

Gap Sense: Lightweight Coordination of Heterogeneous Wireless Devices

Xinyu Zhang

Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Email: xyzhang@ece.wisc.edu

Kang G. Shin

Department of Electrical Engineering and Computer Science
University of Michigan-Ann Arbor
Email: kgshin@eecs.umich.edu

Abstract—Coordination of co-located wireless devices is a fundamental function/requirement for reducing interference. However, different devices cannot directly coordinate with one another as they often use incompatible modulation schemes. Even for the same type (e.g., WiFi) of devices, their coordination is infeasible when neighboring transmitters adopt different spectrum widths. Such an incompatibility between heterogeneous devices may severely degrade the network performance. In this paper, we introduce *Gap Sense* (GSense), a novel mechanism that can coordinate heterogeneous devices without modifying their PHY-layer modulation schemes or spectrum widths. GSense prepends legacy packets with a customized preamble, which piggy-backs information to enhance inter-device coordination. The preamble leverages the quiet period between signal pulses to convey such information, and can be detected by neighboring nodes even when they have incompatible PHY layers. We have implemented and evaluated GSense on a software radio platform, demonstrating its significance and utility in three popular protocols. GSense is shown to deliver coordination information with close to 100% accuracy within practical SNR regions. It can also reduce the energy consumption by around 44%, and the collision rate by more than 88% in networks of heterogeneous transmitters and receivers.

I. INTRODUCTION

The explosive growth of wireless and mobile networks continues proliferation of heterogeneous network devices on the unlicensed spectrum. These devices are often built on distinct PHY layers to serve specific types of applications, and hence cannot directly communicate with one another. For example, WiFi (or IEEE 802.11) [1] and ZigBee (or IEEE 802.15.4) [2] are often deployed in residential environments for mobile Internet access and home-area networking, respectively. Both share the 2.4GHz ISM band, but adopt different modulation schemes and even different MAC protocols (ZigBee allows for TDMA). Heterogeneity may occur even in the same family of networks. For example, besides the standard 20MHz bandwidth, WiFi has evolved to have 40MHz bandwidth in 802.11n [3] and 160MHz bandwidth in 802.11ac [4] to support high data-rate applications, in addition to the 802.11-2007 [1] with narrower bandwidths (5MHz and 10MHz) supporting applications with low data-rate but high energy-efficiency. This trend towards heterogeneous MAC/PHY layers is likely to continue with the evolution of wireless networks.

Ideally, each network should independently manage its associated devices. Interference between different networks can be avoided with CSMA-style MAC protocols. However, network heterogeneity poses significant challenges on existing protocols.

High-speed devices may preempt low-speed devices and cause severe collision even with CSMA enabled [5], [6]. Hidden terminals worsens the problem, since heterogeneous transmitters cannot exchange the RTS/CTS style signaling messages between them. Lack of a coordination mechanism also renders it infeasible to realize protocols that put the receiver in a narrower spectrum (*i.e.*, lower sampling-rate) than the transmitter in order to save energy [7].

A straightforward way to enable coordination between heterogeneous network devices is to enforce common modulation scheme and message format. However, this requires substantial modification to the PHY layers and loses the unique advantages of each device. For example, to enable ZigBee to decode a WiFi-modulated packet, it needs to increase the spectrum width and clock-rate, thus increasing the energy cost. On the other hand, WiFi loses its throughput advantage if it chooses to coordinate using a ZigBee-compatible PHY layer. Therefore, it is essential to develop a lightweight mechanism that does not require modification of the hardware and communication algorithms used in existing devices.

In this paper, we propose *Gap Sense* (GSense), a coordination mechanism for heterogeneous network devices with two salient features: *i*) it leverages the energy-sensing capability and does not modify existing communication modules; *ii*) it works well in spite of the MAC/PHY heterogeneity in legacy devices. GSense prepends a customized preamble to each legacy packet. The preamble contains multiple energy pulses and the gaps between them are used to convey coordination information. A GSense-enabled transmitter adds the preamble as raw signals without specific modulation (DSSS, OFDM, *etc.*), but it can still be detected by the receiver based on the signal patterns created by the gaps. Both the preamble and the detection algorithm are designed to be resilient to ambient interference, exploiting the fact that in legacy wireless networks, the frame length and the inter-frame space have a limited set of values.

We have implemented GSense on the GNURadio/USRP2 software radio platform [8]. Our experimental evaluation shows that within a practical SNR range, GSense can identify the coordination preamble with close-to-zero mis-detection and false-alarm rates, even if the receiver and the transmitter are operating over different bandwidths. Furthermore, we use GSense to design three protocols to improve the performance of heterogeneous networks, each of which alone could be of significant interest to the wireless community.

- Coexistence of heterogeneous wireless networks. To avoid interference between ZigBee and WiFi, we design a protocol that leverages GSense to coordinate both networks, allowing them to share spectrum and multiplex the shared band over time. With GSense, the collision rate can be reduced from 90% to below 10%.
- Coexistence of heterogeneous spectrum widths. As WiFi's spectrum widths become diversified, the traditional RTS/CTS-based signaling protocol is no longer effective in preventing collisions. By replacing RTS/CTS with a GSense mechanism, we show that the collision caused by heterogeneous hidden terminals can be reduced from around 80% to 5% under saturated traffic loads.
- Energy-efficient clock-rate management between heterogeneous transmitter-receiver pairs. It has been shown in [7] that by downclocking a WiFi receiver during idle listening, a significant fraction of energy can be saved. Such schemes require the transmitter to trigger the receiver even if both are configured to operate at different sampling clock-rates. We show how GSense can be used to achieve this objective. GSense demonstrates more than 44% of energy savings, with lower complexity and better compatibility than existing approaches.

In summary, this paper makes the following contributions.

- We identify the importance of coordination between heterogeneous wireless networks/devices, and then propose GSense as a generic mechanism to achieve this coordination.
- We design a simple preamble construction and detection algorithm for GSense that can accurately convey control information between heterogeneous devices. We also implement the algorithm in a software radio testbed and experimentally evaluate its performance under a variety of parameter settings.
- Using GSense, we design three network protocols, which can substantially reduce collision between heterogeneous networks/devices and save energy via intelligent coordination of transceivers.

