Evaluation of a Low-Cost IEEE 802.11n MIMO Testbed

Anatolij Zubow and Robert Sombrutzki
Humboldt University
Rudower Chaussee 25
Berlin, Germany
Email: (zubow|sombrutz)@informatik.hu-berlin.de

Abstract

This paper presents and evaluates a configurable and inexpensive MIMO mesh network platform based on IEEE 802.11n and open source software for research purposes. The requirements on such a research testbed are twofold. On the one hand a highly configurable solution is desirable where the researcher is able to make modifications on each layer of the hard- and software solution. On the other hand to make sound conclusions on the performance of protocols for mesh networks a large-scale testbed consisting of hundreds of nodes is necessary. Therefore, a single mesh node has to be inexpensive. Thus a tradeoff between these two opposed targets has to be made. The proposed solution is based on off-the-shelf 802.11n hardware using Atheros WiFi chips together with the open source WiFi driver ath9k and the Click Router API. This solution represents a good tradeoff where the procurement cost for a network node is below 100 $ while still allowing a variety of adjustments to be made due to the used open-source driver and a highly configurable WiFi hardware. With the help of measurements, the suitability of the platform is evaluated.

Keywords

IEEE 802.11n, MIMO, Testbed, Mesh Network

I. Introduction

Wireless mesh networks (WMNs) [1], [2] are currently a hot research topic in industry and academia. Significant efforts in the academic world are made to provide real-world prototypes and testbeds based on open source software and off-the-shelf technologies mostly based on standards like IEEE 802.11. The advantage of a non-proprietary solution is that results which were found by one research group can be easily verified or validated by another group using a testbed with the same software and hardware platform. A major drawback of an off-the-shelf solution is the limited ability to make modifications on lower layers of the protocol stack (mostly MAC and PHY) which significantly reduces the research field of application. The majority of testbeds based on 802.11 are using the widely deployed 802.11a/b/g standard. However, the upcoming 802.11n standard offers lots of improvements. Therefore, a software and hardware platform based on 802.11n is desirable.

The main contributions of this paper are as follows. First, we identify requirements for a software and hardware solution for building experimental mesh testbeds. Second, we present our configurable and inexpensive mesh network platform based on 802.11n and open source software. A comparison with a solution based on 802.11a/b/g and the MadWifi driver is given. In addition we give an overview on the integration of the ath9k driver with the Click Router API. Finally, we present measurement results from our testbed highlighting the strengths and weaknesses of the proposed solution.

II. IEEE 802.11n

The aim of this section is to give an overview on the improvements from 802.11n. Here it is important to know which improvements from 802.11n are mandatory and which are only optional. The IEEE 802.11n standard promises faster networks with an increased WiFi coverage. At the physical (PHY) layer, it is the introduction of multiple antennas at the receiver as well as the transmitter (Multiple Input Multiple Output) together with advanced signal processing and modulation techniques and the use of wider channels. At the Media Access Control (MAC) layer, protocol extensions like frame aggregation and block acknowledgement reduce significantly the MAC layer overhead and therefore allow a more efficient use of available bandwidth.

A. PHY Improvements

The most important improvement of 802.11n on the PHY layer is the ability to receive and/or transmit simultaneously on multiple antennas. The improvements from multiple antennas are two-fold. First, using multiple antennas at the receiver and transmitter side offers an antenna diversity gain which improves the reliability of a wireless link by reducing the error rate. Second, the MIMO channel can be used to simultaneously transmit multiple data streams through different antennas and therefore significantly increasing the maximum data rate.

In the following we will present the most important signal processing techniques introduced by 802.11n to exploit multiple antennas. Spatial Multiplexing (SM) is a MIMO transmission technique to transmit independent and separately encoded data signals, so-called streams, from each of the multiple transmit antennas. Therefore, an outgoing signal stream is subdivided into multiple parts before being transmitted through different antennas. The gain from SM comes through the reuse of the space dimension. Whether SM is possible or not depends on whether the spatial streams have a sufficiently distinct spatial signature so that the receiver is able to back-calculate the original signal streams. In theory multiplexing two spatial streams onto a single channel effectively doubles capacity. Space-Time Block Coding (STBC) provides a diversity gain by sending a signal stream redundantly, using up to 4 coded spatial streams, each transmitted through a different antenna. STBC improves reliability of a wireless link by reducing the error rate experienced at a given Signal to Noise Ratio (SNR). The use of STBC is especially interesting in environments with presence of high RF interference and distortion. STBC is an optional feature in 802.11n. Transmit Beamforming (TxBF) is a signal processing technique where the outgoing signal stream is steered towards the intended receiver by concentrating transmitted RF energy in a given direction by make use of constructive interference. To be able to steer a signal the transmitter needs to know channel state information (CSI). CSI can be obtained implicitly (by assuming channel reciprocity) or explicitly (by obtaining CSI feedback from the receiver). This optional 802.11n feature is not yet widely implemented.

Moreover, another important optional 802.11n feature is the use of wider channels. According to 802.11n channels having a bandwidth of 40 MHz can be used which effectively doubles throughput. The 40 MHz channels can be used in the 2.4 GHz ISM as well as the 5 GHz UNII band.

Finally, the available Modulation and Coding Schemes (MCS) where extended. MCS is the selection of a given RF modulation, coding rate, and guard interval. In 802.11n a new coding rate of 5/6 is added. Furthermore, an OFDM short guard interval ($0.4\,\mu s$ instead of $0.8\,\mu s$) can be used. Note, the guard interval is necessary to offset the adverse effects of multipath that would otherwise cause Inter-Symbol Interference (ISI). A shorter guard intervals may lead to more interference and reduced throughput in environments with a large multi-path delay spread, while a longer guard interval is inefficient due to unused idle time. An optional feature of 802.11n is the possibility of using a different MCS on each spatial stream called unequal modulation. To further improve the spectral efficiency the number of OFDM data subcarriers was increased from 48 to 52 which effectively increases the data rate by around 8%.

Table 1 shows the relationships between the variables that allow for the maximum data rate. According to 802.11n APs are required to support at least MCS index 0 through 15, while 802.11n stations must support MCS index 0 through 7. All other MCS values, including those associated with 40 MHz channels, SGI, and unequal modulation, are optional.

Index	Streams	Data rate (Mbit/s)			
		20 MHz channel		40 MHz channel	
		800 ns GI	400 ns GI	800 ns GI	400 ns GI
MCS Index 0	1	6.50	7.20	13.50	15.00
1	1	13.00	14.40	27.00	30.00
2	1	19.50	21.70	40.50	45.00
3	1	26.00	28.90	54.00	60.00
4	1	39.00	43.30	81.00	90.00
5	1	52.00	57.80	108.00	120.00
6	1	58.50	65.00	121.50	135.00
7	1	65.00	72.20	135.00	150.00
8	2	13.00	14.40	27.00	30.00
...	...	...	...	...	...
15	2	130.00	144.40	270.00	300.00
...	...	...	...	...	...
31	4	260.00	288.90	540.00	600.00

TABLE I
RELATIONSHIP BETWEEN MCS INDEX, GUARD INTERVAL, BANDWIDTH AND THE CORRESPONDING DATA RATE.

