Creating Wireless Multi-hop Topologies on Space-Constrained Indoor Testbeds Through Noise Injection

Sanjit Krishnan Kaul, Marco Gruteser, and Ivan Seskar
WINLAB /Electrical and
Computer Engineering
Rutgers University
Piscataway, NJ 08904
Email: {sanjit,gruteser,seskar}@winlab.rutgers.edu

***Abstract*— To evaluate routing protocols on a controlled indoor wireless testbed, the radio range must be compressed so that larger multi-hop topologies can be mapped into a laboratory-size area. We propose noise injection as a more flexible option than hardware attenuation and consider methods for mapping a free-space propagation environment onto the testbed. While more realistic propagation models are also needed, experiments in a free-space propagation environments are useful for protocol testing and analyzing differences between simulation and experimental results. Our experimental results show that additive white Gaussian noise effectively reduces the radio range, without the need for hardware attenuation and careful shielding of wireless cards. By selecting node positions through an automated procedure, we were able to create a 4-node/3-hop string topology and a random partially connected 8-node topology in a 8m by 8m area with off-the-shelf IEEE 802.11 hardware.**

I. Introduction

Wireless networking researchers increasingly emphasize experimental analysis of wireless systems and experimental validation of simulation models. Unfortunately, controlled experiments with wireless LAN systems are difficult and costly to conduct, primarily because experiment setups often require large distances between senders, receivers, and interferer. In particular, multi-hop topologies, where the network is not fully connected, require that the distance between some transmitter-receiver pairs is large enough to prevent direct communication. For example, off-the-shelf IEEE 802.11 radios have a transmission range of several hundred meters in a controlled line-of-sight propagation setting. While multi-hop wired testbeds can be set up in a small laboratory space, creating larger wireless multi-hop topologies, say with 5 hops, for experimentation with routing protocols would easily require a laboratory space of one kilometer length. Therefore, creating such topologies is usually only possible in outdoor settings with an uncontrollable radio propagation environment, which is affected by weather patterns, foliage, moving obstructions, among other factors.

To facilitate controlled experimentation with larger wireless networks, researchers have frequently resorted to attenuating the transmission power [1]. While some off-the-shelve hardware provides transmission power control, the minimum setting is usually insufficient to create large network topologies in a laboratory space.[1] Thus hardware attenuation is necessary, but inserting hardware attenuators requires manual labor and careful shielding for all nodes. Unless expensive variable attenuators are used, it also severely restricts experimenters flexibility in creating different topologies.

An alternative approach to attenuating transmission power is increasing the noise floor. Since the transmission range is primarily defined by a signal-to-noise ratio threshold, increasing noise has a similar effect on radio range as reducing transmission power. This approach is widely used in testing single radio links. Lei and colleagues [2] have also applied it to an access point network with multiple clients. They introduced the radio mapping problem, which seeks to map realistic radio environments onto indoor testbeds through controlled noise injection. This can be achieved by changing sender, receiver, and interferer positions and their power levels until the communication channels between access point and each client meet their desired signal-to-noise ratio (SNR) constraint.

In this paper, we extend this approach to ad hoc networks with multiple senders and receivers, but with a slight different goal in mind. Rather than mapping realistic radio environments, we seek to create a radio environment that resembles those described by a free-space propagation model commonly used in wireless network simulations. This provides an easier migration path for experimenters when porting their protocols and systems from a simulation environment to the more complex emulation environment of a testbed. Specifically, key contributions include:

- an automated mapping algorithm that selects suitable nodes based on packet error rate measurements.
- an experimental study of the effects of interference sources on the received signal strength, packet error rate, and symmetry of 802.11 channels.

[1]For Atheros 5212-based IEEE 802.11 wireless cards the minimum setting is 1mW, which still results in a transmission range greater than 20m. Requiring the use of the minimum transmission power setting would also prevent experimentation with power control algorithms.

- experimental evidence that 4-node/3-hop networks can be configured with only 4 interference sources.

The remainder of the paper is organized as follows. We begin by introducing controlled indoor wireless testbeds and the challenges in evaluating wireless protocols in section II. Section III defines the free-space radio mapping problem and describes methods to map free-space topologies onto a testbed. Section IV describes our experimental methodology for evaluation the effectiveness of noise generation and the mapping algorithms. The results are presented in section V before we conclude.

II. Wireless Network Testbeds and the Radio Mapping Problem

To evaluate novel network protocol ideas, researchers usually resort to a simulation environment where the complexity of underlying simulation models is manageable and results can be easily reproduced. Even on relative performance comparison of two protocols, say routing protocols, the underlying physical and MAC layer models can have a major impact. In the field of wireless networking, the art of choosing adequate simulation models is still being refined. Therefore, validation of simulation results on different wireless network testbeds is necessary to understand system performance.

A. Controlled Indoor Testbeds

We can distinguish between application-oriented testbeds and controlled indoor testbeds. Application-oriented testbeds are deployed in a realistic setting and may be used as a production network. Examples are MIT's Roofnet [3], or Dartmouths campus network [4]. These testbeds capture radio propagation effects of their deployment environment and allow experimentation in a realistic setting. They tend to provide lower reproducibility of results because of a constantly changing radio environment. Controlled testbeds are tailored for earlier stage evaluation and are deployed in a shielded indoor environment to increase reproducibility of results.

