Long-Distance 802.11b Links: Performance Measurements and Experience*

Kameswari Chebrolu
Dept. of EE, IIT Kanpur
Kanpur, INDIA 208016
chebrolu AT iitk.ac.in

Bhaskaran Raman
Dept. of CSE, IIT Kanpur
Kanpur, INDIA 208016
braman AT iitk.ac.in

Sayandeep Sen
Dept. of CSE, IIT Kanpur
Kanpur, INDIA 208016
sdsen AT iitk.ac.in

ABSTRACT

The use of 802.11 long-distance links is a cost-effective means of providing wireless connectivity to rural areas. Although deployments in this setting are increasing, a systematic study of the performance of 802.11 in these settings is lacking. The contributions of this paper are two-fold: (a) we present a detailed performance study of a set of long-distance 802.11b links at various layers of the network stack, and (b) we document the various non-obvious experiences during our study.

Our study includes eight long-distance links, ranging from 1km to 37km in length. Unlike prior studies of outdoor 802.11 links, we find that the error rate as a function of the received signal strength behaves close to theory. Time correlation of any packet errors is negligible across a range of time-scales. We have observed at least one of the link to be robust to rain and fog. But any interference on the long-distance links can be detrimental to performance. Apart from this however, such long-distance links can be planned to work well with predictable performance. During our measurements, we have observed a few hardware/driver quirks as well as system bottlenecks apart from the wireless link itself. We believe that our measurements and the documentation of our experience will help future network planning as well as protocol design for these networks.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Measurement, Performance, Experimentation

Keywords

802.11 mesh networks, Link-level measurements, Wireless link characteristics, Application throughput, Point-to-point links

*This work was supported by Media Lab Asia, IIT Kanpur.

MobiCom'06, September 23–26, 2006, Los Angeles, California, USA.

1. INTRODUCTION

No solution for pervasive networking can ignore the rural areas of world, which account for the majority of human population. IEEE 802.11 [3] (WiFi) has been proposed as a cost-effective option to provide wireless broadband in rural areas [7, 9]. In the developing and developed world alike, 802.11 links are being used in long-distance (up to several tens of kms) settings. Some examples are: (a) the Ashwini project [4] in Andhra Pradesh, India, (b) the Akshaya deployment [1] in Kerala, India, (c) the Digital Gangetic Plains testbed [7] in Uttar Pradesh, India, (d) DjurslandS.Net: a deployment in Denmark [2].

Although 802.11 does not specify operation in long-distance settings, there are several vendor products available for such scenarios (e.g. Cisco, SmartBridges, iBridge, etc.), with proprietary MAC protocol modifications. There has been at least one research effort looking at protocol design for such long-distance networks [12, 13]. However, there has so far been no systematic performance study of such links. While there have been detailed studies of WLAN deployments [11] as well as 802.11-based community networks [6], these do not necessarily apply to long-distance 802.11 links. In fact, our measurements show several differences from those in [6].

To our knowledge, this paper is the first to present a detailed performance study of long-distance 802.11 links. The questions we seek answers for are as follows.

- What are the packet error-rates seen on the long-distance links, and how do they vary with received signal strength?
- Is there any dependence of the packet error rate on the link length?
- What is effect of packet size and transmit rate (modulation) on the packet error rate?
- Is there any time-correlation in the packet errors seen? At what time scales?
- What effect do weather conditions (rain/fog) have on the link performance?
- Are there any MAC-level ACK timeouts on the long-distance links? What effect does this have on the application throughput?
- What is the effect of inter-link or external interference?

Answers to the above questions have implications on the planning of long-distance links, protocol design, as well as application design.

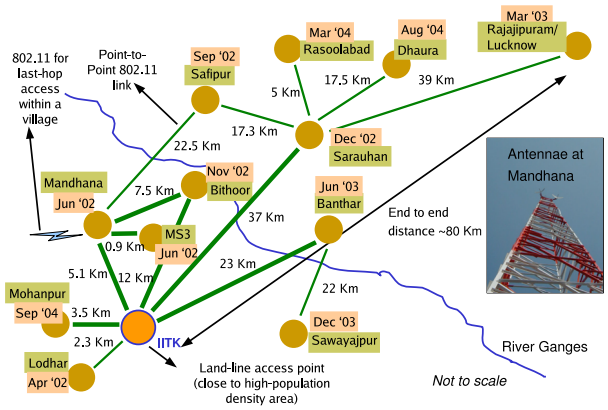

Figure 1: The Digital Gangetic Plains testbed

We use eight different links for our study, the shortest of which is 1km and the longest 37km. Seven of the links were in the Digital Gangetic Plains (DGP) testbed [7] and one was in the Ashwini project deployment [4]. The links were setup using high-gain directional antennae on top of antenna towers or tall buildings. Fig. 1 shows the DGP testbed, with the seven links used for our measurements drawn bolder than the rest. For most of our experiments, we used off-the-shelf Senao 2511CD plus ext2 PCMCIA cards, based on the Prism2 chipset. Although our study consists of only eight links, the consistency of the results across the links indicates that our conclusions are indeed meaningful. Our main results are as follows.

- The dependence of the packet error-rate on the signal-to-noise ratio (SNR) is close to theory. There is a small window of SNR, spanning about 4 to 6dB, below which the error rate is close to 100%, and above which the error rate is less than 1%. This observation is in sharp contrast with what is known for outdoor 802.11 deployments in community networks [6].
- The packet error rate does not (directly) depend on the link length, only on the received SNR.
- The effect of packet size on the error rate is noticeable, but not significant.
- There is a definite dependence of the packet error rate on the transmit rate (modulation used); the dependence suggests a simple and robust rate adaptation algorithm based on the SNR at the receiver. This again is in contrast with [6].
- The time correlation of any packet errors is insignificant in the absence of external interference.
- We observed the performance of one of the links (5km long) during two days with periods of rain and fog; weather conditions had little effect on the link.
- We observed MAC-level ACK timeouts only on the longest link (37km) of our testbed; application throughput is affected critically in the presence of such timeouts.
- Any external interference on these links is detrimental to performance. This too is in contrast with the conclusion presented in [6].
- Operation of adjacent links independently even in the so-called non-interfering channels (e.g. 1 and 6, or 1 and 11) involves subtleties. Phenomena such as the antenna *near-field* effect and *leakage* from RF connectors come into play.

Our results do not uncover any previously unknown phenomena, but indicate which phenomena kick-in for the long-distance links, to what extent it affects the link performance, and which phenomena are not significant. The main implication of the above results is that *long-distance links can be planned well for predictable performance*. The "link abstraction" [8] is natural for long-distance links, and it does hold. The last two results in the list above underscore the importance of *planning*. This has technical implications as well as non-technical implications.

The technical implications are multi-fold. Unlike in urban community networks, since the link abstraction holds, efficient routing is unlikely to be an important issue. In contrast, there is a need for algorithms and tools to plan such long-distance networks. Similarly algorithms are also needed to diagnose any cases of interference. The knowledge-base for such functionality exists (and evolving) for enterprise or campus deployments of WiFi, but not for long-distance mesh networks.

In the course of our measurements, we have modified the open-source HostAP driver to pass up various diagnostic information to the user level. Such a mechanism is likely to form an integral part of any tool for planning/diagnosis.

The non-technical implications arise from the fact that any RF pollution (interference) is detrimental in long-distance networks. This means that unless there is some legal or semi-legal control in the use of the spectrum, commercial operators are likely to desist from investing in the infrastructure required to provide any service.

