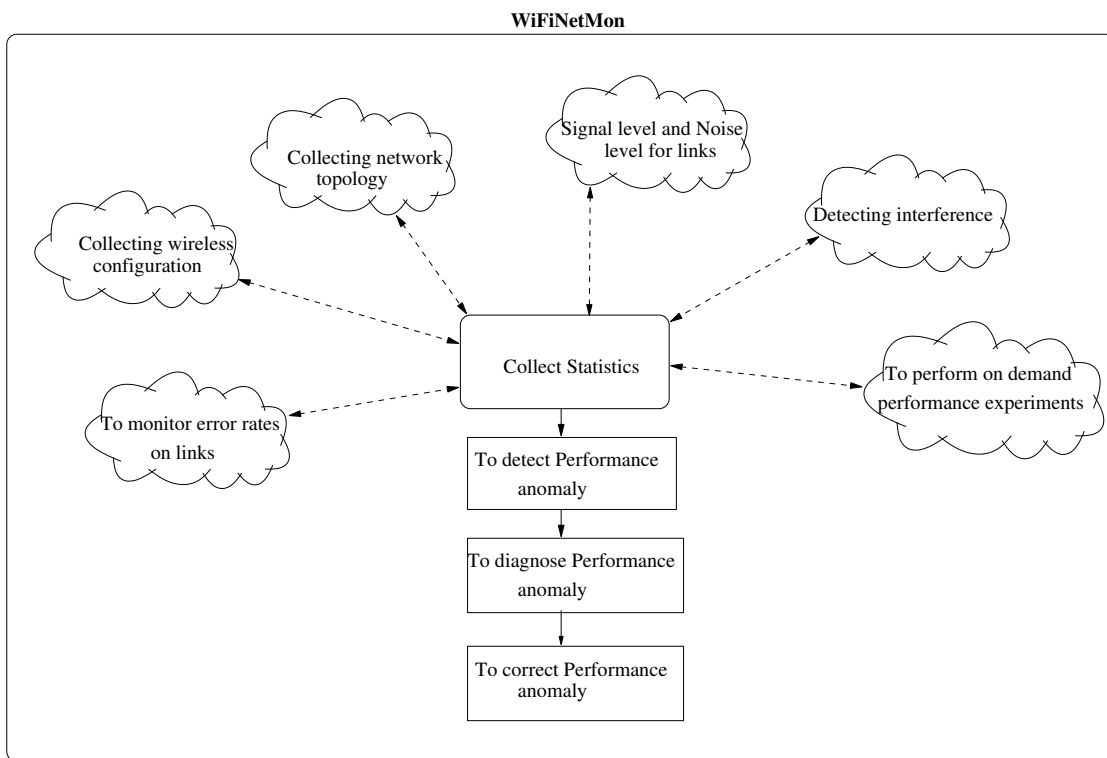


Figure 4.1: WiFiNetMon overview

4.3 Overview of the Architecture

In long distance WiFi network most of nodes are specialized networking devices. These networking devices are poor in terms of available memory and computational power. For storing the history of monitored data and interference information we need machine with greater memory available with it. In addition, for collecting the interference information we

need the centralized view of the complete network. This is needed so that the centralized location can perform controlled experiments with the entire network. The central location needs to infer interference information from experiment results, this requires the knowledge of complete network topology. With these requirements we opted for an architecture in which light weight daemon runs at all nodes of network. All nodes send the monitoring information and there network configuration to a central location periodically. The central location receives this information from all nodes from the network and constructs the network topology from it. The central location also infer the interference information on the basis of controlled experiment performed.

The Figure 4.2 gives the overview of the architecture for interference measurement and monitoring. Architecture consists of the following two modules:

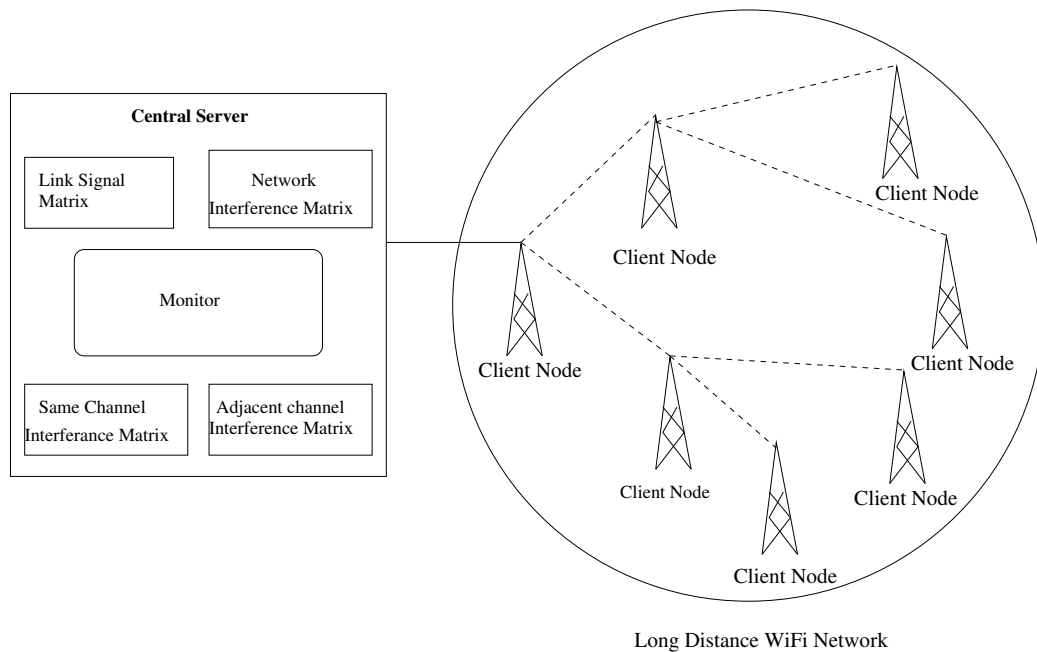


Figure 4.2: Overview of proposed system

1. Client module:

this module runs on all the client nodes of the network. We are assuming Soekris boxes being used at the network nodes. The client module monitors both the link and interfering packets. The metrics monitored by a client module are RSSI values, noise floor values, transmit rate and packets size. The RSSI values for link packets gives measure of links health and, the RSSI value for the interfering packet gives measure of

interference caused by sender radio. Noise floor value gives the indications related to presence of interference, high and varying noise floor value is indication of presence of interference [16]. Transmit rate used give indication about packet error rates. 802.11 uses transmit rate fallback in presence of packet losses. The packet size information helps in computing throughput and characterizing the user applications.

The client module segments the monitored data based on the MAC address of the received packet. It then creates summary of the monitored data for each received MAC address and, periodically sends summary of monitored data to the remote server. In addition to periodically sending this trace summary to central server, client node receives experiment requests from central server for performing experiments related to interference measurement.

2. Server module:

this module runs on a machine that have more computational and memory resources at its disposal as compared to Soekris machines. The central server periodically receive trace summary from the nodes of the network. It keeps the history of these trace summaries. This history can be used to generate a complete picture of functioning network at the central server. History can also be used to query about the links health over a period of time. In addition to keeping history of link performance central server maintains following structures.

- **Link signal matrix:** this matrix has a row corresponding to each link in the network. This matrix has RSSI value and noise floors for each link averaged over past packets. This matrix is updated each time trace summary is received.

Link signal matrix track the health of each link belonging to network. it identifies the links that need diagnosis as they have insufficient signal strength over the period of time. It also identifies the presence of interference on the basis of high and variable noise floor values.

- **Network interference matrix:** this matrix has a row for each node in the network. The entries include the nodes which are interfering with this node and with what signal strength they are interfering. It also includes total interference being seen by this node which is sum of individual signal strength values in milli

Watts.

Network interference matrix keeps record of the amount of interference each node is facing. This matrix will be useful in deciding on the optimum transmit power values as discussed in Section 2.2. In addition it gives a useful input for diagnosing performance anomaly in the links.

- **Same channel interference matrix:** this matrix shows interference values that a node sees assuming whole of the network is operating on the same channel and with maximum transmit power. Each row entry in this matrix is like entries in network interference matrix but are for nodes operating on same channel and at maximum transmit power.

This matrix is used in computing the missing interference values as discussed in Section 4.7.2. It has the interference values at maximum transmitted power which is used in calculating some of undetected interference. This matrix contains useful information to work as input for deciding channel assignment in the network, this is already discussed in Section 2.2.