One example of a controlled indoor testbed is provided by the Open Access Research Testbed for Next-Generation Wireless Networks (ORBIT) [5]. A prototype indoor testbed comprises 128 IEEE 802.11a/b/g radio interfaces attached to 64 static nodes arranged on an 8 by 8 grid, as shown by Fig. 1. The antennas are mounted on the sides of crates, at 45 and 225 degree positions when looking at the topside of a node. The antennas are connected through shielded cables to the Atheros-based wireless cards. Every node is a small form factor PC with 1GHz Via C3 CPU, 512 MB RAM, and 20 GB hard disk. All nodes contain a chassis manager with a 100BaseT Ethernet connection to allow remote management, such as booting and shutdown, of the nodes. The nodes also have two 1000BaseT Ethernet ports, which are connected to each other and to external servers through a 2-layer hierarchy of switches. Through one of these networks the system collects experiment data during run-time. The other network is free for arbitrary use in experiments.

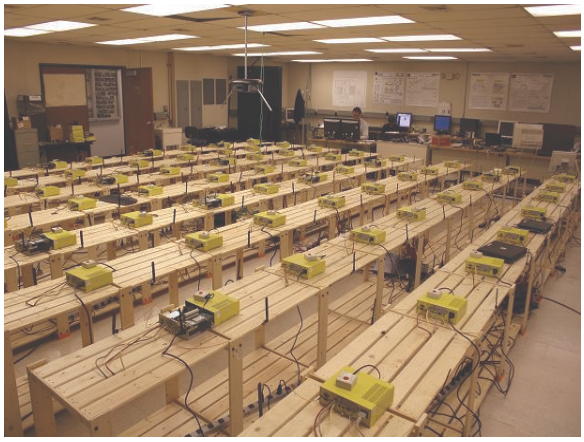


Fig. 1. The prototype ORBIT indoor testbed comprises 64 nodes (small form factor PCs) arranged on a grid. Each node has two IEEE 802.11a/b/g network interfaces, whose antennas are mounted on the sides of the tables.

B. Radio Mapping Problems

As such indoor testbeds can only be of limited size, mere pathloss is not sufficient to create multi-hop topologies where some nodes cannot directly communicate with each other. One foreseeable use that requires these multi-hop topologies is the emulation of ad-hoc networks on the grid. The radio environment is also insufficiently complex to understand protocol performance under real-world conditions. For example, multipath and attenuation due to moving obstructions are absent.

When migrating simulation code to a testbed environment, however, researchers may prefer a simplified propagation environment that resembles the simulation models. The *ns-2* network simulator, for example, includes three radio propagation models as part of the CMU Monarch and ISI extensions [6]: Friis free-space-model, the two-ray ground model, and the shadowing model. The distance-based free-space and two-ray ground models are commonly used. For simplicity, we concentrate here on the free-space model. It assumes that there is a line-of-sight path between transmitter and receiver. It calculates path loss solely as a function of distance. The received signal power P_r at distance d from the transmitter equates to

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

, where P_t is the transmitted signal power, λ is the wavelength, L is the system loss, and G_t and G_R are the transmitter and receiver antenna gains, respectively. The model is further simplified to $P_r(d) = \frac{P_t \lambda^2}{(4\pi)^2 d^2}$ in many simulations.

In addition to a radio propagation environment that differs from the simulation model, evaluation results may be affected by protocol (MAC, routing, transport, etc.) implementations that differ from the simulation models, or affected simply by bugs in the protocol implementation (while much of the code base can be tested in simulation, interfaces with the OS kernel differ from the simulation interfaces).

To isolate the effects of radio propagation patterns from the effects of protocol differences in simulation and testbed, researchers are likely to proceed in the following incremental steps when evaluating new protocols and networked systems.

1) Simulation study
2) Controlled indoor testbed study with free-space propagation environment
3) Controlled indoor testbed study with realistic propagation environment
4) Application-oriented testbed study

This work concentrates on the second step, creating a multi-hop radio propagation environment that matches the free-space propagation models, which are frequently used in simulation studies.

III. Radio Mapping Methods

We assume that an experimenter has access to a number of interferers and a number of nodes that can act as senders and receivers. The interferers are configured to emit additive white Gaussian noise (AWGN) with an experimenter-specified center frequency, bandwidth, and power level. The experimenter can adjust the positions of both the interferers and nodes within a square area (either by physically rearranging nodes or, if working with a stationary testbed, the experimenter may select nodes close to the intended positions from a larger number of candidates). The experimenter can also control bitrate and transmission power on the senders.

We define the *free-space radio mapping problem* as follows: Given a set of n two dimensional virtual node positions $x_1, x_2, \ldots, x_n$, configure the testbed so that the packet error rate (PER) on the links between the chosen testbed nodes approximates the PER in the virtual scenario under free-space propagation assumptions (note that we do not consider packet collisions in this usage of PER). We choose the packet error rate as a link metric because it is a key parameter that affects the performance of higher layer protocols and it can be easily measured on the testbed.

Packet error rates for the free space propagation model are computed based on distance. Given the transmission power P_t and the distance d between two nodes, a packet is received if the received power P_r is greater or equal to a threshold P_{th}, which corresponds to the receiver sensitivity. Thus, the free space propagation model defines a hard distance threshold before which all and after which no packets are received, or in other words, each link will be assigned a packet error rate of either 0% or 100% in stationary networks. We will also refer to these values as link up and link down, respectively.

The desired link status can be captured in a link matrix that defines whether each unidirectional link between a given sender and receiver should be up or down. Table I shows a sample link matrix for a four node string topology. This matrix will be symmetric, because the free-space propagation model can only generate symmetric links. The goal of the radio mapping methods then is to identify a set of nodes, and an interference configuration that approximates this link matrix.

