ICNS 2023: Your contribution 10021 is accepted

"IARIA Confirmation" <confirmation@iariasubmit.org> January 23, 2023 2:32 AM

To: vahid.salmani@sharif.edu

CC: confirmation.log@iariasubmit.org

Dear Vahid Salmani,

Congratulations! Your contribution 10021 to ICNS 2023 titled "Making Most of Packet Collisions towards More Efficient Contention Resolution" is accepted as 1. regular paper [in the proceedings, digital library] / academic research.

Note that for 2023, all events will be held in a hybrid mode with five options: on site or virtual choices (live, prerecorded videos, voiced presentation slides, and .pdf slides).

We hope for better times allowing us to return to the traditional on site scientific events. However, we are ready to adapt any which way the conditions dictate.

There are some fixable issues that must be addressed before the article is ready for publication:

- Scientific and technical;

- English & punctuation, and

- Sections and presentation flow.

To proceed further, please register the paper and pay the registration. IARIA will allocate technical and linguistic resources to assist with the fixing process. The comments received from the reviewers will be the starting point for the interaction with the IARIA representative.

The final revision and fixing process will start immediately after the registration transaction is successfully processed.

Upon receipt of this notification, please proceed to complete your registration:

- go to the "Registration form" on the conference site:
https://www.iaria.org/conferences2023/RegistrationICNS23.html

- fill it out with all required information including the payment information

- submit it

Once the registration is successfully processed, somebody from IARIA will contact you to finalize the camera-ready based on reviewers' comments.

--------- Preparing the camera-ready process ----

1. Registration processed and IARIA will contact you about your article

2. Work on the first round of comments sent after registration (additional rounds will come). Please consider the enforced action for English/punctuation/figures; see: the editorial guide posted at http://www.iaria.org/editorialrules.html

3. Update validation and additional comments.

If you have any questions, contact directly steve@iaria.org

The timeline of this entire process would be:

1. Registration successfully completed by February 5, 2023

2. You acknowledge your availability to work with someone from IARIA to finalize the paper at latest on February 15, 2023

Note that this work may extend beyond the posted camera ready deadline. Arrangements have been made with the publisher in order to accommodate this. Going beyond the official deadline does not jeopardize the publication of the article.

Note that sometimes there are several articles needing assistance, so a reply to your email may be delayed.

On behalf of the organizing committee

Making Most of Packet Collisions towards More Efficient Contention Resolution

Vahid Salmani
School of Computer Science, Institute for Research in Fundamental Sciences (IPM)
P.O.Box 19395-5746, Tehran, Iran
vahid.salmani@sharif.edu

***Abstract*—In this paper, we present an adaptive collision resolution scheme that extracts signal related information from the wireless channel in order to perform slot splitting more intelligently while backtracking to the partially resolved contention to improve efficiency. Contrary to the existing solutions that split the collision domain by a fixed number and into equal-length sub-ranges, we make the splitting decisions adaptively based on the current channel condition. Through extensive simulations, we prove that the proposed scheme outperforms the state of the art, and is well-suited for all contention levels while offering better scalability.**

***Index Terms*—Medium Access Control; Internet of Things; Collision Resolution; Adaptive Protocol; Constructive Interference.**

I. Introduction

One of the practical problems with the Internet of Things (IoT) vision is how to handle dramatic increases in network scale. A critical issue in such environment is the ability of smart devices to communicate adaptively so as to make the best of limited channel bandwidth, and cope with competition which is inevitable as more objects join the network.

WSN constraints such as energy consumption, limited packet length, error-prone wireless channel, and network dynamics impose many challenges in designing communication protocols. More specifically, traffic in WSNs can be quite dynamic and contention variation can be induced by changes in network scale. Moreover, the events being sensed, the sensing application itself, or the protocols being used can also lead to different levels of contention. This demands improved and adaptive communication strategies with WSNs as the main focus of typical IoT applications [1].

MAC protocols coordinate the access to the wireless medium among multiple nodes. The ability to efficiently arbitrate the channel is directly affected by the efficiency of the employed Contention Resolution (CR) scheme. The problem of CR addresses the amount of time until any one of the contenders involved in a collision transmits alone. When we mention collisions, hereafter we mean *"destructive"* collisions throughout the paper unless stated otherwise.