- **Adjacent channel interference matrix:** this matrix shows interference observed by a node when all interfering nodes are operating on adjacent non overlapping channel to the one in which this node is operating and with maximum transmit power. For example if node A is operating on channel 1 then for measuring adjacent channel interference for node A, all other nodes belonging to the network will be operating on channel 6. Each entry in this matrix is interference other nodes operating on adjacent non overlapping channel and maximum transmit power are causing.

This matrix is used for same purpose as same channel interference matrix. The difference is that it provides interference information from adjacent channel. The need of measuring adjacent channel interference is already discussed in Section 2.2.

Assumptions

- **Signal strength stability over time:** we assume that the RSSI values are stable over time. Stability of RSSI value was reported in prior measurement study on long

distance links [3]. In addition to this performance study, RSSI stability both over short and long durations has been reported in [16]. Both these studies point that RSSI variation is within 0-4 dBm band. This assumption is used in detecting link misalignments. Any variation beyond a decided threshold value is attributed to some parameter reset or link misalignment and link is reported for diagnosis.

- **Interference measurement:** interference ranges are normally longer than the transmission ranges (typically more than double). This makes it difficult to detect interference. Receiving a packet successfully requires minimum SINR threshold requirement to be met. This SINR threshold is variable depending on data rate at which the packet is being transmitted. Possibility of receiving a packet is better at lower transmit rates than at higher transmit rates. The work in [3] has measured this threshold for available data rates in links using 802.11b compatible radios. Threshold SINR required at 1 Mbps transmit rate to receive a packet successfully has been reported to be about 5dB. This value for data transmit rate of 11 Mbps has been reported to be about 12 dB. The chances of detecting the interfering nodes increases if nodes are transmitting at maximum possible transmit power. So we assume that we will be detecting all the interference if interfering nodes are transmitting at maximum transmit power, at same channel and at lowest possible data rate.
- **Channel of operation:** the 802.11 standard specify the multiple channels at which 802.11 compatible radio can operate. For example 802.11b specifies 11 channels between 2.412 GHz to 2.462 GHz. Out of 11 possible channels three channels namely channel 1 (2.412 GHz), channel 6 (2.437 GHz) and channel 11 (2.462 GHz) are considered to be non overlapping. We also assume three non overlapping channels for nodes operating in the network. All the nodes of the network are assumed to be operating on one of these three non overlapping channels.
- **TDMA MAC:** we assume that radios operating in the network are working with TDMA MAC. This assumption is important for point to multipoint links. Nodes belonging to same point to multipoint group operate in a time shared manner. Therefore a node belonging to some point to multipoint set is assumed not to be interfering with other nodes belonging to the same set.

The client modules in the described architecture collect the low level per packet details for link packets and, send the summary to the central server. These low level details then helps in creating link signal matrix, which is introduced earlier in this section. The low level per packet details for link packets received by the radio, are reported to driver by the hardware. Currently these low level packet details for 802.11 packets are not passed to the user space from the driver. Collecting these details at user space require a mechanism to pass these details from kernel space to user space. We discuss issues related to this in Section 4.4.

To create the network interference matrix introduced earlier in this section, we need low level packet details for interfering packets. The design discussed in Section 4.4 gives solution for getting low level details for link packets. This solution do not pass low level packet details for the interfering packets. In Section 4.5 we discuss possible design approaches for collecting low level packet details for interfering packets.

To measure the same and adjacent channel interference value for a node, we need interference values while entire network is on same channel or on adjacent channel with respect to this node. This requires the normal operational network to change the channel settings. The design choices for measuring the same channel and adjacent channel interference are discussed in Section 4.5.

4.4 Collecting Link Packet Details

Creating the link signal matrix introduced in Section 4.3 requires the details about received signal strength and noise floor for every link packet received. These low level packet details are exported to the driver from the hardware for each 802.11 MAC packet received. This information is available only up to the driver level, these details are not passed to the user space. Commercially available monitoring tools like tcpdump, ethereal shows MAC level details in monitor mode. These details do not include information about received signal strength and noise floor values. There are *ioctl* calls supported by drivers to report the link quality. For example Hostap driver for Prism chipset based WiFi cards support user space functions like *iwconfig* to report link quality etc but these values are less accurate. We are interested in per packet low level statistics about signal strength and noise floor.

To achieve the low level per packet information we use the driver modifications used in

the measurement study work [3]. These driver modifications are done on Hostap driver. These driver modifications create a buffer in the driver which stores the low level details on per packet basis. These details are made available into user space using /proc file system. /proc file system creates virtual files for communication between kernel and user space. Virtual file in /proc file system can be read by user space processes.

In this work we are using these Hostap modifications patched onto newer version of Hostap. For collecting the low level metrics for link packets, client node module periodically transfers the low level per packet details to a regular file. The frequency of how often this is to be done is decided by the overall aim of monitoring. If the aim is to monitor link for every packet being received then this transfer from virtual to regular file is done before the driver buffer gets filled up. We have worked with driver buffer size of 2000 entries. For collecting details about all the packets we are doing transfer every 5 seconds. With 5 seconds frequency there will not be any buffer overflow for average inter packet duration as low as 2.5 milli seconds.

If aim is just to monitor the signal and noise floors of packets seen we transfer log entries from /proc to regular file at longer intervals. This interval is decided by considering how often the central server wants update about link status.

The file created by transferring /proc to regular file is parsed periodically and a trace summary is sent to central server. While parsing, the log entries belonging to same sender MAC address are combined together. The trace summary created for central server has an entry each for different MAC addresses which the receiver radio has seen in the log file. It has the average signal value and average noise floor values for each distinguish MAC address. It also has the details about the number of packets seen from a particular MAC address. Actually the trace summary includes the entries for interfering nodes also, this is explained in Section 4.5.3. Table 4.1 shows the example trace summary in tabular form.

4.5 Measuring Interference in Operational Network

For collecting current interference values we considered using the RSSI values reported for interfering packets. Each of the node creates a trace summary of the packets it has seen since the last trace summary was generated. At the central server the average RSSI values reported for interfering nodes is added. This gives the measure of the amount of interference

Mac Address	00:0f:b5:96:c5:2b	0:0f:b5:96:c6:70	00:11:95:d8:e3:48	af:c6:ec:f0:1d:ba
Average Noise (dBm)	-92.1	-92.3	-91.8	-92.8
Average Signal (dBm)	-75.1	-63.3	-83.0	-64.8
Max. Noise (dBm)	-91.2	-91.2	-90.5	-91.7
Max. Signal (dBm)	-72.1	-61.2	-80.5	-61.9
Min. Noise (dBm)	-93.2	-93.2	-92.9	-93.4
Min. Signal (dBm)	-78.3	-65.1	-85.2	-66.3
Std. Dev. Noise	0.31	1.09	0.59	0.68
Std. Dev. Signal	2.14	3.19	1.81	1.01
Average Packet Size (KByte)	590	691	323	
Total No. of Packets	120	1339	272	1901
No. of Packets @ 11Mbps	0	1210	0	1743
No. of Packets @ 5.5Mbps	0	0	0	0
No. of Packets @ 2Mbps	0	0	0	0
No. of Packets @ 1Mbps	120	129	272	158
No. of Error Packets	0	5	1	0

Table 4.1: Example of trace summary information

each node is witnessing.

4.5.1 Challenges in collecting interference values

Measuring interference in the above mentioned fashion has two major challenges to be overcome.

- Low level per packet details exported to `/proc` file system are for packets that are addressed to receiver node as a immediate receiver. Packets that are sent on other neighboring link are not included in the `/proc` buffer. These packets are addressed to radio on the other end of link and are not exported to `/proc` file system for radios other than the radio its addressed for. To measure the interference caused by other nodes we need the RSSI values for the interfering packets but this information is not available.
- Interference ranges are much longer than the transmission ranges. If we assume all the links to be operating at the highest transmit rate, threshold SINR needed to receive the packets will also be high. This makes it difficult for radio to receive interfering packets. This increases the possibility of undetected interference. In some cases we could have the interfering source send packets at lowest possible rate, but doing it in an operational network is a challenging task.

4.5.2 Possible approaches for collecting interference values

We considered several approaches to collect interference values in operational network. In this section we discuss these possible approaches.

- **Moving into monitor mode**

One option which would have allowed looking the interfering packet details is to move to monitor mode periodically. We could put the link temporarily into monitor mode when there is not much traffic going on link. In addition to this we could continuously check the `/proc` entries to see if any packet addressed to this radio as immediate receiver is received. As soon as we find a such a packet we could put the link back into normal operational mode.