Apart from the different results, we document various lessons which were not obvious to begin with. We make observations of hardware/driver quirks too, and also of system bottlenecks other than the wireless link.

The rest of the paper is organized as follows. The next section (Sec. 2) describes our experimental setup and methodology in depth. Subsequently, Sec. 3 presents an in-depth analysis of the packet error rate and its dependence on various factors. Sec. 4 then analyzes the application throughput seen on the various links. The effect of interference is then explored in Sec. 5. At the end of each (sub-)section, we discuss the implications of the results presented in that (sub-)section. In Sec. 6, we document some of the simple lessons we learnt the hard way. We conclude in Sec. 7.

2. EXPERIMENTAL SETUP AND METHODOLOGY

In this section, we first describe the links used for the experiments (Sec. 2.1). We then present the hardware and software setup (Sec. 2.2). Sec. 2.3 describes the various logistical challenges we faced, and Sec. 2.4 presents the overall experimental methodology.

2.1 Long-Distance Links Used

A long-distance link in our experiment consists of two WiFi radios communicating with one another between two sites. We use towers or tall buildings for line-of-sight, which

is essential for long-distance WiFi links. Directional antennae mounted atop tall buildings/towers are used at one or both ends to achieve the necessary transmit/receive gains for link setup. This is depicted in Fig. 2.

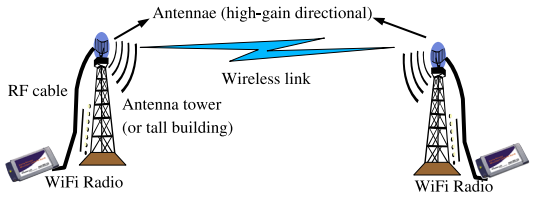

Figure 2: Long-distance link setup

Our experiments involved frequent moving around of our set of WiFi equipment among the different sites. The RF cable (as shown in Fig. 2) connected to the antenna allowed us to connect/disconnect the WiFi radio equipment without having to climb the tower each time.

The set of site locations we used for our study is given in Table 1. The first seven are at a particular geographical location, while the last two are at another location about 1500km away.

Site Name	Notation	Tower arrangement	Mains power supply	Alternate power supply
IITK	A	40m building	Available mostly	–
Mohanpur	B	17m tower	Not available	12V battery + stabilizer circuit
Mandhana	C	40m tower	Available at times	12V battery + stabilizer circuit
MS3	D	30m tower	Unreliable, huge voltage fluctuations	12V battery + stabilizer circuit
Bithoor	E	25m tower on roof of 15m building	Available at times	12V battery + stabilizer circuit
Banthar	F	25m tower	Available at times	12V battery + stabilizer circuit
Sarauhan	G	40m tower	Not available	12V battery + stabilizer circuit, solar panel
Bhimavaram	P	45m tower	Available mostly	–
Kesavaram	Q	30m tower	Available at times	Battery + inverter

Table 1: Set of sites used for link setup

Link	Length (km)	Antennae	RF cables	Remarks
A-B	3.5	ParG-ParG	50ft, 100ft	–
A-C	5	Sec-ParG	50ft, 150ft	–
C-D	1	ParG-Can	125ft, 50ft	Ant at 30m at C, 15m at D
E-D	7.5	ParG-ParG	125ft, 50ft	–
A-F	23	ParG-ParG	50ft, 100ft	–
A-G	37	ParG-ParG	50ft, 150ft	–
A-E	12	ParG-ParG	50ft, 150ft	–
P-Q	16	Sec-ParG	1ft, 1ft	Power-over-Ethernet for radio atop the tower

Table 2: Links used for the measurements

The description of the eight links we used for the measurements is given in Table 2. We use the notation *ParG* to denote a 24dBi parabolic grid antenna, *Sec* to denote a 17dBi sector antenna, and *Can* to denote a 12dBi Cantenna. ParG, Sec, and Can have beam widths of 8^o, 90^o, and 60^o respectively. The Cantenna was locally made, according to the specifications given in [5].

2.2 Hardware and software setup

For most of our experiments, we used the Senao 2511CD plus ext2 PCMCIA cards based on the Prism2 chipset as the WiFi radio. This card has external connectors to connect to an external antenna via a pigtail connector. We also used the miniPCI version of the card in one of the experiments.

We used the Soekris (www.soekris.com) platform with Pebble Linux (www.nycwireless.net/pebble) to insert and use these cards. The Soekris platform is popular in several outdoor community networks. Further, the platform is ideal for us in terms of form-factor, as well as lower power consumption than conventional laptops or computers. We could simply use a 12V battery with a capacity of 32AH, with a voltage stabilizer circuit to power up the Soekris (see Table 1). This setup was sufficient to power the platform for about three days, which was convenient for our long-running experiments. Its physical security could also be more easily guaranteed (as compared to laptops) at the various sites. We used the net4521 model of Soekris for most of our experiments, and net4501 for one of the experiments. All of these use a removable compact flash for system installation and data storage.

We used the open-source HostAP driver v0.3.7 in our setup. For our experiments, we required the functionality of being able to learn per-packet information. Such per-packet information includes: (a) the received signal strength, (b) the silence value (energy level just before packet reception), (c) the modulation used for the packet, (d) MAC packet type and sub-type, (e) whether or not the CRC check succeeded, (f) MAC address information, (g) MAC sequence number information.

All of the above information is available at the driver-level from the hardware. The driver exports some information such as the signal strength or the silence value, to the user level, but not at a per-packet granularity. Hence our first effort involved driver modifications to export all the above per-packet information. We used the Linux proc file-system to export this information to the user level. We also pre-calibrated the mapping between the register value exported by the driver and the actual signal/noise value.

Apart from the above, we also needed the ability to turn-off MAC level ACKs to measure the packet error rate independent of any MAC retransmissions. For this, we included (a) an ability in the HostAP driver to change all outgoing packets to be MAC broadcast packets, and (b) at the receiving end, to change incoming MAC broadcast packets to look like unicast packets. We also included a provision to turn off/on this broadcast-unicast conversion at the user level, through a writable proc file.

We also tried to use Atheros chipset based NL-5354MP ARIES2 cards, with the latest madwifi-ng driver. However, our success with these cards on the outdoor links was poor, due to poor/unreliable transmit power settings.

2.3 Logistical Challenges

We have faced several logistical issues in setting up experiments on the eight links, many of which are in remote rural

locations. The primary issue has been that of lack of reliable power at most of the locations. We used battery-based alternate power supply for most of the sites (see Table 1). This limited the time for which we could leave a Soekris on at each site and collect measurements passively (to about 3 days).

Needless to say, none of remote sites have any other form of network connectivity. This further limits any dynamic software update or testing. If anything went wrong at a remote site and it became unreachable, in the least, a diagnosis of the problem is required in terms of: (a) battery down, or (b) physical disturbance of the supply, or (c) incorrect wireless settings. There was no personnel support at most sites even for such limited diagnosis, and no way to communicate with anyone at the site except physical travel. Even arranging a simple reboot was not easy.

The physical inaccessibility of some of the sites has also posed problems. Trips to and from some sites can take as much as an entire day from our location.

These logistical issues have significantly affected our measurement methodology, as outlined below.

2.4 Measurement Methodology

Metrics and parameter space

There are two broad metrics we consider during our measurements: (a) packet error rate, and (b) application throughput: both UDP as well as TCP.