CR techniques fall into the centralized and distributed categories. In the distributed class, the most common approach is backoff-based collision avoidance where contenders transmit only after the medium is found to be clear. Although this approach offers minimal delay when contention is low, it suffers from collisions under high contention, and wasted idle slots due to backoffs under low contention. During bursty traffic, multiple senders have to backoff with an exponentially increasing window, which incurs large delivery latency and energy consumption.

In the centralized class, which is *collision-driven*, the receiver initiates a data transfer by transmitting a *probe* or a Ready To Receive (RTR) packet. If multiple transmissions are triggered in response to the same probe, a collision occurs and the resolution procedure comes into play. As explained in Section II, none of the existing techniques in this category is suited for all contention levels either.

Despite the above efforts, how to handle various levels of contention among low power wireless devices within a single channel is still an open problem. In this paper, we propose an adaptive technique called SLSRQ which distinguishes multiple contenders in one round using the quantity and length of the colliding signals obtained from the cascading RSSI sequence phenomenon. The proposed approach is a delay-bounded resolution procedure suited for various levels of contention, and thus can be used as a complementary component of current MAC protocols.

II. Related Work

The majority of techniques in the centralized class are based on tree splitting or contention tree [2]. The Query Tree protocol [3] is a deterministic variant of the tree splitting algorithm in which contenders respond to the probe if it contains a binary prefix of their IDs. While this technique is efficient for a large number of contenders and under high contention, it may downgrade due to excessive splitting, and thereby idle slots, when the number of contenders is low.

Another deterministic variant of the tree splitting technique is the Binary Search Tree Collision Resolution (BSTCR) scheme [4] that treats the collision domain as a "range" of node IDs, and resolves a collision by splitting the *probing range* in half. Although effective at handling a massive number of contenders, this technique is not as efficient under lower contention levels, or in case of contenders with close IDs. In such conditions, several splitting is needed to break the probing range down and resolve the collision.

Motivated by BSTCR design, Slot Sharing TDMA (s^2TDMA) [5] uses an interval tree search upon collisions instead of a binary search tree. By splitting the probing range into more than two slots it can offer better performance for a large number of *evenly distributed* contenders. However, s^2TDMA introduces even more overhead and delay under low contention due to unnecessary splitting and idle slots. By using a fixed number of sub-slots, the above work does not address adaptability to the contention level. Finally, no optimal value for the number of sub-slots is presented.

The Most-Informative First-Serve (MIFS) splitting algorithm [2] is a slotted-ALOHA protocol designed for decentralized detection. In addition to channel feedback, MIFS uses the data *informativeness* as the splitting criterion, ensuring that more informative data will be collected first. Although this approach yields optimal performance by adaptively performing splitting, it is tied to a specific application. Moreover, the resolution process continues until the amount of data needed by the fusion center reaches a target performance. As a result, not all contenders involved in a collision are necessarily identified. A *reliability-based* splitting algorithm with similar shortcomings is proposed in [6] in which only nodes with a specified range of reliabilities compete for the channel within each frame.

Strawman [7] handles contention based on the principle of drawing *straws*, and the channel access is granted to the sender of the longest contention packet. Since two or more contenders may share the longest request packet, Strawman resolves collisions in steps while keeping the average lengths of straws as short as possible. Only the colliding winners of the current round are authorized to participate in the next round until a successful transmission occurs. Even though this approach works well with a limited number of contenders, it is time-consuming and inefficient under high contention because at most one sender can be identified in each round. Moreover, the maximum straw length cannot scale with the number of contenders due to the limited payload length (e.g. 110 bytes on CC2420 radio).

Designed to solve the aforementioned problem, STAIRS [8] makes it possible to identify more than one winner per round through *Constructive Interference* (CI) [9]. The idea is that RSSI of the superimposed colliding signals with identical waveform exhibits *stairs-like* or *cascading* phenomenon with a different number of contenders [10]. The number of potential senders can be obtained by counting the number of *falling edges*, and their identity can be revealed by the lengths of their transmitted packets. In response to a probe, the simultaneous senders contend for the medium by sending a random length packet on the granularity of 10 bytes, yet with identical payload. The receiver then sends schedule packets including one estimated packet length for each detected edge. Any sender whose length equals to the length carried in the schedule packet wins.