The concern with this approach is that the link becomes non functional from the time the first link packet is received to the time node comes back to the normal mode of operation. During this period all the packets received will be considered to be lost as this node will not be able to send acknowledgment packets back to the sender. We measured the time taken for link to switch from monitor mode to normal mode operation. We calculated this using the linux *time* command, this duration was found to be of the order of 30 milli seconds.

In addition to switching time this approach need other nodes to transmit some packets at lower transmit rate. This increases the chances of interfering packets getting received. This requires nodes to inject some monitoring traffic into the network and to switch to lower transmit rates for transmitting these extra packets. This approach needs to fix up a schedule for each node to transmit at lower transmit rates and for other to move to monitor mode. Designing such a schedule in large network deployment is difficult to do, and this approach thus is not scalable.

In Madwifi drivers for Atheros chipset there is provision of creating multiple virtual interface over the same hardware. This removes the need of putting link into monitor mode. Two virtual interfaces can be created onto same card and one of them can be put to normal operating mode and other can be put onto monitor mode to collect low level packet details for other link packets. Our reservation with this approach is that this would be a driver specific solution. Hostap drivers don't have virtual interface support.

- **Using multiple radios**

As we discussed above, in monitor mode the `/proc` log file have entry for each packet received. This can be used to measure the interference from other nodes. The problem with using monitor mode is that as long as the wireless network device is in this mode the link is no longer functional. A possible way out of this is to use multiple radios and put one of them in monitor mode while the other wireless network card can work in normal mode making link functional and at the same time collecting other link packets also. Multiple wireless card slots are supported in many of commercially available networking platforms like *Soekris* machines.

This approach is simple but has a drawback of increasing the cost incurred for extra WiFi card. This card will be used only for monitoring purpose. Since we are focused on wireless networks in developing regions, cost is an important metric for a network to be deployed widely. In addition to the cost one other concern with this approach was use of signal splitters. Splitters were needed so as to give both the radios the common feed received from the antenna. Splitters would be causing signal loss which will further decrease the chances of receiving packets with low RSSI values. We looked at data sheets of some commercially available splitters and many of them had maximum signal loss of the order of 4 dBm. Such loss will increase the amount of error in measuring the interference.

- **Driver modifications**

We investigated if it is possible to receive the other packets addressed to other nodes by some driver modification. We started with idea that if it is driver which is discarding the interfering packets on the basis of receivers MAC address. We can change the driver code so as to get the low level statistics before the packet is been discarded by the driver. As we looked into the Hostap driver code we realized that interfering packets are never passed to the driver and must have been discarded by the firmware in the card itself. So the idea of doing some driver modifications to export interfering packets low level statistics is not a possibility.

- **Broadcast packets**

The other possibility we considered is to see if broadcast packets can be used, these packets will be received by other nodes in the network. The idea is to ask every node in the network to transmit broadcast packet periodically at lowest possible transmit rate. This calls for added network traffic in terms of extra broadcast packets transmitted at lowest possible data rate. This idea of sending broadcast packets did not work because of the way MAC broadcast works in 802.11. In 802.11 MAC packets contain a field corresponding to BSSID of the network. This BSSID for a network consisting of master node and managed mode clients is by default set to the MAC level hardware address of the wireless device of master node. In long distance mesh network setting each of the links have a different BSSID assigned, so broadcast packets sent by radio of some link are discarded by other links. This is discarded because the driver find its

BSSID different from BSSID in broadcast packet.

4.5.3 Master mode operation in WDS setting

This section describes the approach we are following in this work for measuring the interference values.

Hostap driver allows a 802.11 access point to listen to beacon packets from other access points operating in the vicinity. This provision is available to make it possible to create a distribution system using a wireless link between access points. We confirmed this by doing experiments with Hostap driver on the Soekris machines. We observed that whenever the radio was in master mode it was hearing some packets sent from radios which were not having the same BSSID as this radio. We looked further into the details of packets in the `/proc`. We confirmed that all the packets which were received were beacon packets from other access points. We confirmed this with Hostap documentation also. We use this functionality for nodes to learn about interference from nodes operating in the vicinity.

The advantage of using these beacon packets is that it do not need any extra traffic to be introduced from user space. In addition the beacon packets by the 802.11 standard are to be transmitted at the lowest possible data rate. So this is no longer the duty of user space process to change to lowest possible data rate to transmit packets. Beacon packets are generally of small size of order of 40 Bytes, so it is consuming very less network bandwidth. These features have made interference measurement in an operating network more passive.

WDS operation

The Problem with this approach is that only the nodes which are operating in master mode are able to hear each others beacon packets. So, all the nodes which are associated with the master mode radio are not able to hear each other. Possible remedy to this is using the WDS setting. WDS stands for wireless distribution system and allows master mode radios to interact with other master mode radios. Figure 4.3 shows two access points in WDS to build a distribution system. In WDS setting a MAC level tunneling is done from one master node to other master node. In our long distance settings we could put two ends of a link into WDS setting. Now two ends of the link will be operating in master mode simultaneously. So each node hears beacons from other nodes in vicinity and at the same

time link remains functional also. One node can have WDS to multiple other nodes so this is applicable for point to multipoint links also.

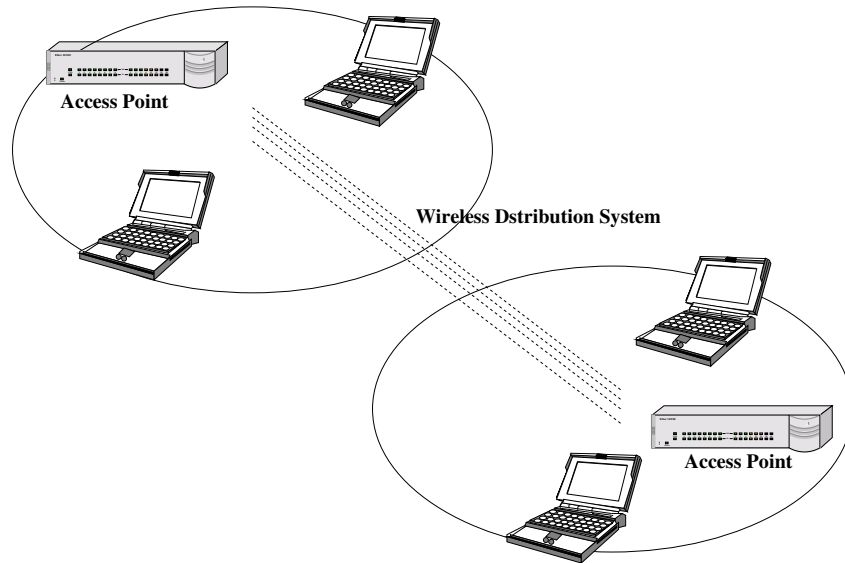


Figure 4.3: Wireless distribution system between two access points

The WDS operation is the basis of our solution to measure interference in the operational network. We have all the links of network in WDS setting. So all the nodes hear beacon packets from each other. The signal strength reported for these beacon packets is the measure of the interference the sender node is causing to the receiver node. These beacons are transmitted at lowest transmission rate. So the possibility of missing out on interference from some node is minimized.

As explained in Section 4.4 the client module creates a trace summary from monitored log data. There is an entry in trace summary for each of the interfering node. Actually the sender of the trace summary do not distinguish between the interfering nodes and the link nodes, while sending the trace summary. Its the responsibility of server to identify the interfering nodes and link nodes in the trace summary. The server identifies the interfering nodes as it have the knowledge about the network topology. In section 4.7.2 we look at how some of the undetected interference is also included in the network interference matrix.

4.6 Measuring Same/Adjacent Channel Interference

In addition to creating the current network interference matrix, server also needs to create the same channel and adjacent channel interference matrices. For same channel interference matrix we need the entire network to operate in same channel. For adjacent channel interference matrix the entire network should be on the adjacent channel with respect to the node for which interference is measured.

Two possible approach we looked at to collect same channel interference value are (1) collecting for whole network at the same time, or (2) collecting for one node at a time. We discuss these alternatives below.

4.6.1 All at the same time approach

In this approach we ask entire network to move into same channel and transmit beacons at full transmit power for some time. During this, each node receive beacons from all other nodes that are interfering with it. Each node sends the information about the interfering nodes to the server. So each node creates an entry in interference matrix showing the nodes that are interfering with this node and with what signal strength. Figure 4.4 shows the network channel assignment while doing such an experiment.