	1	2	3	4
1		up	down	down
2	up		up	down
3	down	up		up
4	down	down	up	

TABLE I

SAMPLE LINK MATRIX TABLE FOR FOUR NODES IN A STRING TOPOLOGY

We consider two methods to address this problem: Select Interference for Fixed Nodes (SIFN) and Select Nodes with Fixed Interference (SNFI).

SIFN selects nodes on the testbed so that the relative distances between the nodes (their antennas) best approximate the relative distances between them on the graph. Interferers are then placed next to the antennas. The algorithm starts by emitting the same power level from all interferers and measures link quality on all links. It then increases the interference in steps until all links marked down in the link status matrix show sufficiently high PER. If this leads to high PER on any of the links that are marked up, it selectively reduces interference power of the antenna next to the receiver node of this link.

The *SNFI* algorithm measures link quality (in terms of PER) for each directional link between any available testbed nodes and an exhaustive search through this link quality graph yields the nodes that best approximate the virtual scenario. It accepts as input the desired link status matrix, the number of nodes in this matrix, and a PER bound which defines when a link is considered down (PER $>= 100 -$ perBound) or up (PER $<=$ perBound).

Interferers are placed at fixed positions close to the corners of the square with an interference power setting that maximizes the dynamic range on the grid. Algorithm box 1 provides a detailed description of SNFI. It begins by capturing a snapshot of the grid as seen by each node. The snapshot is defined as the one hop connectivity between the transmitting node and every other node. We measure the one-hop connectivity in terms of PER. The algorithm then searches a set of nodes that match the desired link status map and reports the first match. Searching is implemented using a backtracking approach. The algorithm may not always find a match, in this case it can be restarted with a higher perBound parameter.

IV. Experiment Setup

The ORBIT grid used is a 8 by 8 square grid consisting of 64 radio nodes, 32 of them fitted with two Atheros 5212-based IEEE 802.11a/b/g cards each. The remaining nodes use Intel -based wireless cards and are not part of our experiments. Let us refer to each node as node(x,y), where x is the column index and y is the row index (both in the interval $[1, 8]$). The Atheros and Intel-based nodes are placed alternately on the grid—the Atheros-based nodes are those nodes for which $x+y$ yields an odd number. Adjacent nodes are separated by a distance of 3ft.

Algorithm 1 Calibration and mapping algorithm. Given the desired link status matrix this algorithm measures per on all testbed links and identifies a set of nodes that approximate the desired link status matrix

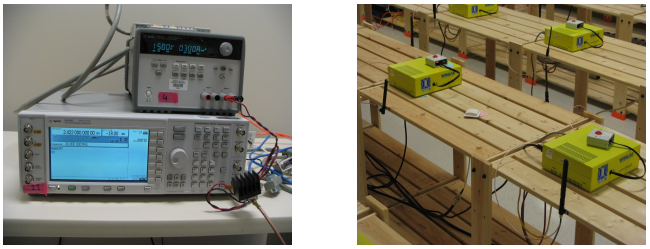
{MapNodes(numDesiredNodes, linkStatus, perBound)}
for each node t **do**
  activate beacon transmission with beacon interval i on node t for duration d
  log received packets on all nodes except transmitter t
  for each receiver r **do**
    count N_r, the number of received beacons in time interval i
    per = $1 - \frac{N_r i}{d}$
    perMatrix$[t, r]$ = per
  end for
end for
findNextNode(emptyList, numDesiredNodes, linkStatus, perMatrix, perBound)
{Procedure findNextNode(selectedNodeList, numDesiredNodes, linkStatus, perMatrix, perBound)}
for all nodes n **do**
  if n in selectedNodeList **then**
    continue with next node
  else
    if perMatch(selectedNodeList, n, linkStatus, perMatrix, perBound) **then**
      add n to selectedNodeList
      if length of selectedNodeList = numDesiredNodes **then**
        output selectedNodeList
        exit
      else
        findNextNode(selectedNodeList, numDesiredNodes)
      end if
      remove last node from selectedNodeList
    end if
  end if
end for
{Procedure perMatch(nodeList, node, linkStatus, perMatrix, perBound)}
for each node n in nodeList **do**
  if (linkStatus$[n, node]$ = up AND perMatrix$[n, node] >$ perBound) **then**
    return false
  end if
  if (linkStatus$[node, n]$ = up AND perMatrix$[node, n] >$ perBound) **then**
    return false
  end if
  if (linkStatus$[n, node]$ = down AND perMatrix$[n, node] <$ 100 - perBound) **then**
    return false
  end if
  if (linkStatus$[node, n]$ = down AND perMatrix$[node, n] <$ 100 - perBound) **then**
    return false
  end if
end for
return true

Each node runs a Debian Linux distribution with 2.4.26 kernel and uses the madwifi stripped driver.[2] This allows generation and processing of raw IEEE802.11 encapsulated frames from the Click Modular Router [7]. One node is configured in Master mode to transmit 802.11 beacon packets, while all other nodes act as receivers. The receivers' driver provides all received MAC frames encapsulated with a so-called Prism header that contains bitrate, received signal strength indicator (RSSI), and other physical layer information. A Click script on the receiver extracts and logs the sequence number and RSSI for each correctly received frame.

We implemented the SNFI algorithm in a Perl script, that executes on a server and can remotely execute commands on the nodes through ssh. Log files were copied back to the server and the packet error rate (PER) at each receiver node is calculated as $1-\frac{N_r}{N_t}$, where N_r is the number of packets in the log file and N_t is the number of transmitted beacons. Since the transmitter sends one beacon per 100ms, $N_t = \frac{d}{100\text{ms}}$, where d is the duration of the experiment in milliseconds.