B. MAC Improvements

In addition to the improvements on the PHY layer 802.11n introduces enhancements on the MAC layer. The most important improvements are Frame Aggregation (FA) and Block Acknowledgement (BACK). FA is the most promising way reducing the MAC layer overhead by sending very large PHY frames. In 802.11a/g the maximum payload per frame cannot exceed 2304 Bytes. In contrast in 802.11n it is possible to bundle multiple frames together for transmission thus increasing the payload size and reducing fixed overhead caused by inter-frame spacing and preamble. The standard distinguishes between two aggregation options: MAC Service Data Unit Aggregation (A-MSDU) and MAC Protocol Data Unit Aggregation (A-MPDU). The former allows the aggregation of multiple MSDUs into a single MAC frame containing one MAC header, followed by up to 7935 MSDU bytes. The later one occurs later, after MAC headers were added to each MSDU. Complete MAC frames (MPDUs) are then aggregated into PHY payloads of up to 65535 Bytes. BACK allows the receiver to confirm reception of multiple unicast frames by a single ACK thus heavily reducing the number of transmitted ACKs. BACK improves the efficiency since the ACK packets are always send on robust MCS (base rate) and therefore cannot gain from the PHY enhancements introduced by 802.11n.

C. Coexistence with Legacy Devices

Coexistence with legacy 802.11a/b/g devices is critical due to the very large deployment of those devices operating in the same frequency bands as the new 802.11n standard. Therefore, interoperability is accomplished using High Throughput (HT) protection and coexistence mechanisms. 802.11n dinstinguishes between three operating modes: HT, Non-HT, and HT Mixed. The optional HT mode, known as Greenfield mode, assumes that there are no nearby legacy devices using the same frequency band. This is the most efficient mode since no interoperability or protection is required. When using Non-HT mode all frames are sent in the old 802.11a/g format so that legacy stations can understand them. This mode offers no better performance than old 802.11a/g devices, however, backward compatibility is ensured.

III. Requirements

The requirements on a research wireless mesh testbed are twofold. On the one hand a highly configurable solution is desirable where the researcher is able to make modifications on each layer of the hard- and software system. On the other hand an inexpensive solution is required so that sound conclusions can be made on the performance of protocols in large-scale mesh testbeds.

To allow a deployment of hundreds of nodes the hardware should be cheap. This excludes a proprietary solution based on expensive Software Defined Radio (SDR). Therefore we favor a non-proprietary solution based on the 802.11 standard. The nodes should have at least two radios for multi-channel and backbone operations. For experiments a fast wired interface (e.g. Gigabit Ethernet) is desirable. Furthermore a fast CPU is required when using high throughput data rates from 802.11n or when evaluating computationally intensive algorithms. (e.g. network coding). Finally, for a testbed consiting of hundreds of nodes a hardware watchdog is inevitable. On the software side a full open-source WiFi driver with an active developer community is indispensable which allows a researcher to make modifications on the MAC layer (e.g. for rate/power/channel control). Moreover, many chipsets (e.g. Atheros) allow to get information on low-level PHY statistics like channel utilization (e.g. for cognitive radio). Protocols involving algorithms on the network, routing or a higher layer can be easily implemented using the Click modular router API [3]. Therefore, a tight interworking between the WiFi driver and the Click API must be ensured (e.g. raw frame injection and monitor mode). Finally, an operating system based on Linux or BSD is required. Finally, a solution is required which can be easily integrated into an existing wireless mesh testbed based on a legacy standards like 802.11a/b/g.

IV. HWL Testbed

The Humboldt Wireless Lab (HWL) is a large-scale wireless mesh network at the campus of the Humboldt University, Germany. It consists of about 100 mesh nodes based on 802.11a/b/g which are deployed indoors as well as outdoors. The indoor nodes, which are placed in several building, form a fully connected wireless network, which can be combined with the outdoor network to improve the connectivity between the buildings. The aim of HWL is to evaluate large-scale mesh networks, since small- and medium-scale mesh networks are already well understood. The upcoming IEEE 802.11n standard promises to significantly increase coverage, reliability, and throughput which comes from the advanced PHY layer technology based on MIMO techniques. Novel protocols and algorithms (e.g. MAC/Routing) are required to get the full advantage at higher layers from these improvements. We are now in the process of extending HWL by new nodes based on the 802.11n standard. Here we are looking for a solution where the existing system and network architecture remains unchanged. In this section we present our existing architecture and show what steps, mainly adaptation of the wifi driver and router API, where required to integrate the new 802.11n based mesh nodes. Furthermore, we discuss the features and limitations of our solution.

A. Platform

The existing HWL platform is as follows. The indoor nodes, mainly MIPSEL based routers, are equipped with a single 802.11b/g wifi card with Atheros AR5212 chipset. The outdoor nodes are Soekris boards with a x86 architecture, which are eqipped with two 802.11a/b/g transceivers also based on the Atheros AR5212 chipset. All nodes are using an adapted version of the MadWifi driver[1]. All mesh related functionality is implemented using the Click Router API.

All indoor nodes are connected via a wired VLAN backbone to a central testbed server, which provides services like TFTP, DHCP, DNS and NFS. The indoor nodes load the operating system (Linux kernel, rootFS) completely from the network. This centralized approach simplifies the maintenance and extension of the network by new nodes, because the software has to be

[1] http://madwifi-project.org/

updated only once on the server and is immediately available to all nodes. In contrast the outdoor nodes are connected by a wireless mesh network backbone and a gateway with the testbed server. Therefore the second wifi interface is used and therefore cannot be used for experiments. Due to the slow backbone the outdoor nodes boot the operating system directly from the internal flash memory.

All experiments are centrally controlled from the testbed server where the data collected in experiments is stored centrally, which simplifies the analysis considerably. The server also performs a monitor function (Nagios).

1) The new Mesh Node: The Netgear WNDR3700[2] was selected as new 802.11n based mesh node. It is an off-the-shelf wireless router equipped with two 802.11n compliant radios, one for 802.11b/g/n (Atheros AR9223) and 802.11a/n (Atheros AR9220) respectively. The WNDR3700 has an Atheros (AR7161 rev 2) MIPS CPU, running at 680 MHz, and 64 MB of RAM and 16 MB of flash memory. Moreover, it has a Gigabit Ethernet adapter (Realtek RTL8366SR), an USB2, serial and Jtag interface. Netgear made the Linux source code (openwrt[3]) publicly available. The uBoot bootloader allows us to boot the operating system in a similar way to our legacy mesh nodes. Additional useful information can be found on the openwrt website[4]. In order to integrate the WNDR3700 in our existing testbed, the latest version of openwrt is used, which allows to load the operating system via network. Therefore, the boot loader was reconfigured. With the help of the ath9k wifi driver, which is part of openwrt, we were able to use both 802.11n interfaces, which are described in more detail in the next section.