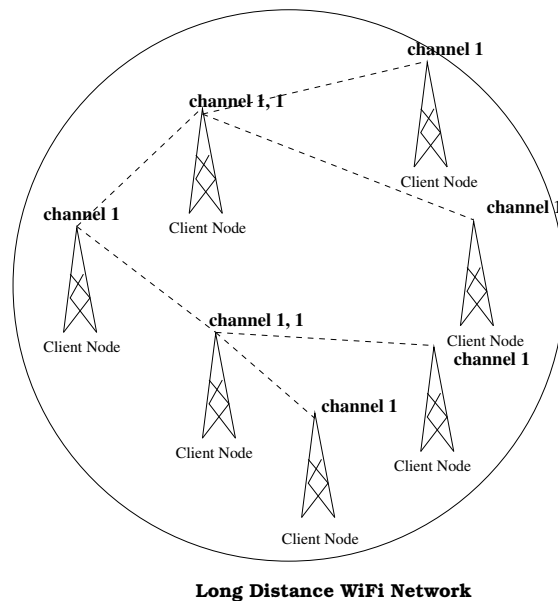


Figure 4.4: Network moving to channel 1 from normal operation

We have considered this approach but have not found it suitable to create interference

matrices. This approach will create the entire matrix in a short period of time compared with the alternative but will suffer from following drawbacks.

- Since we ask whole of the network to come to same channel this hinders with the normal working of the network. The entire network is required to be working on same channel and at maximum transmit power. This makes the network mostly non functional for the duration this measurement is performed.
- For a large network, deployment is mostly done in phases where new nodes are added with time. So with this approach of collecting same channel interference matrix we need the network to become non functional from time to time.
- If some node is getting misaligned then the current channel interference matrix needs to be updated to make the change in interference scenario visible. If misalignment is frequent then network will become non functional regularly for the duration interference measurements are performed.
- In rural settings power is a constraint. This setting requires all the nodes to be available at the same time. This is difficult to guarantee because of links frequently going down due to power constraints.
- This approach for creating adjacent channel interference matrix takes time proportional to number of nodes in the network. This will put network into non functional state for long durations while adjacent channel interference matrix is created.

4.6.2 One at a time approach

We have used this approach in WiFiNetMon for collecting the same channel and adjacent channel interference. In this approach network will be working as in normal scenario, only one group at a time will move to all the three non overlapping channels one by one. In this approach we assume all the nodes of the network to be time synchronized. We will be using NTP for synchronizing the nodes of the network with the server, NTP normally provides time synchronization in order of 100 of milliseconds which is good enough for our protocol requirements.

In this approach the central server keeps track of nodes which have incomplete information about the interference they are facing due to other nodes in the network. From time

to time central server tries to gather the interference by doing interference experiments for such nodes. These experiments are performed for such nodes one at a time. At the end of the experiment for a particular node central server has the information about the amount of interference this node is causing to other nodes and also the amount of interference this node is facing because of other nodes. Whenever the central server first starts no interference information is available for any of the nodes. So to start with, it is required to perform interference measurement experiment for each of the node.

Let us consider an interference measurement experiment performed for node A. Other nodes in this discussion refer to nodes of the network, which do not belong the same group as that of node A. This experiment requires entire node A group to move to all the three non overlapping channels and stay there for the specified time of *duration per channel* . In addition node A is asked to operate at maximum transmit power during the complete experiment. This can be understood more clearly by figure 4.5 and figure 4.6.

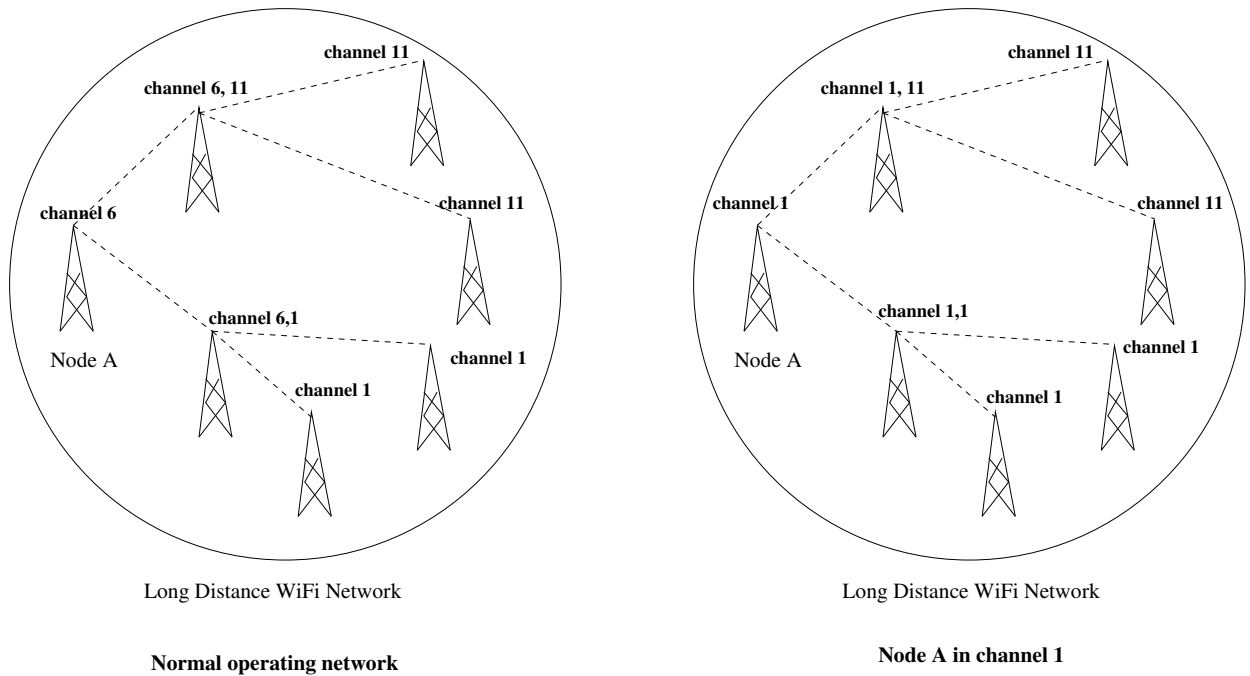


Figure 4.5: Node A moving to channel 1 from normal operation

In these figures node A is moving to all the three non overlapping channels to collect same and adjacent channel interference. Since node A is moving to all the three non overlapping channels, all the other nodes will have scenario, when they are on same channel and when they are on adjacent channel with respect to node A. So this way all other nodes

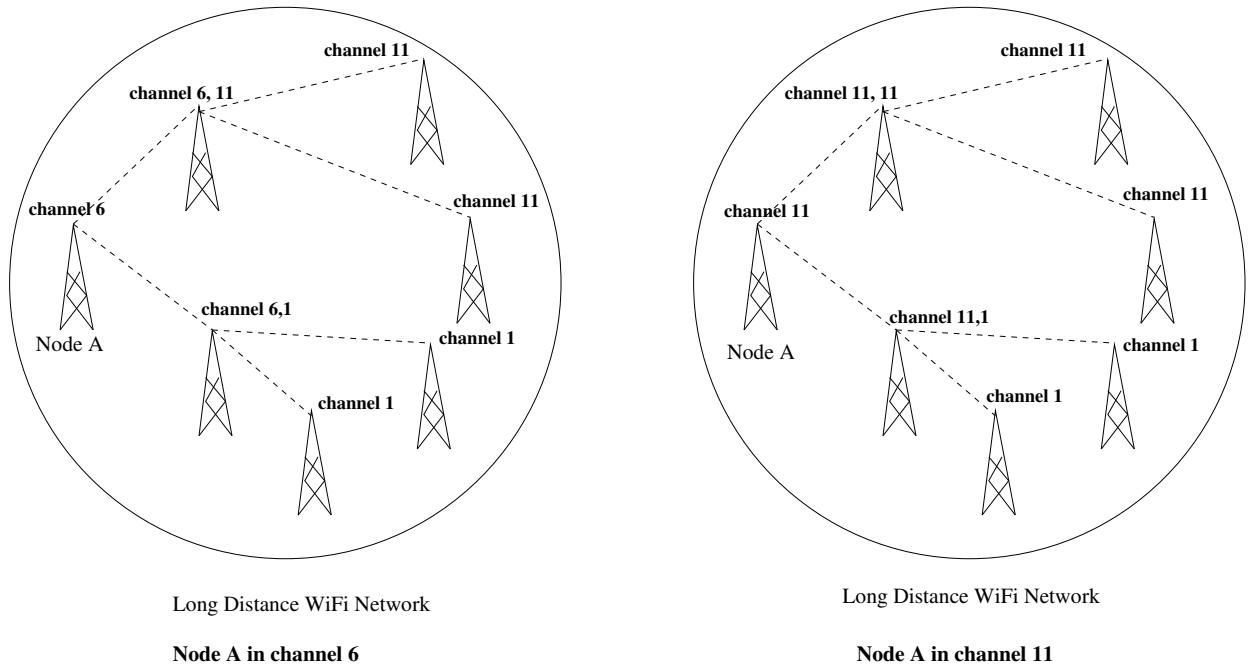


Figure 4.6: Node A moving to channel 6 and 11

get the opportunity to measure interference from node A, this is measured on the basis of signal strength for beacon packets transmitted at maximum transmit power by node A. All other nodes record the interference they see because of node A when node A is on adjacent channel with respect to it and when node A is on the same channel with respect to it. In addition to this node A also records the interference it faces on all the three channels. In addition to node A transmitting at maximum transmit power we need all other nodes also to be transmitting at maximum transmit power. This is needed so that node A can also register interference at maximum transmit power because of all other nodes operating in the network.

