Collision Tolerance: Improving Channel Utilization with Correlatable Symbol Sequences in Wireless Networks

Jumin Zhao, Shaolei Fan, Deng-ao Li, and Baofeng Zhao
Taiyuan University of Technology, Shanxi 030024, China
Correspondence should be addressed to Jumin Zhao; zhaojumin@126.com
Received 17 May 2014; Accepted 1 September 2014
Academic Editor: Nianbo Liu
Copyright © 2015 Jumin Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Packet corruption caused by collision is a critical problem that damages the performance of wireless networks. Traditional medium access control (MAC) protocols resort to collision avoidance to improve the efficiency of channel utilization. According to our investigation, however, collision avoidance causes time delay, which also hurts channel utilization, especially in the dense networks or intensive traffic. Discovering the ability to tolerate collisions with the correlatable preamble at the physical layer implementations of wireless networks, we propose CWM in this paper, a MAC protocol that advocates simultaneous accesses from multiple senders to a shared channel, allowing collisions instead of avoiding them. With the correlatable sequences, CWM addresses the challenges in achieving collision tolerance, such as precise sender alignment and control of the transmission concurrency. We implement the experiments on USRP verifying the feasibility of our protocol and implement CWM in 802.15.4 networks and evaluate its performance with 100 nodes on MATLAB. The results demonstrate that our protocol significantly improves the spatial reuse of the wireless channel by at least 23% compared with the state-of-the-art protocols.

1. Introduction

Wireless networks suffer from collisions seriously; this is due to the communication property of broadcast in wireless network. Multiple transmitters communicating simultaneously in a common channel are likely to interfere with each other; as a result the transmitted packets are corrupted. If the collisions are not handled appropriately, the performance of channel utilization is degraded. Hence how to resolve collisions is a crucial issue in wireless network.

In order to tackle the problem of collisions and improve the space utilization in wireless network, most researchers advocate a mechanism to avoid collision. In other words, there should be one sender transmitting in a common channel in any slot. Many protocols based on collision avoidance [1–3] for medium access control (MAC) have been proposed in the past years. Their common principle is to assign the transmissions in different slots or different channels to limit the chance of collisions. These mechanisms usually come at the cost of sacrificing the efficiency of channel utilization. Those mechanisms can be roughly classified into two categories: contention-based and schedule-based. Schedule-based MAC protocols incur nonnegligible overhead in coordination and synchronization to make schedules. In CSMA, a typical protocol of contention-based mechanism, senders always use the binary exponential backoff algorithm to choose the backoff window. In this way, because of uncertain contention, a lot of idle slots are left unused in the channel. This problem is further aggravated in the scenarios with intensive network traffic.

We can draw from these facts that there is a conflict between collision avoidance and channel utilization. If we can make full use of the collisions in the channel instead of simply avoiding them, the channel utilization will be significantly improved. Following that idea, we make some investigations and attempt to modify the medium access control mechanism. Reference [4] plots the comparison between collision avoidance and collision tolerance; the results show that the utilization is enhanced by 71.4%.

In this paper, the feasibility of collision tolerance comes from the fact that the preamble in the head of the packets is robust to tolerate collision. With the property of correlatable preamble sequence, we propose a collision tolerance protocol, which greatly improves channel utilization. Several
challenges by far exist to realize collision tolerance with corre-
latable preamble sequence. First, collision tolerance requires
the corretable sequences being robust. At the same time, the
corretable sequences must not bring high overhead. Second,
the number of corretable sequences must be enough to meet
the concurrency senders.

In order to address the above challenges, in this paper we
propose CWM, a protocol for medium access control. Using
CWM, concurrent senders select unique corretable symbol
sequence corresponding to their unique ID to optimistically
content with each other to access the share medium. In
this way, CWM limits the concurrency and improves the
channel utilization. The main contributions of this paper can
be summarized as follows.

First, we devise CSS for the preambles of packets, which
are then used to identify different senders. In order to enable
efficient identification of CSS in noisy environments with low
SNR, we design a simple CSS dictionary and use it to convey
the sender’s ID.

Second, in order to support simultaneous identification of
multiple CSS and in turn the simultaneous channel allocation
to multiple senders, we formulate the CSS assignment to
networked sensor nodes into a graph coloring problem, for
which we propose a lightweight greedy algorithm to resolve it.

Third, we implement CWM on MATLAB with 802.15.4
standard. Our performance evaluation results demonstrate
that CWM enhances channel utilization by at least 25%
compared with the state-of-the-art protocols.