The testbed supports additive white Gaussian noise interference generation at center frequencies of 250KHz to 6GHz. An Agilent E4438C ESG vector signal generator provides the interference signal. The signal generator is connected to four noise antennae, placed between node2-1 and node2-2; node2-7 and node2-8; node7-1 and node 7-2; and node7-7 and node7-8. The noise power is split equally amongst the noise antennae. An amplifier is used to amplify the signal from the ESG before it is split amongst the noise antennae. The amplifier approximately compensates for any losses in the coaxial cables that connect the ESG to the antennae. All experiments carried out used I/Q modulated AWGN as the interference. Noise power can be varied between -95dbm and -5dbm (at a granularity of 0.5dbm), and distributed over a noise bandwidth of up to 40MHz.

Figure 1 shows the ESG and the antenna placed between node2-1 and node2-2. Unless otherwise mentioned the wireless



(a) (b)

Fig. 2. (a) The ESG vector signal generator with the amplifier and power supply. (b) The noise antenna (white square) placed between node2-1 and node2-2.

[2]Available from http://www.pdos.lcs.mit.edu/ jbicket/madwifi.stripped/. The madwifi stripped driver is a fork from the regular madwifi driver that provides integration with Click. It also provides a pseudo-ibss mode that we used for our experiments because the ad-hoc mode implementation in the standard madwifi driver is not stable.

cards and the noise generator use the configuration shown in table II. We selected the highest transmit power and lowest available bitrate for these experiments, because they result in largest possible transmission range and so present the most challenging scenario for our approach.

Parameter	Default Setting
Interference Type	AWGN
Interference Power	-5dbm
Interference Bandwidth	20MHz
Interference Center Frequency	2.422GHz
Channel	3
Mode	802.11g
Transmit Power	20dbm
PHY rate	1Mbps

TABLE II
DEFAULT EXPERIMENT PARAMETERS

Unless otherwise mentioned PER was measured over a period of 30sec. We experimentally established that measurement variance only marginally decreases over longer intervals. All measurements show packet losses uniformly distributed across the entire measurement period—no packet loss bursts are observed.

V. Results

The experiments conducted address the following questions:

- How does the injected AWGN affect packet error rate and received signal strength on the testbed?
- How reproducible are specific noise conditions?
- How large are the multi-hop topologies that can be constructed, if any?

A. Effect of Noise Sources on PER and RSSI

Figure 3 shows the packet error rate observed at all nodes for node1-4 as transmitter and the noise set to -5dbm. The transmitter transmits beacons for a period of 5 mins (3000 beacon packets). As is seen from the plot, there are two dominant receiver behaviors: full packet loss (100% PER) and no packet loss (0% PER). A few nodes also show partial packet loss. As expected, the nodes receiving all packets tend to be closer to the transmitter than the others.

Figure 4 shows a boxplot of the correlation between the RSSI and the PER observed by a receiver. Each sample contains a mean RSSI value of all packets received by one receiver within the experiment duration and the receivers packet error rate over this interval (since RSSI is only provided for received packets, samples cannot be obtained for nodes that receive no packets). Such RSSI-PER pairs were collected from transmissions of 27 senders for interference levels -20dbm to 0dbm in increments of 5dbm[3]. The graph shows higher packet error rates and higher variance for lower RSSI values. This indicates that RSSI can be interpreted as the signal-to-noise ratio observed by the receiver, rather than the signal power.

[3]27 transmitters, 26 receivers per transmission, 5 noise levels

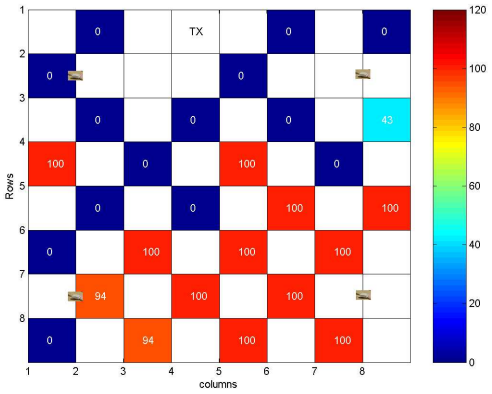


Fig. 3. Percentage Packet Loss with node1-4 as Tx and other nodes as receivers but for the nodes that are in white. Noise is set at -5dbm. The two mini white rectangles in row 2 and row 7are the four noise antennae placed between node2-1, node2-2; node2-7, node2-8; node7-1, node7-2; node7-7, node7-8 respectively.

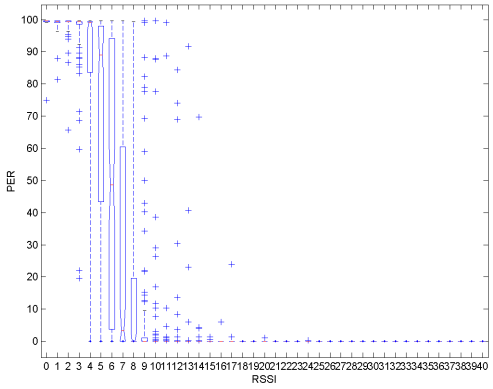


Fig. 4. RSSI PER correlation

The next two figures further show the spatial distribution of the effect of noise across different noise levels. Figure 5 shows the average RSSI received at various nodes for node1-4 as transmitter. The RSSI are plotted for noise levels of -15dbm, -10dbm, -5dbm and 0dbm. The corresponding PER is plotted in Figure 6. Nodes close to the noise antennas (corners) and farther from the transmitter tend to show higher packet error rates, although the effect varies across trials (e.g., compare with Figure 3).

