Adaptive Preambles for Coexistence

Božidar Radunović, Ranveer Chandra, Dinan Gunawardena

June 2011

Technical Report
MSR-TR-2011-15

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Abstract

Wireless protocols in the unlicensed spectrum are developed for different requirements in terms of range and power, which makes them difficult to coexist in the same unlicensed spectrum. One such example is Zigbee and WiFi coexistence where low-power Zigbee nodes are frequently starved by WiFi nodes. Recent standardization efforts of short range IEEE 802.11af and long range IEEE 802.22 in the TV white spaces will make this problem more severe in the future.

In this paper, we propose a novel PHY and MAC protocol for coexistence. Our protocol is decentralized and simple. The key building block is an adaptive preamble support at the PHY layer that allows high-power nodes to detect a low-power transmission even when the difference in transmit power constraints between the two groups of nodes is as high as 20dB. We show that this technique can prevent starvation of low-power nodes in almost all existing scenarios. We further propose a MAC protocol that builds on CSMA MAC and exploits the adaptive preambles functionality. We extensively evaluate our system in a test-bed and in simulations. We show that we can improve the data rates of low-power links by as much as 10x over existing MACs, without sacrificing more than 20%-40% of throughput of the rest of the system.

1. INTRODUCTION

The openness of unlicensed spectrum has fuelled the growth of several new applications and devices over the last decade – from WLANs using WiFi devices, to Bluetooth headsets and low power Zigbee sensors, there are several devices using diverse standards that operate in the same unlicensed spectrum. This proliferation has led to the performance degradation of networks when they are in interference range of each other. While standards, such as IEEE 802.11, ensure that devices following the standard coexist with each other, there is very little support for coexisting with devices outside the standard. Carrier sense ensures that WiFi will not occupy the medium forever, and cede the medium to contending nodes. However, this is insufficient for coexisting with low power devices, such as Zigbee, since their signal might not reach the WiFi node. This problem will become more severe with the likely adoption of Zigbee in home appliances [3].

The coexistence problem will also become important with the recent interest in using the unoccupied TV channels (white spaces) as unlicensed spectrum. The FCC recently passed a ruling [4] to this effect in the US, and several other countries, such as the UK, Singapore and Brazil, are considering the adoption of similar rules. The TV spectrum is very attractive for wireless communication because of excellent propagation characteristics. It not only extends the reach of a transmission, it also enables faster transmissions at short distances because of higher SNR. Both these benefits lead to different applications. The former is useful in regional area networks (WRANs) as is enabled by the IEEE 802.22 standard. The latter is useful in in-home media distribution applications, which have been proposed by Dell, Phillips, and other companies, and will be enabled by the 802.11af standard [2]. Because the regional and in-home networks will operate at different power levels, there is a need to design a mechanism using which they can coexist on the same spectrum.

An additional challenge is that the coexistence between different devices needs to be managed, either in a distributed way or through a central, authorized entity. Decentralized architecture is a compelling approach for network design in unlicensed spectrum. WiFi, Zigbee and 802.11af are all decentralized protocols. However, enabling coexistence in a decentralized protocol is a difficult and open problem, and the issue exist in all aforementioned protocol examples.

In this paper we present Coexistence PHY and MAC design, a fully decentralized architecture for coexistence among nodes with different transmit power levels. We make the following contribution:

Coexistence PHY: We present a new mechanism, called adaptive preambles, using which low power devices can indicate their presence to high power devices without increasing their transmit power. Our technique is based on the observation that longer preambles can be detected at larger distances. We also propose a novel technique that minimizes false detection of interfering packets. Combining these two techniques, we are able to reliably detect low-power transmissions even when the difference between high-power and low-power transmission levels are as high as 20dB.

Coexistence MAC: We build on the adaptive preamble mechanism to design a novel MAC for coexistence. It is an enhancement to the existing CSMA MAC protocols, hence it is simple and fully distributed. Coexistence MAC is able to avoid starvation of low-power links while maintaining the network efficiency.

We have implemented our system on the Lyrtech SDR platform, and shown the feasibility of the adaptive preamble approach and evaluate the Coexistence PHY and MAC in a basic network setting. To evaluate our solution in larger scenarios, we have implemented it in QualNet. We compare our MAC with the existing distributed MAC protocols. Our results show that, unlike the existing distributed MAC protocol, Coexistence MAC avoids starvation of low-power flows and increases the throughput of the worse off links by as much as 10x.

2. PROBLEM DEFINITION

We consider decentralized wireless protocols, and we confine ourselves to the coexistence problem among networks that use carrier sense and operate at different power levels, such as Wi-Fi and Zigbee, city-wide and in-home white space networks [2], etc. When these networks use overlapping spectrum the performance of one of them, most likely the low power network, suffers. Such case is illustrated below, in Figure 1. We run a high-power and a low-power link using a WiFi MAC and move the low-power link on a table with wheels (the experiment setup is described in Figure 7.5), and we see that in many cases the low-power link gets no throughput.

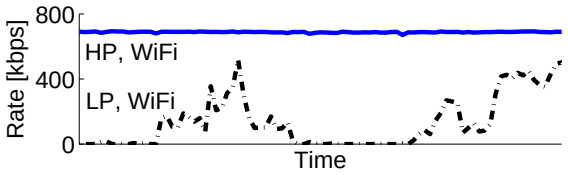

Figure 1: Starvation of low-power (LP) link in presence of high-power (HP) link.

The reason for performance degradation is that carrier sense works best when devices are in communication range of each other. RTS/CTS is a well known mechanism to improve performance when devices are not in communication range of each other. However, this technique does not work well when devices are operating at low power levels, since the RTS or CTS packet of the low power node might not reach the interfering devices in its vicinity. Jung et. al.[8] propose sending RTS and CTS packets at the maximum power. However, low power devices are usually energy constrained with insufficient hardware to send at the maximum power, or their power is constrained by regulators (as mobile devices are constrained in the white space networks [4]).

In this paper, our goal is to enable fair operation between all CSMA devices operating in the same spectrum. Since different nodes may use very different technologies that might not be able to talk among themselves and we seek to use only the minimum signaling necessary.

We divide all nodes according to their transmit power into two groups, low-power and high-power nodes. For example, in the white space scenario, we classify 100mW transmitters as low-power and 4W transmitters as high-power. Similarly, Zigbee nodes classify as low-power and WiFi as high-power.

We want to run the network efficiently and fairly. In particular, we want to avoid starvation of low-power nodes due to high power interference. We also want to avoid starvation of a high-power link when being exposed to a potentially large number of low-power neighbours active in its vicinity. There is no commonly accepted definition of fairness. Moreover, fairness and efficiency are conflicting goals [13]. Our design goal are, in the following order: (a) to avoid starvation of any link, (b) to give more priority to high-power links and (c) run the network as efficiently as possible.

In our design, we propose a novel adaptive preamble detection technique that is PHY-independent and enables signaling between low-power and high-power nodes even when the transmit power constraints are different by as much as 20 dB. We further build on this signaling technique and propose a fully distributed, reservation-based MAC based on CSMA design, which avoids starvation of low-power links and give more priority to high-power links. We use adaptive preambles to signal start of reservation periods for low-power links, and we use CSMA principles to achieve a distributed coordination among nodes.

3. ADAPTIVE PREAMBLES

In this section we present a new technique, called Adaptive Preambles, that allows high power nodes to detect the transmissions of low power nodes.

3.1 Key insight: adapting preamble size

Due to power asymmetry, the high-power node cannot detect the energy of the low-power one, and hence carrier sense does not work. We seek to repair carrier sense in a network operating at different power levels by detecting longer preambles. Indeed, preamble detection is used in digital radios for time synchronization, and it is also used for carrier sensing [7].

In order to avoid hidden terminal problem in non-uniform power networks, we need to reliably detect the presence of a transmission at low SNRs even below 0 dB. This is not possible with current preambles that are designed to operate at SNR levels when packet reception is possible, that is for SNR above 5 dB.

To gain intuition on how to extend the functionality of the preamble detection for lower SNR, let us first consider a generic detection problem. Suppose that we have a known preamble $(P_0, P_1, \cdots, P_{I-1})$ where each symbol in the preamble is $P_i \in \{-1, 1\}$. We wish to detect the preamble in the received signal.