The rest of the paper is organized as follows. Section 2
presents brief discussion on the background of CSS and the
motivation of this work. We elaborate on the design of CWM
in Section 3. Section 4 presents the performance evaluation
results as well as the comparison between our approach
and the state-of-the-art approaches. Section 5 discusses the
related work. We conclude the paper in Section 6.

2. Background

2.1. Correlatable Symbol Sequence (CSS). A corretable
sequences \( x(n) \) is detected via cross-correlation with local
copy; at the reception a dictionary complex signal \( y(n) \) that
may contain \( x(n) \), \( y(n) \) is cross-correlated with the complex
conjugate of the target CSS \( x^*(n) \). Formally, for a CSS of
length \( L \),

\[
R(\tau) = \sum_{i=0}^{L-1} x^*(n) y(n),
\]

where \( \tau \) represents the position of the correlation with respect
to the input signal, that is, the sample for which we perform
the correlation. Note that (i) if \( [y(\tau) \cdots y(\tau+L-1)] \) does
not contain exactly \( x(n) \), the value of \( R(\tau) \) is close to 0; (ii) if
\( [y(\tau) \cdots y(\tau+L-1)] \) contains a copy of \( x(n) \) with sufficient
SNR, the \( R(\tau) \) obtains a large value that is proportional to the
energy of the signal.

We use cross-correlation as a test statistic for the detection
of a target CSS and repeat its computation at each new sample
of the incoming signal. Specifically, detection is performed
by setting a threshold \( T \); if \( R(\tau) \geq T \) (resp., \( R(\tau) \leq T \), the
presence (resp., absence) of the CSS in \( y(\tau+L) \) is declared. The
performance of cross-correlation can be quantified in terms of
false positives and false negatives. Specifically, after decid-
ing on a threshold \( T \), and a false negative is declared when
\( R(\tau) < T \) even though the CSS is present within \( [y(\tau) \cdots y(\tau+L-1)] \), and a false negative is declared when \( R(\tau) \geq T \)
even though the CSS is absent. Cross-correlation detection
provides a processing gain, which is linear in the length of the
related sequence \([y(\tau) \cdots y(\tau+L-1)]\). This means that a sequence of length
\( 2L \) obtains a value of \( R(\tau) \) twice as large a sequence of length
\( L \); that is, it is considerably more likely to exceed \( T \).

3. Protocol Design

CWM is a protocol for medium access control that coordi-
nates the behavior of a receiver and multiple senders. Sup-
ported by the principle of collision tolerance, the basic idea
and key feature of CWM is aligning the packet transmissions
from multiple concurrent senders, so that the channel is fully
utilized. This section presents the design of CWM. We start
with a brief overview of the workflow and then explain the
details of the protocol design. We also analyze the protocol’s
performance and discuss its limitations.

3.1. Overview. Collision in wireless network is serious and
the channel utilization is very low. CWM attempts to utilize
corretable symbol sequences explicitly to contend for the
media access, which improve the channel utilization. Upon
detecting data packets collisions, CWM is activated immedi-
ately and dynamically adjusts the network channel utilization
which increases network capacity by quickly receiving data
from several neighbors one by one like a queue and does not
produce any overhead in other conditions.

Figure 1 depicts the workflow of CWM with an example
of three senders and one receiver. We assume that each
sender has one packet to send for simplicity. The whole
workflow is divided into three periods. In period 1, three
senders send packets to the receiver almost at the same time; as a result the packets on the receiver collided as S1. Detecting the corruption, the receiver realizes the channel is overly crowded and decides to reduce the transmission concurrency. In period 2, the receiver then broadcasts the collision packets as R1 which contains the matching relation between neighbors’ ID and the sequences in the dictionary. The potential senders transmit the sequences mapping their ID to the receiver as S2. The receiver utilizing correlation detection detects all the contenders and allocates different slots to each contender. The decision packets are broadcasted as process R2. With the decision packets, in period 3, S3, S2, and S4 transmit their packets successfully one after another. We explain this protocol in more detail in the following part.

3.2. The Choice of Correlatable Sequences in Dictionary. Correlatable symbol sequences are predefined as pseudonoise binary codewords in a dictionary; namely, while codewords are deterministically generated, they retain the statistical properties of a sampled white noise. For this reason, the cross-correlation of any such sequence with a matching copy obtains spike values, while it appears random to a listener without prior knowledge of the codeword. An example of a correlatable symbol sequences is the 802.11 preamble used for symbol synchronization.

