The Utility of Perceptive Communication between Distant Wireless Nodes

Kimaya Sanzgiri* Ian D. Chakeres‡ Elizabeth M. Belding-Royer*
*Department of Computer Science, University of California, Santa Barbara
‡Boeing Phantom Works, Seattle WA
kimaya@cs.ucsb.edu, ian.chakeres@gmail.com, ebelding@cs.ucsb.edu

Abstract

CSMA-based MAC protocols require wireless nodes to share the transmission medium with other nodes that are within carrier-sensing (CS) range. Hence, operations that depend on and can potentially affect the state of the shared medium, such as admission control, require coordination and information sharing among these nodes. The CS range of wireless nodes is traditionally much larger than the reception range. Nodes that are outside reception range but within CS range can sense packet transmissions; however, they are unable to decode the packet contents. Direct messaging between these nodes is therefore not feasible for information sharing. In this paper, we describe two perceptive approaches for communication with nodes within CS range. Our approaches exploit the ability of a node to detect a change in the strength of the carrier signal when a transmission is in progress. Information is encoded in, and inferred from, perceptible transmission characteristics, such as the duration of a transmission or the silence between adjacent transmissions. This paper evaluates the feasibility and effectiveness of our perceptive communication mechanisms on a hardware testbed. We implement a prototype of our solution on the Mica2 mote platform and test it in different network scenarios. Our results demonstrate that the durations of transmissions and silences can be correctly detected within a small margin of error in most situations, thus verifying the utility of our approach.

1. Introduction

Popular wireless MAC protocols, such as IEEE 802.11 [6] and IEEE 802.15.4 [7], employ a Carrier Sense Multiple Access (CSMA) strategy for medium access. CSMA specifies that when a node transmits a packet, all other nodes that are within carrier-sensing (CS) range of the transmitter (called carrier-sensing neighbors) detect a carrier signal and do not attempt to transmit simultaneously. The medium is shared among carrier-sensing neighbors in this manner.

Operations that depend on and affect the state of the shared medium, such as admission control [2, 14], require coordination and information sharing among carrier-sensing neighbors. For example, when making an admission decision for a new traffic flow, a node must know the bandwidth consumption and priorities of existing traffic flows at all carrier-sensing neighbors to ensure that the new flow does not adversely affect any existing flows of equal or higher priority. Another example is the determination of intra-flow contention [11, 14], which requires a node to know the number of carrier-sensing neighbors that lie on a particular multihop path. These and other applications create a need for a communication mechanism among carrier-sensing neighbors. Further, to share information such as flow priorities and identities of carrier-sensing neighbors, it is essential that the communication mechanism be more sophisticated than the binary presence or absence of a carrier signal.

The CS range of a wireless node is typically much larger than its reception range. Nodes that lie outside the reception range but within CS range can only sense packet transmissions; these nodes are unable to decode the contents of packets. Hence, direct messaging is ineffective for sharing information with these nodes. Other methods for communicating with carrier-sensing neighbors include the use of high power transmissions, forwarding of messages over multiple hops and the use of a lower rate transmission code at the physical layer. These methods have several drawbacks as described in Section 2.

Even though some carrier-sensing neighbors may be unable to decode a transmitted packet, they can all perceive a change in the strength of the carrier signal while a transmission is in progress. This change in signal strength can be used to infer certain characteristics of the transmission. For instance, the transmission duration can be sensed, which in turn indicates the size of the transmitted packet (assuming the data rate is known). If a node encodes information in the packet size, the information can be inferred by all carrier-sensing neighbors simply by sensing the carrier signal. By pre-agreeing on a communication protocol related to packet size, information can be communicated to all carrier-sensing neighbors.

In a previous work [11], we proposed a perceptive communication mechanism based on encoding information in the size of transmitted packets and used this to address the problem of intra-flow contention calculation. Using simulation, we evaluated the costs and benefits of the approach. In simulation, the received signal strength during a packet transmission is modeled as a constant value based on the distance between the transmitter and receiver. On real hardware, however, the received signal strength typically varies in each time slot[1]. The simulation model, therefore, is not realistic for slot-by-slot signal strength measurement and packet size detection. Our goal in this paper is to experimentally evaluate the ability of wireless nodes to correctly infer transmission durations from the detected carrier signal on real hardware. This ability is key for effective perceptive communication.

Similar to detecting transmission durations, carrier-sensing neighbors can also perceive the duration of silences, i.e. the idle time between adjacent transmissions. This characteristic can also be leveraged for perceptive communication. Specifically, if every transmitted packet is preceded by a *pre-frame* and a short silence, the length of the silence can be used to communicate information to all carrier-sensing neighbors.

In this paper, we make the following contributions: we propose a new mechanism for perceptive communication among carrier-sensing neighbors based on detection of silence durations and qualitatively compare it with our previously proposed mechanism based on packet size detection. We also experimentally evaluate the feasibility and performance of both mechanisms on real hardware. To this end, we implement a prototype of our perceptive communication mechanisms on the Mica2 mote platform [4]. We then perform several experiments to evaluate the accuracy and robustness of the mechanisms in different network scenarios. Our results demonstrate that packet sizes and silence durations can be correctly inferred from the received signal strength within a small margin of error in most scenarios.