All nodes including node A send this recorded trace to central server. The central server infers the same channel and adjacent channel interference node A faces from other nodes, by looking at the experiment trace sent by the node A. In addition central server infers the adjacent and same channel interference that node A is causing to other nodes, by looking at the experiment trace sent by all other nodes of the network.

To perform experiment for any selected node, central server informs all the nodes of the network, about the *time of experiment* time at which interference measurement experiment will be performed and *duration per channel* the duration for which it will be performed.

The central server informs the selected node also about the *time of experiment* and *duration per channel*.

4.6.3 Interference measurement at the granularity of group

In our given approach, the whole of the group is supposed to move to three non overlapping channels. This is needed so as the links corresponding to group can remain functional. Using this idea we have modified our solution to collect interference at the granularity of the group. In place of collecting interference measurement for a single node at a time, we will be collecting interference for all the nodes present in the group. This will reduce the number of experiments needed to collect interference for whole network. Earlier this was of the order of number of nodes in network. Doing it at the granularity of group reduces its order to number of groups in the network.

4.7 Issues Involved in Overall Solution

Now we discuss some issues related to architecture. These are related to overall solution of monitoring and interference measurement.

4.7.1 Getting network topology

We wanted our solution to be as automated as possible. We do not want the network administrator to be bothered about tracking the network topology for interference measurement. We have placed the responsibility of collecting network topology on server node. Each client node give details about its neighbors to the central server. Since all the links are operating in WDS setting, details about neighbors is readily available in wds status file available in /proc file system for the wireless interface. This wds list is in terms of neighbors MAC addresses.

In addition to neighbors details each node give details about its own MAC address and IP address of the wireless interface. Host MAC address and IP address helps in creating the IP to MAC mapping for the entire network. This information is used to identify interfering nodes, as /proc gives information about interfering nodes in terms of MAC addresses.

4.7.2 Inserting missing interference values

The Figure 4.7 shows node A and node B both belonging to same network. Node A is interfering with node B but node B is outside node A's transmission range even for the lowest possible transmit rate. This interference caused by node A to node B goes undetected. Now consider node A operating at maximum transmit power. Node B is within node A's transmission range now. We use the interference values at maximum transmit power for detecting such interference cases. Interference value at current transmit power is calculated by decreasing the interference value at maximum transmission value by difference between current and maximum transmit power.

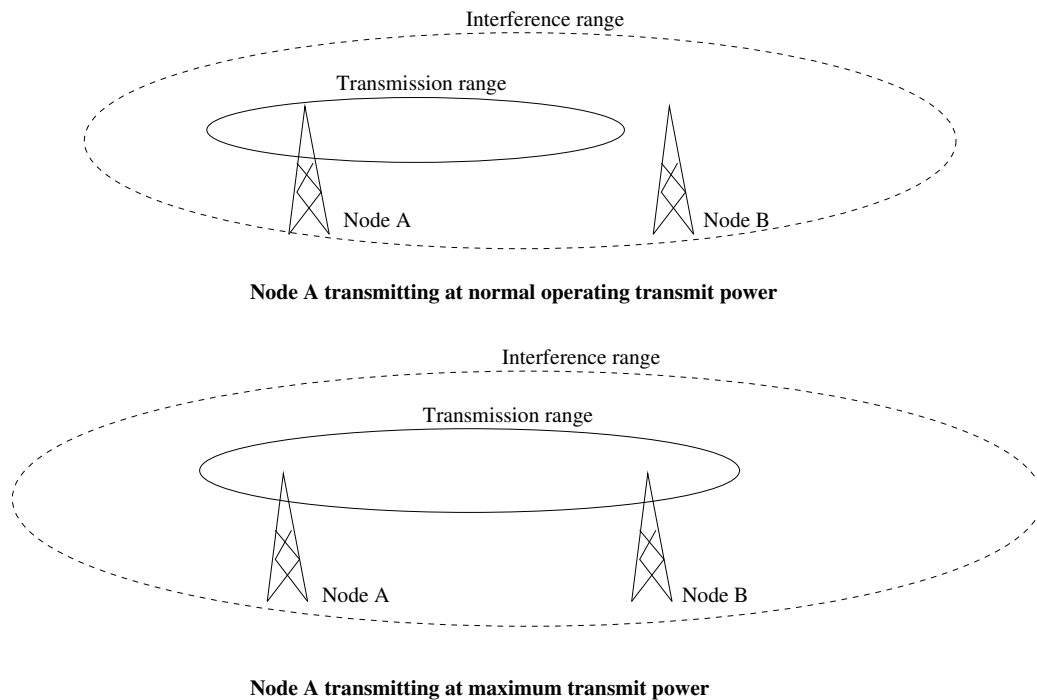


Figure 4.7: Node A transmitting at normal and maximum transmit power

4.7.3 Detecting link misalignment

The work in [4] has pointed to frequent link misalignments in their long distance WiFi deployment. A link misalignment changes the interference scenario in the network. This introduces the need of repeating interference experiments to make changes transparent to interference matrices. We use our first assumption mentioned in Section 4.3 about RSSI stability over time for updating interference matrices. We assume that the change in RSSI

over the assumed threshold is because of link misalignments. We constantly monitor the link signal strength in link signal matrix. We keep the signal strength for each link when last time the link was properly aligned, we call this value *link signal history* . The current RSSI value for link is calculated as weighted moving average. Whenever current RSSI value changes from *link signal history* by more than a threshold value, we attribute this to link misalignment. The threshold value we have worked with is 8 dBm. the work in [3, 16] point to variation of RSSI to be in band of 0-4 dBm in normal scenario. We add a guard of 4 dBm more to avoid false link misalignment call and use the value of 8 dBm as threshold.

4.7.4 False link misalignment call

In Section 4.7.3 we discussed about detecting link misalignments. Consider a scenario when a node changes its transmit power to a new value. Now the current signal strength for links in which this radio is involved will change to new values. Slowly the weighted moving average signal strength will change. This new signal strength value might change to a value differing with *link signal history* by more than threshold value. This will be detected as link misalignment. To avoid these false calls we perform a link check test to confirm if link misalignment is true. In this test server ask the link to operate at maximum transmit power. Reported signal strength is then compared with signal strength reported last time interference experiment was performed for the link. If this value change is more than the threshold than this is a link misalignment and network administrator is informed to diagnose the corresponding link. Otherwise *link signal history* is changed to current signal strength value the link is reporting.

4.7.5 Identifying zero interference

Consider a scenario when a node A is not interfering with node B because node A is outside node B's interference range. Other scenario is when node B is switched off when we are performing the interference experiment for node A. In both scenario's node A see no interference from node B. To distinguish these two scenarios server keeps track of nodes which send the experiment result. If node B has participated in experiment it would have sent result to central server. This confirms that node A has zero interference from node B. If node B is absent from list of nodes participated in experiment, this confirms that

interference between node A and node B is yet to be computed.

4.8 Implementation Details

We have designed the server module using PHP. PHP 5.0.4 engine is used to run server scripts. Client modules are designed using C++. GCC 4.0.0 is used to compile the client scripts. Soekris machines running the client modules were having voyage 0.2 distribution. Soekris machines were running with linux kernel version 2.6.15. Wireless driver supporting the wireless interface was Hostap 0.4.4. XML was used for communication between the client and server modules. Choice of XML was because of its human readable format, ability to modify easily and extensive support for it in various programming languages.