In this paper, CWM uses some correlatable symbol sequences to form a dictionary; these correlatable symbol sequences are orthogonal among each other. Every sender randomly selects a pattern preamble to contend to for the medium. Receiver use codewords correlation to detect each ID, which is corresponding to the sequence. In order to reduce the probability of false positive and false negative, the correlatable symbol sequence have the following properties:

(i) resilience to noise: the sequence must have a strong autocorrelation property; namely, it should produce the best correlation output only when correlating with itself;

(ii) the cross-correlation between two different sequences must be very low as far as possible;

(iii) the number of different sequences must be large enough in order to reduce the probability of some nodes choosing the same sequence.

A correlatable sequence $x(n)$ is detected via cross-correlation with local codewords in the dictionary at the reception; if the dictionary contains the sequence $x(n)$, it outputs a peak magnitude. There are always many nodes in the network and the CSS must be matched uniquely with its ID in one neighborhood, so all the CSS must be reliably detected and discerned. In order to encode the nodes ID and decode the ID at the reception, CWM should construct a unique dictionary between sender and receiver. The most important task is to construct a dictionary in which all the sequences have the above-mentioned properties.

There are several universal synchronization sequences such as Golay codes, PN codes, and ZC codes whose cross-correlation is robust. Different families of sequences provide large sets of codewords and high cross-correlation value between any codeword. We compare the properties among this sequences and draw a conclusion that the Gold sequences, a kind of PN sequences, present strong properties that meet our requirements and thus can be used in our protocol, as Figure 2 shows.

The choice of Gold sequences permits us to support more than 60 nodes by assigning distinct CSSs to each node. But if the number of sequences in the dictionary is too many, delay will be incurred in the detecting process. From the consideration on the other hand, if the number of sequences is too little, the probability of collision will increase the probability of the sequence selected by multiple nodes. We suppose that there are $m$ nodes in a neighborhood, and we did large amount of experiments in the fourth part and then choose $K$ length-$L$ sequences to form the code library, where $m \leq K$.

3.3. Align Senders Packets Algorithm. Considering the dynamics of networks, even there is a method to figure out the total number and identities of the senders, it still means considerable overhead at the receiver to align senders packets. In order to align senders packets, we propose an algorithm to encode ID and detect nodes’ ID.

3.3.1. Encoding ID. In our protocol, all the contenders should select a unique sequence mapped with each ID from the dictionary to contend for the medium when collision occurred. It is the optimal choice to assign unique sequence to all the nodes in the wireless network, but the overhead is quite large in the large-scale network. For example, if we utilize the 127-length sequence in the network of size of $N$, the dictionary size will be set to $N$ and we need a $127 \times N$ bits of memory. The hardware complexity of detection preprocessing at the receiver in our system is $O(N)$; the overhead cannot be acceptable in wireless sensor network. Furthermore, the topologies in sensor wireless network are almost based on cluster and tree, which are based on the communication radius $r$ of nodes. The approximate radius $r$ is used to decide whether the two given nodes are neighbors or not in the networks. So it will be enough to make sure that every node
Input: $V_c = 1, 2, 3, \ldots, N$, maximum clique $G(\bar{V}, \bar{E}) = (V', E')$;  
Output: One color assigned to each node $(1, \bar{c}_1), (2, \bar{c}_2), \ldots, (N, \bar{c}_N)$ in which $\bar{c}_i \in 1, 2, \ldots, k$ and $k$ is the number of colors. 

1. **begin**;
2. Order the nodes as $(n_{i_1}, n_{i_2}, \ldots, n_{i_N})$;
3. for $i = 1$ to $N - 1$ do  
4. $i = 1$;
5. if $\exists j$ assigned to color $i$ at $(j, n_j) \in G(\bar{V}, \bar{E})$ then
6. $i = i + 1$;
7. assign color $i$ to $n_j$.

**Algorithm 1**: Basic operation of decision making.

Within a neighborhood is mapped with unique sequence. Let $m$ denote node degree (the number of nodes in the receiver’s transmission area). The sequences assignment problem for wireless sensor network is stated as follows: a sequence assignment $C(S_i)$, $S_i \in U$ is a mapping $C : U \rightarrow \{1 \cdots m\}$ from the set of all nodes in the network to the sets of sequences; the nodes in the receiver communication area are interfered by each other.

So this problem can be modeled as a graph coloring problem in which all the nodes are the vertices of a graph. The link between two nodes is represented by an edge in the graph. The goal of this graph coloring problem is to assign a set of distinct sequences to the nodes in the neighborhood. To enable the most efficient use of these sequences, the objective of this problem will be coloring the graph with the minimum number of colors, a problem which is known as NP-hard for general graphs (Figure 3).