Suppose that we receive a wireless transmission represented as a sequence of symbols $Y_0, Y_1, \cdots$. Each received symbol is

$$Y_i = S\,X_i + N_i, \tag{1}$$

where S is the transmit power, X_i is the transmitted symbol and $N_i \sim \mathcal{N}(0, \sigma^2)$ is the added white noise with variance σ^2.

The transmitter starts transmitting the preamble at some instant n, hence we have $X_{n+i} = P_i$, for all $0 \le i < I$. The assumption here is that the transmitter either trans-

mits a preamble, or it is idle, hence $X_{n+i} = 0$, for all $i < 0$ and $i \geq I$. We are interested in detecting whether preamble transmission has started at time n.

A standard way[1] of detecting the preamble is to *correlate* each observed subsequence $(Y_n, Y_{n+1}, \cdots, Y_{n+I-1})$ with the preamble and obtain

$$C_n = \sum_{i=0}^{I-1} Y_{n+i} P_i \quad (2)$$
$$= \sum_{i=0}^{I-1} S X_{n+i} P_i + \sum_{i=0}^{I-1} N_{n+i} P_i = X + N.$$

We say that a preamble is *detected* at time n if $C_n \geq IT$, where IT is some threshold (which we also take to scale linearly with the size of the preamble I).

If the sequence $(Y_n, Y_{n+1}, \cdots, Y_{n+I-1})$ indeed contains the preamble (hence $X_{n+i} = P_i$) then for each summand in the first part $X_{n+i}P_i = 1$. They all add coherently and we have $X = \sum_{i=0}^{I-1} S X_{n+i} P_i = SI$, where I is the length of the preamble. Variable $N = \sum_{i=0}^{I-1} N_{n+i} P_i$ is the sum of the I Gaussian random variables (the fact that they are multiplied with $+1$ or -1 does not play any role), hence N is also a Gaussian random variable $N \sim \mathcal{N}(0, I\sigma^2)$ with variance $I\sigma^2$. Consequently, $C_n = X + N$ has a mixed Gaussian distribution, as illustrated in Figure 2.

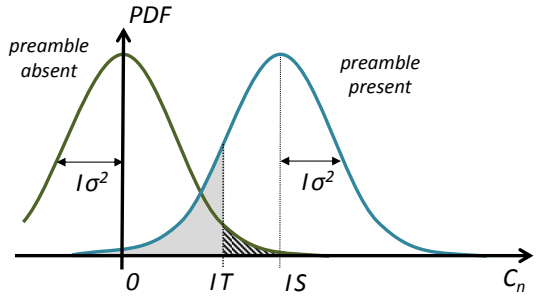

Figure 2: The PDF of C_n when the preamble is present and when absent.

There are two types of errors the detector can make. The first one is the *false positive*, meaning that it declares that the preamble is present at time n ($C_n \geq IT$) when in fact it is not. The second one is the *false negative*, meaning that the preamble is present at time n, but the detector misses it ($C_n < IT$).

The probability distributions of C_n when the preamble is present and when not are given in Figure 2. The probability of the false negative corresponds to the surface denoted with light gray in Figure 2, and it is

$$P_{fn} = \frac{1}{2}\left(1 + \text{erf}\left(\frac{I(S-T)}{\sqrt{2I\sigma^2}}\right)\right). \quad (3)$$

Similarly, the probability of the false positive corresponds to the dark gray surface and it is

$$P_{fp} = \frac{1}{2}\left(1 + \text{erf}\left(\frac{IT}{\sqrt{2I\sigma^2}}\right)\right). \quad (4)$$

From (3) and (4) we see that the enumerator grows linearly with the preamble length I, and the denominator grows sublinearly. Hence, we have the following observation

OBSERVATION 1. *For any fixed signal strength S and noise power σ^2, we can arbitrarily decrease the probabilities of false positives and false negatives by increasing the preamble size.*

A caveat though is that increasing preamble size increases the per-packet overhead, and should be carefully used.

3.2 Existing preamble detection algorithms

We next analyze the the existing preamble detection algorithms, and we focus on 802.11a/g as a typical example of a synchronization algorithms for OFDM PHY. The 802.11a/g preamble consists of 4 OFDM symbols (16μs overhead in total). The first OFDM symbol (which we call $\mathbf{A}$) is used for an adaptive gain control algorithm, the second (we call $\mathbf{P}$) is used for a coarse-grained time synchronization, and the last two are used for channel estimation[2].

The most common algorithm for time-synchronization in OFDM is Schmidt-Cox algorithm [14]. However, its performance can be significantly improved if correlated with a known preamble [15]. We base our algorithm on Tufvesson algorithm from [15] and we start by briefly reviewing its main idea.

One of the main issues when applying correlation to OFDM is that the transmitter's and receiver's clock are not synchronized, which introduces a random phase offset in the received signal. Thus, instead of (1), we have $Y_i = S X_i e^{i\theta} + N_i$ where θ is an unknown constant phase shift. The issue is that, in order to apply (2) directly, we would need to know the phase and use the modified coefficients $P_i e^{-i\theta}$ to cancel the phase effect.

In [15], the preamble $\mathbf{P} = (\mathbf{P}', \mathbf{P}')$ is divided into two equal parts $\mathbf{P}' = (P_0, \cdots, P_{S/2-1})$, where S is the number of samples per OFDM symbol. The correlation is then performed in the following way

$$C_n = \left(\sum_{i=0}^{S/2-1} P_i^* Y_{n+i}\right)\left(\sum_{i=S/2}^{S-1} P_i^* Y_{n+i}\right)^*.$$

where $*$ denotes complex conjugate. The main intuition is that, when the preamble is being transmitted, the first

[1] for this simple model, it is also the optimal approach

[2] The IEEE 802.11g standard also specifies the use of a long preamble, which is meant for communicating with legacy 802.11b devices that use 1 and 2 Mbps data rates

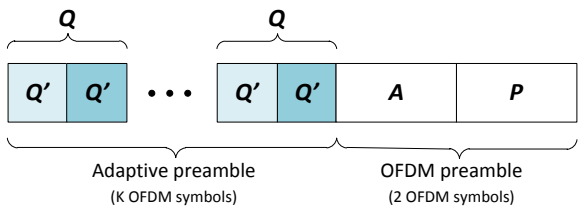

Figure 3: The structure of the PHY header, comprising an adaptive preamble and a standard OFDM preamble.

and the second sum will be the same since the first half of the preamble is the same as the second half. But since the second sum is conjugate, when multiplied with the first, it will cancel the unknown phase θ. For more details on the derivation, please see [15].

3.3 Adaptive preamble for detecting low-power transmissions

We next present our novel adaptive preamble design for PHY that (a) allows preamble detection at negative SNRs, (b) has an adaptive detection range, (c) is simple and cheap to implement (in terms of silicon area, and hence power consumption), and (d) works well in presence of interference. Our design has three distinct parts. The first part is a repetitive preamble, that allow detection at low SNRs, the second is a support for preamble length adaptation and the third is a detection mechanism that is robust to interference.

3.3.1 Repetitive preambles

One of the main issues when designing a correlator for detecting longer preambles is its complexity. In order to calculate the correlation in (2) we need to implement I multiplications with constant coefficients and an adder of size I. Hence the size of the correlator circuit scales linearly with the size of the preamble I, which is not desirable.

In synchronization algorithm design for OFDM, described in Section 3.2, the main goal is to identify precisely the time instant n (at sub μs precision) when the packet starts. Thus the preamble sequence $\mathbf{P}'$ is a pseudo-random sequence with as little auto-correlation as possible. In our application, in which the high power node only needs to detect (and not decode) a packet transmission, we are not interested in the detection with such an accurate timing. It is sufficient to detect the packet transmission with a timing precision even as high as a few tens of micro-seconds. In order to simplify the complexity of the receiver design, we propose to use repetitive preambles.

In our design, we use a symmetric preamble $\mathbf{Q} = (\mathbf{Q}', \mathbf{Q}')$ as the main building block. Sequence $\mathbf{Q}' = (Q_0, \cdots, Q_{S/2-1})$ is a pseudo-random sequence, different from the one used for timing synchronization ($\mathbf{Q}' \neq \mathbf{P}'$). S is the number of samples per preamble, hence the size of $\mathbf{Q}$ is one OFDM symbol. Our repetitive preamble $\mathbf{L} = (\mathbf{Q}, \cdots, \mathbf{Q})$ consists of K repetitions of $\mathbf{Q}$ and lasts K OFDM symbols. The preamble is illustrated in Figure 3.