Although STAIRS makes it possible to determine the channel access order among all senders in just one arbitration cycle, it fails if two or more senders pick the same length as in Strawman, requiring more cycles to complete the resolution process. Moreover and more importantly, CI suffers from scalability problems as discussed in [11], and also RSSI jumps measurement works only for a limited (maximum of about 20) number of contenders [12]. Finally, since state of the art WSN platforms tend to use packet-based radios in which payload length is limited, length-based resolution techniques inevitably face scalability issues. Furthermore, longer payloads are more susceptible to noise levels and synchronization errors [11], [13].

Contention Detectable MAC (CD-MAC) [14] employs a lightweight CR mechanism which uses the *temporal diversity* of multiple probe acknowledgments to recognize the potential senders. A reservation frame is initiated by the Base Station (BS)'s probe containing the frame size and an auxiliary parameter used by contenders to compute the slot number in which to send their software ACKs (contention packets). Contenders select slot numbers in a double-modulo process to spread out across the frame so as to minimize the probability of collisions. At the completion of the reservation frame, the BS obtains the quantity of the announced contenders, and starts the packet polling stage. If there are any ACK collisions, they will be given another chance in the next round with an adaptive reservation frame size.

Although CD-MAC outperforms Strawman combined with A-MAC, its CR scheme is not a general purpose one because it follows the contention/scheduled alternation policy. As a result, it may not identify all the contenders in one round which introduces extra latency. Moreover, the collision probability quickly grows as the number of contenders goes up (doubles with every 5 nodes added [14]). Finally, it will take at least one round for CD-MAC to adapt to the contention level and adjust the frame size.

In summary, existing schemes cannot meet both of the adaptability to the contention level and scalability requirements at the same time. The downside with the techniques solely relying on examining the colliding signals, and even non-recursive variations of tree-splitting such as Flip-MAC, is that they can suffer considerable performance degradation under high contention. The issue comes from splitting the same collision domain multiple times, which is analogous to solving a problem from scratch as opposed to leveraging the solutions to partially solved problems (i.e. backtracking). Besides, these techniques suffer from scalability problems.

On the other hand, it is known that deterministic algorithms are typically faster than probabilistic schemes [15]. However, the inadequacy of deterministic tree-splitting approaches has to do with taking several steps to come up with fine-grained ID ranges that separate contenders with close IDs apart, which happens under low contention. That is because contrary to the former class, they do not base their slitting decisions on the knowledge extracted from colliding signals to exhibit an adaptive behavior.

As discussed in Section IV, the novelty of our proposed approach compared to the existing solutions is in its deterministic, delay-bounded and adaptive behavior while being general purpose.

III. Network Model and Assumptions

Regardless of the topology, IoT systems must address *localized competition* for wireless medium [1]. In this context, smart devices are typically clustered in *geographically proximal* groups. Particularly, things concentrate at specific locations, for example where end-users reside or work, such as intelligent houses or smart office environments. As a result, we concentrate on a *many-to-one* communication pattern (also known as *convergecast*), which is dominant in clustered WSNs as well as many IoT applications.

To avoid introducing unnecessary complexity to smart devices, most of the burden should be moved to the *coordinator* node (typically a base station or a cluster head), which is more resourceful and a natural point of central coordination. This, together with the centralized control provided by the coordinator, suggests that the *receiver-initiated* class of protocols is a better fit. The resolution process must lead to identifying *all* of the contenders and not just a subset of them. The whole contention needs to be resolved either in one round, or else in multiple *consecutive* rounds (prior to exchanging any data messages).

The system is assumed to contain a *variable* number of low power IoT wireless sensors that communicate with the coordinator directly. Nodes are not aware of their number in the network at a given instant (but the coordinator is to some extent). Every node in the system owns a *unique* ID, which allows the BS to distinguish between different data transmissions and contenders. The network can be composed of thousands of nodes with a potentially *large number* of contenders (hundreds [16] or more) transmitting on a shared communication channel.