As Figure 1 shows that S2 is the process of the collision senders using correlatable sequences to contend for the medium, a unique sequence must be assigned to each ID within the neighborhood. The network is represented by a graph $G = (V, E)$, $V$ is the set of nodes, and $N = |V|$ is the number of nodes in $G$. Let $k$ denote the total number of sequences available in the codebook. The $k$ is bounded by $m \leq k \leq N + 1$, where $m$ is maximum clique size (neighbor size) in the graph and the worst case number of colors is $N + 1$. In this protocol the nodes’ neighbor size in the networks is maximum. The maximum clique is represented by the $G'(V', E') = (V', E')$; nodes $i$ and $j$ are assigned different colors if $(i, j) \in G'(V', E')$. Incremental methods appear to be the heuristic choice of vertex coloring [6]: the vertices are colored sequentially with the colors chosen in response to colors already exist in the vertex’s neighbor.

Algorithm 1 is the coloring algorithm. At the beginning of the algorithm, the nodes are ordered according to the decreasing order of degree since high-degree vertices have more color constraints and are more likely to require an additional if inserted late. The algorithm then assigns smallest color to the nodes so that none of nodes of the same color have an edge in the conflict graph. This algorithm assigns a sequence to each node in the neighborhood.

3.3.2. The Detection of Nodes ID. We assume, for now, that the transmitter and the receiver have known the set of Gold sequences, $W_0, W_1, \ldots, W_K$, used to address the nodes’ ID. Multiple Gold sequences sent by the senders will arrive at the receiver at approximately the same time. The receiver discerns the notification amount for a known pattern in an incoming signal. The receiver utilizes the received signal correlated with the sequences in the dictionary one by one; when the signal contains a sequence the dictionary will output a high correlation value as Figure 4 shows. Therefore, by tracking the correlation value for such a spike, the receiver can verify the presence of the node. As Figure 4 shows, this protocol utilizes parallel detection as the processing time in parallel detection is of high efficiency.

However, in fact the correlation values among the different signals from the senders are not always high and the cross-correlation values of different sequences are not always low. The reasons are as follows: (i) although multiple senders transmit signal to the same receiver, the SNR among them is not totally the same; (ii) since the distance between the receiver and different senders varies, the signal power
(1) Detect all the sequences using correlation;  
(2) If the sequences detected are different then go to line (3);  
(3) According to the sequences number from small to large Collide With Me make schedule and broadcast it to all the senders and reset the register;  
(4) Else ordered the different sequences from small to large and load it into register then broadcast the collision sequences to all the senders and then go to line (1).

Algorithm 2: Basic operation of decision making.

However not all the conditions are ideal, when more than one node in a neighborhood selects the same sequence, collisions occurred. How does the receiver cope with this condition? Algorithm 2 summarizes the essence of making decision.

3.4. Discussion. As a MAC layer protocol, CWM can be a good candidate instead of traditional CSMA backoff mechanism for various protocol scenarios, such as data collection, data relay, and data aggregation. When data rate is high and nodes are dense [7], CWM shows its superiority.

However, there are two limitations for CWM. First, CWM is a receiver-initiated protocol [8]. Hence, senders should transmit under the coordination of the receivers. CWM only shows its superiority in the conditions of high contention and high data rates. Backoff mechanism presents the advantages in the scenarios of low data rates, while CWM has poor performance.

4. Evaluation

In this part, we evaluate CWM from two dimensions. First, we conduct experiments to verify the feasibility of CWM. Namely, we evaluate the performance of CSS and the code-book size on USRP. Second, we simulate CWM on MATLAB. We compare the channel utilization with CSMA based on collision avoidance.

We have simulated CWM on MATLAB. In order to obtain lower error code rate, O-QPSK is used in 802.15.4 modulation. In this set of simulations, we evaluate the performance in the single-hop networks. Our results reveal several key findings.

(i) CWM has no overhead when data collisions do not occur; when collisions are resolved CWM is triggered.
(ii) CWM can sustain a range of different traffic loads, quickly reacting to changing conditions as shown in the following.

Based on these results, we argue that CWM is a welcomed design for tolerance collisions. By not imposing any additional overhead in absence of collisions, it allows the MAC protocol to run without unnecessary performance penalties. When collisions occur, CWM quickly intervenes to resolve them efficiently and improve the channel utilization.