Let $C_n^S = \sum_{i=0}^{S/2-1} (Q_i')^* Y_{n+i}$ be the correlation with one half preamble $\mathbf{Q}'$. We then correlate in the following way

$$C_n = \left(\sum_{k=0}^{K-1} C_{n+kS}^S \right) \left(\sum_{k=0}^{K-1} C_{n+S/2+KS}^S \right)^* . \quad (5)$$

In other words, we take the first halves of each of the K repetitions of the preamble (lightly shaded squares in Figure 3), and sum them. We sum the second halves as well (darker shaded squares in Figure 3), we conjugate the sum of the second halves and multiply with the first ones. Intuitively, with (5) we have achieved both goals: firstly, both factors in (5) grow with K, as explained in Section 3.1; secondly, by multiplying with the complex conjugate of the second, identical copy we get rid of the unknown phase bias, as explained in Section 3.2.

The reason for this particular way of calculating the correlation is the complexity. The sooner we calculate the sums and the multiplications, the less data we have to store in the delay elements for later processing. The complete design of the detector is given in Figure 4. To implement the design we only need a single correlator of length $S/2$ (the same complexity as a standard OFDM correlator). We implement the sums using three delay elements, as illustrated in Figure 4. Thus, for each of the four parallel correlators $K = \{2, 6, 10, 14\}$ (explained in the next section), we need three delay elements and one complex multiplier.

We also note that the design presented in Figure 4 is adapted to OFDM physical layers. However, the general idea described in Section 3.1 can easily be generalized to other types of physical layers and correlation algorithms.

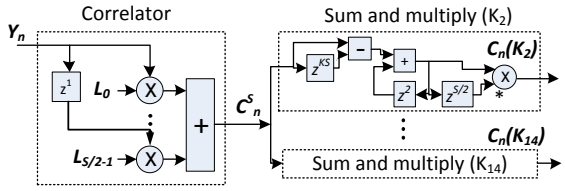

Figure 4: Adaptive detector for different preamble repetitions $K = \{2, 6, 10, 14\}$. Symbol z^n denotes a delay element of n cycles.

3.3.2 Adapting preambles

By choosing the number of the repetitions K, a transmitter can effectively control how far the preamble can be heard, and it is the responsibility of the transmitter

to choose the appropriate K (we present adaptive algorithm for choosing K in Section 4.2).

The detector at the receiver does not know apriori the number of repetitions K a transmitter has decided to use in the preamble. It has to be able to detect the preamble if possible, regardless of the choice of K. Therefore, the detector correlates with preambles of several different repetition lengths in parallel. We choose to implement four detectors in parallel, and set $K \in \{2, 6, 10, 14\}$. This gives us an appropriate balance between granularity of detection sensitivity and receiver complexity. We predefine four detection thresholds T_2, T_6, T_{10}, T_{14} for each possible of the values of K (the thresholds scale with K, as discussed in Section 6.4). The same set of thresholds is used by all high-power nodes. A preamble is *detected* if any of the four detectors exceeds its corresponding threshold.

It is important to notice that the detector does not need to know which value of K the transmitter has choosen. Suppose that for example the transmitter chooses $K = 10$ repetitions. If the correlator $K = 10$ detects the preamble, a detection is declared. If the correlator $K = 10$ misses the preamble, then it is very likely that the correlator $K = 14$ will also miss the preamble. This is because, in addition to 10 repetitions of the preamble $\mathbf{Q}$, correlator $K = 14$ also admits 4 other arbitrary OFDM symbols, and the correlation level is lower then expected for a preamble of length 14. Similarly, the correlators $K = 2$ and $K = 6$ are also likely to miss the preamble if $K = 10$ misses it. Hence, the performance of the correlation is as good as if we used only the correlator $K = 10$.

3.3.3 *High-power and low-power preambles*

Low-power nodes will use adaptive preamble to protect their ongoing transmissions, so it is important that the high-power nodes can detect the preambles even when the received signal is low. However, it is even more important to avoid false positives than is detecting a preamble when there is no ongoing low-power transmission. Any such false positive will decrease the efficiency of the network.

There are two types of false positives. The first one is a *false positive against the background noise*. This denotes an event when a correlation against background noise happens to exceed the correlation threshold. The other type of false positive is *the false positive against high-power interference*. High-power links are not allowed to use adaptive preambles in their transmissions. Still, a distant high-power transmission without an adaptive preamble may still happen to trigger the correlation.

We illustrate this observation with a simple experiment in our test-bed (described in Section 6.2). We position two nodes close to each other, and we put a variable attenuator at the transmit antenna of the transmitting node. In one experiment, the transmitter transmits adaptive preambles. In the second experiment the transmitter transmits regular OFDM packets without adaptive preambles. In the third experiment the transmitter is switched off and we only receive the background noise. In all three experiments we measure the maximum correlation value C_n observed in a fixed time interval. We plot each observed C_n against the SNR at the receiver (calculated from the corresponding attenuation values) in Figure 5. False positives have more negative impact on the system performance, hence we plot 90% for confidence intervals for the signal and 99% confidence intervals for the noise.

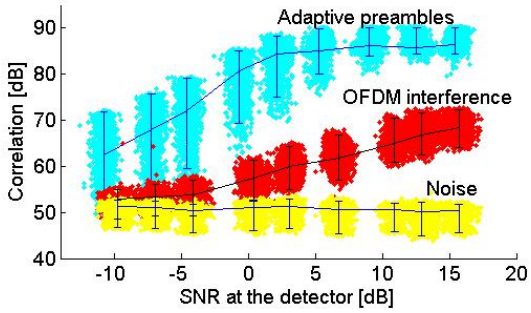

Figure 5: Correlation as a function of SNR for the loopback link and preamble length 10 OFDM symbols ($K = 10$).

We see that the correlation against the background noise, when the transmitter is idle, is very low. It visually seems that the signal can be reliably detected versus the background noise for SNRs all the way to -10dB (we postpone the detailed evaluation of this claim to Section 6.4). However, we also see that the correlation with the OFDM interference can be much higher.

There are two reasons for this. Firstly, the background noise and an interfering packet have different statistical properties. We see that when the power of the interfering signal is at the level of noise (0 dB point at the x axes in Figure 5), the correlation with an interfering packet is by approximately 5dB larger than the correlation with the background noise. Secondly, the received power of the interfering signal may be stronger than the noise. Indeed, as the power of the interfering packet increases, so does the correlation (due to a linear scaling with the signal level, shown in (2)). This is also clearly visible in Figure 5[3].

For example, suppose that a high-power interferer is positioned such that its signal at a high-power receiver is received at SNR = 2.5 dB We want to make sure that we do not detect these transmissions as adaptive preambles. Then, we want to set the correlation threshold above 99%-th percentile of the observed correlation with an

[3]The signal correlation stops the increase at high SNRs due to the saturation of the receiver's dynamic range

interferer at SNR = 2.5dB. According to Figure 5, this value is at about 65 dB. But the same value of the correlation threshold is above the lower 90%-th percentile of the observed correlation with the adaptive preambles at SNR = -5dB. Hence, if we set the threshold so high, we will not be able to detect the adaptive preambles at very low SNR.

To address this problem, we define a short preamble $\mathbf{H}$ for high-power nodes. It has the same structure as $\mathbf{L}$, but it is not adaptive and consists of two repetitions $\mathbf{H} = (\mathbf{R}, \mathbf{R})$, hence lasts $K = 2$ OFDM symbols. Preambles $\mathbf{L}$ and $\mathbf{H}$ are mutually independent pseudo-random preambles ($\mathbf{Q}$ and $\mathbf{R}$ are different). Each high-power transmission is prepended with $\mathbf{H}$ (similarly to low-power transmissions illustrated in Figure 3). Also, low-power nodes that don't need extra protection prepend $\mathbf{H}$ preamble.