The remainder of this paper is organized as follows. In Section 2, we briefly review some related work. Section 3 describes how perceptive information can be inferred from received signal strength measurements under ideal conditions. In Section 4, we give examples to illustrate how our perceptive communication mechanisms can be used to facilitate information sharing among carrier-sensing neighbors. Section 5 describes the implementation of our prototype, while Section 6 presents our experimental evaluation. Finally, Section 7 concludes the paper.

1 We define a time slot as the time required to transmit one byte over the radio.

2. Related Work

Previously proposed methods for communication with carrier-sensing neighbors include the use of high power transmissions and forwarding of messages over multiple hops [14]. These approaches have several drawbacks. High power transmissions lead to increased power consumption and reduced spatial reuse, while multihop message forwarding requires a relaying node and is therefore not guaranteed to reach all carrier-sensing neighbors in all scenarios. Another possibility is the use of a lower rate transmission code at the physical layer, which results in a larger reception range. However, a lower rate code may not be supported by the hardware. Also, it is impossible to make the increased reception range match the original carrier-sensing range in order to exactly reach all carrier-sensing neighbors.

Our perceptive communication approaches modify data transmissions to support communication with carrier-sensing neighbors. Hence, the additional energy consumption is negligible in comparison with high power transmissions. Further, since communication is based on the fundamental ability of carrier-sensing neighbors to detect a change in the strength of the carrier signal during a transmission, the communicated information is guaranteed to reach all carrier-sensing neighbors irrespective of their location and no relaying node is necessary.

Communication of information through time duration encoding has been explored previously in the context of covert timing channels [1, 5, 9, 10, 12]. Specifically, several researchers have explored the encoding of information in the inter-arrival times of packets into a queue [1, 5, 12]. The focus of this work has been primarily on the security aspects and capacity of such a communication system.

Our perceptive communication approaches are similar in concept to the covert timing channels. However, covert channels are typically intended to provide an additional means for secret communication where a regular communication channel already exists. Perceptive communication, on the other hand, enables communication between nodes that cannot otherwise communicate directly. Further, the challenges faced by perceptive communication are different from those relating to covert channels. Specifically, the detection of time intervals between packets is trivial for a node that is able to receive and decode the packets, as in the case of covert channels, but may not be so for a node that can only sense the transmissions. Our focus in this paper is on evaluating the feasibility of such detection by carrier-sensing neighbors in a wireless environment.

Zhu and Sivakumar [15] introduce the paradigm of communication through silence (CtS) for wireless sensor networks. To communicate information using CtS, a sender transmits a start signal and then remains silent for a specific

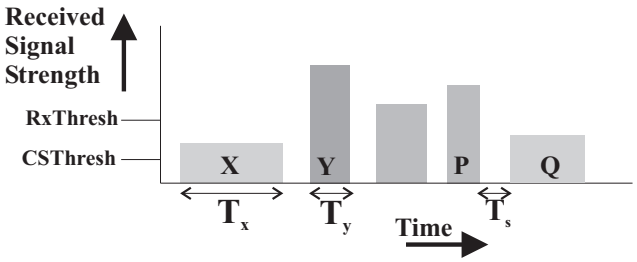


Figure 1. Diagram of received signal strength versus time.

number of time slots before sending a stop signal. The number of silent time slots is derived from the bit sequence to be transmitted. The receiver can then infer the bit sequence by counting the number of silent time slots and doing a reverse mapping. This scheme is intended for generic communication between sensors with the objective of saving energy. The authors present a simulation-based evaluation that assumes a lossless channel and perfect synchronization between nodes. In a realistic environment, the scheme can potentially suffer from a significant decrease in throughput and an increase in delay. Also, existing MAC protocols are incapable of supporting such a communication scheme. A new MAC mechanism is required, which presents several challenges as pointed out by the authors.

The purpose of our perceptive communication approaches is different from the CtS scheme. Instead of generic energy-conserving communication between nodes that are within transmission range, our approaches are designed for application-specific communication between carrier-sensing neighbors. As described in later sections, the duration of transmissions and silences intended for perceptive communication can be limited through the use of an appropriate codebook, and so the throughput degradation and delay increase are minimized. Further, our approaches can be directly integrated into existing CSMA MAC protocols. Most importantly, we evaluate the feasibility of our approaches through a real testbed implementation, and do not assume a lossless environment or perfect synchronization between nodes.

3. Carrier Sensing and Detection of Perceptive Characteristics

The duration of a transmission, as well as of the silence between adjacent transmissions, may be detected by all carrier-sensing neighbors by sampling the carrier signal. In this section, we describe how this detection can be accomplished in ideal circumstances. Later sections present our experimental evaluation in a realistic environment.

Figure 1 is an idealized graph of received signal strength over time at a given node. If there are no ongoing transmissions and the channel is idle, the received signal is comprised of noise, and its strength is typically small. When a transmission occurs at a node within CS range, the received signal strength is greater than the carrier-sensing threshold ($CSThresh$) and the receiving node is able to detect that a transmission is in progress. In the figure, the received signal strength of packet X is above $CSThresh$, so the node can detect this packet transmission. If the strength of the received signal is greater than the reception threshold ($RxThresh$), the contents of the packet can be decoded; this happens when the receiver is within reception range of the sender. Referring to the figure, packet Y can be received and decoded by the node since its received signal strength exceeds $RxThresh$.