The rest of this paper is organized as follows. Sec. II motivates the need for efficient coordination of heterogeneous wireless devices while Sec. III presents the design of GSense. Sec. IV evaluates the performance of the GSense preamble detection algorithm. Sec. V presents three GSense-based protocols and evaluates their performance. Sec. VI discusses related work and finally, Sec. VII concludes the paper.

II. MOTIVATION

Why is an efficient coordination mechanism necessary for heterogeneous networks/devices? We answer this question by investigating three prominent cases where such a mechanism is critical to network performance.

Coexistence of heterogeneous wireless devices. Different network devices may have distinct PHY-layer profiles and MAC protocols. For example, WiFi runs CSMA, whereas ZigBee allows for TDMA operations to support real-time applications.

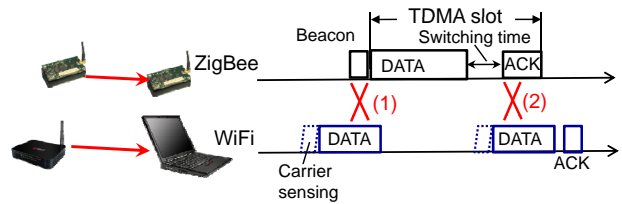


Fig. 1. Collisions when WiFi and ZigBee devices coexist.

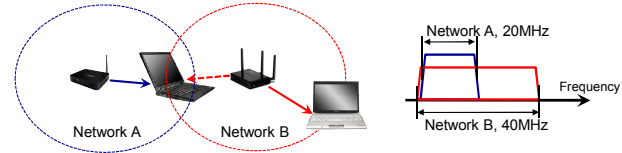


Fig. 2. Collisions caused by hidden terminals with heterogeneous spectrum widths.

Although WiFi will defer its transmission when it overhears ZigBee packets, ZigBee may dispatch its TDMA-scheduled packets during an ongoing WiFi transmission, thus causing collision (case (1) in Fig. 1). Collision may also occur due to the disparate time resolution of different devices. To reduce power consumption, ZigBee has a low clock-rate and a coarse time resolution of $320 \mu s$, *i.e.*, switching between states (carrier sensing, transmission and receiving) takes at least $320 \mu s$ [2]. In contrast, WiFi has a fine resolution of 9 to $20 \mu s$, and may complete a carrier sensing operation during the ZigBee's switching time, thus colliding with a ZigBee packet that follows immediately (case (2) in Fig. 1). Under moderate to high traffic loads, such a collision occurs frequently, rendering ZigBee's TDMA mode unusable [6].

As a natural solution, ZigBee may inform WiFi of upcoming TDMA slots so that WiFi can temporarily defer its transmission. Such coordination can significantly reduce collision at a low cost of channel time, since ZigBee operates with a very low duty-cycle [9]. But direct information sharing between WiFi and ZigBee is not possible due to their incompatible PHY-layer modulation/demodulation methods. It is thus preferable to create a lightweight *side channel* independent of the existing MAC/PHY modules.

Coexistence of heterogeneous spectrum widths. Emerging WiFi standards allow for wider (40MHz or 160MHz) or narrower (5MHz or 10MHz) spectrum widths than the standard 20MHz channel. When two WiFi links with different bandwidths coexist and share part of the spectrum, the conventional message-exchange mechanisms cannot coordinate their transmissions. In particular, when hidden terminals exist in a 'homogeneous' network, a receiver can use the CTS message (containing the duration of an upcoming data packet) to reserve the channel, warning interferers who cannot hear the transmitter. But when the interferer has a different bandwidth and cannot communicate directly with the transmitter/receiver, the CTS scheme becomes ineffective (Fig. 2). Hence, we need a new coordination mechanism that is oblivious of the spectrum heterogeneity between network devices.

Coordination of heterogeneous transmitter-receiver pairs. Conventional network protocols assume that the transmitter

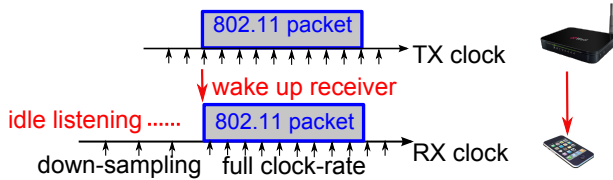


Fig. 3. Coordinating a transmitter–receiver pair operating at different sampling rates, in order to reduce energy consumption.

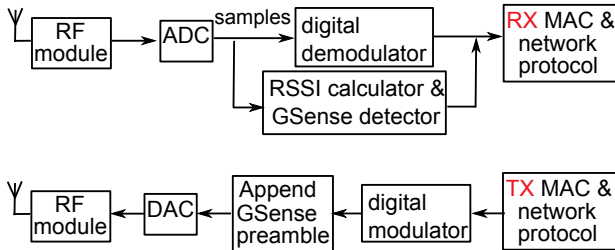


Fig. 4. Integrating GSense into existing radio devices.

and the receiver can communicate only if their PHY-layer configurations (modulation/demodulation, sampling clock-rate, etc.) match. Recently, by relaxing this constraint and intentionally introducing heterogeneity, network performance is shown to improve substantially [7]. Specifically, by down-clocking the radio during its idle listening, the receiver can reduce the dominating power consumption in idle listening. For such clock-rate management protocols, the key challenge is to trigger the receiver which is configured to a lower clock-rate than the transmitter (Fig. 3). Again, the conventional message-exchange scheme is ineffective as the receiver cannot decode any notification packets from the transmitter. Therefore, the coordination mechanism should be oblivious to the sampling clock-rate used by different nodes.

III. DESIGN AND DETECTION OF GSSENSE PREAMBLE

GSense is a generic mechanism that creates a side channel between heterogeneous wireless devices to enhance their cooperation. It can be integrated into existing radio devices without modifying their modulator/demodulator. As illustrated in Fig. 4, GSense patches legacy modulated packets with a customized preamble, which is sent directly as raw samples together with the modulated data payload. The receiver leverages the energy level of incoming signals (provided by the RSSI module) to identify the preamble and the coordination information therein. Note that the preamble only serves as a primitive for conveying limited units of information. It is up to the MAC or application protocol to determine the mapping between the preamble’s format and the actual coordination information. Although the RSSI module is usually used by CSMA wireless devices (e.g., WiFi and ZigBee), it is becoming a requirement in new generations of TDMA devices (e.g., Bluetooth 4.0 and 3.65GHz WiMax) with built-in coexistence mechanisms.