In order to deal with the false-positives from interference, we propose the following detection algorithm

$$\begin{aligned} \textbf{Detected: if } & C_n^L \geq T^L \text{ and} \\ & C_n^H < T^H \text{ and} \\ & \text{RSSI } < \text{carrier sensing threshold} \end{aligned}$$

where $C_L(t)$ is the correlation with the preamble $\mathbf{L}$ and $C_h(t)$ is the correlation with the preamble $\mathbf{H}$, T_L, T_H are the corresponding thresholds, and RSSI is the received signal strength indicator.

The main intuition for this algorithm is as follows. The variance in correlation value, as seen in Figure 5, comes from random attenuation in the system (channel, noise, etc). False positive from high-power interference against $\mathbf{L}$ is more likely at times when the random attenuation is low, hence the interfering signal is high. But at the same time, the correlation with the $\mathbf{H}$ is also likely to be high. So the idea is that, if we detect *both* preambles $\mathbf{L}$ and $\mathbf{H}$ during a short time interval, we can conclude that it is an interfering packet and ignore it. Furthermore, we also ignore correlation if carrier is sensed.

4. MAC DESIGN

In this section we discuss an enhanced CSMA/CA MAC protocol that builds on the adaptive preamble capability of the PHY layer. Our goal is to design a decentralized MAC to enable coexistence, that is to avoid starvation of nodes while maintaining the efficiency of the network. Our MAC design consists of two parts. The first part is a reservation protocol for low-power nodes and the second part is an algorithm for adapting the length of the preamble.

4.1 Low-power Reservations

We start with CSMA DCF (802.11 MAC) protocol, as a simple and decentralized wireless MAC. The main challenge in deploying CSMA DCF to our scenario is the fact that the carrier sense does not work, as high-power nodes cannot sense the low-power ones.

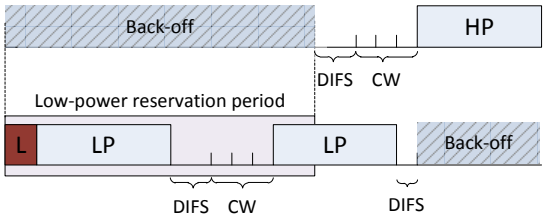

Figure 6: Illustration of the low-power reservation period (HP - high-power transmissions, LP - low-power transmissions, L - adaptive preamble).

To address this problem we propose to use the adaptive preambles to reserve time slots for transmissions of low-power links. Each time a high power node detects a preamble, it interprets it as a *low-power reservation* and refrains from transmitting for a fixed, predefined period of time. Once this period expires, the high-power node starts contending again.

Intuitively, this is a very simple time-division multiplexing algorithm. Since no high-power node is allowed to transmit during a low-power reservation, this can be interpreted as a fixed-duration time-slot reserved only for low-power node, as any TDMA MAC would do. Within a low-power reservation period, low-power nodes contend as usual, using carrier sense and back-offs.

Once a low-power reservation period expires, both high-power and low-power nodes contend for transmission. If a low-power node wins the contention, it sends an adaptive preamble. Then another low-power reservation period starts, and all high-power nodes back off. Otherwise, a high-power transmission starts, and low-power nodes sense it and back off.

To avoid starvation of the high-power nodes, low-power nodes are not allowed to start another low-power reservation period within an ongoing low-power reservation period. Even if a high-power node detects another special preamble during a low-power reservation period, it ignores it.

Another effect, illustrated in Figure 6, gives priority to high-power nodes. If many low-power nodes contend for access, it is very likely that someone will be transmitting at the very end of the a low-power reservation period. However, at this point, high power nodes start contending again, and decreasing the back-off counter. Thus it is very likely that a high-power link will gain access right after a low-power reservation period (but not always; the exact probability will depend on the number of contending high and low power nodes; this is illustrated in Section 7.2).

Note that the winning high-power transmission (HP

in Figure 6) may destroy the last low-power transmission extending beyond the reservation period (the second LP in Figure 6). However, this does not affect the high-power transmission and hence does not reflect to the efficiency of the network.

To avoid starvation of the low-power nodes, each low-power node contends separately for a low-power reservation period. A low-power period will start whenever any of the low-power nodes wins and sends an adaptive preamble. Hence, the more low-power nodes there are, the more likely they are to gain the access.

The duration of the low-power reservation should not be too large, to avoid over-booking the air if there are not too many low-power nodes. It should also not be too small, because each reservation period is preceded by a adaptive preamble, hence short periods incur high overhead. For a network with 802.11a/g PHY, we choose a standard low-power reservation duration of 600 μs. As we illustrate in Section 7, this is a good compromise between fairness and efficiency.

4.2 Algorithm for adapting preambles

Sending an adaptive preamble incurs overhead. It also prevents any high-power node that detects it from transmitting concurrently. It is thus important to carefully decide when to send adaptive preambles, and how long should they be. If a low-power node does not detect anyone interfering with its own transmissions, it should not transmit a long preamble. If it detects a high-power node that interferes with it, it should send the preamble just as long as necessary to prevent the particular interfering high-power node from transmitting in parallel, but it should not block any other node further afield.

We want to use packet losses as a feedback whether to increase or decrease (or switch off) adaptive preambles. However, we need to be able to distinguish losses due to a high-power interferer from other types of losses.

There are three primary reasons why wireless links will lose packets. The first one is due to interference from concurrent transmission. This is the case we want to protect against by using adaptive preambles. The second reason is MAC level contention. In DCF, as the number of nodes contending for medium access increases, so does the number of collisions that are due to two or more links starting transmitting at the exact same slot (and thus not having enough time to detect each other and avoid collision). If packets are lost due to contention, we do not want to use this as a signal to start using adaptive preambles or increase the preamble length. In fact, longer preambles can only make things worse by introducing additional overhead to an already congested medium. The third reason for wireless losses is the link loss, due to wireless channel changes. We do not want to switch on adaptive preambles or increase the preamble size due to these losses either.

We start by observing that in case of the second and third type of losses, we are not very likely to see many consecutive losses. For example, we measure that the average number of consecutive losses with 16 contending node is 2.5. Similarly, most modern rate adaptation algorithms are able to adapt the rate with minimal channel losses (less than 10%-15%), hence wireless losses are unlikely occur consecutively.

On the contrary, losses due to interference are very likely to occur in a sequence. Namely, if a high-power node has data to send, and if it does not hear a low-power node, it will continuously transmit and kill several subsequent low-power transmissions. Also, consecutive packet losses are particularly bad for the link performance, as they will exponetially increase the back-off counter, and cause link starvation.

We propose an adaptive preamble tunning algorithm based on the additive-increase, multiplicative-decrease (AIMD) principles. We use the number of consecutive packet losses as a measure of interference. The pseudocode of the algorithm is given below.

// After every packet transmission
$consecutive \leftarrow 0$;
$counter \leftarrow 0$;
if consecutive loss **then** $consecutive$++;

// After every packet transmission
if consecutive loss **then** $consecutive$++;
else $consecutive \leftarrow 0$;

if $consecutive = 6$ **then** $counter$++; $consecutive$++;
else $counter \leftarrow counter \times 0.9$;

if $counter \leq 2$ **then** $preamble\_size \leftarrow 0$;
else $K \leftarrow (2, 6, 10, 14)$;
. $preamble\_size \leftarrow K[\lfloor counter \rfloor - 2]$;

Due to the scale of the power difference between the low-power and the high-power nodes, low-power nodes that contend among themselves are likely to see the same high-power interferers. Although each of them runs the adaptive algorithm on its own, it is very likely that they will see the similar number of collisions with the same high-power interferers, and hence use similar level of protection within the same low-power reservation period.

We also note that the proposed algorithm will only kick in when detecting high-power nodes that persistently interferes with our transmissions and cause starvation to a low-power node. If the high-power nodes interfere only sporadically, we will not use the preamble protection. This in spirit with our design goals to avoid starvation of low-power nodes but give priority to high-power nodes and minimize the overhead and keep the network efficient.

5. IMPLEMENTATION

We build a test-bed with the Lyrtech Software Define Radio SFF-SDR platform [12]. It is a DSP and FPGA-based software defined radio platform with fully programmable PHY and MAC layers. We set the carrier frequency of the radio transmission to 500 MHz. We have obtained a test license to operate in the white space frequencies. Our channel bandwidth is 10 MHz.