2) 802.11n Interface: The WNDR3700 has to 802.11n radios. The first radio is dual-band (Atheros AR9220) allowing operating in the 2.4 as well as 5 GHz band, whereas the second radio can only be operated in the 2.4 GHz band (Atheros AR9223). Both radios support 2x2 SM-MIMO (2 spatial streams), channel bonding (40 MHz channel) as well as the possibility to use short guard interval (SGI). The maximum data rate at the PHY layer is 300 Mbps. In the ath9k driver SGI may be used only in conjunction with 40 MHz wide channels. Both wifi chips also support space-time block codes (STBC), thus achieving a transmit diversity gain. The optional transmit beamforming is not supported. In addition, the forward error correction (FEC) type can be set, with both binary convolutional codes (BCC) and the more efficient low-density parity-check codes (LDPC). Both chips provide a fine-grained rate adaptation. A unicast packet can be annotated with up to 4 rate index (MCS) values and the number of retries to be used. The chip has, like the Atheros 5212 chip, 4 so-called performance registers which can be used to determine the dwell time in the states receiving, transmitting and busy[4]. It is also possible to get the information of the error vector magnitude (EVM) per spatial stream. This information can be used for designing better rate control algorithms[5]. Moreover, it is also possible to read out the received signal strength for each antenna element. However, the majority of the mentioned features is not or only partially supported by the wifi driver. The required modifications are explained in the next section.

B. Software

1) 802.11n Driver: The biggest disadvantage of Netgears MadWifi driver for the WNDR3700 is that the full source code is not publicly available. Furthermore, the shipped version supports only the AP infrastructure mode. The linux-wireless project[6] develop the ath9k, an open source driver for Atheros 802.11n chipsets, which support raw packet injection. All packet have an Radiotap header[7] to server additional information, e.g. transmission rates from user- to kernelspace and backwards. This offers more control over the packet transmission and enables the development of cross layer protocols in userspace. Currently, the fields of the radiotap header only supports to set one rate per transmission and they also serve brief information about received packets. We extend the ath9k, so that additional fields which can hold up to four rate and a second field containing the number of retries per rate. Therefore it is possible to implement a more fine-grained rate selection in userspace. Furthermore, the Radiotap header provides additional information, e.g. signal strength of both antennas. The content performance register can be read using a additional entry in the sysfs of linux.

Table II gives an overview of the supported features, which are available with the Atheros AR5212 and the MadWifi driver. All of them can be controlled from userspace, preferably using packet annotations. Currently only two features are supported by AR9220/9223 and ath9k.

2) Router API: The mesh software is implemented using the Click router API [3]. A Click router is built by sticking together several packet processing modules, called elements, forming a directed flow graph. Each element is responsible for a specific task such as packet classification, scheduling, or interfacing with networking devices. Click comes with an extensive library of elements supporting various types of packet processing. Such a library allows to easily write new router configurations by simply choosing the elements to be used and the connections among them. Finally, a router configuration can easily be extended by writing new elements.

Due to our changes on the ath9k driver, it becomes necessary to adapt several click elements, especially for rate selection. Each rate is specified by the MCS index, channel width and the length of the guard interval. Additional to the 4 possible rates per packet, the number of retries per rate and the type of FEC can be specified. The elements related to the radiotap header

[2]http://www.netgear.com/home/products/wirelessrouters/high-performance/WNDR3700.aspx

[3]http://www.openwrt.org/

[4]http://wiki.openwrt.org/toh/netgear/wndr3700

Feature	Status
Multirate support	✓
Performance register	✓
Set interframe spaces	○
Set backoff per queue	○
QoS per frame	○
Transmit power control (per frame)	○
Disable carrier sensing	○
Channel selection (per frame)	○
MAC address switching (per frame)	○

TABLE II
OVERVIEW OF SUPPORTED FEATURES OF THE AR9220/9223 AND ATH9K.

use the additional fields in the Radiotap header the serve this information to the driver. The information of received packets, e.g. signal strength of each antenna is annotated at the packet for further usage in other elements.

V. EVALUATION

In order to evaluate the proposed testbed platform we conducted measurements. First, the capability of the selected hardware and software was evaluated. Afterwards we present results showing the saturation throughput in an isolated hotspot scenario. Thereafter we evaluated the impact of power control on the received signal strength. We conclude this section by presenting results from an extensive link-level measurement.

A. Supported 802.11n Features

In 802.11n several improvements are optional. Furthermore, by using an open source driver which is currently under heavy development some additional missing capabilities might exist. Therefore, we give an overview of the supported 802.11n capabilities. The following measurement setup was used. A sender was transmitting to a close-by receiver using a low packet rate. We evaluated MAC layer broadcasts as well as unicasts whereas for the latter one also the multirate support was evaluated. All three modes were correctly working. The most important PHY improvements from 802.11n like advanced MCS, SM-MIMO, wider channels and the use of short guard interval (SGI) are supported. The only restriction was that SGI was not working together with 20 MHz channels. Note, that we were not able to find out whether the STBC and LDPC codes were supported. This is because the current version of the ath9k driver doesn't report whether such a feature was used or not.

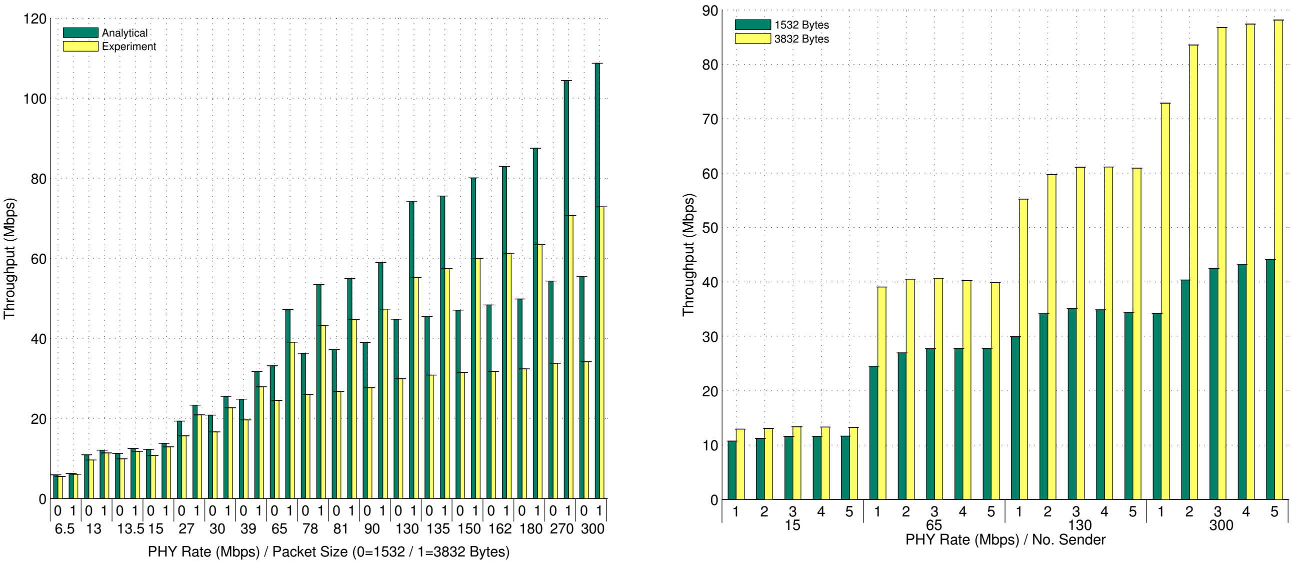
Index	20 MHz		40 MHz	
	800 ns GI	400 ns GI	800 ns GI	400 ns GI
1	✓	×	✓	✓
7	✓	×	✓	✓
8	✓	×	✓	✓
15	✓	×	✓	✓

TABLE III
SUPPORTED MCS-RATES.