Chapter 5

Evaluation

In this chapter we evaluate our system for correctness and overheads involve in interference measurement and link monitoring. Section 5.1 talks about the emulation performed to check the correctness over a 8 and 16 node deployments. The next important metric we consider is the overheads involved in channel switching during interference experiments. These overheads are discussed in Section 5.2. Subsequently we look at CPU utilization and memory overhead in Section 5.3. Additional traffic introduced for network monitoring and interference measurement is the topic of discussion in Section 5.4. Finally, we measure the time taken to create interference matrices in Section 5.5.

5.1 Emulation for WiFiNetMon

To check the correctness of WiFiNetMon architecture we performed the emulation of wireless network over the Soekris machines. During the implementation we tested the architecture modules over a four node, two link wireless deployment only. This deployment was over the real Soekris machines. We felt that this small deployment was not sufficient to test the architecture completely. We wanted to test our implementation in scenarios where all the links were not available at the same time, links were coming up and going down. We wanted to test if network topology is being correctly detected for a bigger deployment.

A physical deployment of a wireless network at large scale is too cumbersome in terms of the effort needed in collecting and managing resources. Our aim was to check the functional correctness of the architecture for a bigger deployment of 8 or 16 nodes. We choose to setup network emulation for such a study, as explained below.

For emulating a wireless network deployment we ran multiple instances of the client

modules on the same Soekris machine. These multiple instances were bound to different ports. We made appropriate changes in both client and server modules of the original architecture implementation, to allow such emulated operation. The IP and MAC addresses that client module reports to the server were modified to include the port number at which client module is bound, this gave each instance of running client module a unique IP and MAC address. In addition to this, port numbers were appended to the MAC address for each entry in the /proc driver log file. These port numbers were randomly chosen from the set of port numbers at which client modules were running. This gave the interference values from the different client modules which were running over same machine and were using the same wireless card.

We ran the emulations for 8 and 16 node deployments. The emulation setup for 16 nodes is shown in Figure 5.1. As shown in figure three Soekris machines and a laptop were used to run the client modules. The wireless interfaces of Soekris 4521 and laptop was using D-Link DWL-650 PCMCIA cards and Soekris 4826 was using a NETGEAR mini PCI card. The emulations helped us in checking the correctness of architecture. We also found a bug in the implementation of architecture which was not transparent for small 4 node deployment. This was related to server module running out of shared memory. This error was not flagged for small deployment as shared memory allocated was sufficient for a 4 node deployment.

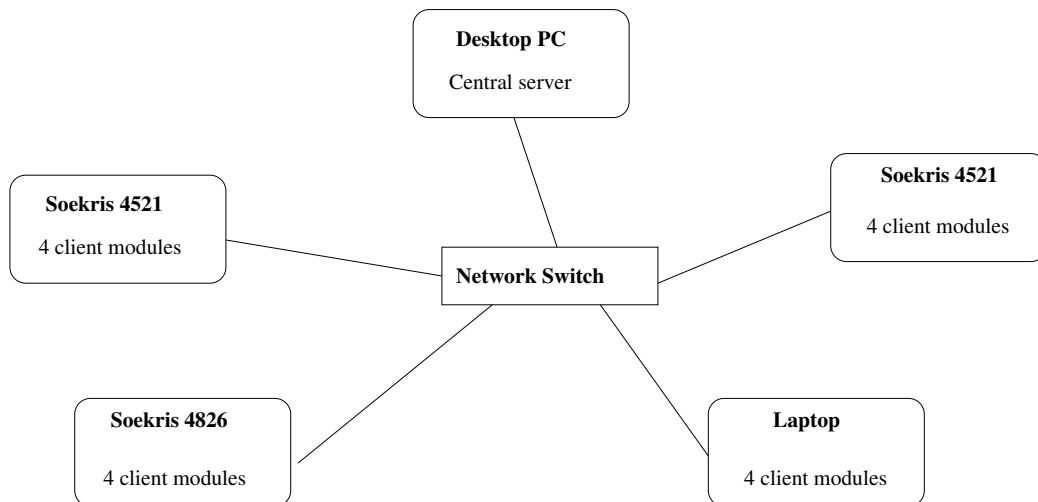


Figure 5.1: Setup for emulation of 16 node deployment

5.2 Channel Switching Overhead

As explained in Section 4.6.2 measuring same/adjacent channel interference requires a link to change its channel settings. We were interested in measuring the time for which a link is unavailable during a channel changing operation. To better understand the overhead involved in terms of link availability during interference measurement, we ran some experiments on the real Soekris machines.

The metric we chose to measure this overhead is in terms of the time for which link becomes non functional. The Figure 5.2 shows the setup of the experiment. To measure this non functional duration we transmitted packets at fixed inter packet duration over the link. The link between Soekris machines was asked to change its channel during this transfer. The number of packets being lost at the receiver gave us the duration for which the link was non functional. This experiment required us to establish a wireless link with close to zero packet error rate during normal operation, so that all the packet losses can be attributed to channel change. We set up the link in the WDS setting, as in our WiFiNetMon system design. The transmit rate was set to the minimum value of 1Mbps, so as to reduce the possibility of packet loss because of reasons other than channel switching. We used MAC level broadcast to remove link layer retransmissions. MAC level broadcast functionality was achieved using the modified Host-ap driver used in [3]. We used UDP traffic to avoid any transport layer retransmissions. The purpose of avoiding retransmissions was to directly infer non functional duration from number of packets lost.

The radio at one end of the link was asked to transmit the 1000 byte packets at regular inter-packet interval of 10 milli seconds. The radio on the other end was keeping track of number of packets received. At the end of experiment the receiver was asked to report the number of lost packets. Both the sender and receiver radios were asked to operate at channel 1 initially. They both were asked to move to channel 6 at the same specified time. Both the radios were regularly synchronized with the NTP time server every 1 second by a separate daemon process.

The Table 5.1 gives the number of packet losses reported for 10 different runs of the experiment. On an average 57 packets were lost. These all the packet losses were continuous sequence numbers in the receiver's log. The loss of 57 packets means that link was non functional for close to 570 milli seconds. A link will be required to do the same channel

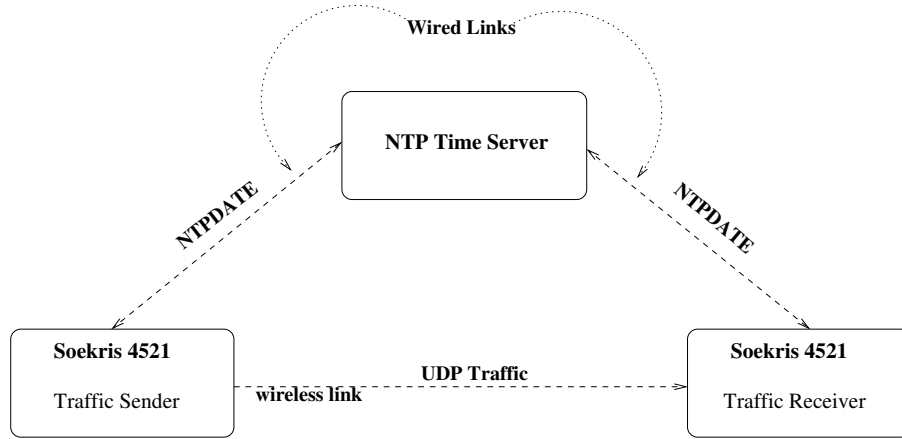


Figure 5.2: Setup for channel switching experiment

interference measurement only very infrequently. So an overhead of 570 milli seconds once in a while is tolerable. The main reason for this loss of link was because two radios of the link were not tightly time synchronized. We monitored the adjustments made for time synchronization with the server clock. These adjustments were found to be varying from 10 of milli seconds to few hundred milli seconds.

In addition to channel switching we performed similar tests for changing transmit power and changing beacon interval. No packet losses were reported for these two operations. We checked the time taken to change these two parameters using *time* command. It reported close to 15 milli seconds for both these operations. Since we have not seen any packet loss, so we infer that link remains functional during these parameter settings.