When simultaneous transmissions occur, the received signal strengths of the packets overlap. However, the signal strength of the highest power packet dominates this measurement. The ability to correctly receive a packet in the presence of noise, collisions or other transmissions depends on the capture threshold of the wireless hardware.

Given the received signal strength measurements, a node can construct a graph of the channel state over time, similar to Figure 1. From this graph, it can determine the duration of transmissions. For example, in Figure 1, the node measures the duration of the received signal corresponding to packet X as T_x. Note that although packet X cannot be decoded, its transmission duration can still be determined by all nodes within CS range. Further, if the data rate is known, the transmission duration can be used to infer the size of the transmitted packet.

The channel state graph can also be used to measure the duration of silences, i.e. the time between adjacent transmissions, when the medium is idle. For example, in Figure 2, the node detects a silence between the transmissions of packets P and X and measures its duration as T_s.

Thus, in theory, a wireless node should be able to detect the length of most transmissions by carrier-sensing neighbors, as well as the durations of silences between these transmissions. Our goal in this paper is to determine whether this holds true on real hardware. Specifically, we examine how closely graphs of actual received signal strength versus time resemble the idealized graph of Figure 1.

4. Application of Perceptive Communication

Before proceeding to our experimental evaluation, we briefly discuss how our perceptive communication mechanisms can be applied in real wireless networks. Perceptive communication is useful in various scenarios that require information to be shared among carrier-sensing neighbors.

Examples of shared information include the priorities and maximum bandwidth requirements of flows for admission control and the identities of nodes for intra-flow contention calculation. Perceptive communication also enables sharing of other types of information, such as MAC layer congestion windows and queue states, which can be used by applications to better view and manage the shared medium.

In this section, we describe how our perceptive communication mechanisms may be used to accomplish information sharing among carrier-sensing neighbors. We focus on the packet size detection mechanism in Section 4.1, while Section 4.2 describes how the silence detection mechanism can be used. The strengths and weaknesses of the two mechanisms are analyzed in Section 4.3.

4.1. Packet Size Detection

In Section 3, we described how the transmission duration, and thereby the size, of a packet can be detected by all carrier-sensing neighbors. Communication with carrier-sensing neighbors can thus be achieved by encoding information in the size of transmitted packets. One simple application of this mechanism is discovering the identities of carrier-sensing neighbors. In multihop wireless networks, nodes often broadcast *Hello* messages to establish and maintain connectivity. The Hello messages can be sensed, but not decoded, by carrier-sensing neighbors outside the reception range. To communicate the identity of the transmitter to these nodes, the size of the Hello message can be modified by appending a *tail*, i.e. additional bytes, to the packet. Note that the tail only serves to modify the size of the packet and need not contain decodable information. On entering the network, each node selects a random unique size for its Hello message, which it communicates to all other nodes by flooding a packet through the network. All nodes maintain a mapping of network identities to corresponding Hello message sizes. Then, by monitoring the sizes of sensed packets and mapping them back to network identities, a node can determine the identities of its current carrier-sensing neighbors. Note that the set of permissible sizes for the Hello messages should exclude the sizes of regular data and control packets. This is to avoid incorrect inferences drawn by carrier-sensing neighbors from packets not intended for perceptive communication.

In our previous work, we used the packet size detection mechanism to address the problem of intra-flow contention calculation. Our solution involved modifying the sizes of route discovery packets (RREQ/RREP) by appending tails of appropriate lengths. Details of the solution can be found in the corresponding publication [11].

Other applications can similarly leverage the packet size detection mechanism for perceptive communication. Each application has its own codebook of packet sizes, where specific sizes are mapped to corresponding meanings. The codebook may either be pre-defined or created by the network nodes during the initialization of perceptive communication. Information sharing is then accomplished by measuring the sizes of sensed packets and looking up their meaning in the codebook. The codebook should exclude the sizes of regular data and control packets.

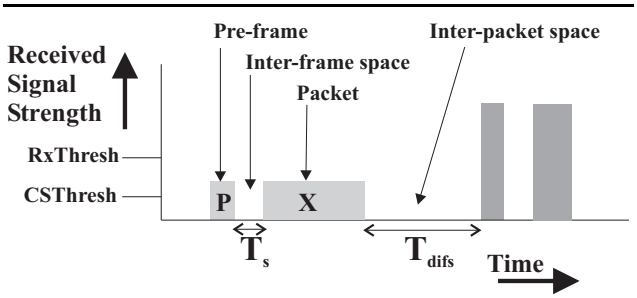


Figure 2. Use of silence detection.

4.2. Silence Detection

As described in Section 3, carrier-sensing neighbors can determine the duration of silence between transmissions by continuously monitoring the received signal strength. A node can therefore communicate with carrier-sensing neighbors by encoding information in silence durations. To accomplish this, every transmission is preceded by a *pre-frame* and a short silence that we refer to as the *inter-frame space*. The pre-frame need not contain any useful information; its purpose is to distinguish the inter-frame space preceding the actual transmission. The transmitter can then vary the length of the inter-frame space to communicate information to carrier-sensing neighbors. Figure 2 illustrates this mechanism. In the figure, the transmission of packet X is preceded by a pre-frame P and an inter-frame space of duration T_s. The value of T_s can be varied to communicate information.