B. Saturation Throughput

Next we present results showing the saturation throughput (UDP) in an isolated hotspot scenario. All nodes were placed within a short distance (1-2 m) to each other to make sure that the observed throughput was not negatively affected by weak signals. Moreover, we used an unoccupied 5 GHz channel to avoid problems like co-channel interference as well as competition for the medium. Thus the results show the best achievable performance under optimal conditions.

At first the saturation throughput for a single sender and receiver depending on the used MCS (PHY rate) as well as the used packet size is analyzed. The results from the experiment are compared with analytical results. In the analytical model we calculated the air capacity from the payload size and the frame start interval. The frame start interval is the sum of DIFS ($34\,\mu s$), average contention window ($15/2 \times 9\,\mu s$), 802.11n PLCP header ($28\,\mu s$), payload frame duration which depends on the MCS, guard interval and channel width, SIFS ($16\,\mu s$) and the ACK ($32\,\mu s$). The following observation can be made (Figure 1(a)). The difference between the analytical and the experimental results is small only for low PHY rates, i.e. robust MCS. For efficient MCS and wider channels the difference is large, e.g. for small packets at 300 Mbps PHY rate (MCS=15,SGI,HT40) the achieved throughput is only 62% of the expected. During the measurement we observed only a very small number of corrupted packets indicating only a small number of collisions. We believe that the difference between the expected and



(a) Saturation throughput - 1 sender and 1 receiver.

(b) Saturation throughput - 1 to 5 sender(s) and 1 receiver.

Fig. 1. Saturation throughput and channel utilization in hotspot scenario.

achieved throughput is due to the slow CPU or slow memory access. During the experiment we observed a very high CPU load at the transmitter of 95% at high MCS. So it is very likely that the CPU was the bottleneck.

Next we increased the number of sender nodes. From the theory we would expect that the total throughput will decrease due to increased collision probability between the competing sender nodes. Figure 1(b) shows the results. On our case, however, this only true for low MCS and large packet sizes. E.g. large packets at 300 Mbps PHY rate the throughput increases by up to 21% when the number of sender nodes is increased from 1 to 5. Thus a single sender seems to be unable to fully saturate the medium. This is an additional indication that the CPU or the memory is the bottleneck at high MCS. The CPU load at sender side was 95% for a single sender and small packets and 78% for 5 senders and large packets. In contrast the CPU load at receiver side was never beyound 45%.

Finally, we take a look at the channel utilization during the experiment. From Figure 2(a) we can see that for high MCS the channel cannot be fully utilized which is again connected to the slow CPU or memory.

C. Power Control

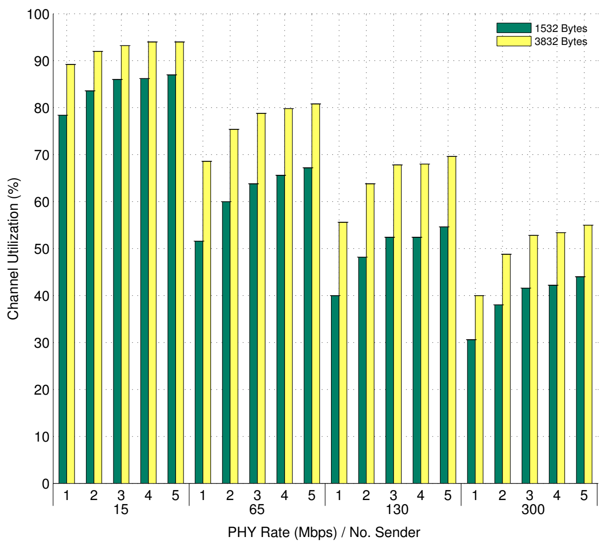
The ability to control the transmission power is essential when designing power control algorithms. Next we will analyze the relationship between transmit and receive power. Therefore two nodes where placed 7 m apart from each other. The inter-node distance was varied by some carrier wave length to average out any multipath effects. From Figure 2(b) we can see that the used Rf channel, MCS and channel width have a significant impact. Especially the transmission power at 2.4 GHz is low compared to 5 GHz. This is mainly due to regulation requirements. Besides that the power is adjustable to some kind of degree and can therefore be practically used.

D. Link-level Measurements

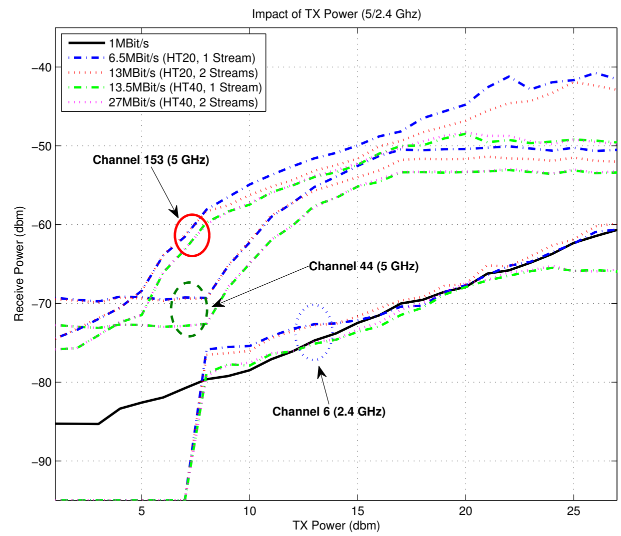
The aim of this section is present link-level results obtained from our indoor testbed. The testbed resides in two buildings on 4 different floors of the computer science department of the humboldt university. The exact node locations of the selected 37 nodes are given in Figure V-D.

In this section we present results from link-level measurements in our indoor testbed. The testbed resides in two buildings on 4 different floors of the computer science department of the Humboldt University. For the experiment 37 nodes were selected. The following setup was used. Each node sends 1000 packets at a rate of 10 Hz MAC broadcast packets of 1000 Bytes size on each available MCS, guard interval and channel width combination. The experiment was conducted in the 2.4 and 5 GHz band. All received packets were stored for later analysis.