The parameter space involves several dimensions:

- *Transmit power (txpower):* The primary dependence of the packet error rate is on the transmit power (txpower). Our calibration of the Prism2 cards showed that they have transmit power in the range of 0dBm to 20dBm. For each of the links, we experiment with up to four different transmit powers, subject to the received signal strength being in the range $-70dBm$ to $-90dBm$.

- *Transmit rate (txrate) or modulation:* The next known dependence of the packet error rate is on the transmit rate (txrate), or the modulation. 802.11b has four transmit rates: 1Mbps, 2Mbps, 5.5Mbps, and 11Mbps. We experiment with all four of these.

- *Packet size:* The direct effect of the received signal strength is on the bit-error-rate (BER). The packet size thus has an effect on the packet-error-rate (PER). We use three different packet sizes in our experiments: 100, 500, and 1400 bytes. These sizes represent the UDP payload. The first and second represent small and medium sized VoIP or video packets, while the last represents a packet size close to the ethernet MTU.

- *Packet inter-arrival duration:* To study time correlation of packet errors, we chose different packet inter-arrival durations. We used 2ms, 100ms, and 500ms packet inter-arrivals to study correlation at different granularities.

- *Broadcast vs. unicast:* As mentioned earlier, the driver can be optionally made to convert all outgoing packets to have the MAC broadcast address. This avoids MAC-level ACKs. We optionally turn on/off this feature for the various experiments.

- *Channel of operation:* The Prism2 cards support 11 channels of operation. We did not vary this parameter in most experiments since we did not expect any change in the behaviour of the link due to the channel of operation. For all links except A-F, we chose a channel of operation to avoid interference from other WiFi radios in the vicinity. For the A-F link, such a choice was not available, but it offers us a chance to look at the effect of external interference.

Experiments

In our measurements, we have explored the above parameter space extensively. For the rest of the paper, we define an *experiment (expt)* as follows. An expt is either a *UDP expt* or a *TCP expt*.

In a UDP expt, we choose a specific value for the transmit power, transmit rate, packet size. Within an expt, we first send UDP packets at the maximum possible rate, for 8 seconds. This is for UDP throughput measurement. We then send packets with: (a) 2ms inter-arrival for 8 sec, (b) 100ms inter-arrival for up to 15min, and then (c) 500ms inter-arrival for up to 30min. We excluded the 2ms case for scenarios where a single packet could not be sent within that duration (e.g. 1400 byte UDP packet at 1Mbps). In some expts, we had the 100ms and 500ms inter-arrival patterns only for 2min and 4min respectively. This was for non-technical reasons: we had to schedule the start/finish of a series of experiments on a link depending on personnel/transport availability. This however does not affect our overall conclusions below.

All UDP expts have the MAC broadcast feature turned on (i.e. no MAC-ACKs). Also, we chose to put the receiver in *monitor* mode (promiscuous). In this mode of operation, the hardware passes up errored packets and MAC management packets too to the driver. Further, it also allows us to look for any interference packets from other WiFi sources.

In a TCP expt, we choose a specific value for the transmit power and transmit rate. The packet size is the default ethernet MTU (1500 bytes). Data is transferred over a TCP connection for 25 seconds. We try the two cases of with and without MAC-level ACKs in separate TCP expts.

For all the experiments, we used the *pseudo-ibss* mode of operation of the HostAP driver. In this mode, there are no management packets needed for link formation. This was necessary in the case where the receiver was in monitor mode.

The data collection proceeds in three phases. (1) In the first phase, the two ends of the link come up in with default settings, and form the WiFi link. One end of the link then determines which expt (combination of the above set of parameters) to conduct next. It communicates this to the other link using a control TCP connection. (2) The two ends of the link then perform the expt and record the results. (3) Finally, the two ends of the link store their data. In the experiments where one of the ends was able to connect (through a LAN) to a laptop/PC, we stored the data there. In others, we had to use the Soekris's flash memory for such storage. In some cases, we had to compress the data to accommodate it within the limited flash capacity available.

We now present the various measurement results.

3. ANALYSIS OF PACKET-ERROR RATE

The primary characteristic of a link is its packet error rate. This section explores this aspect in depth. In Sec. 3.1, we look at the dependence on the Signal-to-Noise Ratio (SNR), transmit rate, and the packet size. Then in Sec. 3.2 we look at the possible time correlation of packet errors. Sec. 3.3 discusses the effect of weather conditions.

3.1 Dependence on SNR

To study the effect of SNR on the error rate, we proceed as follows. For each of the eight links, we consider the set of UDP expts. These expts are for different values of txpower, txrate, and packet size. From each expt, we compute the average error rate across the 2ms, 100ms, and 500ms inter-arrivals. We specifically *do not* consider the error rate observed during the UDP throughput (full-rate) measurement, the reason for which will become apparent in Sec. 4. The average SNR is computed from the per-packet readings exported by the driver.

Fig. 3 and Fig. 4 show the error rate as a function of the SNR across the dimensions of txrate and packet size respectively. We had first plotted these for all of the eight links, but observed that two links were outliers. These were the links A-B and A-F. As we elaborate in Sec. 3.2, A-B is likely affected by a non-WiFi interference source, and A-F has interfering WiFi sources. We consider these two links as special cases and exclude these from Fig. 3 and Fig. 4.

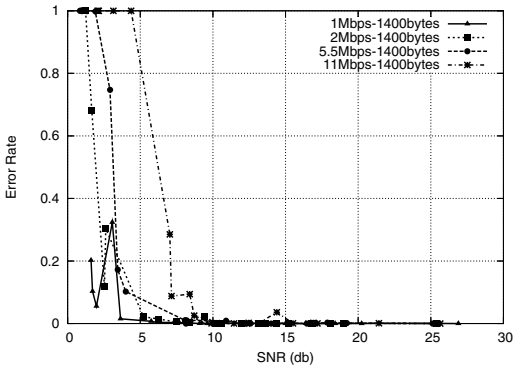

Figure 3: Error Rate vs. SNR at 1400 bytes

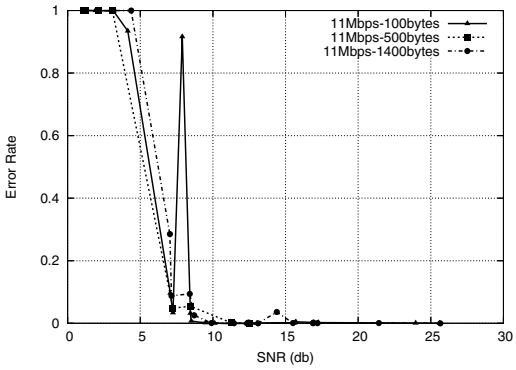

Figure 4: Error Rate vs. SNR at 11Mbps

Effect of txrate: In Fig. 3, we see that the error rate falls sharply with increasing SNR for each of the txrates. The threshold at which the error rate falls is higher for higher txrates. The value of SNR at which loss rate falls below 1% is about 6db at 1Mbps as opposed to 10db at 11Mbps. The value of SNR at which loss rate reaches 100% is about 2db for 1Mbps as opposed to 4db for 11Mbps. The jump from 100% to 1% happens within 4 to 6dB for all the txrates.