Experiment Number	No. of Packets Lost	Non functional duration (milli seconds)
1	47	480
2	78	780
3	72	720
4	70	700
5	21	210
6	68	680
7	68	680
8	74	740
9	6	60
10	67	670
Average	57	570

Table 5.1: Packet lost on link during channel switching

5.3 CPU Utilization and Memory Overhead

The client module is meant to be executed on nodes which in the most cases will be resource poor in terms of CPU computational power and memory. This makes the computational overhead of the client module a crucial factor to consider. We measured the overhead in terms of the CPU utilization at the client nodes. We did this for two different hardware platforms, Soekris 4521 and Soekris 4826. The Soekris 4521 has a 133 MHz AMD Elan SC520 micro controller while Soekris 4826 has a 266 MHz AMD Geode SC1100 processor. The Soekris 4826 has 128 Mbytes of SDRAM and 128 Mbytes of soldered flash. The Soekris 4521 has 64 Mbytes of SDRAM and 256 Mbytes of flash memory.

We monitored the CPU utilization by the client module when node was receiving packets from another node over the wireless interface. We did it for cases when node was receiving packet at inter packet duration of 0 milli seconds and 10 milli seconds. Packet size of 1000 byte was used. UDP with MAC level broadcast was used to send these packets. The experiment with 0 milli second inter packet duration was done to test the client module in the worst case scenario, when the wireless channel is fully occupied. The CPU utilization was measured using unix *top* command. The CPU usage for 0 milli second inter packet arrival time is shown in Figure 5.3. The average CPU utilization for Soekris 4826 and Soekris 4521 is 39% and 41% respectively. The high CPU utilization values were mainly due to processing of large amount of log data, in full speed blast large number of packets are received by wireless radio and client module needs to process low level details for all of them. These CPU utilization value were for the worst case scenario.

The Figure 5.4 shows the CPU utilization at 10 milli second inter packet interval. The average CPU utilization is 10% and 12% for Soekris 4521 and Soekris 4826 respectively.

The above reported CPU utilization value for Soekris 4826 and Soekris 4521 are close to each other. This is in spite of them being of different processor speeds. Its difficult to compare CPU utilization between these two Soekris platform. These platforms differ from each other in both processor and memory configurations. In addition the CPU utilization will also depend on number of packets received by wireless interface. The two platforms were using different type of wireless cards, Soekris 4826 was using a NETGEAR mini PCI card while Soekris 4521 was using a D-Link PCMCIA card. This again makes it difficult to compare the CPU utilization on these two platforms.

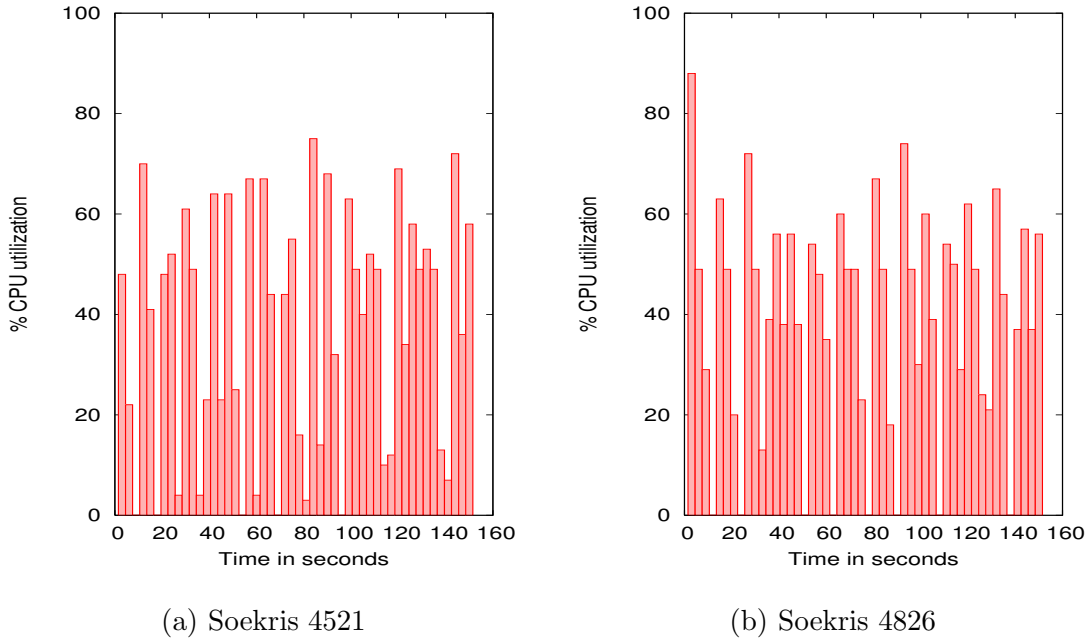


Figure 5.3: CPU utilization at 0 millisecond inter packet gap

The CPU utilization's reported in this section are for scenario when client module is monitoring all the packets it receives. This is needed only when we want to monitor the link for throughput evaluation and packet error rates. If the purpose of monitoring is to collect the interference or link signal strength information only, We can avoid transferring all the /proc log entries to user space. We can periodically transfer the /proc log file without bothering about /proc buffer overflow at the driver. In this case CPU utilization will be dependent on the frequency at which we transfer the /proc log file to user space. If we read the /proc file once in every 10 minutes then the CPU will be used for 1.5 seconds (time taken to create trace summary) every 10 minutes. This will amount for average CPU utilization of 0.25%.

The memory overhead introduced by the client module is for the driver buffer to store the per packet low level details. In current implementation we were using driver buffer for 2000 entries. Each log entry present in the driver buffer takes 44 Bytes of memory, so driver buffer requires about 88 Kilo Bytes of memory. In addition to this, 400 Kilo Bytes of disk space is required to store the entries from the driver buffer into user space file. The space requirement by client module is negligible as compared to available memory on the Soekris machines.

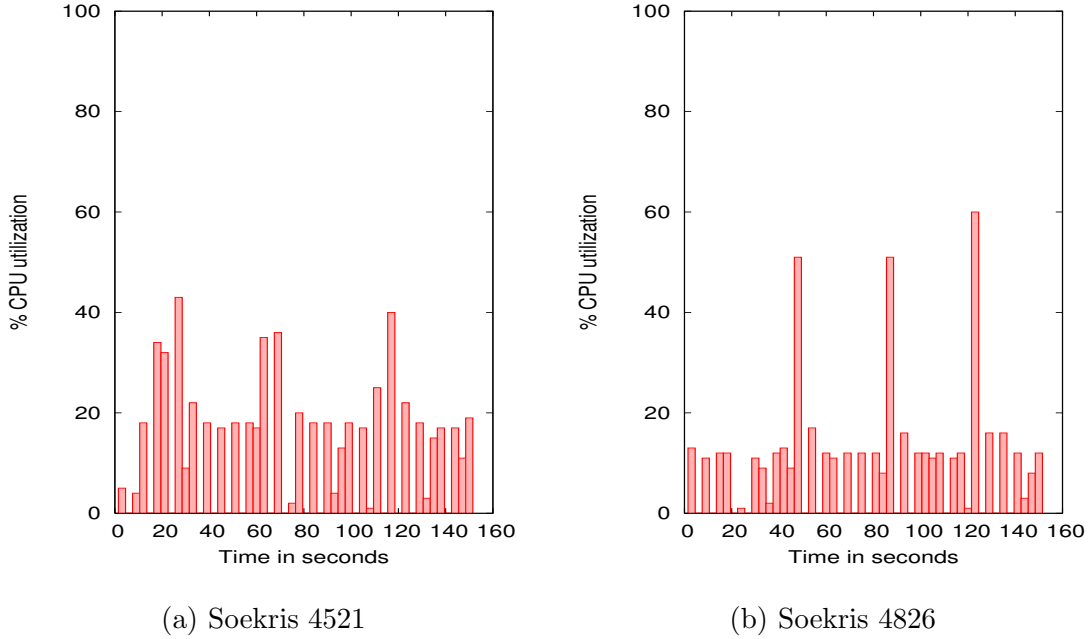


Figure 5.4: CPU utilization at 10 millisecond inter packet gap

5.4 Network Overhead

The additional traffic introduced by interference measurement and monitoring is for broadcasting beacon packets, sending trace summaries and experiment results to the server and sending experiment requests to the client nodes. Out of this, only the beacon packets are part of regular traffic, while the others are transmitted only once a while and can be neglected. The number of beacon packets transmitted depends on the beacon interval used. The beacon interval we are using in current implementation is 100 milli seconds. This is chosen so as to limit the additional traffic introduced. Assuming the beacon packet size to be 88 bytes, the beacon packets will be occupying the radio for 0.7% of time.