We implement the adaptive preambles and the detection algorithm (as given in Figure 4) in FPGA, and we incorporate them in an OFDM receiver. This implementation applies directly to any single-channel OFDM PHY. It can be extended to the WiFi - Zigbee coexistence case by implementing a Zigbee receiver within a WiFi device (as demonstrated in [17]).

We also implement the low-power reservation functionality (Section 4.1) in MAC. Thus, we are able to evaluate the Coexistence PHY and MAC in a real-world deployment. In order to test the scalability of our design, we further implement the full PHY and MAC in Qualnet. We use the measurements from Section 6.3 to tune the realistic PHY model parameters for long preamble detection.

6. EVALUATING ADAPTIVE PREAMBLES

6.1 Real-world Challenges

The main goal of this section is to evaluate the performance of the proposed adaptive preamble detection technique in a realistic setting. There are three main issues to deal with. The first one is the *fast fading*. During a transmission of a single packet the signal quality will vary very fast, and a few dBs around the mean. In general, this is not a problem for packet transmission due to channel coding. However, it is a problem for the detection. The preamble is relatively short and if the fast fade occures during its transmission, it may be missed, and the signalling will fail. We measure and model the effects of fast-fading on the detection.

The second major issue is the *multi-path propagation*. Different channel profiles with different multi-path propagation characteristics will cause different signal transformations. Some may deform the signal in such a way that it decreases the correlation: for two links with the same SNRs, the correlation may be different due to different channel characteristics. Changes in channel profile also cause slow-fading. We need to make sure that the detector will be able to detect the preamble in a wide range of possible channel conditions. We evaluate the detector's performance in a large number of scenarios and we show how to select a threshold, as a function of the number of repetitions.

The third issue is the *interference*. A detector should only detect the proper preamble, not an arbitrary interference sample, such as a regular OFDM packet. We evaluate the design of the detector, presented in Section 3.3.3, in presence of interference.

We next analyze the impact of the above effects on the performance of our detector. We start by describing the experiment, and then present the evaluation results.

6.2 Description of The Experiments

We deploy two nodes, one is a high-power node, transmitting at 4W. The other one is a low-power node, transmitting at 100mW. These power settings are selected according to the FCC ruling on White Space devices [4]. The difference in transmit powers is 40 times (16 dB). Similar power difference is between Zigbee devices (1mW) and WiFi devices (50mW).

The high-power node initiates the experiment. It sends a packet containing a sequence number and the length of the preamble used in the experiment. For every packet successfully received the mobile, the low-power node responds with an adaptive preamble whose length corresponds to the control information in the packet. The low-power node also logs the packet sequence number and the SNR (called RX SNR).

After transmitting a packet the fixed, high-power nodes spends 20ms trying to detects preambles. At the end of this period it logs the packet sequence number, the experiment parameters, and whether it has detected at least one adaptive preamble or not. The high-node also logs the SNR at which it has received the preamble (TX SNR). Note that in many cases it cannot log the TX SNR accurately because it is below zero. In these cases, we estimate the TX SNR based on the RX SNR and the difference in transmit powers.

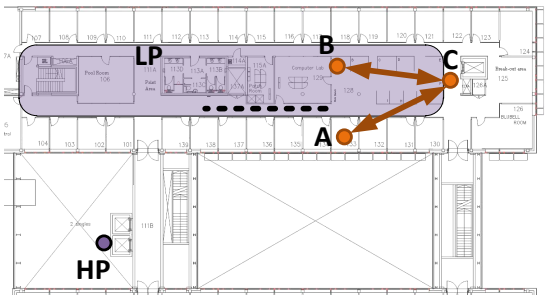

Figure 7: Floorplan of our building with the locations of measurements. Building dimensions are approximately 60m $\times$ 40m.

We perform two groups of measurement. The first group are controlled measurements on three carefully choosen locations, performed at night. There are two reasons we do the controlled experiments. The first one is that in each TX-RX pair of locations we can limit the randomness due to channel changes, and we are able to compare the effects of channel profiles accross the experiments (line-of-sight versus no line-of-sight). The

second one is that the signal strengths throughout the controlled experiments are fairly constant. We use variable attenuators to attenuate the transmit power of the low-power node, and this allows us to obtain accurate measurement of received SNR even when the SNR is below 0 dB.

The first controlled experiment is a *loopback* link, where both the high-power and the low-power nodes are placed at the location A in Figure 7, very close to each other. The second one is *line-of-sight* (LOS) link, where high-power node is placed at position C and low-power node at position B. The area in between is an open plan, and there is a clear direct path between the two nodes. The third one is the *no line-of-sight* (NLOS) link, where high-power node is placed at position C and low-power node at position A. Half-way between A and C there is a large metallic cupboard that blocks the line-of-sight between the two points.

The second group of measurements we call *real-world measurements*. These are measurements performed at different locations with the low-power node, moving slowly all around the shaded area on the first floor of our office building. We place a high-power node in the atrium on the ground floor, on the location denoted with HP. Floors and walls towards the atriums are made of glass with traces of metal that severely attenuate electro-magnetic waves. We run the correlation experiment continuously while moving the LP node. We also perform real-world experiments during night time to have a stable channel response during each subset of experiments. However, we vary nodes' placement to evaluate the performance over a large number of different realistic channel responses.

6.3 Effects of Fast Fading

We first look at the effects of fast fading. We observe a link in a static conditions (at night, no movement, no slow fading), and we analyze the effects of the fast fading. We first verify that the auto-correlation function of correlation C_n in time is very low, hence C_n can be assumed independent from the past realizarion. Moreover, for a specific channel profile, C_n can also be assumed i.i.d. Based on the measured correlations, we propose a model of the performance of the correlation with fast fading. We observe that $C_n \sim \text{Exp}(\lambda)$ is an exponentially distributed random variable[4] with parameter

$$log(\lambda) = k_1\, log(\text{preamb. len.}) + k_2\, \text{SNR [dB]} + f(\text{link}),$$

where k_1 and k_2 are fixed constants that are independend on the topology, and $f(\text{link})$ is a function of the actual channel profile of a link.

[4]This is expected: if the amplitude is distributed as Rayleigh, then the correlation C_n, a square of the amplitude, has exponential distribution

We demonstrate that the model fit very well to our measurements, as depicted in Figure 8 (a). As we see, for the two types of links the slope of the line is the same, hence k_1, k_2 do not depend on the channel profile. The horizontal position of the lines depends on the link parameter $f(\text{link})$. Note that, as the measurements are done at night with static channels, so we observe the same link parameter $f(\text{link})$ stays constant for each link throughout the entire measurement. We will use this model for our Qualnet simulations in Section 7.

6.4 Effects of Noise and Interference

We next look at the effects of noise and interference in the environment. The goal is to measure how does the correlation value depends on the received signal strength and to establish a reasonable value of the correlation threshold. We start by looking at the level of correlation for different SNRs for a fixed length of the preamble of 10 OFDM symbols. The results are presented in Figure 5 (b) and (c), and discussed in Section 3.3.3.

We next analyze how to set the threshold level T to have an acceptable level of false positives and in Figure 8 we plot how the observed correlation values C_n depend on the preamble length K. False positives can have severe negative impact on the performance. We want to guarantee there will be no false positive detection of the noise. We plot the maximum correlation value triggered by the noise and we want to set the threshold level T higher than the maximum level. We also want to minimize the false positives triggered by interference. We plot the upper 99-th percentile. This represent the value of the threshold T we need to select to see at most 1% of false positives generated by interfering packets. Finally, we plot the lower 10-th percentile for the signal as the value of the threshold T we need to select to see at most 10% of false negatives when actually receiving the preamble.

From Figure 8 we observe that all percentiles approximately behave as $O(K^3)$. We select the correlation threshold $T = 150K^3$ (where the constant depends on the actual parameters of our hardware) as the lowest threshold where we do not see any false positive correlation with the white noise. We see that with this threshold we can easily detect signal with 90% accuracy at as low as -9 dB (for the loopback scenario, preamble length $K = 10$ or more).

