Exploiting The Capture Effect For Collision Detection And Recovery

Kamin Whitehouse   Alec Woo   Fred Jiang   Joseph Polastre   David Culler
{kamin,awoo,xjiang,polastre,culler}@cs.berkeley.edu
Department of Computer Science
University of California, Berkeley

Abstract

In this paper we evaluate a technique to detect and recover messages from packet collisions by exploiting the *capture* effect. It can differentiate between collisions and packet loss and can identify the nodes involved in the collisions. This information is provided at virtually no extra cost and can produce significant improvements in existing collision mediation schemes. We characterize this technique using controlled collision experiments and evaluate it in real world flooding experiments on a 36-node sensor network.

1 Introduction

The *capture* effect, also called *co-channel interference tolerance*, is the ability of certain radios to correctly receive a strong signal from one transmitter despite significant interference from other transmitters. This property violates the "collision as failure" assumption that packet collision results in packet corruption, which is often seen in simulation [25], theoretical analysis [18], and collision avoidance schemes [23]. However, capture can also be beneficial; it has been exploited by many MAC and networking protocols to prevent packet collisions, increase network throughput and decrease delay. In theory, capture could provide additional gains through collision detection and the recovery of the stronger packet from collisions.

In this paper, we demonstrate and empirically evaluate collision detection and recovery using capture on a standard low-power radio that is commonly used in sensor networks. The collision detection scheme explored in this paper has three important properties:

- It can differentiate between a packet collision and packet loss.
- Collision detection happens at the receiver.
- It is often possible to identify the transmitters involved in the collision.

This is different than the information provided by Acknowledgments, with which the transmitter can only positively identify packet *reception*. If an acknowledgment is not received, the transmitter can not differentiate between loss due to a collision and loss due to channel noise. Furthermore, if the packet was not received for either reason, the receiver does not even know it was sent. Finally, neither the transmitter nor the receiver can identify the other nodes involved in a collision. Collision detection using capture augments MAC-level detection schemes like Acknowledgments and opens doors for completely new collision mediation techniques because it provides new information at virtually no extra cost.

Collision detection is particularly important in wireless sensor networks because collision avoidance schemes such as RTS/CTS do not help in the face of asymmetric links and broadcast-based communication, and many common sensor network protocols such as flooding and data aggregation are especially prone to collisions. Collision detection and recovery using capture can produce significant improvements in several such algorithms. For example, if a transmitter in an aggregation tree could differentiate between a high collision rate and low link quality to its parent, it could decide whether to change its transmission schedule or to find a new parent altogether. Similarly, if a parent could recover the identities of the transmitters involved in each collisions, it could automatically reschedule them. Even without new collision avoidance or scheduling techniques, exploiting capture increases throughput in the flooding experiments performed in this study nearly 10% simply by recovering packets from collisions.

We characterize this collision detection and recovery technique through controlled collisions by generating two packets with a precise time difference Δt. The technique detected 42% of all collisions without making any changes to existing packet structures or mac protocols. However, an extension that does change the packet format can increase detection to nearly 92% of all collisions. Packet recovery is near 92% even without the extension. In a second experiment we verify that this technique works in an uncontrolled network scenario. A 36-node network flood was generated with multiple transmission power settings and the number of collisions, detections and recovered packets was recorded. A network flood was chosen because it is a very common protocol for data and command dissemination in sensor networks. In this experiment, we also empirically characterized the relationship between capture and the distances between nodes in a collision.

The rest of this paper is organized as follows: Section 2 describes the capture phenomenon, ways in which it has been exploited, and other techniques for collision detection. Section 3 describes the process of collision detection and recovery. Section 4 describes controlled collision experiments where we characterize the detection and recovery techniques. Section 5 evaluates how often collision detection and recovery may take place in natural multi-hop networking environments. Section 6 describes an analysis of the relationship between capture and distance. Section 7 discusses ways in which the new information provided by this technique can improve existing protocols. Section 8 concludes and describes the types of networks in which this technique is most valuable.

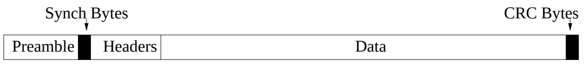
2 Background

The *capture* effect is the ability of some radios to receive a signal from one transmitter despite interference from another transmitter, even if the relative strengths of the two signals are almost the same [19]. Capture is commonly believed to occur in FM receivers but not in AM receivers; it is the reason why FM radios have a clean crossover point between regional radio stations while AM radios combine overlapping stations and create static. However, studies have shown that, when designed properly, OOK/ASK radios can actually exhibit the capture effect to a greater extent than FSK radios [3]. Capture has also been demonstrated on a wide variety of transceivers including television [24], radio broadcasting, Aloha networks [6], the DARPA packet radio, 802.11 radios [8], Bluetooth radios [9], and cellular systems [21].