4.1. The Performance of CSS. Whether the mechanism is feasible depends very much on the property of correlatable
symbol sequences (CSS). We implement the transmission of CSS only instead of the packets as 802.15.4 frame transmissions on USRP. USRP is a software radio platform which can be defined by computer. On every node, we preload a unique CSS in the outgoing buffer in order to allow multiple transmitters synchronously to transmit sequences.

In this section we present an experimental evaluation of correlatable symbol sequences using MATLAB. Our evaluation covers the following issues.

(i) Exploring the performance of sequences in different length: we try to find the optimal sequence compromised among the sequence of overhead, processing gain, and robustness. Our simulation results show that the sequence of 127 bits provides a good trade-off between overhead (25.55 us) and robustness (the probability of successfully detecting sequence can reach no less than 95% even at −6 dB SNR).

(ii) Confirming the size of codebook.

The experiments in this part need to be designed from two aspects as follows.

(i) Evaluating the time to detect sequences, when multiple sequences arrive at the same time.

(ii) Confirming the probability of successful detecting sequence at different SNR.

4.1.1. Codebook Size. In CWM, multiple CSSs nodes used by nodes need to be able to be detected and reliably discerned. In this experiment, we explore the number of distinct CSSs that can be practically used. For a given CSS length, a trade-off exists between the number of CSSs that CWM uses and the senders number in turn influences the probability of false positive. If the number of senders is more than the number of CSSs in the codebook each sender will be matched with a unique sequence. In this situation, the false positive by the conflict that more than one sender selects the same sequence can be avoided, but large codebook size brings large overhead. If the codebook size is as large as to assign a unique sequence to the senders in a neighborhood, the protocol is feasible. As the graph coloring in Section 3, the codebook size \( k \) must meet the condition \( m \leq K \).

To confirm the codebook size, we investigated the GreenOrbs system [9] in which there are 1200 nodes and collected the data in one month. We randomly select four sets in one month; statistics suggest that, as Figure 6 shows, the neighbor size is between 2 and 10, the general neighbor size is about 9, and the maximal neighbor size is only 10. So the codebook size larger than 10 can ensure one sender with a unique sequence if possible.

4.1.2. CCS Length versus Robustness Trade-Off. The next task in our experiments is to find the trade-off between the CSS length and robustness under different SINR. We quantify robustness in terms of the number of CSS overlying which still can be detected. In this paper, we define the probability of correctly detecting the sequence at different SNR as the stand of robustness. Three aspects should be considered to confirm the length of sequences: robustness, the number of correlation sequences, and the time to detect sequences. The number of sequences does not have to be considered here because the sequences of 64 bits are enough over maximum node degree.

Setting In this part, we evaluate the performance on USRP. We simulate a network with a single receiver and two senders. Node a is regarded as a receiver, while node b and node c act as senders. When node b transmits different lengths of Gold sequences to node a, the received power of node a is maintained fixed to −82 dBm. In order to confirm the channel SINR, node c sends packets to node a as an interference source. In these group experiments, we repeat the experiment for different combinations of SINR (−6, −3, 0, 3, and 6) and CSS lengths (63, 127, 255, and 511).

Experimental Results. Figure 7 shows that the robustness of different CSS lengths at different SINR varies from −6 dB to 6 dB as a function of SINR and CSS length. Specifically, the \( x \)-axis denotes the SINR on link b-a, while the \( y \)-axis denotes the probability of false negatives. The different colors of curves correspond to different CSS lengths. The figure shows that longer CSSs are more robust due to the number
of collided CSSs being detected; for example, at −6 dB, CSSs of 63 can be detected by only 10% (90% of false negatives in the figure), while the CSSs of lengths 127, 255, and 511 can be detected by approximately 95%, 98%, and 99%. However, increasing the CSS length involves an overhead penalty, while a 63-symbol CSS can be delivered in about 3.15 μs; 127-, 255-, and 511-symbol CSSs require 6.35 μs, 12.75 μs, and 25.55 μs, respectively. On the other hand, the delay at the receiver is the processing time to find the CSS. In order to save time, CWM realizes continuous parallel detection. With sufficient FPGA resources for full parallelism, the search can be completed in (3/2)L clock cycles. Increasing the CSS length involves an overhead at the reception.

As an example, with the 166 MHz FPGA, while a 127-symbol CSS can be detected in approximately 1.147 μs, 255-symbol CSS and 511-symbol CSS require 2.304 μs and 4.617 μs, respectively. We conclude that 127-symbol CSSs provide a good compromise between overhead (7.497 μs the sum of delivered time and the detecting time) and resilience as they can be detected by about 95% at −6 dB.