Since we are assuming a TDMA scheme, we need to take into account the number of members in a group for calculating network overhead. In current Ashwini deployment shown in Figure 2.4, the average number of members per group is 3.5, we round it out to 4. So the network overhead due to beacon packets in Ashwini deployment will be 2.8% of time. The total bandwidth network bandwidth consumed by the beacon packets will be 2.8% of the total available bandwidth.

In the proposed architecture the client modules send trace summaries to the server once in every 10 minutes. This duration is chosen to have average signal values computed over a

large enough elements. In between two trace summaries each node will be transmitting close to 6000 beacons (assuming a beacon interval of 100 milli seconds). This number of 6000 is a large enough size to take care of any outliers in signal strength values while reporting the average values to the server.

The network overhead can be reduced further by increasing the beacon interval. In that case trace summary interval should also be increased to a higher duration so as to collect enough packets while calculating average signal strength values.

5.5 Time Taken to Create Same/Adjacent Channel Interference Matrices

This metric is dependent on the number of groups in the network. As explained in Section 4.6.2, to collect interference details each of the groups is required to perform the interference measurement experiment. Thus creating complete interference matrices for network would require time of the order number of groups in the network. This is assuming that whole of the network is active. Time taken to create interference matrices will also depend on, *time out duration* used while trying to connect and *duration per channel* parameter used in the interference measurement experiment. In addition to above mentioned parameters server keeps some free duration between two consecutive experiments. We call this guard time. This is used to ensure that all the results of previous measurement experiment are received and processed before new experiment is performed.

The *duration per channel* parameter should be chosen so that, experimenting node has enough time to collect interference information on each channel. The *duration per channel* parameter is important, as during this duration a group will be operating on channels other than, the one that is assigned during channel allocation. This will be creating unnecessary interference in the network. Decreasing the beacon interval is an option to spend less time per channel. By decreasing the beacon interval node can learn about interference faster. They will get to hear more number of beacon packets from interfering nodes in the shorter span of time. But this will increase the network overhead in terms of additional traffic introduced in the network.

We have worked with *duration per channel* of one minute. Along with this we are

working with increased beacon interval of 10 milli seconds. With beacon interval of 10 milli seconds each of the node transmits 6000 beacons per minute. So while reporting the same/adjacent channel interference value, each node sends the value averaged over 6000 packets. This increases the accuracy of the reported interference value. We have worked with time out interval of 30 seconds. This is chosen so that central server tries sufficiently for establishing a connection with active client nodes.

We now calculate the time our architecture will take for measuring same/adjacent channel interference matrices for current Ashwini deployment shown in Figure 2.4.

Total time = Number of groups * (3 * *duration per channel* + 2 * *time out duration* + guard time)

Number of groups in Ashwini = 6.

duration per channel = 1 minute

time out duration = 30 seconds

guard time = 1 minute

Total time = 6 * (3 * 1 + 2 * .5 + 1)

Total time = 30 minutes.

During this duration of 30 minutes the network continues to work normally. Some links might become non functional for negligible time as discussed in Section 5.2. This 30 minute duration is needed in scenario when the server module starts at first and, do not have previous knowledge of the interference scenario. If server module has complete knowledge of interference scenario and, a new link is added then, interference measurement experiment is needed to be performed only for this new link. Similarly, if some link misalignment happens then we need not repeat interference experiment for all the nodes. We perform interference measurement experiment only for this affected link.

Chapter 6

Conclusion and Future Work

In this thesis work, we considered the problem of measuring the interference scenario in long distance WiFi networks. We have successfully designed, implemented, and evaluated a system to

- monitor the link signal strength for each link.
- measure the interference experienced by each link in the operational network.
- measure the interference experienced by a link if all other links were running on the same channel (same channel interference).
- measure interference experienced by a link if all other links were running on a non-overlapping adjacent channel (adjacent channel interference).

The network overhead for monitoring and measuring interference is found to be near 3% of the total available bandwidth. The duration for which some part of network become non functional during interference measurement is found to be less than 1 second. The CPU utilization for client module on link nodes is found to be close to 10%, assuming 100 packets transmitted per second over the link and all the received packets are monitored. The CPU utilization falls to meager 0.25% if client modules are used to monitor only 2000 of the received packets per 10 minutes.

There are several avenues for future work. The system is still to be tested in real long distance WiFi deployment. There is need of system which can take interference information as input and could decide on optimum transmit power and optimum channel assignment for a long distance deployment. This will add interference management functionality into

WiFiNetMon. The current CPU utilization by client module are on higher side, there is need to reduce the CPU utilization. There is scope of work on exploring, if CPU utilization can be reduced by using more suitable data structure and algorithm while creating trace summary.

Bibliography

- [1] Project Ashwini: Virtual Delivery Services. <http://www.byrrajufoundation.org/ashwini/home.htm>.
- [2] Akshaya: Gateway to Opportunities. <http://210.212.236.212/akshaya/index.php>.
- [3] KAMESWARI CHEBROLU, BHASKARAN RAMAN, SAYANDEEP SEN. Long-Distance 802.11b Links: Performance Measurements and Experience. *In MOBICOM*, 2006.
- [4] SHETH, A. NEDEVSCHI, S. PATRA, R. SURANA, S. BREWER, E. SUBRAMANIAN, L.. Packet Loss Characterization in WiFi-Based Long Distance Networks. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications, IEEE*, May 2007
- [5] ANMOL SHETH, CHRISTIAN DOERR, DIRK GRUNWALD, RICHARD HAN, DOUGLAS SICKER. MOJO: a distributed physical layer anomaly detection system for 802.11 WLANs. *MobiSys*, june 2006, 191-204
- [6] R. DRAVES, J. PADHYE, AND B. ZILL. Routing in multi-radio, multi-hop wireless mesh network. *In MOBICOM*, 2004.
- [7] DANIEL AGUAYO, JOHN BICKET, SANJIT BISWAS, GLENN JUDD, ROBERT MORRIS. Link-level Measurements from an 802.11b Mesh Network. *In SIGCOMM*, Aug 2004.
- [8] A. RAINWALA AND T. CHIUEH. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. *In INFOCOM*, 2005.
- [9] TILL KLEISLI. Channel Allocation for IEEE 802.16 Mesh Networks. *Masters Thesis, Swiss Federal Institute of Technology*, October 2006.
- [10] SAYANDEEP SEN. Topology Planning for Long Distance Wireless Mesh Networks. *Master's Thesis, Indian Institute of Technology, Kanpur*, May 2006.
- [11] PAUL IPE. Power Allocation Issues in a Wireless Mesh Network. *Bachelor's Thesis, Indian Institute of Technology, Kanpur*, May 2004.

- [12] D. DE COUTO, D. AGUAYO, J. BICKET, AND R. MORRIS. High-throughput path metric for multi-hop wireless routing. *In MOBICOM*, 2003.
- [13] Clarity & Understanding: The High-speed WLAN standards debate. *www.wi-fiplanet.com/tutorials/article.php/990101*.
- [14] K. JAIN, J. PADHYE, V. PADMANABHAN, AND L. QIU. The impact of inter-ferece on multi-hop wireless network performance. *In MOBICOM*, 2003.
- [15] M. KODIALAM AND T. NANDAGOPAL. Charaterizing achievable rates in multi-hop wireless newtorks: The joint routing and scheduling prob- lem. *In MOBICOM*, Sep. 2003.
- [16] DATTATRAYA Y GOKHALE. The Feasibility and Usefulness of Link Abstraction in Wireless Networks. *Master's Thesis, Indian Institute of Technology, Kanpur*, May 2007.
- [17] JITENDRA PADHYE, SHARAD AGARWAL, VENKATA N. PADMANABHAN, LILI QIU, ANANTH RAO, AND BRIAN ZILL. Estimation of Link Interference in Static Multi-hop Wireless Networks. *In IMC*, Oct. 2005.
- [18] ATUL ADYA , PARAMVIR BAHL , RANVEER CHANDRA , LILI QIU. Architecture and techniques for diagnosing faults in ieee 802.11 infrastructure networks. *In MobiCom 04: Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004, ACM Press, pp. 3044.
- [19] CAMDEN C. HO, KRISHNA N. RAMACHANDRAN, KEVIN C. ALMEROOTH, ELIZABETH M. BELDING-ROYER. A scalable framework for wireless network monitoring. *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, October 01-01, 2004, Philadelphia, PA, USA
- [20] LILI QIU , PARAMVIR BAHL , ANANTH RAO , LIDONG ZHOU. Troubleshooting multihop wireless networks. *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, June 06-10, 2005, Banff, Alberta, Canada.