The capture properties of a particular radio depend on its specific implementation. Some radios can only tolerate interference that is much smaller than the amplitude of the captured signal while others can differentiate between signals of almost exactly the same amplitude. Several theoretical models of capture have been proposed and compared with empirical traces [8]. Depending on the type of modulation and decoding scheme, the signal strength difference required for capture to occur is estimated to be as large as 1-3dB [2] or as little as 0.17dB [26]. Several techniques have been shown to tune the capture properties of a receiver [12, 13].

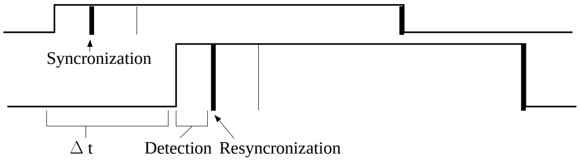
Several studies have explored the impact of capture on the performance of common networking protocols. For example, capture has been shown to increase throughput and decrease delay in Aloha [7] and 802.11 networks [30, 10, 14]. Theoretical work has shown that, in the presence of capture, the network achieves optimal throughput and energy efficiency when all nodes transmit at maximum power [28]. Other studies proposed techniques to deliberately exploit capture. For example, CDPA exploits capture to increase spatial reuse of the same channel among multiple radio cells [11] and several protocols exploit capture to increase throughput and delay in ALOHA networks [6].

Perhaps the closest protocol to the one presented in this paper is Power Control during Transmission (PCT), which increases transmission power over time from the beginning to the end of each successive packet [17]. This technique prevents collisions in the presence of capture by trying to ensure that the end of a packet will never be overpowered by the beginning of another packet. In this scheme, the first of two overlapping packets is almost always received properly. However, this technique can not perform collision detection, which is a key benefit of the technique presented in this paper.



(a) **PacketFormat** A packet contains a preamble, a synchronization byte, the packet headers, the data and CRC bytes.

(b) **Stronger-first** the radio synchronizes with the stronger packet and receives it normally.



(c) **Stronger-last** the radio synchronizes with the weaker packet, after Δt it detects the preamble of the stronger packet and resynchronizes.

Figure 1: **Collision Detection and Recovery**

Wireless MAC-level techniques such as Acknowledgments can achieve limited collision detection, but can not differentiate between collisions and packet loss and cannot identify the nodes involved in a collision. The collision detection scheme presented in this paper is a link-level decoding technique that can be used to compliment wireless MAC protocols with this information.

Techniques such as CDMA perform a superset of the collision detection techniques described here by receiving multiple messages on the same channel simultaneously. However, the technique described in this paper can be performed on extremely simple low-power radios such as those used in sensor networks.

3 Collision Detection and Recovery

In the presence of capture there are two types of collisions: *stronger-first* in which the stronger packet comes first and *stronger-last* in which the stronger packet comes last. In stronger-first collisions, the stronger packet is received normally because the radio synchronizes with the stronger packet and the weaker signal does not cause interference due to capture. In stronger-last collisions, however, the radio synchronizes with the weaker packet but reception fails because stronger packet later captures the channel and corrupts the tail end of the first packet. Thus, stronger-last collisions result in the loss of both packets even in the presence of capture.