There is a single outlier in the graph, corresponding to the 1Mbps txrate plot. On examination of this data point, we found that the SNR was about 3dB. In comparison with other data points in this range, we found that this data point had a lower noise floor. The signal strength itself was low too, about $-88dBm$. This is close to the sensitivity limit of the Senao cards for the 1Mbps transmission: $-89dBm$. Hence this data point has high error rate.

Apart from the single outlier, the plot is close to the expected theoretical behaviour. We also verified this by performing controlled experiments in indoor settings with RF cables and step attenuators.

Effect of packet size: The three plots in Fig. 4 are for different packet sizes, all at the 11Mbps txrate. The graph shows that at a given txrate, the variation with packet size is not very significant. In fact in the figure, the 500-byte plot seems to have a better error rate than the 100-byte plot. This is due to lack of sufficient data points in the steep region of the plot. The difference is noticeable between packet sizes of 100 and 1400 bytes. The value of SNR at which loss rate falls below 1% is about 8db for 100 byte packets as opposed to 10db for 1400 byte packets. The value of SNR at which loss rate reaches 100% is about 3db for 100 byte packets as opposed to 4db for 1400 byte packets.

Once again, there is a single outlier point (100 bytes, SNR=8dB). On examination, we find this also to be a case where both the signal strength as well as the noise readings are low. The signal strength was $-87dBm$, below the specified card sensitivity for 11Mbps reception.

Readings on a specific link: The above graphs looks at data across different links. To examine the behaviour within a link closely, we now plot the variation in error rate as a function of packet size, txrate, and txpower. Fig. 5 shows such a plot for the A-C link. The behaviour for other links was similar.

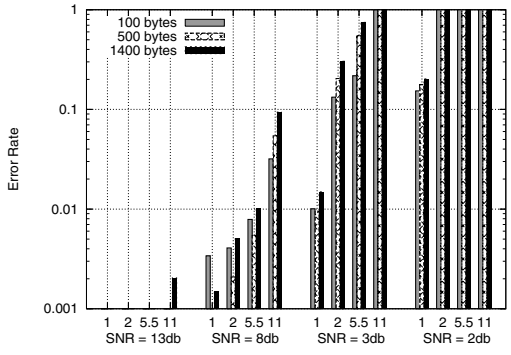

Figure 5: Error rate variation on link A-C

As is shown in the figure, as SNR decreases, the loss rate increases. For error rates below 1%, the dependence of er-

ror rate on the packet size is not fully consistent. This is because, we observed that many of the errors are caused by a hardware quirk, as explained below. The effect of the hardware quirk becomes negligible when the SNR is low. So, in these cases, as packet size increases, the loss rate increases. In all cases, as the transmit rate increases, the loss rate increases.

A hardware quirk: We observed many instances where even at high SNR, the error rate was not really zero: there were some packet losses. We examined the receiver log for these expts and found that many of the losses were actually packets received with CRC errors. Further, we saw that the received signal strength of these specific packets was significantly lower (15dB or more at times) than most of the others.

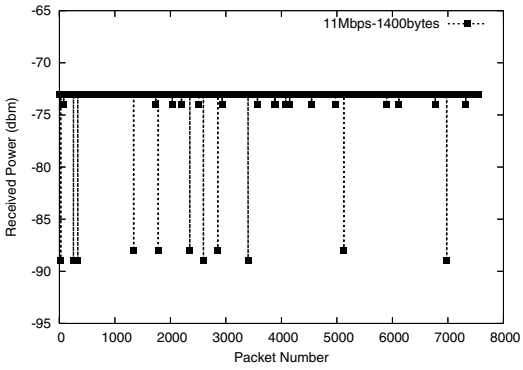

Figure 6: Signal strength variation in a long run

We repeated the experiment in a controlled setting, with an RF cable and step-attenuators connected between the transmitter and the receiver cards. We observed that such signal strength drops happened even in this case. Fig. 6 shows a plot of the signal strength for a sequence of packets.

We believe that this is a hardware quirk with the Senao cards. We also repeated the experiment with a Prism2 Senao transmitter and an Atheros chipset based receiver. In this case too, we observed similar sudden drops in the received signal strength. Hence it is likely that the quirk is at the Senao transmitter, and not the receiver. We have observed this quirk over a range of txpowers, and with different cards of the same model.

Implications: The results in Fig. 3 and Fig. 4 are in sharp contrast with what is known for outdoor WiFi networks [6]. The link abstraction *does* hold. There is a definite and predictable relationship between the SNR and the packet error rate. The links can potentially be planned and transmit powers provisioned such that the error rates are low.

Given the lack of a link abstraction in an unplanned style of community network, researchers have focused on alternate approaches to routing [8]. However, in contrast, in long-distance mesh networks, the above results indicate that it is unlikely that anything sophisticated is required as a routing protocol.

Fig. 5 also suggests how to do transmit rate adaptation in such long-distance links. The adaptation can simply be based on the SNR value of the link. For example, when using 1400 byte packets, if the SNR of the link exceeds 10dB, 11Mbps transmit rate can be used to achieve better throughputs since loss rate is very low. Below an SNR of 10dB, we enter the steep region of the error rate for the 11Mbps txrate. In this region, any minor 1-2dB variation can result in a drastic increase in the error rate. We have observed in our receiver logs both indoors as well as outdoors that variations of 1-2dB are common across multiple packets. Hence below an SNR of 10dB, it is not advisable to use the 11Mbps txrate.

We revisit rate adaptation in Sec. 4. We now look at the time correlation of packet errors on the various links.

3.2 Time Correlation of Packet Errors

As explained in Sec. 2.4, we have measurements with three different packet inter-arrival durations: 2ms, 100ms, and 500ms, for various durations, up to a maximum of 30min. We use the *Allan deviation* metric to explore time correlation of packet errors, i.e. to see if the packet errors are uniformly spread or are bursty. Given a series of values $x_i, i = 1..N$, the Allan deviation is defined as $\sqrt{\frac{\sum_{i=2}^{N}(x_i - x_{i-1})^2}{2N}}$. The study in [6] also uses the same metric. The values x_i are the error rates averaged over a chosen time interval T. We calculate this metric for various values of T. The Allan deviation has a high value near the characteristic error burst length.

We compare the metric against a case where the sequence x_i is computed from an artificially generated packet error sequence. The artificial sequence has packet errors independent of time, but with the same probability as the error rate of our measured packet error sequence. We call this the uniform (over time) error rate sequence.

The averaging interval T is varied over 10ms, 20ms, 50ms, and 100ms for the data with 2ms packet inter-arrival. It is varied over 200ms and 500ms for the data with 100ms packet inter-arrival; and over 1s, 2s, 5s, 10s, 60s, for the data with 500ms packet inter-arrival.

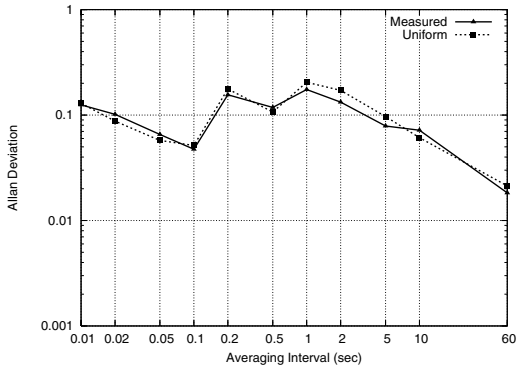

Figure 7: Allan deviation: A-C link

Fig. 7 shows the Allan deviation metric for various values of T. This data is shown for 1400 byte packets sent at the 11Mbps rate, for a received SNR of about 9dB. We chose this SNR since it has an error rate of about 10%: neither close to 0% nor to 100%. In the graph, the metric for the uniform error-rate sequence is also shown alongside for comparison.