The maximum length of the inter-frame space is limited by the *inter-packet space*, i.e. the minimum separation between packet transmissions as specified by the MAC protocol. For example, in the IEEE 802.11 protocol, the inter-frame space could be at most as large as DIFS. This restriction is necessary in order to prevent other nodes from accessing the medium during the inter-frame space. The inter-packet space, T_{difs}, is indicated in Figure 2, where $T_s < T_{difs}$.

The silence detection mechanism can be used for perceptive communication in a manner similar to the packet size detection mechanism. Each application can define a codebook that maps inter-frame space lengths to corresponding meanings. For example, consider the communication

of packet priorities to carrier-sensing neighbors. A specific inter-frame space length can be defined and used for each priority level, thereby enabling nodes to determine the priorities of all packets transmitted by carrier-sensing neighbors by simply sensing the medium. This knowledge can be useful for admission control and bandwidth allocation.

4.3. Analysis

The applicability of each perceptive communication mechanism to various scenarios depends on certain factors. As noted in Section 4.1, when using the packet size detection mechanism, the codebook should not include the size of regular data or control packets. If the size of regular packets vary significantly and cannot be pre-determined, it may be difficult to design an appropriate codebook.

The silence detection mechanism does not have the above limitation. However, in this case the maximum length of the inter-frame space is limited by the inter-packet space T_{difs} as explained in Section 4.2. This in turn restricts the size of the codebook. The packet size detection method, on the other hand, can potentially have a very large codebook, limited only by the maximum packet length, which is typically much larger than the inter-packet space.

In other words, the packet size detection mechanism is appropriate for applications that require a large codebook, provided the typical size of regular data and control packets in the network is known. The silence detection mechanism may be preferable when the codebook is small since it leaves the actual packets unmodified. The network cost/benefit trade-off for each approach depends on the particular application.

The use of multiple data rates in the network, such as when the auto rate adaptation [8] feature of the IEEE 802.11 protocol is used, restricts the applicability of the packet size detection mechanism. Without knowledge of the data rate, the size of a packet cannot be inferred from its transmission duration. However, information can still be communicated perceptively by encoding it in the transmission duration instead of the transmitted packet size. The codebook in this case maps transmission durations, instead of packet sizes, to specific meanings. The silence detection mechanism is still applicable in this scenario.

5. Implementation

We have created a prototype implementation of our perceptive communication mechanisms on the Mica2 mote. This platform was chosen since it is the only commercially available platform that exposes an API to access the received signal strength during each time slot[2], which is essential for our packet size detection and silence detection mechanisms.

The Mica2 mote is produced by Crossbow [4]. It consists of a 7.38MHz Atmel 128 microprocessor with a 4KB EEPROM, 128KB program flash memory and 512KB flash data memory. For wireless communication, a CC1000 [3] radio operating at 914MHz with a whip antenna is included on the mote. Additionally, a sensor board may be attached to the computation and communication unit. For our experiments, we do not make use of this sensor board.

The Mica2 motes run the Tiny OS [13] operating system, which we utilize to develop and test our mechanisms. Tiny OS provides infrastructure to access most of the capabilities of the mote. Of importance to our implementation, a CSMA MAC layer and a simple interface to the received signal strength indicator (RSSI) are available.

For the inter-frame spacing tests, we modified the Tiny OS MAC protocol to transmit an 8-byte pre-frame followed by a short inter-frame silence prior to every packet transmission. A minimum inter-packet spacing of 24 slots is also enforced, i.e. nodes do not begin a pre-frame transmission until the medium is sensed to be idle for at least 24 continuous slots. As mentioned in Section 4.2, the maximum length of the inter-frame silence is limited by the inter-packet space length and therefore cannot exceed 24 slots. The inter-packet space is necessary to prevent other nodes from accessing the medium during an inter-frame space.

The RSSI value indicates the strength of the detected carrier signal, which increases when a nearby node is transmitting. The RSSI is used by the MAC layer to perform carrier sensing, as well as by our perceptive communication code. Our code samples the RSSI value in each time slot. The resulting series of RSSI values is analyzed to detect packet transmissions. We describe our packet detection algorithm in Section 5.1. Silences are inferred from the packet detection output, so no separate silence detection algorithm is necessary.

Since the Mica2 mote has relatively low computation power and memory, we do not directly analyze the RSSI data on the mote. The mote collects and periodically transmits sets of RSSI values to an attached laptop computer via the MIB510 mote programming board and a serial cable. The laptop computer reads the stream of values from the serial port and performs an offline analysis. Our analysis algorithm is fairly simple, and it can be executed in real time on a more powerful wireless device such as a PDA or a laptop.

2 The duration of a time slot is 216 microseconds on the CC1000 radio used in the Mica2 motes.

An interesting artifact of the mote's weak computation power is that the recording of RSSI values in every time slot, together with the periodic transmission of the recorded values over the serial cable, causes the mote to become overloaded. In the overloaded state several RSSI readings are lost. As a compromise, we record the RSSI values in every other time slot, which reduces our detection accuracy. Although this problem should not arise on a more powerful device, we choose to work with the Mica2 motes because of their support for signal strength sampling.