The collision detection and recovery technique we

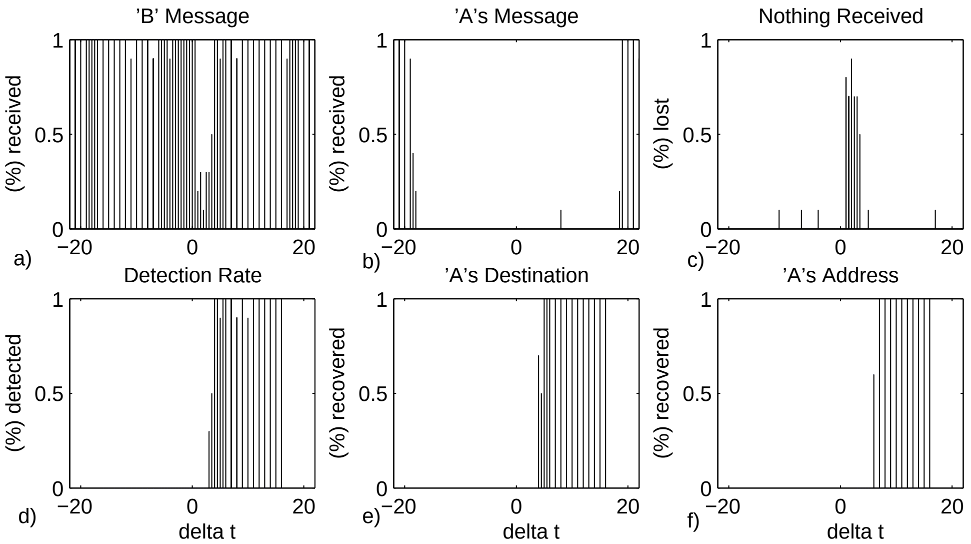


Figure 2: **Three Node Experiment** Nodes A and B are senders and C is a receiver. The 'X' axis indicates the time Δt between transmissions from node A and B. The figure shows a-b) the percent of packets received at C from B and A, respectively, for each value of Δt. c) the percent of collisions in which both packets were lost. d) the percent of collisions that were detected e-f) the percent of collisions where the destination and source address, respectively, were recovered.

use in this paper is to continually search for a new preamble even during packet reception and to resynchronize to the new packet in the stronger-last scenario. This is illustrated in Figure 1(c). A collision is *detected* if a preamble is found during the reception of another packet. If the receiver can properly stop receiving and resynchronize, it can also *recover* the second packet. Because the first packet is already partially received, the information in the header fields can often be recovered and used to identify the transmitter of the corrupted packet. This is only true if the colliding packet arrives after the header of the first packet. This simple technique requires no change to the packet format or the mac layer and requires only a modest change to the link-level receiver processing.

The basic form of this technique can only detect stronger-last collisions, yielding at most a 50% detection rate because stronger-first collisions are not detected. This basic form is useful in existing systems today because it does not require any changes to existing packet structure or MAC protocols. A straightforward extension of this work would be to use a termination symbol at the end of each message. The receiver would then be able to detect stronger-first collisions by searching for extra termination symbols after receiving each message to see if other, weaker messages had been arriving. If the termination symbol contains the source address, the receiver could also recover the identity of the transmitter in stronger-first collisions. This extension requires a modification to the original packet structure but would increase our detection rate to a theoretical maximum of nearly 100%. This extension, however, has not been implemented and the experiments in the following sections evaluate only the basic technique.

One limitation of this technique is that it relies on information at the data level and so it is not as transparent as techniques that might be implemented completely at the physical level. For example, the transmitter must perform byte-stuffing to prevent sending messages that inadvertently contain a preamble.

Another limitation is that the certainty of the collision detection decreases as the time difference between the transmissions decreases. For example, if the last packet begins during the preamble of the first packet, the receiver will observe an extra long preamble followed by a start symbol. This can be interpreted as a collision, but not with high certainty. If the two packets arrive at exactly the same time, the collision can not be detected at all.

4 Experimental Characterization

We characterize the collision detection and recovery techniques using controlled collisions by generating two packets with a precise time difference Δt. The goal of this experiment is to identify how well and under what circumstances collision detection and recovery will work. Two transmitters A and B and a receiver are placed in a one-meter isosceles triangle. The receiver employs collision detection and recovery as described in Section 3. The transmitters are synchronized to transmit packets at times t_A and t_B, such that the time between them was $\Delta t = t_B - t_A$. Thus, when Δt is positive A transmits first and when it is negative B transmits first. Δt is varied from $-23ms$ to $23ms$ at 1ms intervals and 10 collisions were generated at each value of Δt. The packets are $17.9ms$ long, so 0.5ms intervals are used around 0ms and ±18ms for higher resolution data. Time synchronization was only accurate to $1ms$.

This experiment uses the Berkeley Mica2 hardware platform and TinyOS. The Mica2 platform consists of a 7.38MHz Atmega 128 microprocessor and a 433MHz Chipcon CC1000 [1] radio transceiver. Its transmit power can be changed from -20dBm to +10dBm and RSSI values can be obtained through the on-board 10-bit ADC on the processor. Manchester encoding is used at the physical layer, and delivers 19.2kbps encoded bandwidth. No extra byte-level coding or forward error correction is used in the default networking stack in TinyOS. The 36 byte packet format is illustrated in Figure 1(a) and contains a 5-byte preamble, a 2-byte start symbol, a 5-byte header, and a 2-byte CRC checksum. The destination address is the first 2 bytes in the header. In our experiments, we place the source address in the first 2 bytes of the payload. The antenna used is a standard quarter wave-length whip antenna which, during these experiments, is in a random orientation.