In what follows, we introduce the detailed design of the GSense preamble and the corresponding detection algorithm.

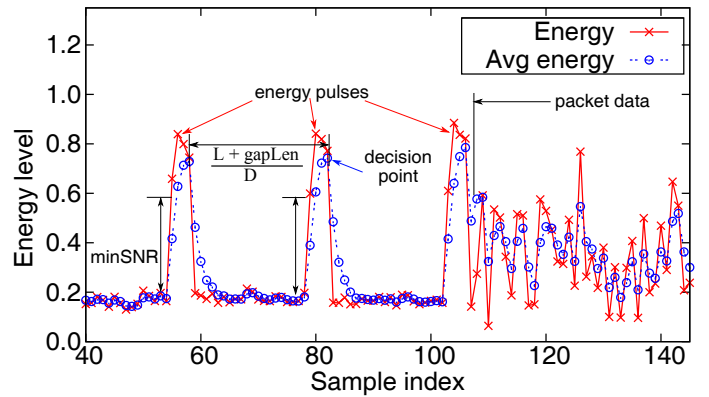


Fig. 5. Traces of received signals’ energy levels with a GSense preamble.

A. The GSense Preamble

The GSense preamble consists of multiple energy pulses with quiet periods (gaps) in between (Fig. 5). The preamble is designed in a way to maximize the SNR and hence the detection probability, while ensuring sufficient capacity to contain the coordination information.

1) Design of a sequence of energy pulses:

a) *Sequence length*: The energy pulses are a sequence of raw samples (complex numbers). The sequence must be sufficiently long so as to be detectable by a heterogeneous receiver. This is particularly important when the receiver has a lower sampling clock-rate (or narrower bandwidth) than the transmitter. Suppose the transmitter’s sampling clock-rate is D times that of the receiver’s, then each energy pulse must contain at least D samples, such that at least one of them can be received.

However, the energy pulse should not be too long either; otherwise, the receiver may confuse it with legacy packets. On the ISM band, legacy network packets consist of a header (used for synchronization between transmitter and receiver) and payload. So, their duration is larger than the header duration (e.g., $160\mu s$, $100\mu s$ and $8\mu s$, for Bluetooth, ZigBee and WiFi, respectively). When constructing an energy pulse, the transmitter ensures that it is shorter than the minimum of the header durations in co-existing networks.

A gap is simply a sequence of zeros in the preamble. Similar to the energy pulses, the minimum gap length must be larger than D , the ratio of spectrum width (or sampling rate) between the transmitter and the receiver. On the other hand, it should be shorter than the carrier sensing duration of neighboring CSMA devices. Otherwise, these devices may deem the gap as a hint for an idle channel and start transmission, thus colliding with subsequent energy pulses. The carrier sensing duration is well defined in existing CSMA networks ($128\mu s$ for ZigBee [2], and $28\text{--}50\mu s$ for WiFi [1]), and the minimum of them is used as the upper-bound of gap duration. The maximum gap length (i.e., the maximum number of discrete samples) equals the maximum gap duration divided by the transmitter’s sampling period.

b) *Deterministic or random sequence?*: In digital communication systems, sequences of bits are intentionally randomized

using a scrambler to improve the accuracy of channel estimation [10]. Existing correlation-based packet detection algorithms also require the sequence of bits in the preamble to be randomized (as in the 802.11 preamble [1]), so that a correlation peak can be identified only if the sequence matches a known pattern. However, in GSense, a deterministic sequence is needed in order to maximize the SNR of energy pulses.

To see this clearly, suppose $d[n]$ ($n \in [-\frac{N}{2}, \frac{N}{2}]$, where N is an even number representing the pulse length) are the samples in the pulse emitted from the transmitter. After traversing the RF front-end at the receiver, at time t , the signal involves noise and a composite of residual signals from other time instants as:

$$r(t) = n(t) + \sum_{n=-N/2}^{N/2} d[n] \cdot h(t - nT_s), \quad (1)$$

where T_s is the sampling period, $n(t)$ the noise signal and $h(t)$ the channel transfer function at time t . After the ADC, the k -th digital sample can be expressed as:

$$r[k] = n(k) + \sum_{n=-N/2}^{N/2} d[n]h[k - n] \quad (2)$$

$$= n(k) + d[k]h[0] + \sum_{n \neq k} d[n]h[k - n]. \quad (3)$$

When $d[n]$ is a random bipolar sequence (assuming elements are randomly chosen from 1 and -1), the resulting SNR is:

$$\frac{|h[0]|^2}{\sigma^2 + |\sum_{n \neq k} d[n]h[k - n]|^2}, \quad (4)$$

where σ^2 is the noise variance. The second term in the denominator represents the interference caused by other samples in the preamble. Due to the multipath effect, the tail of each sample can smear into adjacent samples with opposite phases, thereby weakening the signal strength and degrading detection performance. In contrast, if $d(k)$ is a deterministic sequence, *i.e.*, $d[n] = d[m](\forall m, n \in [-\frac{N}{2}, \frac{N}{2}], m \neq n)$, then the SNR:

$$\frac{|h[0] + \sum_{n \neq k} h[k - n]|^2}{\sigma^2} > \frac{|h[0]|^2}{\sigma^2} \quad (5)$$

which is larger than the SNR in the random case (4). The inequality (5) holds because the channel transfer function is unlikely to undergo abrupt phase changes within the short period of one energy pulse.

2) *Embedding information in gaps*: Once two consecutive energy pulses are detected, the gap between them can be determined according to how many digital samples the receiver counted in between the pulses. Multiple energy pulses can be concatenated, and each of the corresponding gaps represents one field of information, which is customizable by the application.

The gap length can be mapped to various control or coordination information, depending on the objective of higher-layer protocols. For example, it can be directly mapped to a transmitter's spectrum width and a receiver's address, such that a transmitter can trigger a specific receiver even if it is operating

on a narrower bandwidth. More examples of its applications will be provided in Sec. V.