Figure 7 and Figure 8 report on the variation in link asymmetry with increasing interference power (-20dbm to 0dbm). The link asymmetry is shown in terms of PER in Figure 7 and in terms of RSSI in figure 8. In terms of PER the asymmetry is defined as the difference in PER of the forward link and the reverse link between two nodes. Similarly, the second plot shows the difference in the RSSI of the forward and reverse links. As both the plots illustrate, link asymmetry increases for higher interference levels. The asymmetry can be explained by the different interference levels encountered

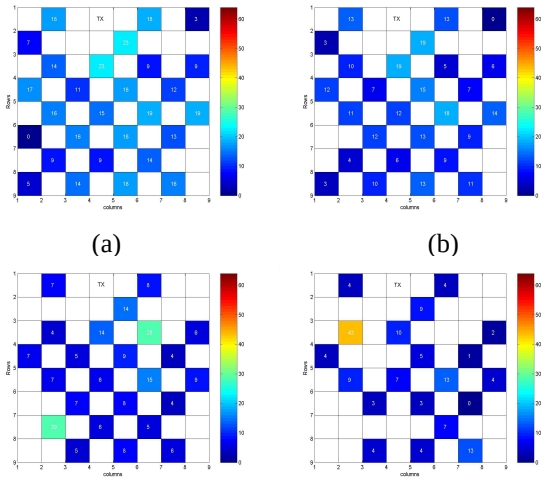


Fig. 5. Average RSSI measured at Atheros Nodes with node1-4 as transmitter for different interference levels. Figures (a) -15dbm (b) -10dbm (c) -5dbm (d) 0dbm. RSSI values are not plotted for nodes not a part of the experiment or those who did not receive any packet during the experiment i.e., the nodes with 100% PER in Figure 6.

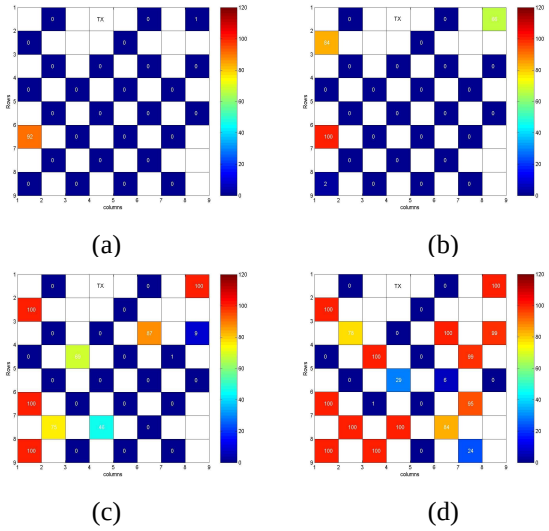


Fig. 6. Average PER measured at Atheros Nodes with node1-4 as transmitter for different interference levels. Figures (a) -15dbm (b) -10dbm (c) -5dbm (d) 0dbm. For the same experiment the RSSI values are shown in Figure 5

by a node pair on the grid. If we consider a bi-directional link between node A and node B, B may be closer to a noise antenna and hence more affected by interference. Hence, the link A-B will show a higher PER than B-A.

B. Reproducibility

Figure 9 shows the results of the same experiment as reported in Figure 3 when repeated on the next day. While most of the nodes show consistent behavior across the two trials, the PER for nodes (7,2), (8,3), and (3,8) exhibits large differences.

Figure 10 highlights the spatial distribution of the 10 out

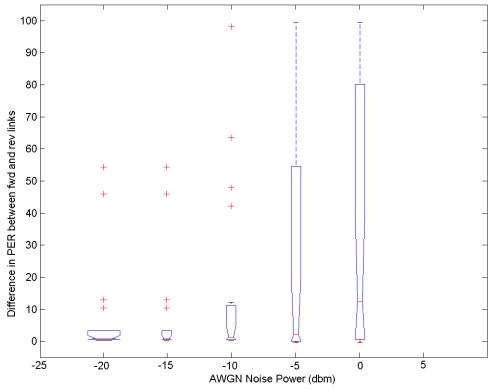


Fig. 7. Asymmetry of Links as difference in PER of fwd and rev links

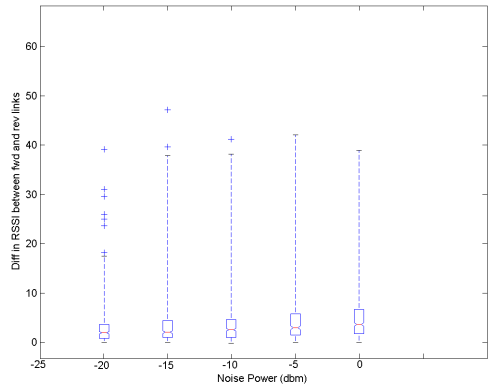


Fig. 8. Asymmetry of Links as difference in RSSI of fwd and rev links

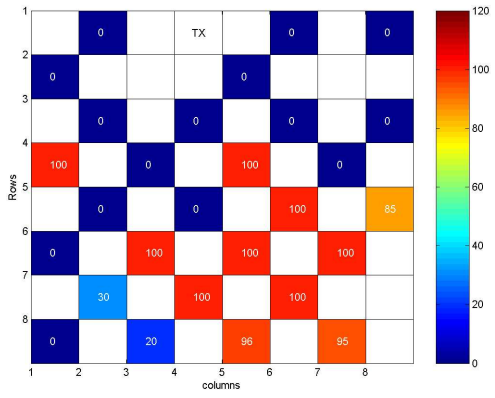


Fig. 9. Another trial of the same experiment as Figure 3. Node1-4 as Tx and other nodes as receivers but for the nodes that are marked Intel, marked white. Interference is set at -5dbm.

of 32 nodes with high variance in PER across 16 trials over a two week period. Blue nodes show no or little variance while red nodes show high variance. The maximum per-node PER difference in these trials was 70%.