However, we also see that we need to set the threshold much higher in order to avoid correlating with the interference, which can severely impact the performance of our receiver. This problem is addressed in Section 3.3.3. We next evaluate the detection algorithm from Section 3.3.3 and present the results in Figure 9. We gather all interference packets transmitted during our real-world experiment with preamble length $K = 10$. On the x axis we

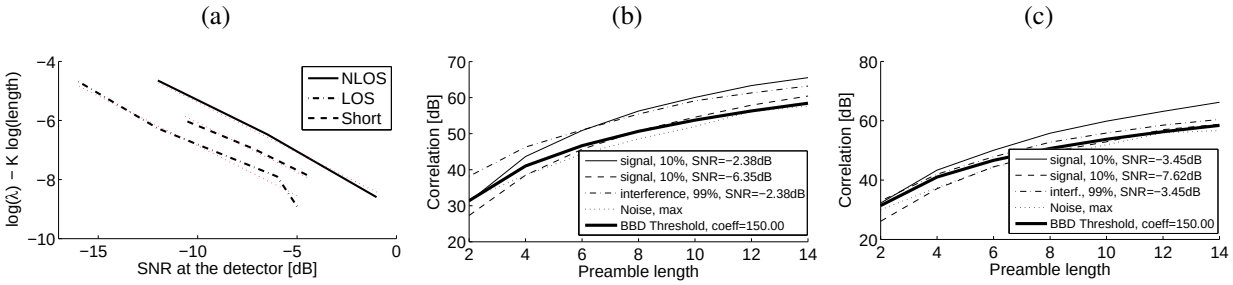

Figure 8: **(a): Fitting the correlation C_n as a linear function of SNR (in dB) and log-linear function of the preamble length; (b) and (c): Percentiles of the observed correlation values as a function of preamble length K for different SNRs for the (b) LOS link and (c) NLOS link.**

plot the correlation with the **L** preamble and on the y axis with the **H** preamble. The lower shaded box is the area of false positives - a packet that detects the high preamble but fails to detect the low preamble. The observed probability of false positive is 10^{-4}.

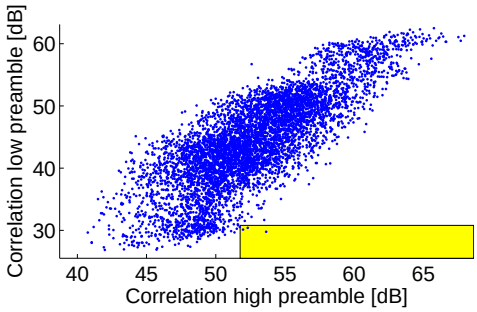

Figure 9: **False positives with H and L preambles.**

6.5 Effects of Multi-path fading

Finally, we look at the effects of channel profiles. During our real-world experiments we collect correlation information for a large variety of channel profiles. Since the experiments involves constant channel changes and the received SNR is very low (below 0dB), we cannot measure the signal strength TX SNR at the high-power node as we did previously in the controlled experiments. However, the signal strength measured at the low-power node, RX SNR, is the metric of interest. This is equivallent to the level of interference the low-power node will experience from the high-power transmission. Therefore, we plot the probability of detection as a function of RX SNR.

We divide the range of the recorded RX SNR into bins, and calculate the aggregate signal detection rates for the entire duration of the experiment for each bin, for different preamble length. This is plotted in Figure 10. Note that the curves are not strictly decreasing in RX SNR because different channel profiles have different performances for the same RX SNR.

We see that for adaptive preamble lengths $K = 10 - 14$ for RX SNR > 3 dB, we can have detection probability of 70-80%. A typical carrier sensing threshold is of the same order, so we see that the virtual carrier sense can guarantee the same level of interference protection as the real carrier sense, with 70%-80% success rate.

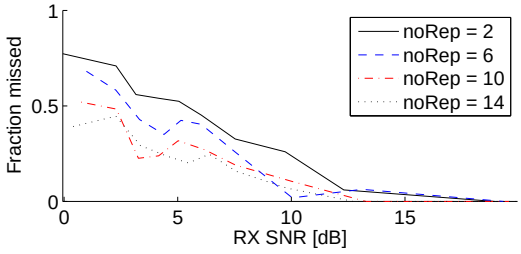

Figure 10: **Average number of missed adaptive preambles as a function of RX SNR and the preamble length K.**

Next, we look at how the probability of detection depends on link conditions. We divide all the received packets in batches of 20. Since the packets are sent back-to-back, we see that all packets within a batch see the same RX SNR. For each batch we record a single probability of detection (out of 20 packets in a batch). We then aggregate batches with the similar RX SNR. We plot the CDF of the probability of misdetection for different RX SNR in Figure 11. Clearly links with different channel profile will take different position on these curves. However, we see that except for the 20% most unfavorable links can have more than 70% success detection rates with RX SNRs as low as 3.5 dB (meaning we are able to prevent interferers with SNR above 3.5 dB).

Finally, the question is whether a high-power interference will affect the reception of low-power packets in cases when preambles are missed. We defer this discussion to Section 7.5 where we evaluate Coexistence PHY and MAC in the test-bed.

7. EVALUATION OF MAC

7.1 Description

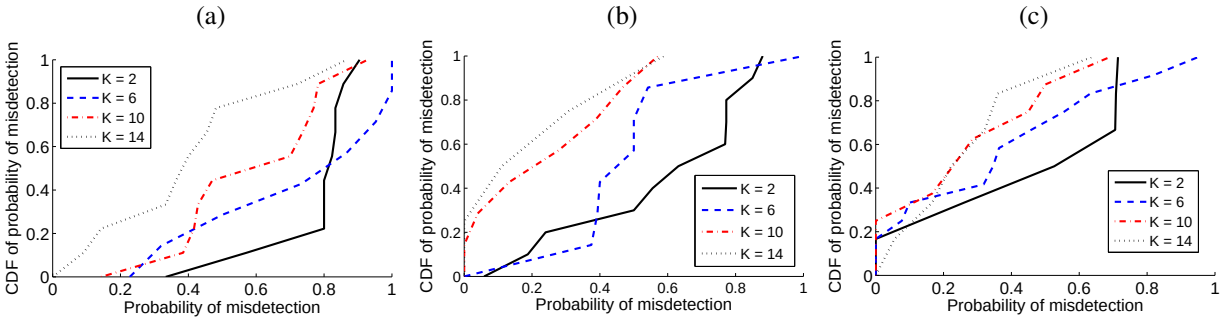

Figure 11: **CDF of probability of misdetection across different channel profiles, for different RX SNR: (a) RX SNR = 1.5 dB, (b) RX SNR = 3.5 dB, (c) RX SNR = 4.5 dB.**

We evaluate our MAC design in Qualnet. Where not otherwise specified, we fix the transmission rate to the maximum rate that yields packet loss rate below 5%. In Coexistence MAC we select the low-power preamble duration to be 600us. We verify in simulations that the overhead is less than 5% in all cases.

We use WiFi and frequency-division multiplex (FDM) as baseline designs to compare our MAC against. In our WiFi implementation, all nodes use the standard WiFi MAC, regardless of the transmit power. In our FDM implementation we divide the available channel into two sub-bands, half a bandwidth each, and assign one of sub-bands the to low-power and the other to high-power nodes. Within each sub-band, node compete using WiFi MAC. We double the DCF time-slot in the FDM implementation to keep the overhead proportional to the packet size.

7.2 Performance Examples

We first look at a few handcrafted scenario that illustrate different issues of the MAC design. The first scenario is *the distant links scenario*, illustrated in Figure 12, with $n = n_1$ and $n_2 = 0$. The power of the low-power nodes $P_{LP} = 0$ dBm is fixed, and we vary the power of high-power nodes P_{HP}. The transmission rates are the same for all links in the scenario, and we vary them across the experiments. We also vary the number of low-power nodes n, high-power nodes m and the distance d.

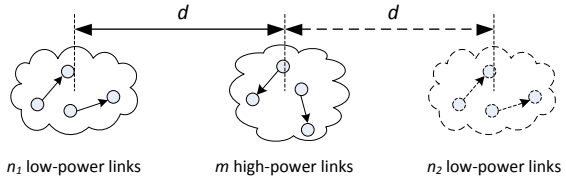

Figure 12: **Example topologies for MAC performance evaluation. Nodes in each cloud are very close to each other and can sense carrier of each others.**

In order to gain understanding on how Coexistence MAC resolve contention among links, we start by looking at the sum of rates of all low-power and all high-power links. This is depicted in Figure 13 (b) for $n = 4$ and $m = 1$.