Although the approach of gap sensing masks PHY-layer heterogeneity, it has low communication efficiency, and is applicable only if its low-efficiency is tolerable by the higher-layer protocols. Suppose the maximum gap length is G_m samples, then up to $\log_2 G_m$ bits of information can be carried in each gap. With a sampling rate of F Hz, the resulting information capacity is $\frac{\log_2 G_m}{G_m/F} = F G_m^{-1} \log_2 G_m$ bits/s. The capacity increases linearly with sampling rate F , but decreases with gap length. Hence, a wider gap may waste more channel resources, albeit more information bits can be delivered. Therefore, GSense is only suitable for delivering low-rate control/coordination information.

B. Detecting the Preamble

We now describe how the GSense preamble can be detected despite the PHY-layer heterogeneity between the transmitter and the receiver. Suppose the receiver's sampling rate is R_r and the transmitter's DR_r . With a deterministic sequence at the transmitter, the corresponding received energy pulse is:

$$r[k] = n(k) + h[k \frac{DR_r}{R_r}] = n(k) + h[kD]. \quad (6)$$

The energy pulse spans a short period (on the order of microseconds), so the channel transfer function remains relatively stable, *i.e.*, $h[kD] \approx h[k]$. Consequently, the receiver can detect the same level of energy even if it downsamples or upsamples the incoming signals.

Due to noise and fading effects, the received energy pulse no longer retains sharp edges, which causes variation of gap length and degrades the reliability of coordination. To solve this problem, GSense first smooths the incoming signals' energy level with an exponentially weighted moving average, with the window size equal to the minimum gap length. In addition, it ensures each energy pulse contains the same sequence of raw samples. This way, each transition from gap (or noise floor) to energy pulse shows a similar pattern, and the detected starting times of adjacent pulses have the same deviation from the true edges (Fig. 5). At the receiver side, the detection decision tends to be made at the same position of consecutive energy pulses, and hence, the gap length can still be identified by subtracting the known pulse length $\frac{L}{D}$ from the inter-pulse period $\frac{L+gapLength}{D}$.

Algorithm 1 shows the preamble construction and detection procedure in GSense. When constructing the preamble, the deterministic sequence uses a complex number $A + Aj$ for each sample, where A is the maximum signal amplitude for the radio. GSense identifies the energy pulses using level detection. It declares a pulse only if multiple samples (H_1 fraction of an ideal energy pulse) with sufficient SNR are detected, which is more reliable than edge detection. An important feature of the detection algorithm is that its detection threshold need not be adapted to the radio environment. To detect the energy pulses, a fixed threshold $minSNR$ is used, which is the minimum detectable SNR. Below this SNR level, interference signals are

Algorithm 1 Preamble construction and detection.

```
1. /*Transmitter: preamble construction*/
2. foreach gap field
3. /*Insert energy pulse with length L */
4.   for k = 1 to L
5.     s(k) = A + Aj
6.   endfor
7.   for k = L + 1 to L+ gapLen
8.     s(k) = 0
9.   endfor
10. endfor
11. /*Receiver: preamble detection*/
12. /*Energy level of incoming signal at time stamp k*/
13. e(k) ← |r(k)|2
14. /*Update average energy level, β = 1/minGapLen*/
15. a(k) ← βe(k) + (1 - β)a(k)
16. /*Collect sufficient high SNR samples to decide on a
    pulse*/
17. if 10 log10 (a(k) / a(k-L/D)) > minSNR
18.   decisionQ ← push 1
19. else decisionQ ← push 0
20. endif
21. /*Edge detection of energy pulse. D*/
22. if sum(decisionQ) > H1 · L/D
23.   if minGapLen/D ≤ k - lastPulseEdge ≤ maxGapLen/D
24.     /*Pulse detected. Determine gap length*/
25.     gapLen = k - lastPulseEdge - L/D
26.   endif
27.   lastPulseEdge = k;
28. endif
```

too weak, and hence unlikely, to cause collision. Thus, inter-device coordination is unnecessary. In addition, note that for each time stamp k , only a fixed number of operations is needed, and thus the detection algorithm has a linear complexity.

IV. ACCURACY OF GSENSE DETECTION

This section describes our implementation and experimental evaluation of the GSense algorithm on a GNURadio/USRP2 software radio platform [11].

We have built the GSense preamble construction and detection algorithm on the GNURadio library [8]. GNURadio provides the driver interface to the USRP2 software radio, which delivers modulated packets together with the GSense preamble over its RF front-end. The receiver terminal, a USRP2 node, down-converts the incoming signals to the baseband, quantizes them into discrete complex samples, and then delivers the samples to a PC via Ethernet. The PC continuously runs the GSense detection algorithm and extracts the gap length information from preambles.

We conducted experiments in an indoor office environment with obstacles (concrete walls, server racks, *etc.*) which create rich multipath reflections. To incorporate the effects of interference from non-GSense devices, the radio's center frequency is

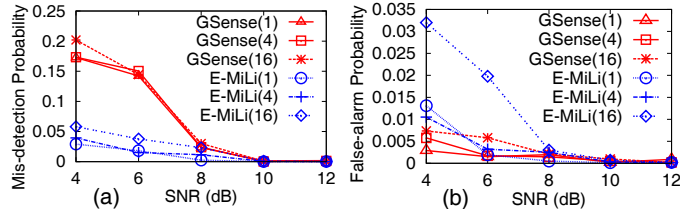


Fig. 6. Mis-detection and false-alarm rates when D varies between 1, 4, 16. D represents the ratio of the transmitter's sampling rate to the receiver's.

configured to 2.452GHz, a channel shared with ambient WiFi networks.

Detection accuracy in a noisy environment. We evaluated the mis-detection and false-alarm probabilities of GSense under various SNR levels. As a benchmark, we also implemented the E-MiLi algorithm [7], which exploits the self-correlation between consecutive random sequences to detect a preamble. In our experiments, SNR is measured by the mean energy level of an energy pulse over the noise floor preceding it. Different SNR values are created by varying the link distance and the signal amplitude (the parameter A in Algorithm 1). The device heterogeneity is represented by D , the ratio of the transmitter's sampling rate to the receiver's. By default, the transmitter sends 3000 preambles per second, each followed by random samples representing the packet payload. The pulse length (L) is set to 64, and the minimum detectable SNR (minSNR) to 4dB.