1) Connectivity: At first we take a look at the connectivity between the testbed nodes. The number of links was used as a measure. A link exists between two nodes if the packet delivery ratio (PDR) exceeds some arbitrary threshold which was in our case 0.5. Figure 4(a) shows the results for the 5 GHz band. With 37 nodes the maximum number of links is 666. When using a low MCS the number of links is up to 325. However, when using a high MCS together with 2 spatial streams (SM-MIMO)



(a) Channel utilization - 1 to 5 sender(s) and 1 receiver.



(b) Impact of transmission power.

Fig. 2. Channel Utilization and Power Control.

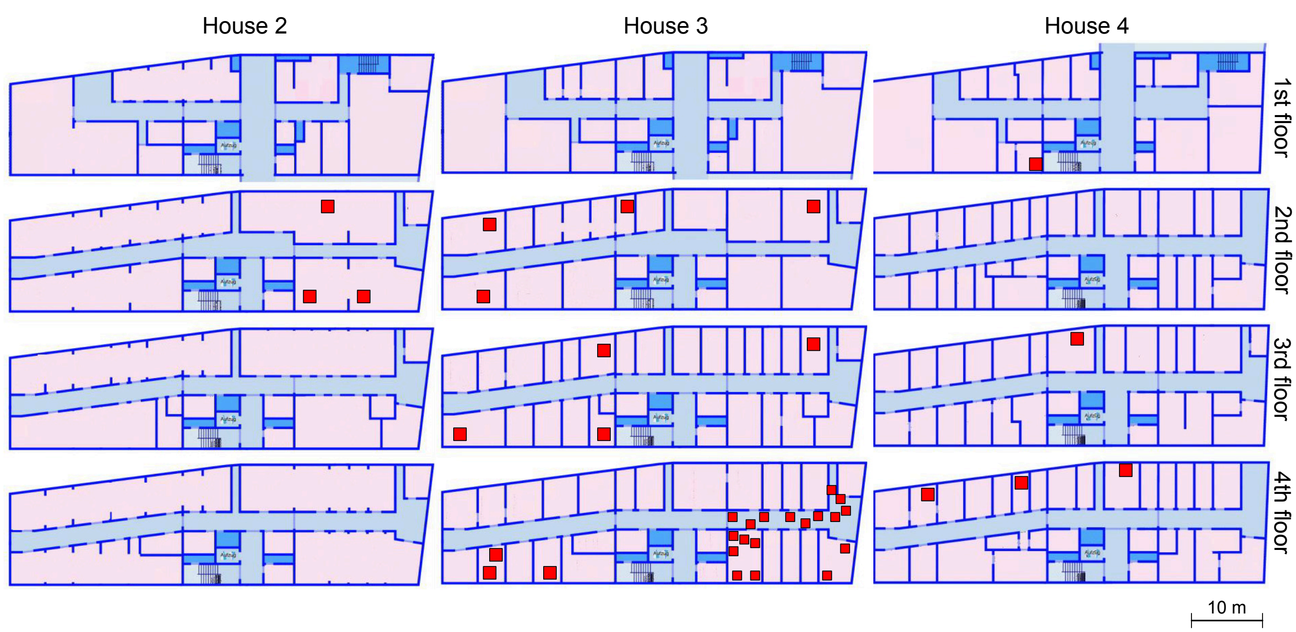
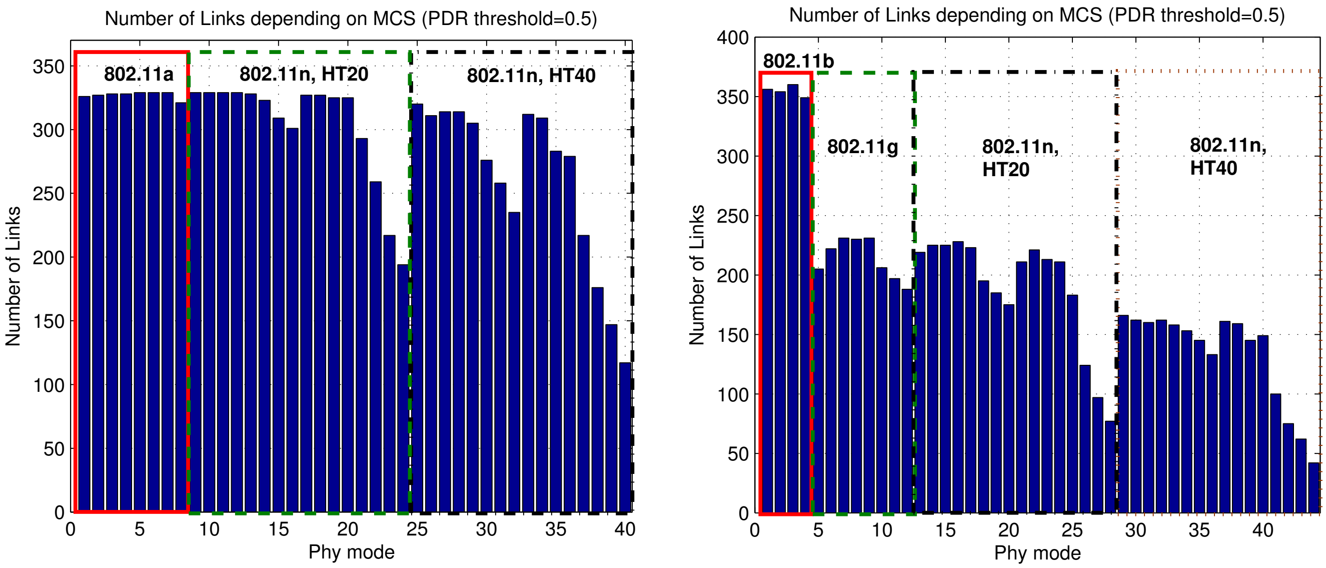


Fig. 3. Location of the 37 indoor nodes during the measurement.

the number of links decreases below 200. The impact is much greater when using wider channels - 40 MHz instead of 20. Here the number of links is only 120 or one-third of 802.11a.

Very interesting is the comparison with 2.4 GHz. From Figure 4(b) we can see that despite the reduced transmission power in 2.4 GHz the number of links when using 802.11b is slightly higher (360 vs. 325). This can be explained by the used single carrier modulation. The situation, however, changes when using 802.11g. Here the number of links never exceeds 225. When using 802.11n together with high MCS and wide channels the number of links decreases to 85.