Figure 11 shows packet error rates observed by node (7,4)

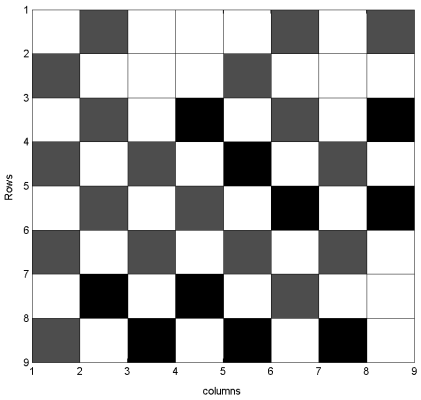


Fig. 10. Spatial distribution of nodes with high PER variance across 16 trials. The transmitter is located at node (1,4). The nodes in dark black are the ones showing high variance.

at different interference levels from -25dbm to 4dbm (note that the interference level is set at the signal generator and does not represent the exact interference power observed by the receiver). Each curve represents one daily trial with packet error rates normalized to unity. A steep increase in packet losses occurs between -8dbm and 0dbm interference. Within this region, a small change in interference power can generate a large difference in packet error rate.

The curves for different trials are slightly offset from each other. For example, the packet error rate for an interference level of -5 dBm ranges from 10% to 80% across trials. This graph explains why large difference in packet error rates are observed on some nodes, while other nodes show hardly any variance. The nodes with high variance are operating with SNR levels close to their minimum sensitivity, thus small changes in environmental conditions may cause large changes in packet error rates.

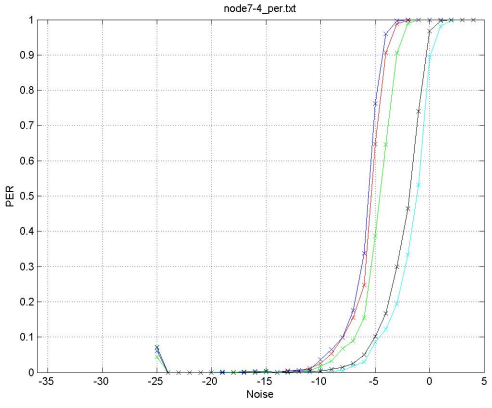


Fig. 11. Packet Error Rates at node7-4 for noise varied between -25dbm and 4dbm at a granularity of 1dbm. The Tx is node1-4. The five differently colored lines correspond to five different repetitions of the experiment in time. Observe the steep increase in the packet losses as the noise generator setting changes from -8dbm to 0dbm.

Figure 12 shows the changes in PER over time relative to per-node mean. The x-axis shows the trial index—results were collected for two trials per day over a period of eight days. Each point represents the packet error rate observed at one receiver, measured relative to the mean for this receiver across all trials. No clear trend can be observed over time. During each trial some nodes show higher PER, while others show less PER than usual. External interference would likely affect all nodes in a similar way, thus this explanation can be ruled out. We observed similar behavior for other transmitters, although the nodes that exhibit high variance change with the position of the transmitter. Thus, these variations are also unlikely to be caused by variations in transmission power.

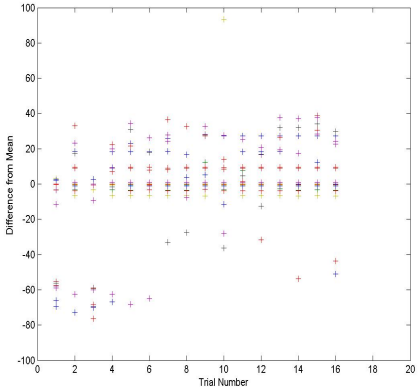


Fig. 12. Packet error rate, plotted as difference from mean, for a total of 16 trials with node1-4 as Tx and other nodes as receivers. Noise is set at -5dbm. For a trial number, each '+' corresponds to a different receiver node.

C. Feasibility of Multi-hop Topologies

We first manually executed the SIFN method to determine the feasibility of mapping a 4-node, 3-hop string topology onto the testbed. We selected nodes (1,2),(1,4),(1,6), and (1,8), which are equally spaced and placed along an imaginary straight line. After placing noise antennas next to the node antennas, we found the desired topology at an interference level of -10dbm. We also moved the noise antenna besides node1-2 by approximately 7cms in a direction perpendicular to the line connecting the nodes, because the noise level was too high at this node. To verify the multi-hop path we ran a ping session over the AODV routing protocol for a period of 20 mins. Ping showed 2% packet loss and an average delay of 89.8 ms. During this experiment we consistently saw three hops between the participating nodes. Figure 13 shows the link packet error rates for this setup, measured as described in section IV. This configuration creates a near-perfect mapping of a free-space 4 node string topology.