While a 0dBm transmit power was used on both nodes, slightly moving one of the senders or adjusting the antenna orientation changed the power relationship between the two senders at the receiver. It was fairly difficult to find a "null" point at which neither transmitter was received due to the difference in received energy from the transmitters being below the SNR threshold of the receiver. In fact, we confirmed that the power difference required to cause one transmitter to be received over the other one was unobservable using the 10bit ADC to sample the RSSI pin on the radio. In our experiments, we deliberately move node B such that its signal was stronger than node A at the receiver.

Figure 2 summarizes the findings from this experiment. The Y axis is the percent of packets received while the X axis is the Δt between packet start times. In the left half of the graphs B sends first while on the right half of the graphs A sends first. At the two edges of the graphs where $|\Delta t| > 17.9ms$ the messages do not overlap in time. At $x = 0$ they are both sent at the same time.

Figure 2.a shows that B is almost always received. Figure 2.b shows that A is also almost always received when its message does not overlap in time with B. When B's message arrives after the preamble and start bytes of A's message, ie. when $\Delta t > 3ms$, a collision is almost always detected because two sets of preambles and start bytes are received. If Δt were uniformly distributed in real network scenarios, collision detection would have a 42% success rate. Figures 2.e and f show that, even though A's message is overpowered, the destination and address fields can be reliably recovered when B's message comes sufficiently late, ie. when Δt was large enough.

Packet recovery yields similar results, recovering one packet in 92% of all collisions. This is up from the 42% detection rate because recovery works in both stronger-first and stronger-first collisions. Recovery should produce a higher yield than detection because it does not need to receive two start symbols in order to work. In our implementation, recovery does not work for $0 < \Delta t <= 3$ms, as shown in Figure 2, because once the preamble of A is detected, the software waits up to five byte times for the arrival of the start bytes. However, they never arrive because they are overpowered by B's preamble. By the time B's start bytes arrive, the software has gone back to looking for preamble bytes and the radio does not synchronize to B's packet. This is a software problem in our implementation that reduces recovery rates from 100% to 92% of all collisions, but it is not a fundamental limitation of the technique.

5 Evaluation in Flooding Scenario

In this section, we identify how often collision detection and recovery applies in an environment where 1) there can be more than two transmitters and 2) Δt is not carefully controlled but is determined by the CSMA protocol. We create a broadcast storm [22] scenario because of 1) its common usage in sensor networks and 2) its high likelihood of collisions due to the many correlated transmissions. The results of this section may generalize to any traffic load with correlated transmission, such as event-detection, multi-hop routing, etc. that would also generate a large number of collisions.

Thirty-six nodes are deployed in a 9x4 grid with approximately two meter spacing. The nodes are on the floor in a large room with many obstacles such as tables, chairs and couches. One node in the corner emits a message with a sequence number to trigger the network flood. Every other node retransmits the message with that sequence number once, with a random time up to 120ms after it first received that message. If the radio channel is busy, it takes another random backoff of at most 12ms and retries, as implemented by the CSMA MAC protocol available with TinyOS [20]. We use transmission powers of -18, -10, -4, 0, and 4.5dBm and 5 floods are performed at each power setting. At -18dBm, the radio range is approximately two to three meters while at 4.5dBm the radio range covers the entire network. Each node records the timestamp of each message it sends and receives and collisions are analyzed after the experiment. A *collision* is said to take place at a receiver if two or more transmitters that have good connectivity with a receiver transmit messages that overlap in time.

Figure 3 summarizes the findings from this experiment. The Y axes are percents while the x axes are power settings (dBm). These plots are box and whisker plots, which indicate a data set with four statistics. The central bar is the median while the box indicates the upper and lower quartiles. The whiskers extend to cover the rest of the data except for the outliers, which are indicated by $+$ marks. Outliers are points that fall outside the box by more than 1.5 times the inner quartile range. Where there is no central bar, the median falls on the quartile mark.

5.1 Collision and Capture Frequency

With 36 nodes, each transmitting once per experiment, five experiments per power setting and five power settings, the experiments could have included up to 900 transmissions if none of the floods died prematurely. In reality, we saw 857 transmission, 12687 receptions, 2036 collisions and 1142 instances of capture and recovery.