We start with the case where the receiver needs to detect the address information embedded in one preamble gap. The E-MiLi algorithm is configured to achieve the same objective, but using a longer preamble (3 repetitions of the same random sequence, as exemplified in [7]). Fig. 6(a) shows the mis-detection probability as a function of SNR and the spectrum heterogeneity. Compared to E-MiLi, GSense shows a higher mis-detection rate at low SNR (below 8 dB). This is because the pulse detector is more susceptible (than the self-correlation detector) to noise corruption and the fading effects, which occasionally reduces the SNR of energy pulses to below the minSNR. E-MiLi's preamble and self-correlation tends to preserve themselves even under a low SNR. The E-MiLi's downside is that it incurs higher computational complexity, and the correlation operation is not natively supported by legacy radios.

We would like to stress that in most common cases, wireless networks operate in a high SNR region to ensure a high packet-delivery ratio, and weak interference signals can easily be overwhelmed by the desired data signals. For OFDM-based wireless networks, around 7dB of SNR is needed to achieve the lowest bit rate (with BPSK modulation), and most existing wireless links have a much higher SNR than the lowest margin in order to combat the uncertainty caused by fading [12]. For DSSS-based networks such as 802.15.4 and 802.11b, at least 10dB of SNR is needed to maintain a low packet-loss rate [13]. Within such high SNR regions, GSense achieves a close-to-zero mis-detection rate. More importantly, its detection performance is insensitive to the spectrum heterogeneity, and can thus serve as a reliable means of inter-device coordination.

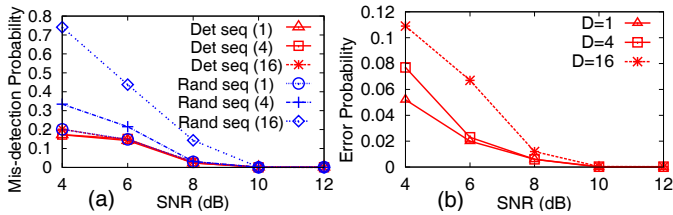


Fig. 7. (a) Effect of a random sequence; (b) Error probability of detecting the gap length.

With respect to the false-alarm probability (Fig. 6(b)), GSense is found to consistently achieve below 1% error even under low SNR, which is even superior to E-MiLi. This is essentially due to its implicit gap length constraint: a gap is declared only if two consecutive energy pulses are identified, with a separation between the minimum and maximum gap lengths. It is rare to find such cases falsely created by noise.

Effects of sequence design. To validate the importance of a deterministic sequence in GSense, we compared it with a scheme that uses randomly generated energy pulses (the real and imaginary components in each sample are randomly chosen between A and $-A$). Fig. 7(a) shows the resulting mis-detection rate, where each SNR value corresponds to the same signal amplitude and link distance for both schemes. The random scheme suffers from a sharp rise of mis-detection rate, with up to 14% at 8dB and 73% at 4dB. It is also more vulnerable to spectrum heterogeneity: the detection performance may differ by multiple folds with different spectrum widths or sampling rates. Therefore, to achieve high detection accuracy, a deterministic sequence should always be used to generate the energy pulses for GSense.

False detection of gap length. Since GSense uses gap length to convey meaningful information, it is important to assess the probability that one gap value is mistaken as another. Fig. 7(b) shows that the error probability is close to 0 with above 10dB SNR, around 1% at 8dB SNR, and increases sharply when SNR goes below 8dB. Therefore, application protocols can employ GSense reliably at high SNRs, but use it with caution when SNR is below 8dB.

V. COORDINATION OF HETEROGENEOUS NETWORKS WITH GSENSE

In this section, we use GSense to design three prevalent protocols that coordinate heterogeneous network devices, in order to reduce interference and improve energy-efficiency.

A. Coexistence of Heterogeneous Wireless Devices

We design the GSense preamble for ZigBee networks, so that they can notify a coexisting WiFi device of forthcoming TDMA packets, thereby preventing the two types of collision due to lack of coordination (Sec. II).

The unique characteristics of ZigBee allow GSense to realize such coordination at low overhead. ZigBee has a periodic frame structure. Each frame starts with a beacon sent by a coordinator node, followed by a number of contention-free slots with a fixed duration, in which TDMA packets can be delivered between

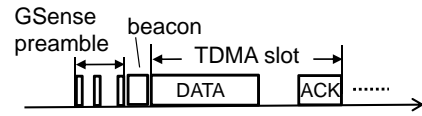


Fig. 8. Using GSense to coordinate ZigBee and WiFi.

the coordinator and clients [2]. CSMA packets may also be delivered afterwards, but are beyond the scope of this paper.

To enable the coordination, it is sufficient for ZigBee to send a bootstrap GSense preamble containing general information about its schedule, and then periodic preambles before each frame of TDMA slots comes. Fig. 8 illustrates the basic operations. In the bootstrap stage, the GSense preamble contains 3 fields, indicating (i) type of preamble (*i.e.*, a bootstrap preamble), (ii) the TDMA slot duration (in the unit of ZigBee time slot, *i.e.*, $320 \mu s$), and (iii) number of TDMA slots in each ZigBee frame. The bootstrap preamble is sent proactively. It is triggered only when ZigBee experiences high interference indicated by unusually high packet losses. It may be repeated to ensure reliable delivery to WiFi interferers, and re-initiated if parameters in its information fields change.

After the bootstrap, a notification preamble is sent before each frame, warning WiFi about the forthcoming beacon and TDMA slots. The notification preamble comprises consecutive energy pulses with the same gap length that is known to both WiFi and ZigBee. The number of pulses (*i.e.*, the preamble length) should be adjusted to strike a balance between the overhead and performance (*i.e.*, collision-rate reduction). If it is too long, it may waste too much channel time; otherwise, the WiFi transmitter may be in the middle of a transmission and miss the GSense preamble before the ZigBee's TDMA frame starts, thus causing collision. To resolve this dilemma, the ZigBee node can increase the preamble length gradually until the observed collision-rate is acceptably low.

It should be noted that the preamble is sent without carrier sensing and may still collide with ongoing WiFi transmissions. However, ZigBee applications typically have low data-rate and low duty-cycle (below 1% [9]), so the collision occurs infrequently, and WiFi can easily recover from collision with MAC-layer retransmissions.