Figure 14 shows that the automated SNFI algorithm with fixed noise antenna positions (as described in section IV) can also identify mappings for 3-hop string topologies. The graph shows the number of node mappings identified by the algorithm for different string-topology lengths, noise levels,

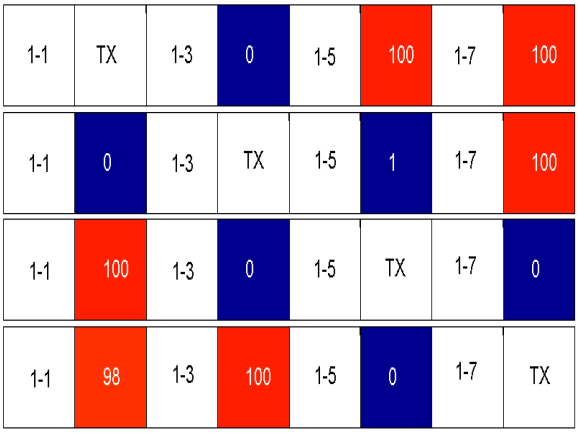


Fig. 13. Three hop experiment PER plot

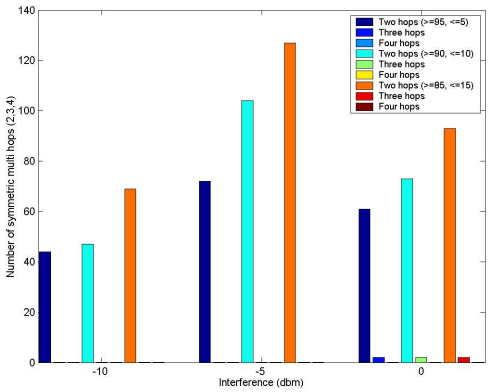


Fig. 14. Figure shows the number of two, three and four hop string topologies obtained for noise levels of -10, -5 and 0dbm. For each noise level three different values of perBound are selected.

and per bound parameters. The noise levels were set to -10, -5 and 0dbm and the per bound parameter to 5, 10, and 15% (which corresponds to a maximum PER of 5, 10, and 15% for a link that should be up and a minimum PER of 95%, 90%, and 85% for a link that should be down) for each noise level. For noise levels equal to and below -5dbm a maximum of two hops were possible. The number of two hops peaks at -5dbm. While the higher 0dbm noise level shows fewer 2-hop links it creates two 3-hop links. For example, one of the three hop links created was that between node2-1, node4-1, node8-3 and node8-7. Referring back to figure 3, it can be seen that the noise antennae are very close to the nodes 2-1 , 8-3 and 8-7 (the noise antennae are the ones between node2-1 and 2-3, node7-1 and 7-2, node7-7 and 7-8, respectively). Thus this setup shows similarities to the manually created configuration. The experiment was repeated on three different days. The total number of string topologies found were similar to the reported numbers. The number of 3-hop topologies varied, however. On the second repetition one topology was found for a noise setting of 0dbm and $perBound$ of 10%. On the third repetition no three hop string topologies were found. Reproducibility may, however, be improved by slightly adjusting the interference power level.

More complex topologies are also possible. Figure 15 shows the largest topology that was extracted from the SNFI experiment with 0dbm noise and perBound set to 5%. The eight nodes in the topology are selected so that all links are symmetric 'up' or 'down' links (using a PER bound of 5%). All nodes with asymmetric or partial connectivity were eliminated. Any two nodes may not be connected to each other. Table III shows the number of distinct topologies obtained. Topologies consisiting of four to eight nodes were found.

Size of topology	4	5	6	7	8
Number of occurences	5	2	13	1	3

TABLE III

THE NUMBER OF DIFFERENT SIZED TOPOLOGIES OBTAINED AT 0DBM, PER BOUND IS SET TO 5%.

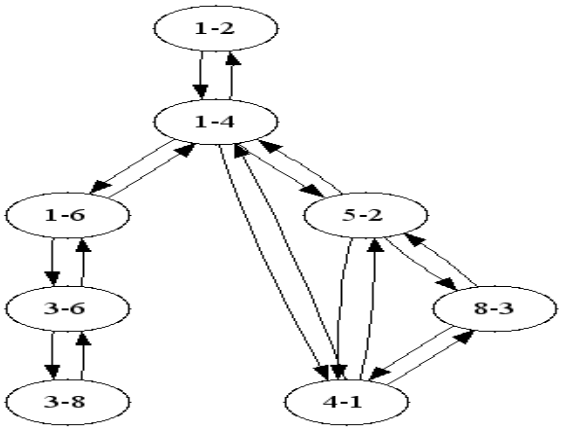


Fig. 15. Figure shows the largest symmetric topology that does not include any marginal links. Note that the graph only indicates connectivity of nodes, it does not imply relative position among them.

D. Discussion

The results show that multi-hop topologies can be successfully created with only four noise sources. Given the limited number of sources, the SNFI approach appears preferable, because it is more flexible and no manual intervention is necessary. It can automatically select suitable nodes for a number of different 2- or 3-hop topologies. The large number of possible 2-hop topologies also indicates that it may be possible to combine some of them into more complex topologies. While the SIFN approach could also be fully automated through a variable attenuator for each antenna, this approach will likely be limited to the nodes next to the noise sources. The availability of many noise sources would reduce this disadvantage, however.

The results also show that long-term reproducibiltiy of results is reduced if nodes with marginal SNR are involved. Simply configuring the same noise level days later is not sufficient to guarantee the same result. While this is also the case in real-world wireless experiments, a controlled testbed

should aim for increased reproducibility. This problem could be addressed by executing the mapping algorithm before each experiment. The algorithm then may map the topology onto a different set of nodes. However, the PER characteristics of these nodes will be similar to the earlier experiments, which leads to improved reproducability. An open question is which of the radio mapping methods leads to better reproducibility of results. SIFN may provide more fine-grained control and thus allow a more exact reproduction of the desired PER rates between nodes, even if the radio environment has slightly changed.