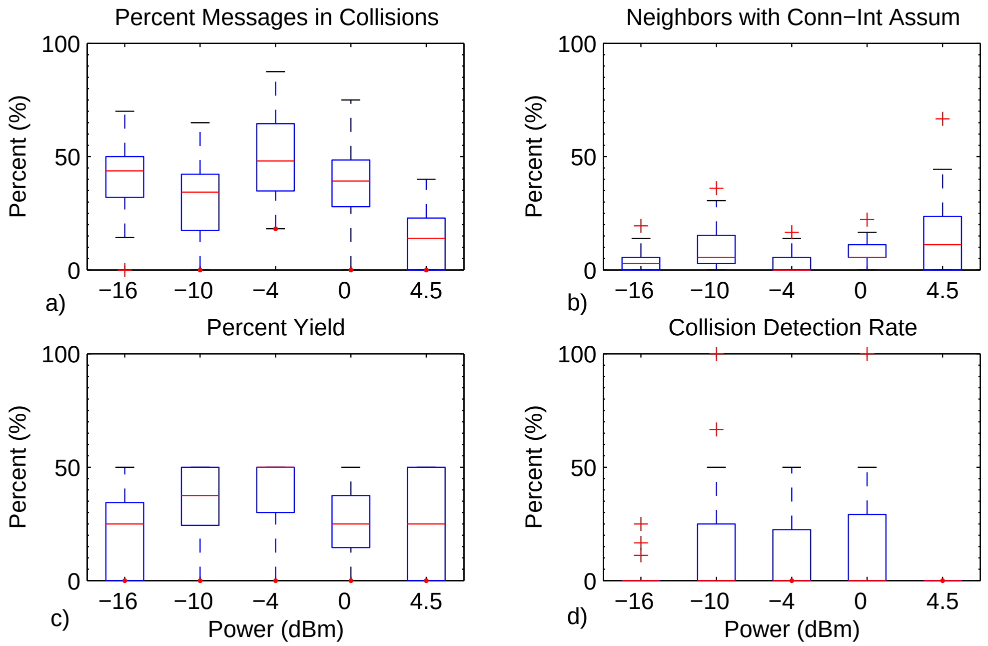


Figure 3: **Broadcast Storm.** Part a) on the upper left corner shows the extent of collision. Part b) shows the percent of neighbors that holds the connectivity-interference assumption. Part c) shows the yield, and part d) shows the rate of detecting collision. The boxes indicate the upper and lower quartiles, with the central dotted bar indicating the median. The whiskers show the extent of the outliers indicated as $+$.

Thus, almost 10% of all received messages were received during a collision.

Figure 3.a shows the percentage of all packets transmit that participated in a collision at at least one receiver. The figure shows that up to 40-50% of all messages transmit in the flooding experiment were involved in at least one collision. This is a very high number caused partially by the fact that, in a flood, each message is destined to so many receivers. It does, however, represents an upper bound on the number of messages that can be lost due to collisions in bursty network traffic. Of course, the number of collisions is a function of the 120ms delay chosen, and experiments have shown that a delay of about 1000ms was enough to allow all messages to be sent without any collisions.

Figure 3.b indicates that the percentage of neighbors a node has for which it can always capture the channel is extremely low, between 0 and 10%. That is to say that, for almost all of node A's connected neighbors, there exists at least one other transmitter B which can overpower A at that receiver.

5.2 Detection and Recovery Frequencies

Figure 3.c shows the percent yield of collision recovery to be between 25% and 50%, where yield is defined to be the ratio of the number of received packets in a collision, either due to stronger-first capture or stronger-second capture and collision recovery, to the number of contenders in a collision:

$$\text{Yield} = \frac{\text{Recovered} + \text{Received Normally}}{\text{Collided}}$$

The value of the denominator is an overestimate because many of the colliding messages may have been lost due to normal link failure. Even so, this data suggests that capture can recover a packet from between 50% and 100% of all collisions. In this scenario the upper bound on yield is actually slightly higher than this because two or three messages can be recovered from a sequence of many colliding messages, which actually did occur on a few occasions.

Figure 3.d shows the detection rate of collision detection to be nominally 20% and up to 50%, where detection is defined to be the ratio of detected or recovered messages to the number of contenders during a collision,

$$\text{Detection} = \frac{\text{Recovered} + \text{Detected}}{\text{Collided}}$$

This value represents the number of collisions that are correctly identified by collision detection.