We see that the time correlation of error rate is insignificant. Note that unlike in [6], the Allan deviation metric does not necessarily decrease uniformly with increase in T. This

is because the sample size does not increase with different values of T. We observed similar behaviour with other links as well, which did not have any external interference.

Variations at larger time scales: To study the time variation of packet errors at time scales of several minutes to a few hours, we collected data across 24 hours on the A-C link. This was arranged as a sequence of 30min expt runs, with 11Mbps txrate, 1400 byte packets, and 100ms packet inter-arrivals. We performed such a sequence of runs under two conditions: (a) SNR above the threshold as given in Fig. 3, and (b) SNR in the steep region of Fig. 3.

In either case, we observed that across the set of 30min runs, the signal strength showed variations of about 1-2dB, but not more. Such variation did not have much of an effect on the packet error rate in case (a) above. The error-rate remained within 0.1% across the various 30min runs.

The 1-2dB had a significant effect on the packet error rate in case (b) above (steep region of the curve). The error rate varied from as low as 1.5% in some 30min runs, to as high as 45% in others.

Implications: For links without external interference, we have found that the error rate is independent of time. This is useful input for any simulation modeling or protocol study.

3.3 Effect of weather conditions

As mentioned above, we had conducted close to 2 days of repeated 30min expt runs on the A-C link. For the first day, we had set the transmit power such that the received signal strength was $-75dBm$. The second day, we set the transmit power such that the received signal strength was about $-85dBm$. At the start of either set, we had no abnormal weather in terms of rain or fog. But during each 24 hour run, there was significant rain as well as fog. This weather condition was aseasonal and our study of the effect of rain/fog on the link was really unplanned.

During each day, there was at least several hours of heavy downpour and thunderstorm. There was fog during the night and early morning. The fog on the second night was in fact quite thick, with visibility reduced to a few metres only. These anecdotal points were observed at site-A. We do not attempt to quantify the rain or fog any further since there is no meaningful way to do this. Weather reports for the city/region are unlikely to be applicable to the specific area between the tower sites A and C.

Water is known to be a good absorbent of 2.4GHz radiation. Given the heavy fog and hours of rain, we had expected to see significant effect on the link performance. To our surprise, we found none. As already observed in Sec. 3.2, we found not more than about 1-2dB variation in the received signal strength. Such variation is present even in our expts during clear weather. There is no significant effect on the packet error rate as well.

Implications: That the effect of weather is not significant is good news for those deploying such networks. However the result above should be taken with a pinch of salt, since it is only on one link, and at a particular geographic location.

4. THROUGHPUT ANALYSIS

In this section, we present measurements of UDP as well as TCP throughput on the various links.

4.1 UDP Saturation Throughput

Fig. 3 defines the relation between error rate and a given SNR. From this figure, it follows that at higher SNRs, the error rate is close to zero and therefore the UDP saturation throughput (packets sent as fast as possible) should be close to the theoretical maximum. A quick calculation considering the effect of backoffs and the DIFS interval, reveals that the maximum throughput that can be achieved for 1400 byte packets on these links is about 0.92, 1.79, 4.42, and 7.63 Mbps for transmit rates of 1, 2, 5.5 and 11Mbps respectively. This calculation does not include MAC level ACKs since UDP packets were sent as MAC broadcast.

Fig. 8 shows the UDP throughput observed on the various links as a function of SNR for 1400 byte packets sent at different transmit rates. At transmit rates of 1, 2 and 5.5Mbps, the UDP throughput is close to the theoretical maximum, at high SNRs. For 11Mbps, though the error rate is low at high SNR, the UDP throughput is much lower than the theoretical maximum. This is because other bottlenecks come into play at 11Mbps as we describe below.

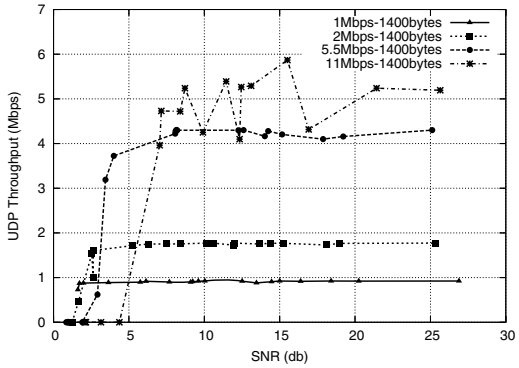

Figure 8: UDP throughput vs. SNR at 1400 bytes

Other system bottlenecks: We repeated the 11Mbps throughput measurement in an indoor setting too, and observed similar numbers. We suspect that the reason is the following. Neither the Prism2 Senao cards, nor the HostAP driver, has support for DMA (Direct Memory Access). Hence all data transfer has to happen with the intervention of the Soekris's CPU. The net4521 model we used for most of our expts has only a 133 Mhz 486 class processor. Note that the CPU is also involved in the per-packet system calls and the user-to-kernel space memory copies. Since the scheduling of these operations is also imperfect, we see fluctuations in the throughput in the graph. We have observed the above bottleneck in indoor settings too.

We also measured the UDP throughput at different packet sizes. Again, at lower transmit rates, the UDP throughput for different packet sizes is close to the theoretical maximum. However at the 11Mbps transmit rate, a packet size of 100 bytes (UDP payload) shows a throughput of only 0.77Mbps. Whereas, the theoretical maximum for this case is more than twice as much: 1.53Mbps. A throughput of 1.53Mbps does not seem very high to hit any system bottlenecks, but it turns out that this is not the case.

A closer examination of the above expt run reveals yet another bottleneck. We looked at the receiver side log and found that several MAC sequence numbers were missing! The loss could not have been due to wireless packet errors

since the SNR was very high. So the receiver side was likely losing packets between the hardware and the driver. This was probably because the rate of interrupt generation by the hardware to clear its buffer was not (or could not be) fast enough.

Implications: Given that the Soekris platform is used commonly for outdoor mesh networks (not necessarily long-distance links), the above bottlenecks are significant. The measurements with the smaller packet size have implications for VoIP traffic. It is well known that 802.11b is not really efficient for small VoIP packets. Our measurements point out that there are system bottlenecks other than the wireless interface itself.

Another implication of the plots in Fig. 8 is with respect to rate adaptation. The steep increase in application throughput with increasing SNR re-emphasize the SNR-based rate adaptation mechanism discussed in Sec. 3.1.

4.2 TCP Throughput

Our primary interest in TCP throughput on these long-distance links is to study the effect of any MAC-level ACK timeouts.

In our prior experience with the various long-distance links, we had used Cisco a350 WiFi bridges. These bridges allow us to configure the link distance. The hardware presumably uses this to compute and adjust the ACK timeout. We have observed that whenever the link length was underestimated (in the above configuration) by a value more than 5km, it failed to form. This implied that the ACK timeout was approximately the round-trip propagation time in a 5km link: about $33\mu s$ (speed of light).

Given this data point, we had expected to see ACK timeouts on links larger than 5km. We were however surprised to see ACK timeouts only on the 37km A-G link (round trip of $250\mu s$) and no other. The 802.11 specification does not tell what the ACK timeout value should be. But the Prism2 cards we used showed such timeouts only on our longest link. On this link, the two WiFi radios failed to associate with one another in AP-client or in ad-hoc mode. This was also likely due to a timeout. Because of this, our use of the *pseudo-ibss* mode was imperative for experimentation on this link.