From the two main approaches to receiver-side collision detection, namely, preamble detection and carrier-sensing, the former may not detect all types of collisions and it also suffers from scalability problems as discussed in [17]. On the other hand, the *carrier-sensing based* approach has no hard limit on the number of simultaneous transmitters as well as no restriction on the type of collisions, which is why we use it in this work. Moreover, this approach can be implemented using RSSI which has the great advantage of simplicity as it can be extracted by commercial off-the-shelf transceivers.

This work considers a *single-channel* network where all contenders belong in the same collision domain. A “collision domain” is defined as a group of mutually conflicting links where every two nodes interfere with each other pairwise. As such, our study focuses on a *single* collision domain.

IV. Design of SLSRQ

A. Protocol Overview

The more information extracted from colliding signals, the more effective resolution schemes can be devised by designing smarter splitting strategies. In order to distinguish multiple contenders in one round, we use the cascading RSSI sequence phenomenon where every time the response of a node terminates, a drop in RSSI can be detected [10]. In addition to the ability to obtain the number of potential senders by counting the number of falling edges, the lengths of transmitted packets could provide some hints about their senders’ identity.

The proposed *“Signal Length and Strength based Range Query”* (SLSRQ) method represents the collision domain as a range of node IDs. In other words, each probe is in fact a *range query* composed of the minimum and maximum IDs of the range, and any transmitter whose ID falls into this range may contend for the medium. The coordinator must keep the radio on to sample the channel and process the concurrent contention packets. This section provides an implementation-independent description of the proposed mechanism.

Having broadcasted the probe, the BS splits the collision domain (i.e., ID range) based on the feedback received from the channel as described in the following sections. By splitting the probe range into several smaller sub-ranges based on falling edges, we not only speed up the resolution process but also minimize the probability of idle slots (sub-ranges), which is what an optimal splitting strategy should do [2]. In this sense, the proposed approach can be viewed as a complementary component of current receiver-initiated MAC protocols. Furthermore, this work includes several error handling features to cope with uncertainties (Section IV-F).

B. Obtaining Channel Feedback

The proposed protocol operates based on the feedback from the shared wireless channel. Figure 1 shows how the coordinator’s RX module distinguishes among the four possible states; namely, (1) idle, (2) reception, (3) collision only, and (4) detection of colliding packets’ quantity and length using falling edges. Once a probe is broadcasted, the BS listens for incoming packets while measuring RSSI values. In the first condition, the BS performs a CCA to sample the channel energy and check for a valid packet. If no signal is detected on the channel, it indicates an idle slot. Otherwise, an activity is detected on the channel, and thus the BS first examines the RSSI readings to see if two or more falling edges are detected. Edge detection is performed at this point so that the protocol works regardless of whether a packet is successfully received or not. This makes our work more resilient to clock drifts, maximizing

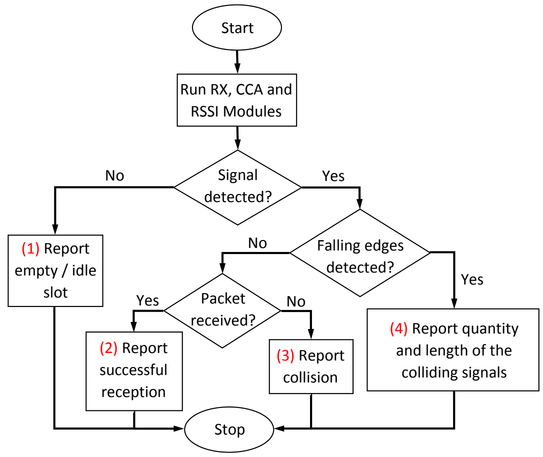

Fig. 1: Flowchart of the RX module depicting how the quaternary feedback is obtained from the RSSI channel

the possibility of correct edge detection. In this case, if two or more falling edges are detected, the length and quantity of the collisional packets are returned. If one or no falling edge is detected, the BS checks if a packet has just been received successfully. If the RX module reports no packet reception, it indicates the presence of a collision. In this case, there are two or more colliding signals on the channel whose length cannot be determined. Thus, a distinctive feature of our work is that it makes use of both constructive and destructive collisions. In case of collisions or falling edges, the Algorithm 2 is invoked.

C. Contention Packet Transmission

This work makes use of CI, that is the ability of certain radio transceivers to decode a concurrently transmitted packet. It is a promising phenomenon, that we believe, has not been fully exploited yet. Even though in CI a packet is received successfully, it is not a requirement in SLSRQ to decode such control packets as described in the previous section.