5.1. Packet Detection Algorithm

The trace of RSSI readings collected by the mote is analyzed to detect packet transmissions. We use a very simple algorithm for this analysis. The objective of the algorithm is to identify groups of consecutive RSSI readings that satisfy certain criteria qualifying them as possible packet transmissions. The length of the group then indicates the size of the transmitted packet. The RSSI readings that lie in between successive detected packets constitute silence; the silence duration is given by the number of such consecutive readings.

The qualification criteria for a valid group are the following. First, the maximum and minimum RSSI readings in the group should not differ by more than a certain limiting value. We observed that RSSI samples fluctuate widely when the medium is idle and remain relatively stable when a transmission is in progress. This criteria thus helps us to distinguish between valid packet transmissions and noise. By experimenting with different values, we found that the limiting value should be set to about 2.34 dBm; this setting resulted in the highest detection accuracy in our experiments. We also observed that the weaker the received signal from a packet transmission, the more it fluctuates while the transmission is in progress; this is because a weak signal is more significantly affected by ambient noise. Therefore, to improve our accuracy of detecting weaker packets, we allow up to two outliers, i.e. samples that violate the first criteria, as long as these samples do not lie on the edges of the packet. This heuristic was also determined empirically.

Second, the average signal strength of a group should be sufficiently higher than the ambient noise. This helps us avoid mistaking noise for a valid transmission.

Unmodified Tiny OS imposes an upper limit of 29 bytes on the payload size of a packet. This limit, together with the size of the headers appended to the packet by the operating system, places upper and lower bounds on the size of a valid packet. Our packet detection algorithm takes advantage of these bounds. Specifically, if the length of a detected group does not satisfy these bounds, the group is not considered to be a valid packet. Protocols designed to use our mechanism to communicate with carrier-sensing neighbors could place similar bounds on their packet sizes to improve detection accuracy. This would not limit the size of all transmitted packets, rather only those that are meant to convey information to carrier-sensing neighbors.

The above criteria are used to identify groups of RSSI readings that potentially represent packet transmissions. The packet detection algorithm starts building a group at the first RSSI reading and continues adding subsequent readings to the group as long as the first criteria is not violated. When a reading violates the first criteria, the group is considered to have terminated. The algorithm then checks whether the group satisfies the remaining criteria. If yes, it represents a valid packet. If no, the process is restarted from the second reading in the group. Although this algorithm is simple, it performs well as indicated by the results presented in the following section. A more complex algorithm would further enhance detection.

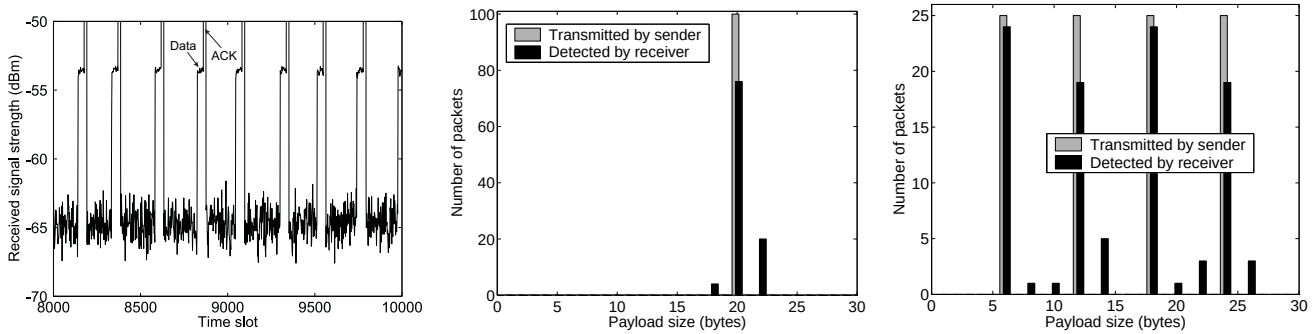
Given packet detection as described above, inter-frame silence duration is simply measured as the number of slots of silence between each detected transmission. Note that the maximum length of an inter-frame silence is limited by the minimum inter-packet spacing (24 slots in our experiments). Silences of longer durations are not inter-frame silences and are therefore ignored.

6. Experimental Evaluation

In this section, we present the experimental evaluation of our perceptive communication mechanisms on the Mica2 motes. Our objective is to determine the correctness and robustness of the mechanisms in different network conditions. Specifically, we evaluate the performance by varying three important parameters: received signal strength, traffic load and probability of packet collisions. The significance of these parameters is as follows.

The strength of the received signal is decreased by an increase in the distance between the transmitter and receiver, obstacles in the transmission path and a decrease in transmission power. To enable correct communication with all nodes within CS range, it is critical that transmissions be detected correctly so long as the received signal strength is greater than the CS threshold. For most devices, the CS threshold is set well above the ambient noise. It is therefore desirable that our mechanisms should perform correctly when the received signal strength from a transmission is greater than the ambient noise level. Hence, we evaluate our mechanisms under varying received signal strengths.

As the network load increases, the idle time between subsequent packet transmissions diminishes. This makes it more challenging for our detection mechanism to correctly identify the start and end of each transmission. Further, if multiple transmissions collide, it becomes still more difficult to detect individual packets. We therefore test our



(a) RSSI readings. (b) Results with uniform packet size. (c) Results with varying packet size.

Figure 3. Results for packet size detection in simple test scenario.

prototype under increasing traffic load and probability of packet collisions.