Due to the interface delay between USRP2 and its PC host, it is infeasible to implement MAC protocols that require real-time feedback. Thus, we evaluate GSense-based protocols by implementing its MAC operations in ns-2, and integrating the empirical PHY-layer results in Sec. IV. Following the 802.15.4 specification [2], we have developed a TDMA module in ns-2, in addition to the existing CSMA library for ZigBee. The GSense preamble scheduling and transmission schemes are built on top of this module. By default, GSense uses a preamble length (number of energy pulses) of 5. To avoid WiFi's confusion of the gap with an idle period, the gap length between adjacent energy pulses is set to $49 \mu s$, below the required 802.11g carrier sensing duration (the DIFS). We modify the ns-2 PHY-layer by incorporating the empirical noise floor and preamble detection rate from our testbed measurements

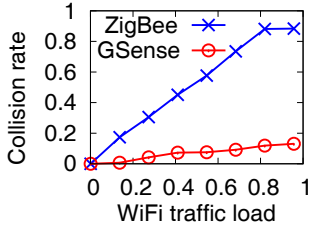


Fig. 9. Collision rate when ZigBee coexists with WiFi.

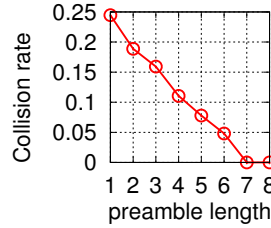


Fig. 10. Collision rate in ZigBee-WiFi coexistence.

(Sec. IV). An empirical propagation model (defined by the 802.15 Group [9]) is used for calculating the received signal power. Our experiment focuses on a benchmark scenario with a WiFi link and a ZigBee link, each of 10m length, and the transmitters are separated by 10m. WiFi uses a standard power level of 15dBm and ZigBee 0dBm. Without loss of generality, ZigBee uses one TDMA slot per frame, and one beacon is sent every 100 ms. WiFi runs an 802.11g UDP file transfer, with a data rate of 12Mbps and packet size 1KB.

Fig. 9 plots the collision rate under various WiFi traffic rates (application throughput normalized by data rate). The collision rate rises sharply as traffic rate increases until the WiFi link becomes saturated. With a moderate traffic of 0.4, the collision rate goes up to 45%, whereas with GSense, the collision rate is below 8%. Even when WiFi traffic gets saturated, GSense suffers around 10% of collision, 88.9% lower than the legacy ZigBee protocol.

Fig. 10 shows the effects of preamble length. When a single energy pulse is used, the preamble may still collide with ongoing WiFi transmissions. But the collision hazard decreases as more energy pulses are incorporated. With a preamble length of larger than 7, the collision probability is reduced to close to 0. Note that this preamble length is equivalent to less than 11 bytes of ZigBee frames, whereas a ZigBee frame may contain multiple TDMA slots of hundreds of bytes of data. Hence, the overhead of GSense is negligible, considering the fact that it can eliminate the risk of losing TDMA data and ACK packets due to collision with WiFi.

B. Coexistence of Heterogeneous Spectrum Widths

We employ GSense to prevent the collision due to hidden terminals with heterogeneous spectrum widths (Sec. II). Before delivering the data payload, the transmitter sends an RTS and the receiver replies with a CTS. Interferers with the same spectrum width can easily decode the CTS as in the legacy 802.11 protocol. In order to deliver the CTS information to the hidden terminal with a different bandwidth, a GSense preamble is appended to the CTS as an implicit coordination message (Fig. 11).

The preamble contains multiple gaps, representing similar information as a legacy CTS, including data rate and packet size, which is used to calculate the duration of transmission such that the hidden terminal can set its NAV and defer its transmission. The data rate can be represented by its modulation order.

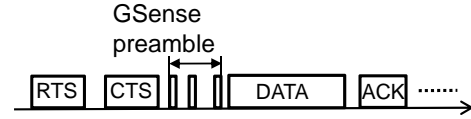


Fig. 11. RTS/CTS mechanism to combat hidden terminals with heterogeneous spectrum widths.

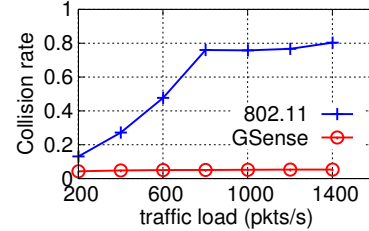


Fig. 12. Combating hidden terminals with different spectrum widths using GSense.

For 802.11a/g/n, the lowest order 1 corresponds to 6Mbps, and a maximum of 32 [3] modulation orders are available. Observing that the maximum packet size is 1500 bytes, the packet size can be represented by 4 gaps, each corresponding to one decimal digit of its value. Thus, the maximum gap overhead is $(32+1+9+9+9) = 60$ samples. Consider the coexistence of 802.11n and 802.11a/g, the maximum spectrum heterogeneity is $40\text{MHz}/5\text{MHz}=8$. The energy pulse length should be at least 8 (we use a larger value of 16 to improve reliability). So, the total preamble overhead is $(60 \times 8 + 16 \times 5) = 560$ samples. At 40MHz, the preamble lasts for $14 \mu\text{s}$, which is even shorter than the 802.11 preamble ($20 \mu\text{s}$), and costs negligible channel time compared to the data payload.

We have implemented the above GSense coordination scheme on the ns-2 simulator and evaluated its performance for a benchmark scenario with two links, running over 20MHz and 40MHz spectrum, respectively (Fig. 2). The transmitters cannot sense each other, and collision occurs at the 20MHz receiver whenever their packets overlap in time. Both transmitters send UDP packets following a Poisson model. A conservative SNR of 8dB is assumed for GSense, corresponding to a mis-detection rate of 5%. Fig. 12 shows that for legacy 802.11 nodes, the packet collision probability increases sharply with the traffic rate (up to 0.8 when the traffic becomes saturated). With GSense, collision is reduced to as low as its mis-detection rate. Note that the actual benefits of GSense depend on the fraction of hidden terminals in the network. An adaptive protocol should enable GSense only when a heterogeneous hidden terminal is suspected to exist, but is beyond the scope of this paper.