VI. Related Work

Lei and colleagues [2] described the a method to emulate the signal-to-noise ratio for several clients associated to an access point. This work, to our knowledge, first introduced the link mapping problem for wireless indoor testbeds. It evaluated the feasibility of mapping several access point to client links based on a simulation study. We extend this work to multi-hop, bidirectional links and present proof-of-concept experimental results.

The concept of introducing additive white Gaussian noise is a standard technique to evaluate communication system performance between a single sender and receiver. For example, Pajukovski and Savusalo [8] present experimental results for a CDMA system that were obtained by injecting AWGN into the channel. These standard techniques do not apply to large-scale multiple sender and receiver scenarios.

Other testbeds have addressed the spatial scaling problem of wireless testbeds through attenuation at the transmitter. Sanghani [9] and colleagues have build a testbed with heavily attenuated antennas that can be repositions to create different topologies. Kaba and Raichle also use attenuation but instead of antennas they connect the radios by wire and control topology through resistive splitters and combiners [1]. These approaches require careful shielding, which can be very time consuming for large numbers of radios. They are also less flexible since attenuators (and especially splitters and combiners) must be manually configured, whereas noise level control can be automated.

Other testbeds have been built and experiments conducted without scaling the communication range. These experiments range from a group of students carrying laptops on a football field or in a larger building [10], [11] over experiments with radios in vehicles (e.g., [12], [13] to in-city deployments of mesh networks [3]. These deployments substantially differ from a stationary indoor testbed in the degree of reproducibility of results. The indoor testbed is not affected by sporadic external interference, foliage, moving obstructions, weather conditions, and other parameters. Thus, it provides a more convenient environment for detailed scientific study before systems are evaluated in a less predictable outdoor environment.

VII. Conclusion

This paper described an approach to configure the placement and transmission power of additive white Gaussian noise sources to create multi-hop wireless network topologies in a constrained indoor space. In particular, the mechanisms aimed at creating topologies that resemble those obtained from a free-space propagation model. This provides a natural first step in migration of protocols that were evaluated and tested in network simulators on such a simplified propagation model. This approach allows isolating effects of different propagation environments from the effects of the protocol stack implementation. Specifically, we conclude that

- A 20 MHz additive white Gaussian noise signal on the 802.11 channel center frequency can effectively reduce SNR at the receiver. This can be observed from the received signal strength indicator and through increased packet error rates.
- 3-hop topologies with symmetric channels can be created with four noise sources in an 8m by 8m area, by placing one noise source next to each radio that is part of the experiment. They can also be efficiently created through an automatic search procedure that selects nodes whose link PER approximates the desired link qualities.
- Network and MAC-layer experiments that include nodes with marginal SNR exhibit some variability in topology when repeated after longer time periods (e.g., 24 hours), they do not show this variability over short time intervals. Better reproducibility could be achieved through an automatic calibration procedure that seeks to recreate a similar topology.

This work lead to three promising ideas for future work. First, many interesting experiments involve more than four nodes. Thus these methods must scale to hundreds of nodes and low-cost noise sources. Second, the system could provide an improved interface that let's users specify desired topologies taking into account different modulation schemes and carrier sensing ranges, rather than requiring node positions as input. Finally, indoor testbeds should provide channel conditions that are similar to common real world deployments of wireless networks such as an office environment. Mapping real-world scenarios onto the testbed through noise generation remains an open challenge.

References

[1] J. T. Kaba and D. R. Raichle, "Testbed on a desktop: strategies and techniques to support multi-hop manet routing protocol development," in *ACM MobiHoc*. ACM Press, 2001, pp. 164–172.

[2] J. Lei, R. Yates, L. Greenstein, and H. Liu, "Wireless link snr mapping onto an indoor testbed," in *IEEE Tridentcom*, Feb 2005.

[3] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2005, pp. 31–42.

[4] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," in *Proceedings of the eighth annual international conference on Mobile computing and networking*. ACM Press, 2002, pp. 107–118.

[5] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE WCNC*, 2005.

[6] W. Ye, "The ns manual, ch 18: Radio propagation models," http://www.isi.edu/nsnam/ns/doc/index.html.

[7] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, “The click modular router,” *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, Aug 2000.

[8] K. Pajukoski and J. Savusalo, “Wideband CDMA test system,” in *Symposium on Personal Indoor and Mobile Radio Communications*, 1997.

[9] S. Sanghani, T. Brown, S. Bhandare, and S. Doshi, “EWANT: The emulated wireless ad hoc network testbed,” in *IEEE WCNC*, vol. 3, Mar 2003, pp. 1844–1849.

[10] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, “Experimental evaluation of wireless simulation assumptions,” in *ACM/IEEE MSWiM*, October 2004, pp. 78–82. [Online]. Available: http://www.cs.dartmouth.edu/~dfk/papers/kotz:axioms.pdf

[11] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordstroḿ, and C. Tschudin, “A large-scale testbed for reproducible ad hoc protocol evaluation,” in *IEEE WCNC*, 2002.

[12] D. Maltz, J. Broch, and D. Johnson, “Experiences designing and building a multi-hop wireless ad hoc network testbed,” Carnegie Mellon University, Tech. Rep. CMU-CS-99-116, Mar 1999.

[13] M. Moške, H. Fuśsler, H. Hartenstein, and W. Franz, “Performance measurements of a vehicular ad hoc network,” in *Vehicular Technology Conference (VTC-Spring)*, May 2004, pp. 2116–2120.