In each experiment, we have one receiver mote and one or more transmitter motes. The receiver mote is connected to a laptop computer. Each transmitter transmits a fixed number of packets of a known size. For the silence detection experiments, the inter-frame space duration is also known. The RSSI sampling code executes within the MAC layer of the receiver mote and periodically transmits sets of RSSI readings to the laptop. We then perform an offline analysis of the readings to detect packet sizes and inter-frame space lengths.

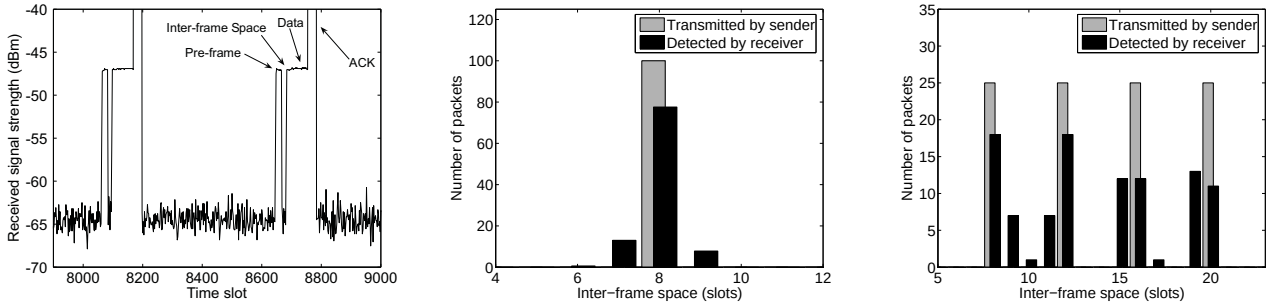
Performance is measured by the number of packets whose duration or inter-frame space are correctly detected and the number of incorrect detections. Incorrect detections are comprised of detections that either do not correspond to an actual transmission, or incorrectly estimate the packet size or inter-frame space duration. In the following subsections, we describe our experiments and present our results. All experiments were performed in an indoor environment with obstacles such as furniture, walls and people.

6.1. Simple Test Scenario

We first consider a simple test scenario that helps us gain insight into the system and establishes a baseline for the performance of our mechanisms. In this experiment, we have a single transmitter placed within a few inches of the receiver mote. The transmitter transmits 100 packets at the rate of 10 packets per second. Each packet has a payload of 20 bytes. The total size of each transmission, including the preamble and sync codes appended by the radio, is 36 bytes. For the silence detection experiments, a pre-frame of 8 bytes followed by 8 slots of silence (unless otherwise stated) is employed prior to normal data transmission. A minimal inter-packet space of 24 slots is enforced.

Figure 3 presents the results of our experiment for packet size detection. In Figure 3(a), we plot the RSSI values sampled by the mote over 2000 time slots. The figure shows the data for a subset of the total experiment time in order to improve clarity; the same RSSI pattern was observed throughout the experiment. As seen in the figure, the ambient noise lies in the -66 dBm to -62 dBm range. The signal strength of each packet is around -54 dBm, well above the ambient noise level. We observe that each packet reception is followed by a few time slots where the received signal strength further increases. This corresponds to the acknowledgments transmitted by the receiver as part of the MAC protocol. During this time, the receiver's radio is in transmit mode so the received signal strength value has no meaning and the sampling API returns a value of zero. Note that the graph is cropped for clarity.

Figure 3(b) illustrates the performance of the packet detection mechanism. As seen in the figure, 100 packets with a payload size of 20 bytes each are transmitted by the sender. Of these, 76 packet sizes are detected accurately. The remaining packets are detected within an error of 2 bytes. This error arises in part because we sample RSSI values in alternate time slots to avoid overloading the motes. The other reason for the error is that the time slots of the transmitter and receiver motes are not perfectly synchronized. As a result, the RSSI readings during the first and last time slots of a packet transmission do not always correctly reflect that a transmission is in progress. The second reason makes it impossible to detect the size of a packet transmission with perfect accuracy even if the RSSI values are sampled in every time slot. An error of +/-1 byte is unavoidable. Further, since we sample only in alternate time slots, this error is effectively doubled in our experiments.



(a) RSSI readings.

(b) Results with uniform inter-frame space duration.

(c) Results with varying inter-frame space duration.

Figure 4. Results for silence detection in simple test scenario.

From these observations, we conclude that an error of +/-2 bytes cannot be avoided, and so we allow for this error in our remaining results. In other words, a detection is considered to be accurate if the error in packet size is no more than 2 bytes.

To verify our observations, we repeat the experiment with varying packet sizes. In this case, the transmitter sends 25 packets each with payload size 6, 12, 18 and 24. The results are presented in Figure 3(c). We observe once again that all packet sizes are correctly detected within an error of 2 bytes.

Figure 4 presents the results for the basic silence detection experiment. Figure 4(a) is very similar to Figure 3(a) except that each packet transmission is preceded by a short pre-frame and a short silence before the normal data transmission and acknowledgment. In the figure, only 600 time slots are shown to improve the visibility of the short inter-frame space.

Figure 4(b) illustrates the performance of the silence detection mechanism. Of the 100 transmitted packets, the durations of 78 inter-frame spaces are detected accurately. The remaining are detected within an error of 2 slots for reasons related to transmitter-receiver synchronization discussed previously.