C. Energy-Efficient Clock-Rate Management for Heterogeneous Transceivers

In adaptive clock-rate management protocols [7], the idle receiver is allowed to operate at a lower clock-rate than the transmitters. A coordination scheme is needed that triggers the receiver to restore its clock. Since GSense can effectively coordinate the transmitter and the receiver even when they use heterogeneous sampling rates, it can be easily adopted as an enabling technique.

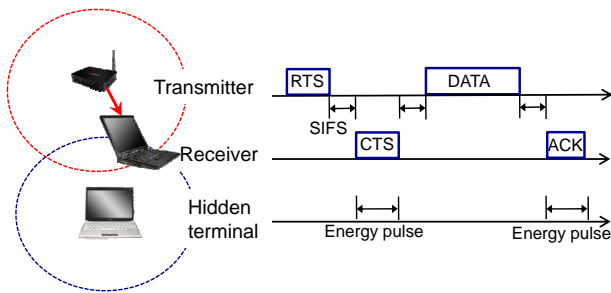


Fig. 13. Realizing virtual carrier sensing using GSense.

To this end, the transmitter needs to prepend the GSense preamble to each legacy packet, with the gap length denoting the intended destination address. A receiver remains in down-clocked mode if no preamble is detected, and restores its radio to full clock-rate whenever a GSense preamble is identified and matches its address.

This mechanism is similar to the E-MiLi protocol [7], but is simpler for implementation as it can leverage the built-in energy sensing module in legacy radio devices. In addition, GSense makes it possible to design an 802.11-compatible virtual carrier sensing mechanism, which is critical for reducing collisions in the presence of hidden terminals. Fig. 13 shows how a downclocked radio can detect the legacy CTS packet using GSense, without appending any customized preamble. GSense treats the CTS packet as an energy pulse with a known duration, since it has a fixed size of 14 bytes and is always sent with the lowest modulation rate. After detecting the CTS, the hidden terminal defers its transmission until it detects an energy pulse corresponding to the ACK (or NACK, which should be sent if the receiver failed to decode the data payload).¹ This mechanism can effectively enable the virtual carrier sensing of hidden terminals configured to different sampling rates, but is infeasible for E-MiLi, which uses a correlation-based detector and needs a customized preamble for every packet [7].

We use trace-driven simulation to verify the above mechanism. The traces are obtained from a public network with a maximum (average) of 31 (7) users [14]. We use the same energy consumption profile as the E-MiLi experiments [7], assuming a conservative SNR level of 8dB for the wireless links. The corresponding detection statistics from the testbed measurement are then fed into the trace-based simulator. The results (Fig. 14) show that GSense can save above 44% of energy for more than 90% of clients. Although GSense has a higher mis-detection rate than E-MiLi, the energy savings is almost the same. As observed in [7], the idle listening accounts for 90+% of the energy consumption for most WiFi clients. Thus, the extra energy cost due to mis-detection and retransmission is negligible compared to the savings in idle listening.

We further evaluated the performance of GSense in ns-2 under various data-rates, which reflects the relative ratio between preamble overhead and payload. Here the transmitter

¹Sec. V-B solves a similar problem using the customized GSense preamble, but requires no NACK.

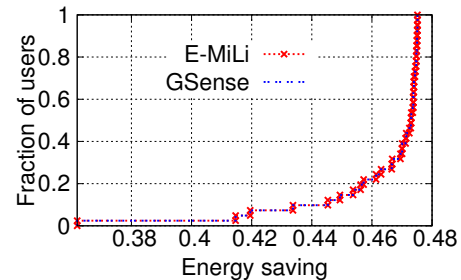


Fig. 14. Energy savings: trace-based simulation.

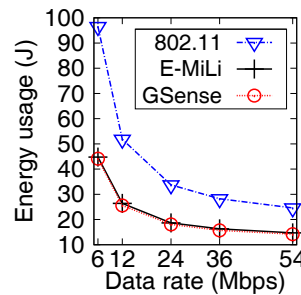


Fig. 15. Energy consumption in FTP application.

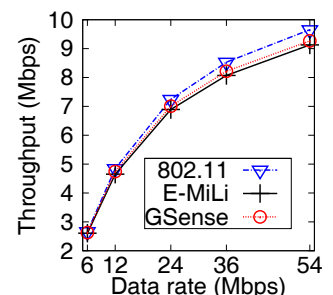


Fig. 16. Throughput in FTP application.

sends a 20MB file to the receiver via FTP. The switching time between low/high clock rate is set to $9.5 \mu s$ [7]. Figs. 15 and 16 plot the resulting energy-savings and throughput, respectively. GSense achieves the same amount of energy-savings as E-MiLi under various data-rates, ranging from 23.1% to 44%. As the data-rate increases, both GSense and E-MiLi incur throughput degradation due to the preamble overhead. At 54Mbps, E-MiLi's throughput is 6.3% lower than the legacy 802.11. GSense's throughput is slightly higher than E-MiLi (1.5% at 54Mbps), as it can use a single gap to deliver information (E-MiLi needs at least two sequences of the same length), thus inducing less overhead.

VI. RELATED WORK

Numerous existing studies addressed the issue of establishing coordination or control channel between wireless devices. Flashback [15] leverages one subcarrier, the smallest spectrum unit in OFDM communication systems, to piggy-back control messages between the transmitter and the receiver. *Side Channel* [16] observes that the typical links have higher SNR than required by the corresponding bit rate, so pulse-modulated signals can be augmented to a high-SNR link to deliver control messages. Both schemes work for the same type of devices that synchronize with each other and adopt the same modulation/demodulation algorithms.

GSense is partly inspired by the Esense [17] and CtS (Communication through Silence) [18] mechanisms. Esense aims at building an alphabet of implicit messages using the packet duration information. It has a high false alarm rate when the packet size varies unpredictably. CtS [18] discussed the basic idea of using the silent period between energy bursts to convey

information, in order to save the transmission energy for sensor networks. It assumes that the transmitter and the receiver are synchronized and the silent period can be detected accurately and without interference. To the best of our knowledge, GSense is the first to address the practical challenges in realizing a *covert* coordination channel, and using it to improve the performance of heterogeneous wireless networks.