These last two statistics are much lower than the 92% and 42% we would expect from the data in Section 2. This is partially due to the fact that the probability of Δt is *not* uniformly distributed; many collisions actually occur within the 3ms window $1 < |\Delta t| < 3$ due to MAC backoff. One interesting observation pertaining to power variation is that the number of collisions, percent yield and detection rate all go up as the transmission power is set such that the radio range is greater than the network grid size but less than the entire grid. This is because, at high power settings there are no hidden terminals and the MAC layer can successfully avoid most collisions. At low power settings the radio range is so small that the

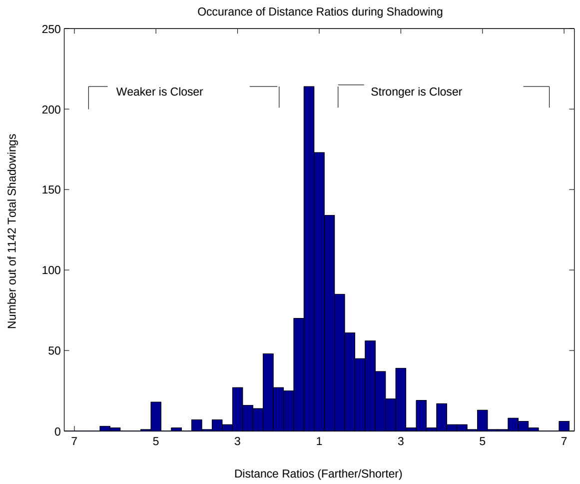


Figure 4: **Distance Ratios of Capture** If a node overpowers another node that is the same distance from the receiver, the distance ratio is 1. If it is 5 times farther, the ratio is 5 and it appears on the left side. If it is 5 times closer, the ratio is 5 and it appears on the right side. The stronger node is almost equally likely to be farther than the weaker node as it is to be closer.

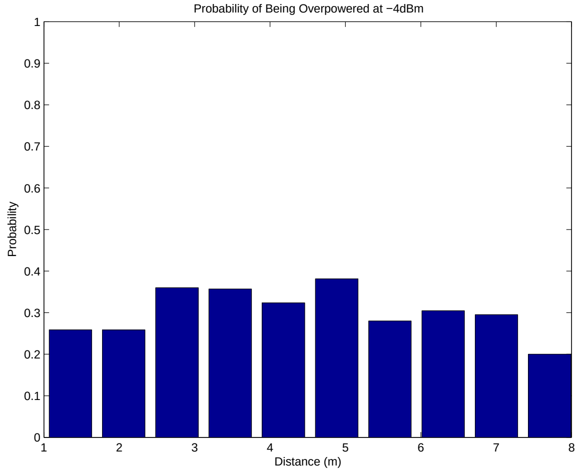


Figure 5: **Probability of Capture Given Distance** The probability of a node being overpowered at a receiver is independent of the distance to that receiver.

number of nodes each transmitter must contend with is small.

6 The Near/Far Effect Indoors

According to the *near/far* effect nearby transmitters are expected to capture a receiver more often that far transmitters [30]. This is due both to increased energy of transmissions from nearby nodes as well as the increased likelihood that a far node will be subject to the hidden terminal problem. This is important for multi-hop sensor networks because of the effective reduction in radio range when the network is under high load. While the theory of the near/far effect is supported by sophisticated models [15], our empirical experiment shows that it is not always true.

Figure 4 is a histogram of the ratios of distances between the two transmitters and the receiver in the 1142 instances of capture observed in this experiment. When the weaker node was the closer node, the ratio was added to the left side of the histogram. When the stronger node was the closer node, the ratio was added to the right side (the graph was plotted this way to avoid ratios with values less than one). One can see that it is almost equally likely for the closer node to be the stronger node as it is for it to be the weaker. The closer node is stronger only 57% of the time while it is weaker 43% of the time.

Similarly, Figure 5 indicates that the probability of a transmitter being overpowered given the receiver's distance is nearly uniform. These results suggest that the capture effect is completely uncorrelated with distance in our indoor experimental setting. This is likely due to multi-path reflections, make RSS uncorrelated with distance indoors, as shown by several empirical studies [27].

7 Impact of Collision Detection

The technique presented in this paper provides information that is not provided by other collision detection schemes. This information can be valuable for both new and existing collision mediation schemes.