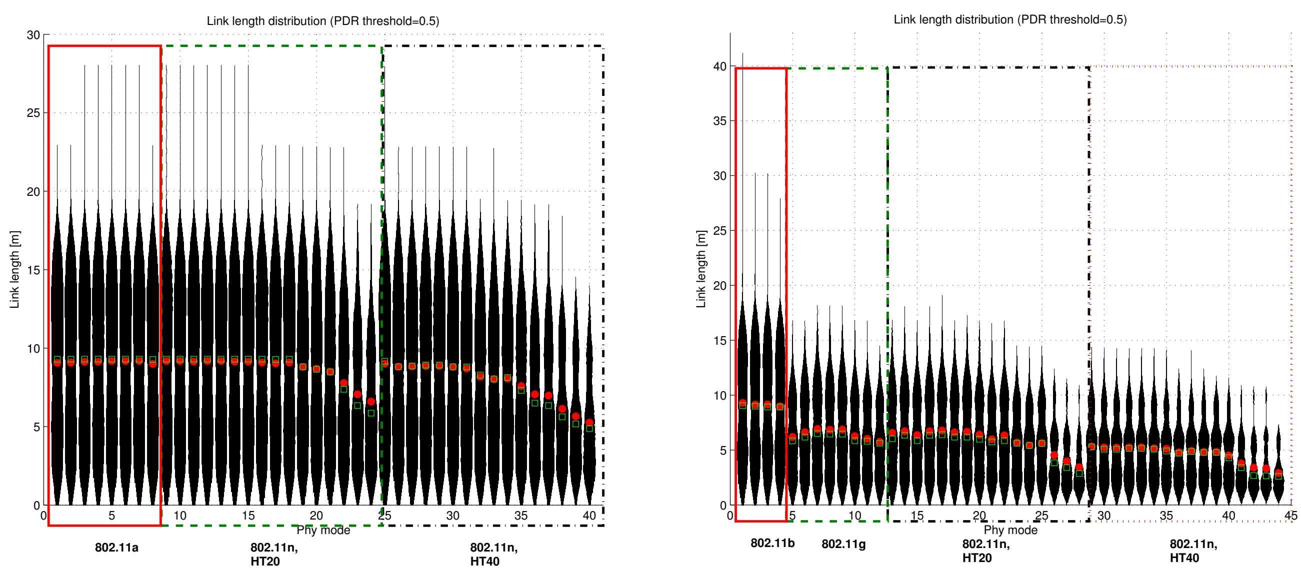
2) Link Length: To get a better understanding on the indoor coverage of 802.11n we estimated the link length for all link having a PDR of at least 0.5. In Figure 5(a) the results are given for 5 GHz. We see that the impact of the used MCS is small; there is only an impact for high MCS. The links at a 40 MHz channel are shorter compared to using 20 MHz channel. The results for 2.4 GHz band are given in Figure 5(b). We can clearly see that the 802.11b MCS are offering the longest links.



(a) Number of links at 5 GHz.

(b) Number of links at 2.4 GHz.

Fig. 4. Number of Links.



(a) Link length at 5 GHz.

(b) Link length at 2.4 GHz.

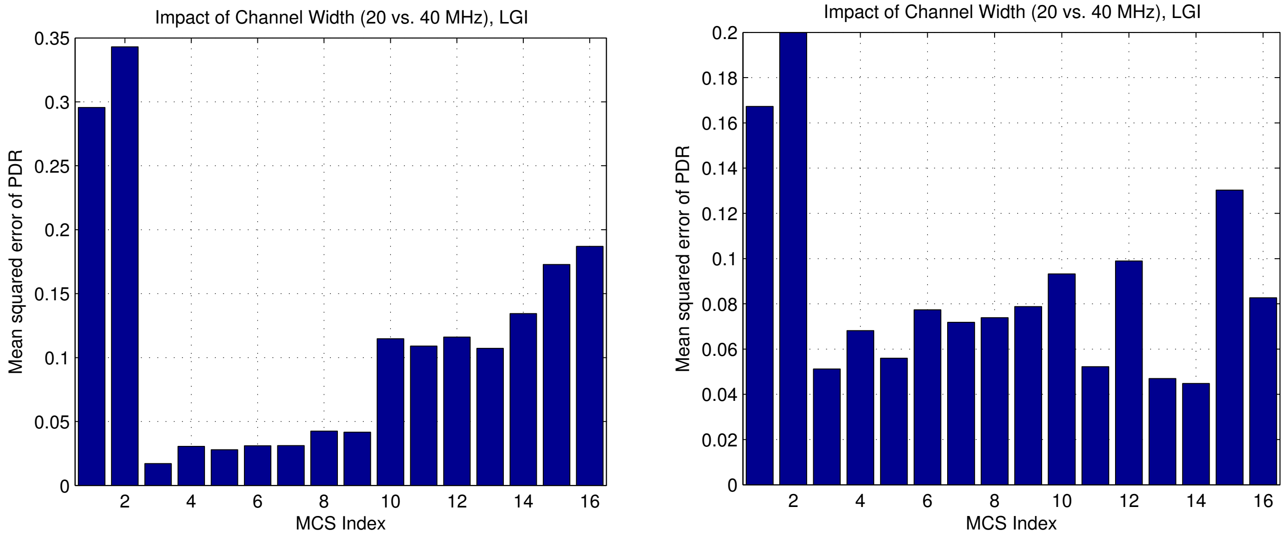
Fig. 5. Length of Links.

The link length for a high MCS together with a 40 MHz channel reduces to only 2-3 m on average. There are only a few links having a length of 7 m or more. This is a very disappointing result.

3) Impact of wider Channel: Next the impact of a wider channel is analyzed. Therefore for each link we computed the PDR using the 20 and 40 MHz channel. The mean squared error (MSE) for each MCS index is presented. The results for 5 GHz are given in Figure 6(a). In general we see that there is a significant impact which is higher when using a higher MCS with the exception of the 2 lowest MCS (BPSK and QPSK with FEC 1/2). The situation is similar in 2.4 GHz (Figure 6(b)) where the MSE of the PDR is high for the first two MCS indexes. Compared to 5 GHz the MSE is a little bit smaller.

4) Impact of Guard Interval: 802.11n offers the possibility to use a more efficient OFDM guard interval - $0.4\,\mu s$ (SGI) instead of $0.8\,\mu s$ (LGI) which effectively increases the throughput by up to 12%. A SGI is sufficient for environments with a small maximum delay spread due to multipath like our indoor environment[5]. However, our results show a significant impact