To verify that the silence detection properly detects various inter-frame space durations, the experiment is repeated with varying inter-frame space lengths. In this case, the transmitter sends 25 packets each with an inter-frame space of 8, 12, 16 and 20 slots. The results are presented in Figure 4(c). We observe that all the inter-frame space durations are correctly detected within our tolerated error.

We now proceed to the more complex test scenarios. In the next section, we describe our experiment for evaluating the detection of perceptive information with different values of received signal strength.

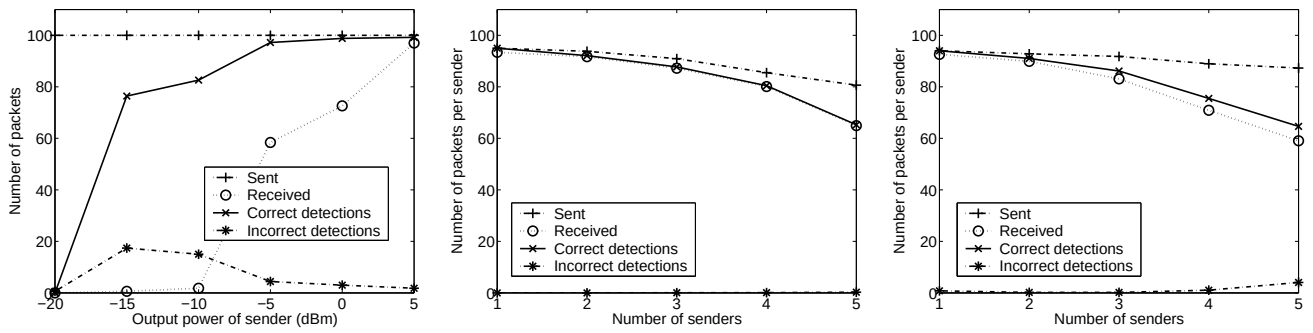
6.2. Effect of Received Signal Strength

To observe performance with different values of received signal strength, we once again have a single transmitter that sends 100 packets, each with a payload of 20 bytes, at the rate of 10 packets per second. The transmitter is placed at a distance of approximately 25 feet from the receiver with several obstacles, such as furniture including metal shelves, obstructing line of sight. We vary the output power of the transmitter mote from -20 dBm to 5 dBm in different tests. This significantly varies the received signal strength at the receiver. Specifically, we observe that at -20 dBm, the received signal is at the same level as the ambient noise, and therefore indecipherable. We execute 5 runs of the experiment at each transmit power level and average the results.

The results of the packet size detection experiment are presented in Figure 5(a). In the figure, we plot the number of packets sent by the transmitter mote, the number of packets received, i.e. decoded, by the receiver mote, the number of packets whose sizes are correctly detected by the packet size detection mechanism and the number of incorrect detections. The transmitter mote transmits 100 packets in each test.

At the lowest power level (-20 dBm), we find that none of the packets are received or correctly detected. This is because the received signal in this scenario is at the same level as the ambient noise and therefore cannot be deciphered. There are no incorrect detections in this scenario either, which shows that the packet detection algorithm does not mistake noise for a packet transmission.

As the output power of the transmitter increases, the number of correct detections improves significantly. When the output power is set to -15 dBm or -10 dBm, the number of received packets is close to zero. This indicates that the receiver is not within reception range of the transmitter. However, we are still able to correctly detect the sizes of about 80% of the packets, even with our simple packet



(a) Effect of received signal strength. (b) Effect of congestion. (c) Effect of collisions.

Figure 5. Results for packet size detection in different network scenarios.

detection algorithm. This shows that the use of packet size detection for communicating with nodes outside reception range is effective. The number of incorrect detections in these scenarios is about 20%. This is because our simple algorithm sometimes miscalculates the packet size. We are confident that the detection performance can be further improved with a more sophisticated detection algorithm.

At output power settings of -5 dBm and 0 dBm, almost all transmitted packets are correctly detected, although many packets are not received. For example, at output power of -5 dBm, only 60 packets are received while 97 packets are correctly detected. The number of incorrect detections is less than 5% in these scenarios. Finally, when output power is set to 5 dBm, the signal is sufficiently strong that almost all transmitted packets are correctly received and detected.

This experiment verifies that carrier-signal-based packet size detection is feasible even when the receiver and transmitter are not within reception range, as long as the received signal strength is greater than the ambient noise level. Our approach is therefore effective for communicating with all carrier-sensing neighbors.

Since the performance of the silence detection mechanism depends on correct detection of transmitted packets, the experimental results for the inter-frame spacing experiments are nearly the same as the results of the packet size detection experiments. Hence we do not include the graphs for these experiments in the paper.

6.3. Effect of Traffic Load and Collisions

To increase the traffic load and likelihood of collisions, we introduce more transmitter motes in this experiment. All the transmitters are placed within a few inches of the receiver mote. Each transmitter generates packets in the same manner as in the previous experiment. Note that some randomness, or jitter, is introduced in the time interval between consecutive packets at each transmitter in order to avoid synchronized transmissions. We vary the number of transmitters from one to five with five runs of the experiment in each configuration.