In TDMA schemes, such as FPS [4], the receiver or parent is often used to schedule transmissions. A parent may advertise which slots it has available to receive data and all transmitters that want a particular slot contend to reserve it. The transmitter that receives an acknowledgement wins the slot and all others must contend again, possibly waiting for the next cycle of slots and advertisements. One of the main challenges with this scheme is to minimize the time it takes for a transmitter to associate with a parent. If the parent could recover the ID of the losing node, it could send an acknowledgement to the node that won a particular slot and a second message to the node that lost to indicate that it use a different available slot. In this way, even the losing node would be able to associate in one try. Once a transmitter associates with a parent it can still regularly lose packets due to collisions from nearby cells and may be forced to try using a different slot. Using the scheme presented in this paper, the parent can differentiate between such collisions and normal packet loss. It could send an ACK to the transmitter when the packet is received sucessfully, a NACK when the packet collides and nothing if no packet is received. If the transmitter receives many NACKs, it knows that a nearby cell is interfering and it must try a different slot. If it receives neither ACKs nor NACKs, it can infer that it has an unreliable link with this parent and no slot will work. It must try to associate with a different parent. In this algorithm, differentiating between collisions and packet loss can drastically improve the actions of nodes using this protocol.

Cricket is a localization scheme that uses ultrasonic ranging. Because collisions between ultrasonic pulses can not be differentiated, Cricket proposed a collision detection scheme in which each ultrasonic pulse is enveloped by an RF pulse [23]. If the radio message is received, the ultrasonic information is used. If it is corrupted, a collision of both the RF and ultrasonic pulses is assumed and all ultrasonic information is discarded. This technique is widely used for acoustic and ultrasonic ranging [27, 29]. However, it implicitly relies on the "collision as failure" assumption discussed in Section 1 and does not work with radio architectures that exhibit capture. Systems that use such architectures will observe higher error rates with ultrasonic ranging due to collisions. To avoid this problem, one ranging system used a special application-level collision detection scheme in which ultrasonic pulses are sent in *batches* of 10, with random intervals between each pulse [16]. Because of the random intervals, the probability that every pulse in a batch collides with every pulse of another batch exponentially decreases as p^{10}, where p is the probability of a single collision using CSMA. If a receiver identifies three pulses in sequence but not in the same batch, it detects a *batch collision* and would discard all ranging information from both batches. This technique is successful at allowing the receiver to detect collisions despite capture. However, the techniques presented in this paper would solve the same problem at the link-layer without affecting application logic or network traffic patterns. Furthermore, it would often provide the header fields of the nodes that caused the collision. If a ranging message collides with a message from a node that is farther than the ultrasonic range, it need not discard the ultrasonic information from the closer node.

With this technique, collision detection information is provided to the receiver, where it is required in both of the applications described above. The ability to 1) differentiate between packet loss and collisions and 2) to identify nodes caused the collision improves both applications at no extra cost. This information could be used for many other existing protocols as well as open doors for completely new collision mediation schemes.

8 Conclusion

Our study demonstrates and evaluates a technique for recovering packets from collisions and, more importantly, detecting collisions. This detection scheme provides important information at almost no extra cost that can be valuable for several collision mediation schemes in sensor networks.

The collision detection and recovery techniques presented in this paper are not only applicable to CSMA. Collision avoidance schemes such as TDMA and RTS/CTS cannot avoid all types of collisions [5] and can be augmented with collision detection and recovery using capture. Furthermore, many of these schemes are *conservative* with respect to collisions, ie. they try to avoid all collisions possible, and the effort to avoid collisions often comes at a significant cost. For example, Acknowledgments, TDMA, and RTS/CTS cannot be applied to asymmetric links or broadcast-based communication, both of which are very common in sensor networks. Therefore, many sensor network applications use only CSMA, which is highly susceptible to collisions. The techniques presented in this paper allow the programmer to more easily embrace collisions when collision avoidance is not feasible.

While our study is based on one particular FSK radio, the literature has abundant evidence to support the feasibility of this technique across a wide variety of other radio architectures. Many radios such as 802.11 and bluetooth do not expose access of the baseband controller to the application to allow continual searching for packet preambles and resynchronization to a new, stronger packet. We feel that this functionality will become critical for low-power radios to enable new and developing collision mediation schemes in sensor networks.

Acknowledgments

This work is funded in part by the National Defense Science and Engineering Graduate (NDSEG) Fellowship, the UC Berkeley Graduate Opportunity Fellowship (GOF), the DARPA NEST contract F33615-01-C-1895, and Intel Research.

References

[1] Chipcon. `http://www.chipcon.com/files/CC1000_Data_Sheet_2_1.pdf`.

[2] ALOHA Packet Systems with and without Slots and Capture. *ACM SIGCOMM, Computer Communication Review*, 1975.

[3] Darrell Ash. A comparison between OOK/ASK and FSK modulation techniques for radio links. Technical report, RF Monolithics, Inc, 1992.

[4] Barbara Hohlt and Lance Doherty and Eric Brewer. Flexible Power Scheduling for Sensor Networks. *Third International Symposium on Information Processing in Sensor Networks*, April 2004.