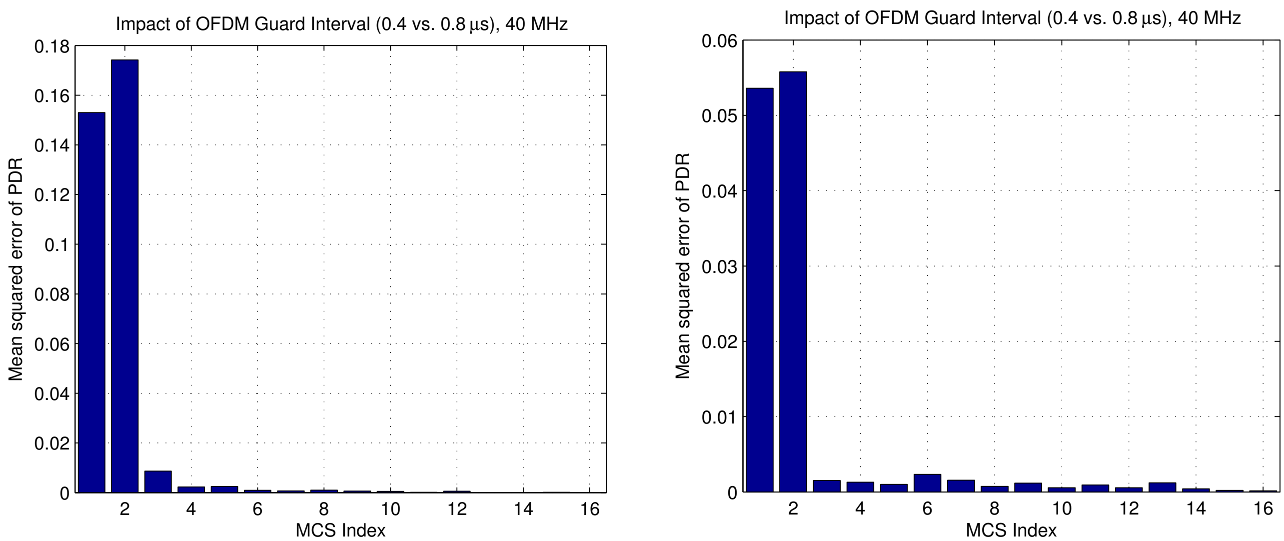
[5] A guard interval of $0.4\,\mu s$ is able to counter inter-symbol interference as long as the difference between the longest and the shortest path does not exceed 120 m.



(a) Impact of channel width at 5 GHz.

(b) Impact of channel width at 2.4 GHz.

Fig. 6. Impact of channel width.



(a) Impact of guard interval: LGI vs. SGI at 5 GHz.

(b) Impact of guard interval: LGI vs. SGI at 2.4 GHz.

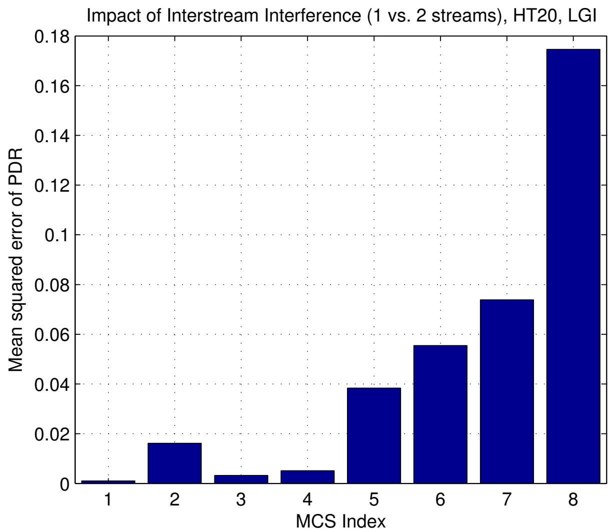
Fig. 7. Impact of guard interval.

for the two lowest MCS (Figure 7(a)). Links using the two lowest MCS are the longest ones and it seems that the SGI is not sufficient here to combat inter-symbol interference. This is weird since the maximum link length never exceeded 30 m. For higher MCS the impact of the guard interval was negligible.

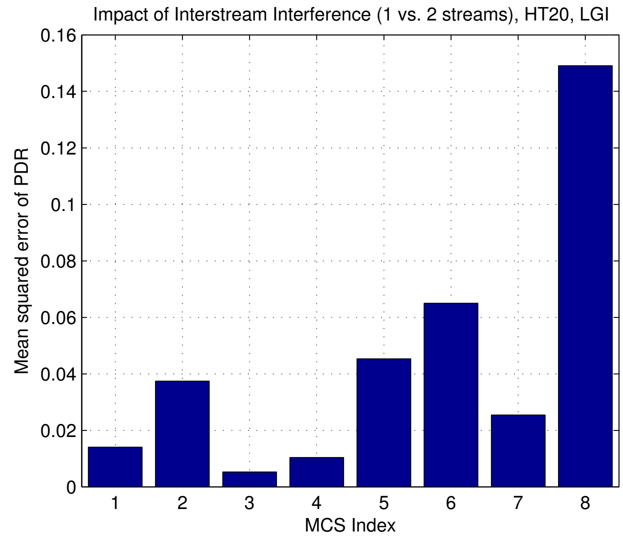
The situation in 2.4 GHz is very similar. Figure 7(b) shows similiar results compared with 5 Ghz with a slightly lower MSE.

5) Impact of Interstream Interference: 802.11n offers a mandatory MIMO mode called spatial multiplexing (SM). With SM it is possible to send multiple data streams using the same time / frequency resource. The receiver is able to decode multiple streams due to their unique spatial signatures. Even in an environment with lots of scatteres inter-stream interference (ISI) may occur. In this section we are trying to quantify this impact. Therefore, for each link we compared the PDR when using a single stream with two streams for each MCS. The results for 5 GHz are given in Figure 8(a). We see that the impact of ISI is small for low MCS but has some significant impact when using high MCS. Again, the situation in 2.4 GHz is very similar.

6) Relation between PDR and SNR: Finally we take a closer look at the relationship between SNR as reported by the WiFi driver and packet delivery ratio (PDR). Figure V-D6 shows the results for the 20 and 40 MHz channels in the 5 GHz band. During the experiment no external co-channel interference was observed, i.e. the channel was empty. The following observations can be made. For MCS index 0 to 11 there is a steep transition from PDR 0 to 1. For higher MCS index ($\geq$12)



(a) Impact of interstream interference: 1 vs. 2 streams at 5 GHz.



(b) Impact of interstream interference: 1 vs. 2 streams at 2.4 GHz.

Fig. 8. Impact of guard interval.

where a spatial multiplexing (2 streams) is applied the correlation between SNR and PDR is only weak. This is especially true for the 40 MHz channel. When comparing the results between 20 and 40 MHz channel for MCS index ≤7 (single stream) we can see that the transition area is thicker for the 40 MHz channel.

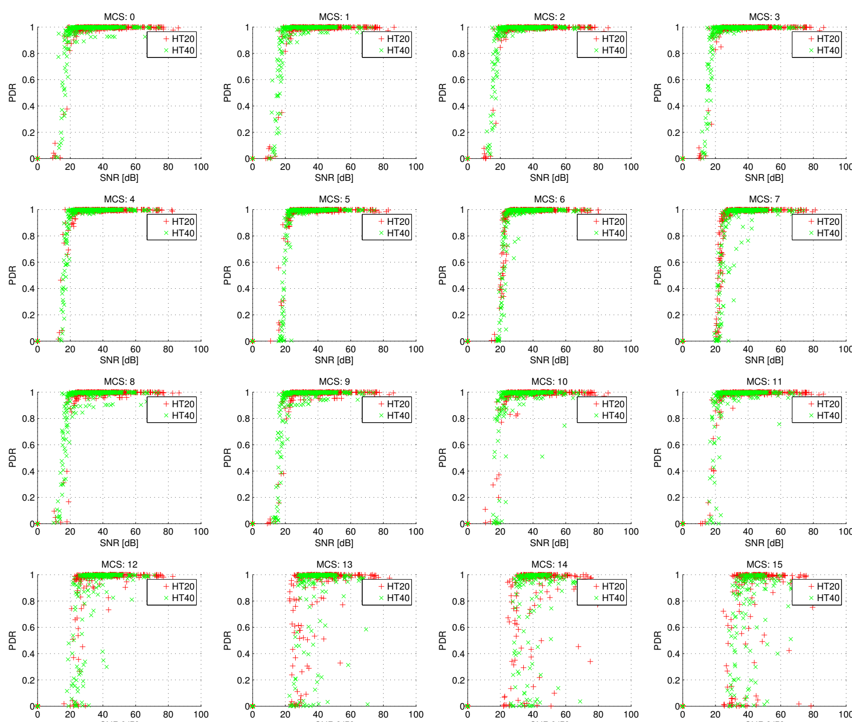


Fig. 9. SNR vs. PDR for HT20/40 in 5 Ghz.