The results of the experiment are presented in Figure 5(b). We plot the average number of packets sent, received and detected per transmitter mote. Note that the average number of packets sent per transmitter decreases as the traffic load increases; this is because the MAC layer is unable to transmit some packets due to congestion. The packet detection mechanism performs very well in this scenario and detects almost all packets correctly. The number of incorrect packet detections is close to zero. This verifies that the detection performs well even when packets are separated by few idle time slots.

Our experiment for examining the effect of collisions is identical to the previous experiment, except that we disable carrier sensing at the transmitter motes. In other words, the motes no longer sense the carrier signal to determine whether the medium is idle before beginning a transmission. In this experiment, as load increases so does the likelihood of collision. Our results, presented in Figure 5(c), demonstrate that the packet size detection also performs well in this scenario. The percentage of received packets drops with increasing number of senders due to collisions. However, the detection mechanism is able to correctly detect some of the colliding packets as well. The average number of incorrect detections is less than five per transmitter for all scenarios.

Again, since our inter-frame spacing measurement depends on packet detection, the experimental results for the silence detection experiments are nearly the same as the packet size detection. We therefore do not include them in this paper.

7. Conclusion

Communication among carrier-sensing neighbors is essential for operations that affect the state of the shared medium, such as medium access and admission control. Although carrier-sensing neighbors may not necessarily be able to decode the contents of a transmitted packet, they can perceive certain characteristics of the transmission, such as a change in the level of the carrier-signal. This perceivable information can be used to infer the duration of the transmission and of the silence between transmissions. By preagreeing on a protocol related to packet size or silence duration, nodes can effectively communicate information to all carrier-sensing neighbors during normal data transmission.

In this paper, we performed an experimental evaluation to determine the feasibility and effectiveness of this idea. We implemented a prototype on the Mica2 motes and tested it under a variety of network conditions. Through our experiments, we found that it is not possible to detect transmission or silence durations with perfect accuracy. An error of +/-1 slot is unavoidable due to the lack of synchronization between the transmitter and receiver, which affects the signal strength measurements at the edges of the packet. However, by accounting for this error when designing the communication protocol, its effect can be mitigated.

Our experiments show that packet size detection and silence detection from signal strength measurements is effective under different conditions of received signal strength and traffic load. We emphasize that the algorithm we use to detect packets from the RSSI traces is simple. A more sophisticated detection algorithm is likely to further improve the results. Thus, this paper verifies that the carrier-signal-based communication mechanism is effective for sharing information with carrier-sensing neighbors during normal data packet transmission. In addition to the motes, we expect that any PHY and MAC protocol that utilizes carrier-sensing [6, 7] should be capable of sharing information in a similar fashion.

Acknowledgments

This work is supported in part by NSF Career Award CNS-0347886 and by NSF NeTS Award CNS-0435527.

References

[1] V. Anantharam and S. Verdu. Bits through Queues. *IEEE Transactions on Information Theory*, 42(1):4–18, January 1996.

[2] I. D. Chakeres and E. M. Belding-Royer. PAC: Perceptive Admission Control for Wireless Mobile Networks. In *Proceedings of the* 1^{st}*International Conference on Quality of Service in Heterogeneous Wireless/Wired Networks (QShine)*, pages 18–26, Dallas, TX, October 2004.

[3] Chipcon. CC1000. http://www.chipcon.com/files/CC1000_Data_Sheet_2_1.pdf.

[4] Crossbow, Inc. Mica2 Sensor Platform. http://www.xbow.com.

[5] J. Giles and B. Hajek. An Information-Theoretic and Game-Theoretic Study of Timing Channels. *IEEE Transactions on Information Theory*, 48(9):2455–2477, September 2002.

[6] IEEE Computer Society. IEEE 802.11 Standard, IEEE Standard For Information Technology, 1999.

[7] IEEE Computer Society. IEEE 802.15.4 Standard, IEEE Standard For Information Technology, 2003.

[8] A. Kamerman and L. Monteban. WaveLAN 2: A High-performance Wireless LAN for the Unlicensed Band. In *Bell Labs Technical Journal*, Summer 1997.

[9] J. K. Millen. Finite-State Noiseless Covert Channels. In *Computer Security Foundations Workshop*, pages 81–86, Franconia, NH, June 1989.

[10] I. S. Moskowitz and A. R. Miller. Simple Timing Channels. In *IEEE Computer Society Symposium on Research in Security and Privacy*, pages 56–64, Oakland, CA, May 1994.

[11] K. Sanzgiri, I. D. Chakeres, and E. M. Belding-Royer. Determining Intra-Flow Contention along Multihop Paths in Wireless Networks. In *Proccedings of the* 1^{st} *Annual International Conference on Broadband Networks (Broadnets)*, San Jose, CA, October 2004.

[12] R. Sundaresan and S. Verdu. Robust Decoding for Timing Channels. *IEEE Transactions on Information Theory*, 46(2):405–419, March 2000.

[13] University of California, Berkeley. Tiny OS. http://www.tinyos.net.

[14] Y. Yang and R. Kravets. Contention-Aware Admission Control for Ad Hoc Networks. Technical Report 2003-2337, University of Illinois at Urbana-Champaign, April 2003.

[15] Y. Zhu and R. Sivakumar. Challenges: Communication through Silence in Wireless Sensor Networks. In *MobiCom*, Cologne, Germany, August 2005.