4.1.3. Detection Threshold Correlation. The choice of the detection threshold is strategic in balancing the trade-off between false positives and false negatives. For example as Figure 5 shows the threshold is denoted by the thick crossing line; corresponding to the chosen threshold, the figure shows 0 false positives and 0 false negatives. In general, a higher value of the threshold decreases the probability of false positives at the expense of a large probability of false negatives; vice versa, a lower threshold increases the occurrence of false positives. The optimal threshold for a target detection SINR [5] is provided in Gaussian noise by the theory of correlation. Specifically,

\[ T = \sqrt{\frac{L \cdot \epsilon \cdot N}{2}} \cdot Q^{-1}(P_{FA}), \]

where \( Q \) is the tail probability of the standard normal function. The formula shows that the optimal \( T \) depends on noise power \( N \), CSS power \( \epsilon \), CSS length \( L \), and the target probability of false positives \( P_{FA} \) that we fix to \( 10^{-8} \). We notice that (i) generally, the power of noise (which may be due to interfering transmission) varies in time; thus the detection threshold should also change; (ii) practically, it is difficult to estimate in advance the power of noise and the power of the signal. In order to address the latter problem, [10] establishes a lower bound on the SINR of the signal that we tune \( T \) correspondingly (i.e., \( T = \sqrt{(L \cdot \text{SINR} \cdot \text{N})/2} \) \( \cdot Q^{-1}(P_{FA}) \)). Unfortunately, this solution is not sufficient because of difficulty to estimate the \( N \). In fact, the receiver can only measure the total power of incoming signal, which may or may not contain the CSS; in [3] it replaces \( N \) with the total signal power; in this situation, when the CSS is actually present, this choice has the effect of tuning \( T \) to a higher value than desired.

In our protocol, there may be more than one sequence in the signal, in which condition the SINR and the power of signal in channel will increase intangibly. In order to select the optimal threshold, we do a lot of experiments to verify the formula suitable for our protocol and confirm the parameters related to the threshold. We evaluate the performance on USRP platform. Seven nodes are deployed in these experiments: node a acts as receiver and node b acts as fixed sender, while other nodes act as interference.

First, node b sends the sequences of 127 bits to node a at different SINR from −6 dB to 6 dB, while other nodes keep quiet. In order to observe the probability of accurately detecting the sequences in different conditions, we increase the number of jamming nodes from one to six at different SINR. The results in Figure 8 show that with the way confirming the threshold on formula the probability of successfully detecting sequences can reach 94% even at SINR −6 dB. We conclude that the way to determine the threshold is appropriate for our protocol.

4.2. Overall Performance Evaluation. We compare CWM with CSMA backoff: in this experiment, we implement exponential backoff in CSMA denoted as CSMA on MATLAB. Experiments are conducted in both saturated networks and unsaturated networks with packet lengths of 80 and 100 bytes.
Figure 9: The performance comparison between CSMA and CWM with different lengths.

We measure time for all senders to finish transmission at the receiver side understanding how CWM handles the extreme end of a sudden traffic bust and compute overall channel utilization.

Figure 9 shows the results. There are several findings to be revealed in these experiments. First, CSMA protocols demonstrate performance degradation to improve channel utilization with hidden terminals. Second, the impact of packet length on CSMA and CWM is different.

Figure 10(a) shows the receiver throughput against traffic loads for both our protocol and random backoff in logarithmic scale. At the beginning, there are only a few of transmitters to transmit data, so almost no conflicts exist and the throughput is rising with the transmitter increase. With the increase of the senders the conflict increased and the CSMA has to back off again and again; as a result the goodput declined sharply. However, throughput of CWM remains almost unchanged with the transmitter increased because the receiver can detect each contender collision and arrange a schedule for the contenders to transmit data packets like a queue one by one. While Figure 11(a) shows the throughput of CSMA and CWM in saturation networks, the sender almost has already reached the maximum transmission rate of the network. At the beginning the sender almost has already reached the maximum transmission rate of the network, so the throughput is the maximum; when more contenders join the network conflicts occur. In the protocol of CSMA the senders must back off to send data packets and with more and more contenders the throughput is reduced quickly, while in our protocol with more contenders more collision occurs, so the throughput decreases slowly. We can draw a conclusion that Collide With Me can efficiently resolve network collisions and significantly improve the transmission spatial reuse in unsaturated networks.