We now look at the effect of such timeouts on TCP performance.

TCP performance on the 37km link: The TCP throughput on the A-G link at the 11Mbps txrate, at an SNR of about 16dB, was found to be 1.9Mbps, with the use of MAC broadcast. With MAC unicast, the throughput fell to as low as 0.5Mbps!

To explain this reduction in throughput, we looked at the driver-level log at the receiver side. We observed several cases where the inter-packet gap was as high as 10ms to 20ms. Note that the transmission of an MTU sized packet at 11Mbps should take about 1.5ms, including all the MAC overheads.

There are likely two reasons for these huge inter-packet gaps. First is the presence of ACK timeouts. On an ACK timeout, the MAC sender retransmits until an ACK is received, or until a retry limit is exceeded. Such repeated retransmissions immediately cut the throughput by a factor equal to the retry limit. Apart from this, ACK timeouts also have the effect of increasing the MAC contention window exponentially. The average backoff when CW is 1024 is $512 \times 20\mu s \simeq 10ms$.

Another reason for the inter-packet gaps could be due to collisions between the TCP data and the TCP ack packets. According to the 802.11 specifications [3], the slot-time during the contention period is supposed to include the maximum expected round-trip time in the system. Otherwise the collision probability increases with higher round-trip time. With such collisions too, the CW increases.

In spite of the ACK timeouts on the A-G link, we still obtain some throughput with MAC unicast (0.5Mbps) since packets are delivered correctly at the receiver.

Another hardware quirk: Another hardware quirk we observed during our experimentation was that in many cases the driver (and hence the TCP layer) was seeing duplicate packets, with the same MAC sequence number! Ideally, the hardware should have suppressed such duplicates based on the MAC sequence number, but it did not.

Performance on other links: Apart from the A-G link, we saw no ACK timeouts for any of the other links, not even on the next longest link A-F, of 23km. The inter-packet gap in the driver logs for these links was always small and showed little variation.

Like the UDP throughput measurements, we also measured the TCP throughput for different transmit rates and transmit powers. Unlike UDP, the Soekris does not seem to be a bottleneck and the throughputs achieved are closer to the theoretical values, for all the links (except those with external interference). This is due to two reasons: (a) the maximum throughput for TCP is lower than for UDP (due to TCP ACK overheads), and (b) the buffering and flow-control mechanism between the kernel and the user space is better for TCP than for UDP.

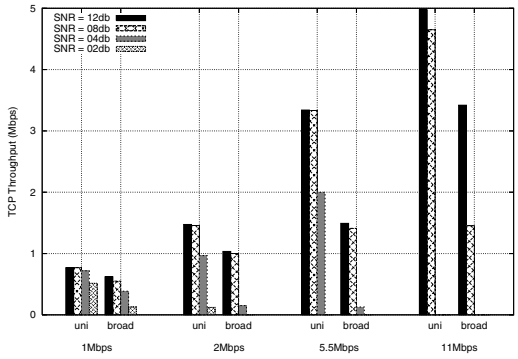

Figure 9: TCP throughput: C-D link

Fig. 9 shows the measured throughput on the C-D link for different transmit rates and powers. The behaviour on other links was similar. *Uni* in the plot refers to the case where the packets were transmitted as unicast over the wireless link. *Broad* refers to the case where the packets were transmitted as MAC broadcast (i.e. no MAC retransmissions).

In the unicast case, the MAC retransmissions are capable of recovering from any stray packet errors at the higher SNR values (some packet errors are seen even at high SNR values due to the sudden txpower drops, as explained in Sec. 3.1). At low SNR, some packet errors are seen by TCP too. For instance, in the 2Mbps txrate case, the number of duplicate SACKs seen at the TCP layer was 5 (obtained from tcpdump log) at the lowest SNR. At other SNR values, TCP saw no packet losses.

Implications: We would have liked to be able to change the ACK timeout value, however the Prism2 cards do not offer such flexibility. Hence we used MAC broadcast packets to avoid any MAC-level ACKs altogether. This of course means that TCP sees any wireless packet losses, and its performance drops. Fig. 9 quantifies this drop in performance. To avoid the performance drop, one could possibly implement a selective ACK scheme at the driver level, so that wireless losses are still hidden from TCP. We did not implement such a mechanism however.

5. EFFECT OF INTERFERENCE

In this section, we first examine the effect of external interference, in Sec. 5.1. We define external interference to be those from non-WiFi sources, or from WiFi sources in neighbouring deployments. We then look at the possibility of interference across adjacent long-distance links within our own deployment, in Sec. 5.2.

5.1 External Interference

We have observed external interference in two of the links. We examine the nature of the interference below.

Interference in the A-B link

The A-B link shows high error rate even at high SNR values. We however did not find any other WiFi sources in the vicinity: neither using the driver-captured log, nor using an active scan. On analyzing using the Allan deviation metric as above, we observed significant correlations below 50ms. A closer look at the data surprisingly revealed rather periodic patterns of errors. This is shown in Fig. 10 for a case with txrate of 11Mbps, packet size of 1400 bytes, and packet inter-arrival of 2ms. The received SNR was about 11dB.

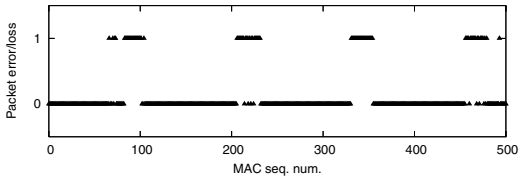

Figure 10: Packet error/loss pattern in A-B link

The burst length of the packet errors spans about 50ms. The periodic pattern suggests that the packet errors are very likely *not* due to multipath. We observed the same pattern in other expts too, on link A-B. We suspect that the interference is from a non-WiFi source near the tower in site-B.

Interference in the A-F link

The A-F link has high error rate due to significant presence of other WiFi sources in the vicinity at site-F. We confirmed this by looking at our driver-exported log, as well as by performing an active scan at F. The active scan showed as many as a dozen different WiFi sources.

Like for the A-C link, we had a series 30min runs for the A-F link. This ran for a duration of 2 days. The error rate averaged over the 30min durations is shown in Fig. 11. The SNR was about 18-20dB for the various runs.

We see that there are lengthy durations of high packet error rates. These error rates are as high as 50%. Expt 20 (about 1am) shows the first large spike in the error rate.

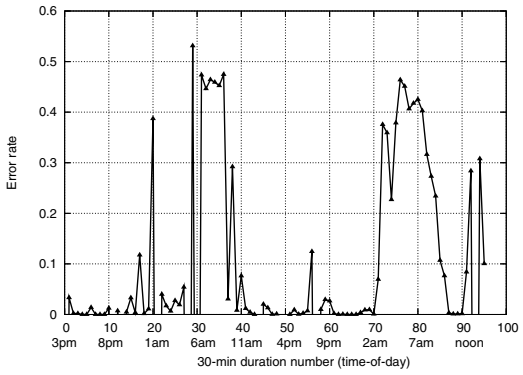

Figure 11: Error rates over various 30min runs: link A-F

We found it strange that other interference sources were operating at this time, but the pattern repeated the next day too. It is likely that the interfering WiFi links are used for late-night data backups in addition to regular Internet access. The other WiFi sources remain active through most of the working day, up to about 4pm in the evening.