When distance d is small, all high-power and low-power nodes can sense each other, hence Coexistence MAC behaves as WiFi MAC, with one exception[5]. When $d \geq 100$m, we see the standard hidden terminal problem. The high-power nodes can no longer sense the low-power nodes. Consequently, the high-power links grab the entire medium and starve the low-power links. This does not happen with Coexistence MAC. Once the Coexistence MAC detects several consecutive losses, its adaptive algorithm will start sending the long preambles. The long preambles will signal the presence of the low-power to the high-power nodes and prevent starvation.

Figure 13 (a) illustrates the flow data rates for the same scenario, for $d = 200$m. We see that WiFi completely starves the low-power links. By design, FDM shares only half of the bandwidth among low-power links. Coexistence MAC is able to give more throughput to each of the low-power links (the high-power link still gets more than each low-power link, as discussed in Section 4.1).

Finally, when $d \geq 300$m, the interference between the low and the high power nodes decreases to the point when both links can operate in parallel. The adaptive preamble algorithm of Coexistence MAC can detect it, and switch off the preambles, allowing the two links to fully use the spatial reuse. In this case, the rates achieved by Coexistence MAC and WiFi are the same. Frequency-division multiplex (FDM), due to its static nature, it is not able to exploit spatial reuse once $d \geq 300$m, and it becomes very inefficient when compared to Coexistence MAC and WiFi.

The second scenario is called *the exposed terminal scenario*, also illustrated in Figure 13 (c), with $n_1 =$

[5]at $d \approx 60 - 80$m, WiFi starves the high-power link due to the EIFS effect

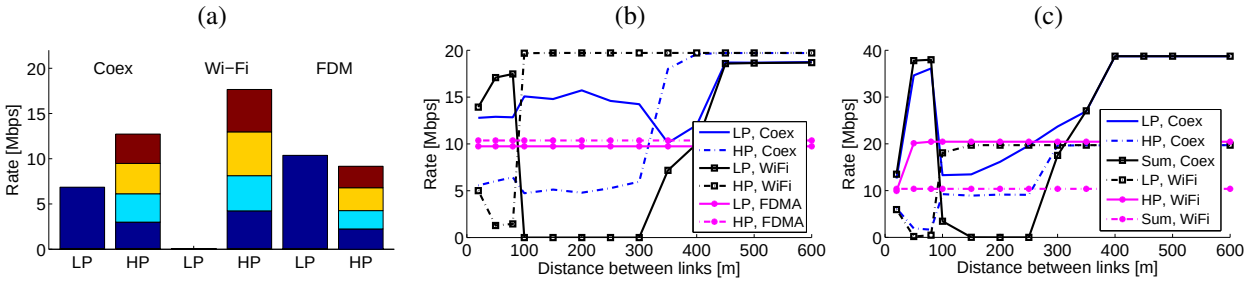

Figure 13: Achieved rates for different scenarios for Coexistence MAC, WiFi and FDM. In all cases $P_{LP} = 0$ dBm, $P_{HP} = 16$dBm and transmission rates are 36 Mbps. Scenarios: (a) Per-link rates for distant link scenario, $n = 4, m = 1$, at $d = 200$m; (b) Sum of low-power and high-power rates for the same scenario, $n = 4, m = 1$, for various d; (c) Sum of low-power and high-power rates, exposed terminal scenario ($n_1 = n_2 = 2, m = 1$).

$n_2 = n/2$. For $d < 100$m Coexistence MAC behaves the same as WiFi MAC, and the high-power link suffers the exposed terminal problem[6]. For $d \geq 100$m, when Coexistence MAC starts using adaptive preambles, it succesfully avoids the exposed terminal problem using the low-power reservation scheme. WiFi starves the low-power links. For $d > 300$m both WiFi and Coexistence MAC outperform FDM due to spatial reuse.

7.3 Performance on random network topologies

Finally, we evaluate the performance of our MAC on a set of random topologies. We consider a 1km $\times$ 1km square area. We randomly place $n = 10$ low-power links and $m = 2$ high-power links in the area. High-power links transmit with 4W transmit power and low-power links transmit either with 40 mW or 100 mW (as in the White Space scenario [4]). Link lengths are selected randomly, but in such way that the achievable link rate is at least 12 Mbps. For each link we select the highest link rate that can achieve less than 5% packet loss rate in isolation. We run a single FTP flow over each link. Flows are single hop. All packets sizes are 1000B.

We select 10 random network topologies. For each topology we execute five runs and each run lasts 20s. We calculate the average rates of flows over all five runs for each topology. The confidence intervals are small and we don't plot them.

In Figure 14 (a) we plot the TCP rate of the flow with the smallest rate in the network. We see that the smallest rates of the high-power flows are comparable for all three MAC designs. However, Coexistence MAC assigns significantly better rate to the smallest low-power flow. Both WiFi and FDM MAC yielded zero throughput to at least one flow in 90% of the cases. This did not happen with Coexistence MAC.

We next look at the number of starved flows, that is the number of flows that achieve TCP rate lower than 100 kbps. This is depicted in Figure 14 (b). We see that in 70% of cases Coexistence MAC does not starve any low-power flow. FDM on average starves around 2 low-power flows (20% of flows), and WiFi around 4 low-power flows (40% of flows). We also note that WiFi starves one of the two high-power flows in 20% of the cases.

WiFi is particularly bad in starving low-power links because high-power nodes cannot detect them. FDM also starves some low-power links because it reserves only half of the bandwidth for 10 low-power links. By design, Coexistence MAC is able to dynamically share the capacity among available links, regardless of how many low-power or high-power links are present.

Finally, we look at the total network throughput for the three MAC protocols. It is well known that efficiency and fairness are conflicting design goals [13]. Coexistence MAC is designed to avoid starvations. Hence, it will occasionally guarantee medium access to a particularly exposed link, at the expense of the total system throughput. It is thus expected that Coexistence MAC will yield lower total network throughput than WiFi or FDM. This is confirmed in Figure 14 (c). However, we see that the loss in total throughput is modest, from 20% to 50%. Hence we are able to maintain a reasonable efficiency for the network while avoiding starvation of links.

7.4 Performance of Zigbee with Coexistance PHY and MAC

We also look at if Zigbee starvation can be prevented with the Coexistence PHY and MAC. We implement Coexistence MAC with adaptive preambles on the top of Zigbee PHY in Qualnet. We randomly place 2 WiFi links and 10 Zigbee links. Each Zigbee link has 1.2Kbps CBR traffic demand and WiFi links run FTP transfer. We plot the CDF of the number of starved links in Figure 15. We see that CSMA MAC (denoted with WiFi)

[6]The WiFi exposed terminal problem is out of scope of this paper.

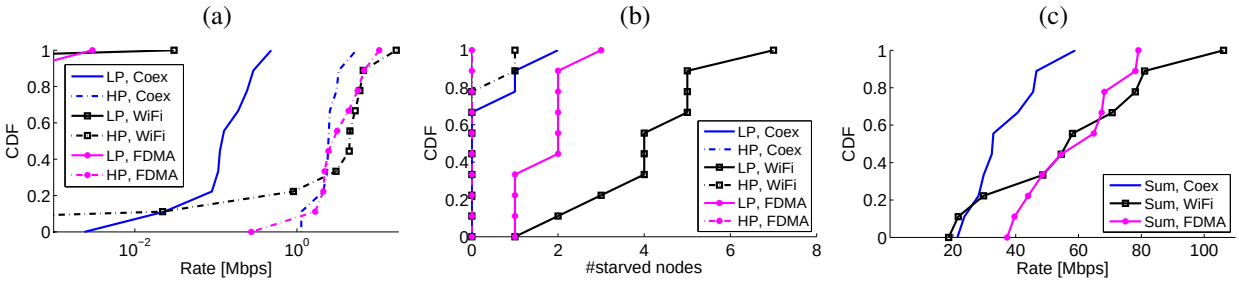

Figure 14: **Performance for random scenarios: (a) TCP rate of the flow with the smallest rate; (b) number of starved flows in the network (flows with the TCP rate less than 100 kbps); (c) Sum of TCP rates of all flows in the network**