[5] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. Macaw: a media access protocol for wireless lan's. In *Proceedings of the conference on Communications architectures, protocols and applications*, pages 212–225. ACM Press, 1994.

[6] Bhaskar Ramamurthi and Adel A.M. Saleh and David J. Goodman. Perfect-Capture ALOHA for Local Radio Communications. *IEEE Journal on Selected Areas of Communications*, 5, June 1987.

[7] C. Namislo. Analysis of MObile Radio Slotted ALOHA Networks. *IEEE Journal on Selected Areas of Communications*, 2, July 1984.

[8] C. Ware and T. Wysocki and J.F. Chicharo . Modelling Capture Behaviour in 802.11 Modems. *IEEE International Conference on Telecommunications (ICT)*, June 2001.

[9] Carlos de Morais Cordeiro and Djamel Sadok and Dharma P Agrawal. Modeling and Evaluation of Bluetooth MAC Protocol. *Computer Communications and Networks*, 2001.

[10] CC Chow and VCM Leung. Performance of IEEE 802.11 Medium Access Control Protocol over a Wireless Local Area Newtork With Ditributed Radio Bridges. *Proceedings of WCNC*, 2000.

[11] Flaminio Borgonovo and Luigi Fratta and Michele Zorzi and Anthony Acampora. Capture Division Packet Access: a New Cellular Access Architecture for Future PCNs. *IEEE Communications Magazine*, September 1996.

[12] I. Bruyland. The Influence of Finite Bandwidth on the Capture Effect in FM Demodulators. *IEEE Transactions on Communications*, 26, June 1978.

[13] James Ward and R.T. Compton. Improving the Performance of a Slotted ALOHA Packet Radio Network with an Adaptive Array. *IEEE Transactions on Communications*, 40, February 1992.

[14] JH Kim and JK Lee. Capture Effects of Wireless CSMA/CA protocols in Rayleigh and Shadow Fading Channels. *IEEE Transactions on Vehicular Technology*, 48, July 1999.

[15] J.P.M.G. Linnartz and R Kekmat and R.J Venema. Near-far Effects in Land Mobile Random Access Networks With Narrow-band Rayleigh Fading Channels. *IEEE Transactions on Vehicular Technology*, 41, February 1992.

[16] Kamin Whitehouse and Fred Jiang and Alec Woo and Chris Karlof and David Culler. Sensor Field Localization: a Deployment and Empirical Analysis. *UCB TR UCB//CSD-04-1349*, April 2004.

[17] Kouichi Mutsuura and Hiromi Okada and Kazuhiro Ohtsuki and Yoshikazu Tezuka. A New Control Scheme With Capture Effect. *International Conference on Communications*, June 1989.

[18] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part I - carrier sense multiple access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23:1400–1416, December 1975.

[19] K. Leentvaar and J. Flint. The Capture Effect in FM Receivers. *IEEE Transactions on Communications*, 24(5):531–539, 1976.

[20] P. Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, and David Culler. Tinyos: An operating system for wireless sensor networks. In W. Weber, J. Rabaey, and E. Aarts, editors, *Ambient Intelligence*. Springer-verlag, 2004.

[21] M. Zorzi and F. Borgonovo . Performance of Capture-division Packet Access with slow shadowing and power control. *IEEE Transactions on Vehicular Technology*, 46, August 1997.

[22] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162. ACM Press, 1999.

[23] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM Press, 2000.

[24] R.D. Stuart and L.B Arguimbau and E.E. Manna and G.M. Rodgers. The Use of Frequency Modulation for Television Transmission. Technical Report RLE TR-259, MIT Research Laboratory of Electronics, 1953.

[25] Sameer Sundresh Wooyoung Kim and Gul Agha. SENS: A Sensor, Environment and Network Simulator. *Annual Simulation Simposium*, April 2004.

[26] P. S. Sang. On capture effect of FM demodulators. *M.S. Thesis Naval Postgraduate School, Monterey, CA.*, March 1989.

[27] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM Press, 2001.

[28] Wei Luo and Anthony Ephremides. Power Levels and Packet Lengths in Random Multiple Access. *IEEE Transactions on Information Theory*, 48, January 2000.

[29] Kamin Whitehouse and David Culler. Calibration as Parameter Estimation in Sensor Networks. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, USA, September 2002.

[30] Zoran Hadzi-Velkov and Boris Spasenovski. Capture Effect in IEEE 802.11 Basic Service Area Under Influence of Rayleigh Fading and Near/Far Effect. *IEEE Internation Symposium on Personal Indoor Communication*, 2002.