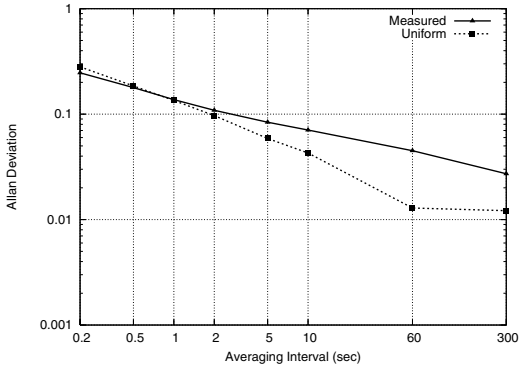

Figure 12: Allan deviation of 30min run # 83: link A-F

For the 30min durations which had errors due to interference, we found that the time correlation of error rate was high at the 10-60 second time scale. For instance, Fig. 12 shows the Allan deviation metric for the 30min duration number 83, which had an error rate of 27%. This is not surprising given that the packet errors are due to other WiFi sources.

Implications

For a long-distance link with directional antennae, interference from any other WiFi source is a classic case of the *hidden node* problem. For instance, in our A-F link, WiFi sources near F are hidden from the transmitter at A. Of course this can be addressed partially by the use of RTS/CTS. However, RTS/CTS has two issues. The first is that it would add to the overhead in the protocol. The second is that it may not necessarily work if the two links are operating in adjacent, overlapping channels (e.g. channels 1 and 3). For example, suppose another WiFi source W were operating in channel-3 near site-F, and A-F were operating in channel-1. Then W may not hear the RTS/CTS exchange between A

and F, and may still end-up interfering with F. It is common for many deployments to be configured in channels other than the three non-overlapping channels 1, 6, and 11. In such a scenario, the RTS/CTS would not necessarily solve the issue.

The extent of the effect seen due to interference on the long-distance links is rather startling. Fig. 11 indicates that error rates can be as high as 50%. Thus, unlike in [6], we see a direct cause-effect relation between interference and packet error rate. Further, unlike in urban community networks, for a long-distance link setup with significant infrastructure investment (antenna towers) to operate with 50% error rate seems rather wasteful.

Naturally, at such high error rates, application performance (TCP/video) will be affected critically. Networks with long-distance links are already being laid out, e.g. the Ashwini project [4], and intended to be used for applications such as high quality two-way video. Video performance is affected significantly even with error rates as low as 1%.

Given this, we strongly feel that dealing with the issue of RF "pollution" needs immediate attention. In technical terms, this means that we should have mechanisms to detect and diagnose cases of interference: both within a provider network as well as across deployments. In non-technical terms this implies that some legal or semi-legal mechanism is required to control mutual interference across deployments.

5.2 Inter-link interference

We now look at an important aspect of the operation of mesh networks with long-distance links. It is common for such networks to have multiple adjacent links setup atop the same tower/building. We now examine whether such adjacent links can operate independently, without mutual interference. If so, under what conditions?

It is known that it is not necessarily possible to operate adjacent links on the same channel [13], even given the directional antenna used for each link: the links have mutual interference and end up sharing the channel. Also, researchers have earlier reported the inability to operate two wireless *interfaces* on the same desktop, without mutual interference, even in the so-called non-interfering channels [10]. In this case, the two radios are within a few centimetres of each other. However, empirical experience in WLAN indoor deployments is that two radios placed "reasonably" apart from one another can operate independent of one another in non-overlapping channels.

We are now left with the question: when two (directional) antennae are mounted atop the same antenna tower (or tall building), is it possible to operate the respective two adjacent long-distance links independently on different channels? To answer this question, we undertook a series of experiments.

Setup to study inter-link interference

We used a setting where two parabolic grid antennae are mounted atop an antenna tower. Although the tower itself is 40m tall, we setup the antennae at a height of 20m only. This is merely for convenience, and does not affect our results below. The setup is shown in Fig. 13. The antennae were on different outer sides of the tower, with the angle between them about, 90^o, and the distance about 1m.

Each of the antennae was connected to a radio: Prism2 card inserted into a Soekris box. We had one of the radios

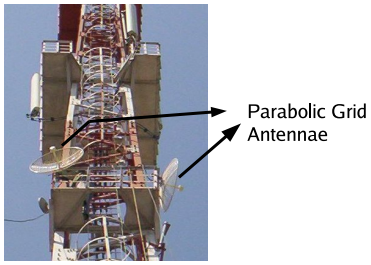

Figure 13: Setup for exploring inter-link interference

act as an Access Point (AP), sending beacons every 100ms, and set at a txpower of 20dBm. The other card was in monitor mode, trying to sniff the beacons from the AP. The AP radio was maintained in channel-1. We changed the sniffer's channel from 1 through 11, recording our observations about the beacons sniffed in each channel.

We tried four different configurations: (1) Both radios kept atop the tower, near their respective antennae (2) The AP radio kept atop the tower, and the sniffer at the base, (3) Both radios at the base, about 1m away from each other, and (4) Both radios at the base, about 5m away from each other. In all configurations, each radio was connected to its respective antenna via an RF cable of suitable length.

Results

We compute the average signal strength of the beacons sniffed, over a duration of 10 seconds, in each of the channels. We plot this against the channel number, in Fig. 14.

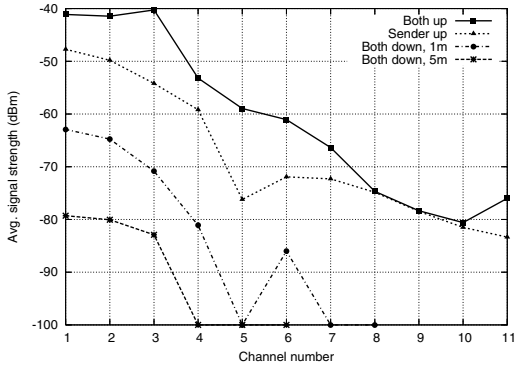

Figure 14: Sig. str. of beacons sniffed at various channels

We make two observations. First, when both radios are up, even channels 1 and 11 cannot be considered mutually non-interfering! This is true even when just the AP radio is up the tower. Note that our transmit power cannot really be considered very high. In fact there are several vendor products which come with up to 400mW (26dBm) transmit power specification for long distance links.

Our second observation is that when the radios are placed at the tower base, interference is reduced significantly. In such a scenario, it is possible to operate channels 1 and 6 independently, provided there is sufficient physical separation too.

The plots are more or less uniform: decreasing received signal strength with increasing channel separation, but for a few exceptions. For instance, when both radios were down and separated by 1m, the received signal strength at channel 6 was higher than that in channel 5. On examination of the data, we found that this was because in channel 6, there was one stray beacon received at a signal strength of $-86dBm$. Whereas in channel-5, no beacons were received. Other exceptions in the graph are also due to similar reasons.

The presence of interference even across large channel gaps can be explained as follows. When two radios (Prism2 Senao cards) are as near as about 1m from one another, there is significant leakage. The Senao cards have two external antenna connectors, of which only one is connected via a pigtail to the external antenna. It is likely that part of the leakage is via the open connector. Leakage is also likely from the pigtail RF connectors. We have observed this leakage at short distances even when operating indoors without any antennae.