Figure 10(b) shows the average time to send a data packet successfully in unsaturated network. At the beginning the average completed time to send a data packet is only 4.76 ms, with the number of contenders being increased the number of data packets increased in a certain period of time; when the number of contenders reached two in these experiments the network throughput reached the maximum in the network. In CSMA, the more data packets being transmitted the more time needed to back off, so the average completion time for sending a packet successfully is increasing. While in our protocol, the average time almost remained 2.693 ms at the same time. While in the saturated network, the more the collisions occur the more the time delay is. The completion time of receiving a packet in both CSMA and CWM increases as the number of contenders being increased; the throughput with payload 100 bytes is 81%. With more contenders joining the network collisions increased; the throughput of CWM with the payload 100 bytes drops to 75% and remains unchanged, while throughput of CWM drops to 70% and remains unchanged. We can draw a conclusion that although the payload affects the network capacity our protocol can quickly react to collisions and improve the network capacity than CSMA.

Figure 12(b) shows the average time to send a data packet successfully in unsaturated network with the payload 80 bytes, which is lower than that with payload 100 bytes. We can also obtain that the delay almost remains unchanged with the number of collisions increase.

5. Related Work

We review several research works that are related to our work in this section.

To tackle collisions and improve the channel utilization, many protocols in medium access control (MAC) layer have been proposed. These mechanisms can be roughly classified into three categories: collision avoidance, collision tolerance, and collision cancellation.

Collision avoidance protocols are the most traditional mechanisms. These protocols can be divided into contention-based and schedule-based. TDMA and FDMA are the representative protocols of schedule-based mechanisms. Transmitters are allocated with different time slots or frequencies to transmit data packets. Many researchers have also proposed many schedule-based MAC, such as B-MAC [11] and T-MAC [12]. This approach greatly reduces idle listening time, but required synchronization introduces extra overhead and complexity, and a node may need to wake up multiple times if its neighbors are on different schedules [8]. Considering the advantages of TDMA and CSMA, many hybrid MAC protocols have been proposed; Z-MAC [13], for example, achieves high channel utilization and low-latency under low
contention and like TDMA achieves high channel utilization under high contention and reduces collision among two-hop neighbors at a low cost. Hybrid protocols are not often used in real applications because of the overhead incurred by schedule, as is the same in TDMA. CSMA is a contention-based MAC layer protocol. Senders perform random backoff for collision avoidance. This approach is successful in many scenarios, but when it fails, as in the case of hidden terminals,
the impact on the interfering senders is drastic [14–17]. Hence, it results in many idle slots that cannot be utilized.

Collision tolerance is a different idea from collision avoidance. Collisions are allowed in the transmission. Flash flooding [18], Chorus [19], and Glossy [20] are the typical collision tolerance protocols. These mechanisms have a common limitation is that they can be only applied in flooding or broadcast scenarios.

Recently, some novel collisions cancellation resolution protocols according to some of the properties of the signal are proposed. SIC [21] is the first technique used to cancel collision. The key idea is that the receiver uses measured properties of the signal to recover and then subtract each packet in turn. Just like SIC, Zigzag proposes the same idea based on interference cancellation that exploits asynchrony across successive collisions. These collisions have different interference-free stretches at their start, which Zigzag exploits to bootstrap its decoding [22]. CRMA [23] is a different approach to achieve collision cancellation compared with SIC and Zigzag. It exploits the property of OFDM and independently selects a few channels for transmission. But it requires fairly accurate channel estimation and it is not efficient when the data rate is high. Although the above-mentioned methods solve the collision packets differently, they all have a common feature that they cancel collision packets when collision occurs.

6. Conclusion

Based on the insight of collision tolerance, this paper presents CWM, a novel contention resolution explicitly utilizing collision to handle collisions. This mechanism works on the physical layer and introduces correlatable symbol sequences which provide robustness and efficiency to detect collision contenders. CWM addresses the challenges in tolerating collisions and improving channel utilization. We simulate CWM in 802.15.4 single-hop networks. The results show that CWM drastically outstand in dealing with network conflicts and significantly improves the channel utilization and throughput.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Youth Science Foundation Project of National Natural Science Foundation (61303207); the Ministry of Education in 2012 Colleges and Universities by the Specialized Research Fund for the Doctoral Program of Joint Funding Subject (20121402120020); Shanxi Province Science and Technology Development Project Industrial Parts (20120321024-01); Shanxi International Science and Technology Cooperation Project (2012081031); Science and Technology Activities Project of Study Abroad Returnees in Shanxi Province in 2012 (funded by Shanxi Province Human Resources and Social Security Hall); and Research Project supported by Shanxi Scholarship Council of China (2013-032).