.

starves more than two links on average. Coexistance MAC only starves one out of 50 runs starved a Zigbee

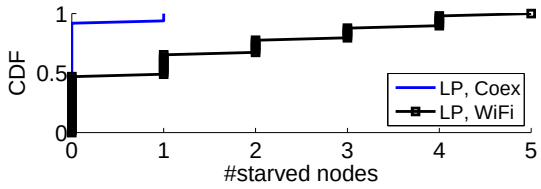

Figure 15: **Number of starved Zigbee links for different random topologies with 2 WiFi (HP) and 10 Zigbee (LP) links.**

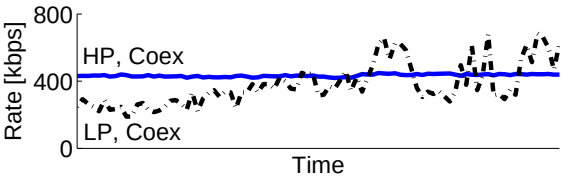

Figure 16: **Rates of LP (low-power link) and HP (high-power link) in the coexistence and WiFi experiment. The time axes of the two experiment are unrelated.**

7.5 Test-bed evaluation

Finally, evaluate Coexistence PHY and MAC in the test-bed. The goal of the experiment is to verify whether we manage to avoid starvation of low-power nodes. We position a 100mW transmitter and a receiver in the office denoted with A, and we put a 1mW transmitter and a receiver on a large table that we move away from link A along the dashed line depicted in Figure 7. In one experiment, called *Coex*, both high-power (HP) and low-power (LP) link implement Coexistance PHY and MAC (except that the low-power link uses a fix preamble of length $K = 14$). In the other experiment, called *WiFi*, both links implement plain CSMA without adaptive preambles. The results for WiFi are depicted in Figure 1 in Section 2. The results for Coexistance PHY and MAC are depicted in Figure 16 .

We see that $K = 14$ preambles are enough to protect the low-power link along the entire path (until LP cannot detect the HP anymore). Moreover, preambles with length $K = 14$ are too long for this particular scenario so HP does not fully exploits the spatial reuse in the right side of the path (we address this problem with the adaptive preambles algorithm, which we evaluate with Qualnet simulations). We also see that without the Coexistence PHY, the low-power link would be starved.

8. DISCUSSION AND FUTURE WORK

Our adaptive preamble design is adapted for OFDM PHY, as one of the most common PHY. However, since we use time-domain correlation, a similar design could be used to detect other types of PHY preambles, such as for example Zigbee or CDMA preambles. The general design principles outlined in Section 3.1 would hold for almost any PHY design.

In our design we have deliberately kept the adaptive preamble separate from the rest of the packet. However, it is easy to see that they might be integrated with the existing OFDM PHY preamble. For example, the current WiFi preamble has four OFDM symbols (Section 3.2). The adaptive correlator could be design to look for the existing OFDM preamble (in particular in case of **H** preamble) and decrease the preamble overhead, although at the expense of the increased correlator complexity.

We also note that our design can be extended to beacon-enabled Zigbee networks (with superframes). In this case the coordinator would contend with the high-power nodes and send the adaptive preamble. The super-frame would coincide with the low-power reservation period.

Adaptive preambles could also be used for different MAC designs. One simple example is a strict prioritization. One could standardize the adaptive preambles for primary users in a cognitive network (e.g. wireless mics), and mandate that the secondary users have to sense and avoid them. This would greatly improve the sensing performance over the secondary users. It would also allow secondary users to multiplex with the primary user at the packet level.

9. RELATED WORK

Most prior work on coexistence can be divided into two groups. In the first one, low-power nodes try to avoid high-power interference. One idea is to use high-power signalling packets to avoid starvation (c.f. [8]). This is impossible in our setting due to power and regulatory constraints. Another idea is to use gaps in WiFi transmissions to opportunistically transmit Zigbee packets (c.f. [5] and the references therein). However, this approach does not prevent starvation of Zigbee nodes in a saturadet WiFi network. Overview of works on Bluetooth vs WiFicoexistence problem can be found in [11]

Another approach is to have high-power nodes be aware of the low-power one and avoid interfering with it. One example is IEEE 802.11h standard [6] that imposes power control to WiFi nodes to avoid interference with satellite communications. Another example are cognitive radio network (IEEE 802.22) where secondary users need to sense and back-off when primaries are present. However, the time-scale of the two approaches is much larger than a packet duration, hence they are less efficient than Coexistance MAC.

Various packet detection techniques based on signal samples have been proposed, e.g. in [18, 1] (including preamble-detection based carrier-sensing [7]), but none of them work at SNRs below 0dB. Cyclostationary detection is a technique used to detect primary signal in cognitive radios at low SNRs. They look at generic features of a signal, hence the detection times are of order of milliseconds [10]. Our detection is several orders of magnitude quicker.

Several papers consider adapting the carrier sensing threshold [16, 9]. However, these techniques cannot be used when the SNR is below the noise floor.

10. CONCLUSIONS

In this paper we have presented a design, implementation and evaluation of a fully decentralized PHY and MAC for coexistence between low and high-power nodes. It avoids starvation of either low-power or high-power links, and it only slightly drops the efficiency of the network. The main components of our PHY design are adaptive preambles. High-power nodes are able to detect low-power preambles at very low SNR levels, hence low-power nodes are able to signal when transmitting. Our MAC layer builds on the adaptive preambles and allows for a distributed coordination between low-power and high-power nodes.

We implement and evaluate our design in a software-designed radio test-bed and in Qualnet simulator. We show that the adaptive preamble detection is able to achieve succesful signalling even when the power difference is as much as 20dB. We also show that Coexistence PHY and MAC avoids starvation in all cases, unlike the existing distributed MACs.

11. REFERENCES

[1] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with wi-fi like connectivity. In *Sigcomm*, 2009.

[2] H.-S. Chen and W. Gao. Mac and phy proposal for 802.11af, 2010.

[3] J. Drake, D. Najewicz, and W. Watts. General Electric Report, Energy Efficiency Comparisons of Wireless Communication Technology Options for Smart Grid-Enabled Devices, December 2010.

[4] FCC. Second memorandum opinion and order, FCC 10-174. 2010.

[5] J. Huang, G. Xing, G. Zhou, and R. Zhou. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In *ICNP*, 2010.

[6] IEEE. Amendment 5: Spectrum and transmit power management extensions in the 5 ghz band in europe, 2003.

[7] K. Jamieson, B. Hull, A. Miu, and H. Balakrishnan. Understanding the real-world performance of carrier sense. In *EWIND*, 2005.

[8] E.-S. Jung and N. Vaidya. A Power Control MAC Protocol for Ad Hoc Networks. In *MOBICOM*, 2002.

[9] T.-S. Kim, J. Hou, and H. Lim. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In *MOBICOM*, 2006.

[10] K. Kyouwoong, I. Akbar, K. Bae, J. sun Urn, C. Spooner, and J. Reed. Cyclostationary approaches to signal detection and classification in cognitive radio. In *DySpan*, 2007.

[11] J. Lansford, A. Stephens, and R. Nevo. Wi-fi (802.11b) and bluetooth: enabling coexistence. *Network, IEEE*, 15(5), 2001.

[12] Lyrtech. Lyrtech small form factor software defined radio (sff-sdr).

[13] B. Radunovic and J.-Y. Le Boudec. Rate performance objectives of multi-hop wireless networks. In *Infocom*, 2004.

[14] T. Schmidl and D. Cox. Robust frequency and timing synchronization for OFDM. *IEEE TRANSACTIONS ON COMMUNICATIONS*, 45(12), 1997.

[15] F. Tufvesson, O. Edfors, and M. Faulkner. Time and frequency synchronization for OFDM using PN-sequence preambles. In *VTC*, 1999.

[16] X. Yang and N. Vaidya. On the physical carrier sense in wireless ad hoc networks. In *INFOCOM*, 2005.

[17] J. e. a. Zhang. Experimenting software radio with the sora platform. In *SIGCOMM Demo*, 2010.

[18] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma. Zifi: Wireless lan discovery via zigbee interference signatures. In *MobiCom*, 2010.