This leakage has a significant effect when one radio is near the other radio's antenna. The *near-field* of an antenna is calculated as $d_f = 2 \times D^2/\lambda$ where D is the largest dimension of the antenna. For the parabolic grid and sector antennae in our measurements, the near-field is about 16m. Within the near-field of the antenna, directionality of the radiation pattern is not accurate.

When the AP-radio is atop the tower, it is in the near-field of the sniffer's antenna. The sniffer's antenna is thus able to get significant radiation from the leakage of the AP-radio. This effect is only magnified when both radios are up the tower. In this case we also have the sniffer radio in the near-field of the AP's antenna.

In Fig. 14, the leakage in the near field of the antenna is strong enough to be seen across a channel gap of 10. When both radios are at the tower base, each is outside the near-field of the other radio's antenna. And although the the two antennae are within the near-field of each other, the leakage component is weak enough to avoid interference when there is a channel gap of 5.

Implications

It is common in deployments to setup multiple radios atop the same tower. From the above, it is clear that it is not only necessary to separate the multiple radios on a tower from one another, but also from each others' antenna. One easy way to achieve this is to have long (20m or more) RF cables coming from the antenna which is atop the tower.

Of course, the use of RF cables has its own set of issues: (a) it adds to the link loss, typically about 3-4dB per 100ft, (b) they add to the system cost, (c) they are clumsy to handle and maintain. In one instance we also found that one of our antennae had its connector damaged because the RF cable dangling from it was swaying in the wind. So one has to take precautions such as the use of ties to secure the RF cable and prevent it from oscillating.

6. AVOIDING MISERY

Here we document some of the simple things which we wish we had known to begin with, but had to learn the hard way. We hope that this list will help others deploying or testing such long-distance links in avoiding the same pitfalls.

Tricky txpower/channel settings: It is rather tricky to set the txpower of the Prism2 cards reliably. The driver (v0.3.7) does not support txpower setting by default, this requires a patch. Apart from this, the *order* in which the wireless parameters are set seems to matter. We observed that the txpower as well channel settings sometimes reset to default values when changing the *mode* of the card. For instance, if txpower were set before changing the card to pseudo-ibss mode, it is not retained after the mode change. We had to discard two days worth of data because of this error.

Reliably setting txpower: Another aspect we have noticed is that the driver-exported transmit power setting (through iwconfig) is not always reliable. It is best to verify the transmit power setting by actually reading/writing the corresponding register: #31 (address 62) for the Prism2 cards.

Cannot force association: Sometimes when we set the remote end of a link to come up and look for a particular essid, we ran into problems. If the radio did not find the specified essid, after a timeout, it associates with any available AP in the vicinity. We faced this issue to a significant extent in the A-F link. We finally had to make both radios come up with WDS (Wireless Distribution System) mode for this link to be established reliably.

Beware of interference: We once had a scenario where we had set the radio at a remote site to use channel-6. We sought to conduct an interference-free experiment. Before setting up the experiment, we had checked using "iwlist wlan0 scan" (active scan) that there was no other WiFi source nearby in channel-6. But after 2 days of experimentation, we realized that there was a WiFi source there on channel-6. But it had not shown up during the active scan. We had to redo those set of measurements. The lesson we learnt from this was that it is better to check for interfering WiFi sources by putting the card in monitor mode, rather than relying upon an active scan.

Interference trouble again: Another unexpected trouble we ran into in our experiment with the A-F link was that the Soekris ran out of space during the series of experiments. We had determined that there should have been more than adequate space given our prior measurements on other links. But we had not accounted for packets from other WiFi sources showing up in our driver log.

Kernel UDP buffer: In our series of experiments, we had programmed the 2ms packet inter-arrival expt to start right after the UDP throughput measurement. This turned out to be a big flaw. During the UDP throughput expt, the sender's kernel buffer gets filled. So even though the user program at the sender switches to a 2ms packet inter-arrival, the kernel continues to send at full rate until its buffer clears. We had to discard yet another 2 days worth of data because of this issue.

RF leakage during calibration: In indoor settings, one has to be careful about the leakage from the various connectors. At short distances, such leakage can result in significant aberrations in any controlled calibration efforts.

7. CONCLUSION

In this paper, we have presented a measurement study of long-distance 802.11b links. To our knowledge, ours is the first such study. Our main conclusion is that *long-distance links can be planned well for predictable performance.* However, any interference within the network or from neighbouring deployments can cause drastic reductions in performance. Interference can occur between adjacent links even in the so-called non-interfering channels. This means that it

is important to develop the knowledge-base for deployment planning and also to diagnose any performance problems after deployment. Living with RF pollution may not be a viable option in these settings. This may also mean that non-technical (legal/semi-legal) support is necessary to promote commercial deployments. Note that none of the four example deployments presented in Sec. 1 are really commercial, for-profit deployments. In fact, the three deployments in India are government funded initiatives.

Although we have presented measurements only on eight links, we feel that our results are meaningful. The results are quite consistent across the links and any aberrations are explainable.

Our study has focused on 802.11b links. Since 802.11g operates in the same frequency band (2.4 GHz), and since 802.11a operates in a higher frequency (5 GHz), we can conjecture that many of our conclusions are likely to be true for 802.11g and 802.11a as well. Of course, actual measurements are required to confirm this.

Outdoor WiFi-based community networks has received a lot of attention in the recent past. The same is necessary for long-distance WiFi deployments too, especially given its potential to provide truly pervasive networking. We believe that our measurement study is a significant step in this regard, and hope that the results will be useful in future deployments and protocol studies.

Acknowledgment

We thank everyone who helped us at various levels with the outdoor experiments.

8. REFERENCES

[1] Akshaya: A Kerala state initiative for creating powerful social and economic e-networks. `http://www.akshaya.net/`.

[2] DjurslandS.net: The story of a project to support the weak IT infrastructure in an low populated area of Denmark. `http://djurslands.net/biblioteket/international/djurslands_net_english_presentation.ppt`.

[3] IEEE P802.11, The Working Group for Wireless LANs. `http://grouper.ieee.org/groups/802/11/`.

[4] Project Ashwini: Virtual Delivery Services. `http://www.byrrajufoundation.org/ashwini_home.htm`.

[5] Radio laboratory handbook. `http://wireless.ictp.trieste.it/handbook/index.html`, 2004.

[6] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level Measurements from an 802.11b Mesh Network. In *SIGCOMM*, Aug 2004.

[7] Pravin Bhagwat, Bhaskaran Raman, and Dheeraj Sanghi. Turning 802.11 Inside-Out. In *HotNets-II*, Nov 2003.

[8] Sanjit Biswas and Robert Morris. Opportunistic Routing in Multi-Hop Wireless Networks. In *SIGCOMM*, Aug 2005.

[9] Eric Brewer, Michael Demmer, Bowei Du, Kevin Fall, Melissa Ho, Matthew Kam, Sergiu Nedevschi, Joyojeet Pal, Rabin Patra, and Sonesh Surana. The Case for Technology for Developing Regions. *IEEE Computer*, 38(6):25–38, June 2005.

[10] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *MOBICOM*, Sep 2004.

[11] Tristan Henderson, David Kotz, and Ilya Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In *MOBICOM*, Sep 2004.

[12] Bhaskaran Raman and Kameswari Chebrolu. Revisiting MAC Design for an 802.11-based Mesh Network. In *HotNets-III*, Nov 2004.

[13] Bhaskaran Raman and Kameswari Chebrolu. Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks. In *11th Annual International Conference on Mobile Computing and Networking paper (MOBICOM)*, Aug/Sep 2005.