References

[1] S. Sen, R. R. Choudhury, and S. Nelakuditi, “CSMA/CN: carrier sense multiple access with collision notification,” in Proceedings of the 16th Annual Conference on Mobile Computing and Networking (MobiCom ‘10), pp. 25–36, Chicago, Ill, USA, September 2010.

[2] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," in Proceedings of the 7th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC ’96), pp. 392–396, Taipei, Taiwan, October 1996.

[3] E. Magistretti, O. Gurewitz, and E. W. Knightly, “802.11e: collision avoidance without control messages,” in Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom ’12), pp. 65–76, ACM, Istanbul, Turkey, August 2012.

[4] X. Ji, Y. He, J. Wang, K. Wu, K. Yi, and Y. Liu, "Voice over the dins: improve wireless channel utilization with collision tolerance," in Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP ’13), October 2013.

[5] D. C. Chu, "Polyphase, Codes with good periodic correlation properties," IEEE Transactions on Information Theory, vol. 18, no. 4, pp. 531–532, 1972.

[6] R. Ramaswami and K. K. Parhi, “Distributed scheduling of broadcasts in a radio network,” in Proceedings of the 8th Annual Joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging, IEEE (INFOCOM ’89), 1989, pp. 497–504.

[7] K. Jamieson, H. Balakrishnan, and Y. C. Tay, "Sift: a mac protocol for eventdriven wireless sensor networks," Tech. Rep., 2003.

[8] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys ’08), pp. 1–14, November 2008.

[9] Y. Liu, Y. He, M. Li et al., "Does wireless sensor network scale? A measurement study on GreenOrbs," in Proceedings of the IEEE INFOCOM, pp. 873–881, Shanghai, China, April 2011.

[10] W. Ye, J. S. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in Proceedings of the IEEE Computer and Communications Societies (Infocom ’02), pp. 1567–1576, June 2002.

[11] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys ’03), pp. 171–180, November 2003.

[12] D. Halperin, T. Anderson, and D. Wetherall, “Taking the sting out of carrier sense: Interference cancellation for wireless LANs,” in Proceedings of the 14th ACM Annual International Conference on Mobile Computing and Networking (MobiCom ’08), pp. 339–350, September 2008.

[13] S. Kim, R. Fonseca, P. Dutta et al., “Flush, A reliable bulk transport protocol for multihop wireless networks,” in Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys ’07), Sydney, Australia, November 2007.
[14] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks,” in Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’10), San Jose, Calif, USA, April 2004.

[15] S. Khurana, A. Kahol, and A. P. Jayasumana, “Effect of hidden terminals on the performance of IEEE 802.11 MAC protocol,” in Proceedings of the 23rd Annual Conference on Local Computer Networks (LCN ’98), pp. 12–20, Lowell, Mass, USA, October 1998.

[16] G. Judd and P. Steenkiste, “Using emulation to understand and improve wireless networks and applications,” in Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI ’05), pp. 203–216, 2005.

[17] C. Ware, J. Judge, J. Chicharo, and E. Dutkiewicz, ”Unfairness and capture behaviour in 802.11 adhoc networks,” in Proceedings of the IEEE International Conference on Communications (ICC ’00), vol. 1, pp. 159–163, IEEE, New Orleans, La, USA, June 2000.

[18] J. Lu and K. Whitehouse, ”Flash flooding: exploiting the capture effect for rapid flooding in wireless sensor networks,” in Proceedings of the IEEE INFOCOM, pp. 2491–2499, Rio de Janeiro, Brazil, April 2009.

[19] X. Zhang and K. G. Shin, ”Chorus: collision resolution for efficient wireless broadcast,” in Proceedings of the IEEE International Conference on Computer Communications (INFOCOM ’10), March 2010.

[20] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, ”Efficient network flooding and time synchronization with Glossy,” in Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN ’11), pp. 73–84, April 2011.

[21] D. Halperin, T. Anderson, and D. Wetherall, ”Taking the sting out of carrier sense: interference cancellation for wireless LANs,” in Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom ’08), pp. 339–350, San Francisco, Calif, USA, September 2008.

[22] S. Gollakota and D. Katabi, ”Zigzag decoding: combating hidden terminals in wireless networks,” in Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM ’08), pp. 159–170, August 2008.

[23] T. Li, M. K. Han, A. Bhartia et al., ”CRMA: collision-resistant multiple access,” in Proceedings of the 17th Annual International Conference on Mobile Computing and Networking—MobiCom ’11, pp. 61–72, ACM, New York, NY, USA, 2011.
Submit your manuscripts at http://www.hindawi.com