The problem of sensing energy pulses is analogous to that of sensing primary users in cognitive radio networks (see [19] for a comprehensive survey). However, GSense needs to identify not only the existence of energy pulses from other users, but also the exact gap between energy pulses. GSense works in the high SNR region (the signal strength is strong enough to enable packet decoding, or the interference is strong enough to cause collision), which is common for short-range communication networks like WiFi and ZigBee. GSense does not assume any distribution of noise or signal power, which is essential for hypothesis testing in cognitive radio networks.

The need for coordinating heterogeneous networks has been discussed previously. Choi *et al.* [20] proposed a rendezvous algorithm to establish a shared channel between different network devices. Many other studies [21], [22] focused on adapting channel allocation (*e.g.*, between ZigBee and WiFi), which is not always feasible (Sec. II). In [6], [23], a dedicated helper node is employed that sends a busy-tone through an opportunistically created channel, in order to make WiFi aware of ZigBee's presence, thus avoiding interference. In contrast, GSense does not need extra devices, and it works even when different networks are sharing the same spectrum.

To avoid the interference between networks with heterogeneous spectrum, conventional solutions [1], [24] force the wideband network to degrade and be compatible with the narrowband network, thus losing its high-throughput advantage. E-MiLi [7] allows a low-bandwidth (*i.e.*, low sampling-rate) receiver to detect packets from normal transmitters, but it has higher complexity than GSense, and cannot realize essential MAC-layer operations (*e.g.*, virtual carrier sensing, Sec. V-C) without hardware modification.

VII. CONCLUSION

In this paper, we have first identified typical scenarios where the heterogeneity of transceivers affects the network performance (*e.g.*, collision probability and energy-efficiency). We have then proposed GSense as a generic solution to such problems by enhancing coordination between heterogeneous wireless devices. GSense creates a side channel between wireless nodes by constructing sequences of energy pulses and sensing the gaps between them. We have addressed important practical challenges in realizing GSense, and evaluated our solution on a software radio platform, which exhibits close-to-zero mis-detection and false-alarm rates in common SNR ranges. Further, we have demonstrated the usefulness of GSense in three prevalent protocols. GSense is shown to reduce the collision between heterogeneous networks by multiple folds. It can also reduce WiFi energy consumption by around 44% by intentionally allowing heterogeneity between the transmitter

and the receiver. GSense is simple and has better compatibility with 802.11 than other alternative approaches. As wireless networks become more diversified, we believe GSense will become an essential mechanism to enable the coexistence and cooperation between heterogeneous wireless networks and devices.

ACKNOWLEDGEMENT

The work reported in this paper was supported in part by the NSF under Grant CNS-1160775.

REFERENCES

- [1] IEEE Standard, "802.11TM: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2007.
- [2] IEEE-SA Standard Board, "IEEE Standard Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," 2003.
- [3] IEEE Standard, "802.11n: Enhancements for Higher Throughput," 2009.
- [4] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std. 802.11ac Draft 1.0*, 2011.
- [5] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving Wi-Fi Interference in Low Power ZigBee Networks," in *Proc. of ACM SenSys*, 2010.
- [6] X. Zhang and K. G. Shin, "Cooperative Carrier Signaling: Harmonizing Coexisting WPAN and WLAN Devices," *to appear in ACM/IEEE Transactions on Networking*, 2012.
- [7] —, "E-MiLi: Energy-Minimizing Idle Listening in Wireless Networks," in *Proc. of ACM MobiCom*, 2011.
- [8] "The GNU Software Radio," <http://gnuradio.org>.
- [9] IEEE 802.15 Working Group, "Coexistence Analysis of IEEE Std 802.15.4 With Other IEEE Standards and Proposed Standards," 2010.
- [10] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice Hall, 2001.
- [11] Ettus Research LLC, "Universal Software Radio Peripheral (USRPI)," <http://www.ettus.com/>.
- [12] K. LaCurtis and H. Balakrishnan, "Measurement and Analysis of Real-World 802.11 Mesh Networks," in *Proc. of ACM Internet Measurement Conference*, 2010.
- [13] M. R. Souryal, L. Klein-Berndt, L. E. Miller, and N. Moayeri, "Link Assessment in an Indoor 802.11 Network," in *Proc. of IEEE WCNC*, 2006.
- [14] A. Schulman, D. Levin, and N. Spring, "CRAWDAD data set umd/sigcomm2008," 2008.
- [15] A. Cidon, K. Nagaraj, S. Katti, and P. Vishwanath, "FlashBack: Decoupled Lightweight Wireless Control," in *Proc. of ACM SIGCOMM*, 2012.
- [16] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. Ni, "Side Channel: Bits Over Interference," in *Proc. of ACM MobiCom*, 2010.
- [17] K. Chebrolu and A. Dhekne, "Esense: Communication Through Energy Sensing," in *Proc. of ACM MobiCom*, 2009.
- [18] Y. Zhu and R. Sivakumar, "Challenges: Communication through Silence in Wireless Sensor Networks," in *Proc. of ACM MobiCom*, 2005.
- [19] T. Yucek and H. Arslan, "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications," *Communications Surveys Tutorials, IEEE*, vol. 11, no. 1, 2009.
- [20] B. Choi and M. Gerla, "Wireless Interrupt: Inter-Device Signaling in Next Generation Wireless Networks," in *Proc. of IEEE INFOCOM Workshops*, 2010.
- [21] S. Pollin, M. Ergen, M. Timmers, L. Van Der Perre, F. Catthoor, I. Moerman, and A. Bahai, "Distributed Cognitive Coexistence of 802.15.4 With 802.11," in *Proc. of CrownCom*, 2006.
- [22] C. Won, J.-H. Youn, H. Ali, H. Sharif, and J. Deogun, "Adaptive Radio Channel Allocation for Supporting Coexistence of 802.15.4 and 802.11b," in *Proc. of IEEE VTC*, 2005.
- [23] X. Zhang and K. G. Shin, "Enabling Coexistence of Heterogeneous Wireless Systems: Case for ZigBee and WiFi," in *Proc. of ACM MobiHoc*, 2011.
- [24] Y. Utsunomiya, T. Tandai, T. Adachi, and M. Takagi, "A MAC Protocol for Coexistence between 20/40 MHz STAs for High Throughput WLAN," in *IEEE VTC-Spring*, 2006.