The results for the 2.4 GHz band presented in Figure 10. In contrast to 5 GHz band we observed lots of external WiFi interference during the experiment. The following observations can be made. For MCS index 0 to 11 there is a steep transition from PDR 0 to 0.8 with a pronounced area (very thick) above PDR of 0.8. The 40 MHz channel needs more SNR to achieve the same PDR compared to the 20 MHz channel (shifted along x-axis). For MCS index ≥12 there is only a weak correlation between SNR and PDR, where with 40 MHz it was not possible to get an error free link.

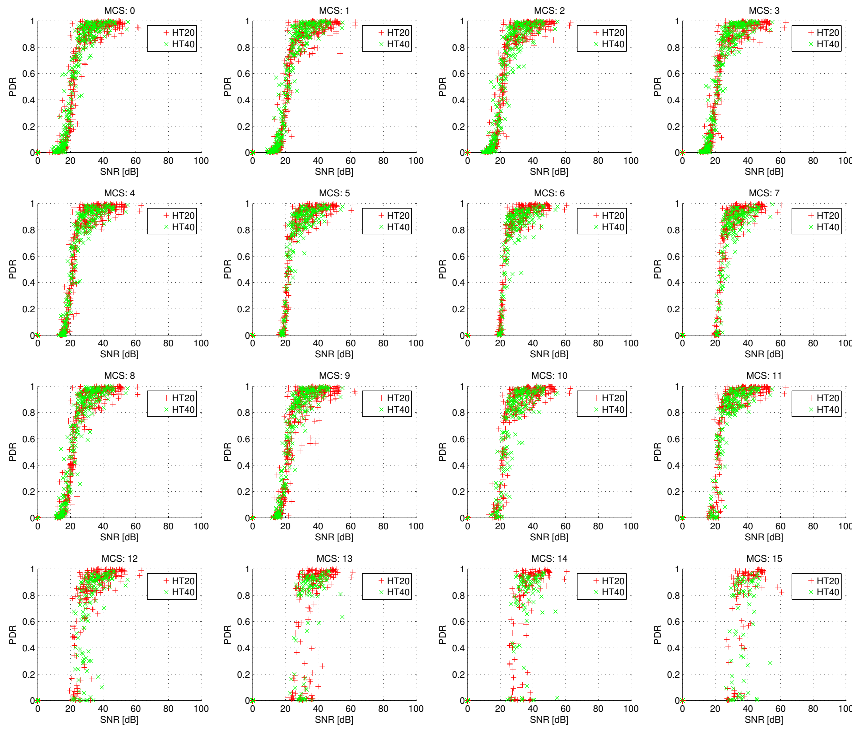


Fig. 10. SNR vs. PDR for HT20/40 in 2.4 Ghz.

VI. Conclusion

In this paper we identified requirements for an experimental mesh testbed. The proposed solution fulfilled most requirements. On the one hand the proposed solution is flexible because of the used open source software for the driver and the router API. On the other hand the hardware is inexpensive while still able to support the most important aspects of 802.11n and thus allowing a large-scale testbed deployment at a reasonable cost. However, we still identified some problems. The proposed solution was unable to achieve the theoretical performance results. From our investigation it emerged that the CPU or the memory was the bottleneck at high PHY rates especially when using small packets. The situation worsens when one wants to use both radios simultaneously (e.g. for multi-channel or backbone operations). Furthermore, the improved coverage promised by 802.11n could not be confirmed. In the 2.4 GHz band the coverage was worse than with 802.11b. The 5 GHz radio outperforms the 2.4 GHz radio due to the increased transmission power. Moreover, we were able to identify a significant impact from SGI, channel width and spatial multiplexing. Finally, a strong correlation between SNR and PDR is present only when using a single spatial stream or two streams together with a robust MCS. These aspects have to be taken into account when developing new protocols for mesh networks based on 802.11n.

VII. Related Work

Besides MIMO testbeds based on off-the-shelf 802.11n hardware [1], [8], there are also lots of solutions based on Software Defined Radio (SDR) [9], [10]. The radios in a SDR testbed is based on FPGAs or DSPs which allows modification also on the PHY layer, which broadens the area of research significantly. However, a solution based on SDRs is expensive, which makes it difficult to set up a large scale testbed. Off-the-shelf 802.11n hardware is a cheap alternative to SDRs which allows the evaluation of MIMO and its impact on higher layer protocols. The authors in [11] evaluated the impact of the different improvements in 802.11n on the throughput and the link quality. Consistent with our results they observed a major impact of frame aggregation on the throughput. Furthermore, they analysed the effect of interference when using channels with larger bandwidth.

VIII. Outlook

neue Protocole, speziell 802.11n

References

[1] "Wings." [Online]. Available: http://www.cs.wisc.edu/wings/

[2] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," 2005.

[3] 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, August 2000.

[4] T. Hühn, R. Merz, and C. Sengul, "Joint transmission rate, power, and carrier sense settings: An initial measurement study," New York, NY, USA, June 2010, p. 1..6.

[5] Q. Sun and P. Petrus, "Rate adaptation using error vector magnitude," Atheros Communications, Inc., 2010.

[6] "Linux-wireless," http://linuxwireless.org. [Online]. Available: http://linuxwireless.org

[7] "Radiotap," http://www.radiotap.org. [Online]. Available: http://www.radiotap.org

[8] "All wireless office mimo testbed." [Online]. Available: http://rvs.informatik.uni-leipzig.de/en/forschung/testbed.php

[9] P. Murphy, F. Lou, A. Sabharwal, and J. P. Frantz, "An fpga based rapid prototyping platform for mimo systems," Signals, Systems and Computers, 2003., 2003.

[10] T. Horseman, J. Webber, M. Abdul-Aziz, R. Piechocki, A. Nix, M. Beach, and P. Fletcher, "A testbed for evaluation of innovative turbo mimo-ofdm architectures," Personal Mobile Communications Conference, 2003., 2003.

[11] V. Shrivastava, S. Rayanchu, J. Yoon, and S. Banerjee, "802.11n under the microscope," Internet Measurement Conference '08, 2008.