CR schemes that apply CI cannot contain regular data in contention packets because they use the same reference payload for all nodes. Although all contenders send packets with identical content, they do not have to send the entire payload. In order to use this mechanism, the phase displacement of contention signals must be less than 0.5 μs [9].

Upon receiving a probe from the BS, any node whose ID belongs in the range contained in the probe and wants to contend for the medium sends a contention packet with the predefined reference payload whose length is determined as explained in the next section.

D. Mapping Node IDs to Signal Length

This section deals with the length and the granularity with which contention packets should be transmitted. Throughout the paper the contention packet length granularity is referred to as *delta*. A larger delta minimizes

Algorithm 1 Pseudo-code of SLSRQ's signal length determination for nodes

getPayloadLength(fromID, toID)

1: $myID \leftarrow$ getMyID()
2: $idRange \leftarrow toID - fromID$
3: $myRelativePosition \leftarrow myID - fromID$
4: $delta \leftarrow 10$ {granularity of signal length (bytes)}
5: $maxLength \leftarrow 11$ {max payload length (110) $\div$ delta }
6: $step \leftarrow \min(1, maxLength \div idRange)$
7: $payloadLength \leftarrow \mathrm{round}(myRelativePosition \times step) \times delta$
8: **return** $payloadLength$

length estimation errors and decreases the possibility of false falling edges in practice. LinearPoll, Strawman and STAIRS use a delta of 4 bytes [10], 7 bytes [7] and 10 bytes [8], respectively. To reduce the overhead in computation and diminish the chance of false positives, we set the length of contention signal on the granularity of 10 bytes. Another consideration that helps avoid length estimation errors is the fact that concurrent transmissions are well synchronized [7].

Contrary to the common length-based techniques, this approach is a deterministic algorithm and does not incorporate any random length signaling. Instead, signal length indicates the *relative position* of the sender's ID in the range being queried (i.e. $[fromID,\ toID]$) on the mentioned granularity, and is calculated according to Algorithm 1. In step 6 of the algorithm, we ensure that the granularity of contention packet length is no more than 10 bytes, which may otherwise happen in case of small ID ranges. Step 7 associates the node ID with a packet length based on the probe range. As an example, consider the range query [362, 407] for which nodes 371 and 386 pick the payload lengths of 20 and 60, respectively. The same nodes will use the lengths of 100 and 110 in the range [0, 400] for their contention packet payload, respectively.

As mentioned earlier, length estimation errors can lead to misdetection of the corresponding edges in the worst case. Thanks to our protocol's dynamicity, such errors are handled in the lower levels of the contention tree. As the collision domain is broken down, the contention packets are further spaced apart, eliminating the impact of length miscalculations. It should be noted that current length based techniques (e.g. STAIRS or Strawman) do nothing to recover from such errors. The next section provides a detailed description of the novel dynamic contention tree proposal which is the key to SLSRQ's adaptability.

E. Dynamic Contention Tree

How the collision domain is split has a significant impact on the efficiency of collision resolution [2]. As mentioned in Section IV-B, four cases can happen in response to a BS probe. In case of a successful packet reception, a contender is identified. An idle slot means there is no contention in the corresponding range, and thus no further resolution is needed. Given the range [$fromID$, $toID$], the third input parameter in Algorithm 2 is a list of detected edges, where $edges[i]$ is the length of the i^{th} edge in bytes. If the only information obtained from the channel is the presence of a collision, the *edges* parameter will contain a single zero when the range is split in half according to Algorithm 2 (steps 5 to 7), and a new probe for each sub-range will be issued. This is where destructive collisions are handled. If two or more falling edges are reported by the RX module, the quantity and lengths of the CI signals are used for finer-grained splitting.

SLSRQ does not rely on the exact number of concurrent contenders but mainly uses the changes in the strength of colliding signals to extract the length of each. As such, information of detected falling edges are used to split the probing range according to Algorithm 2 (steps 8 to 16) where constructive collisions are handled. This will semantically reverse the mapping procedure presented in the previous section. Intuitively, if there are more than two contenders involved in a collision, instead of splitting the range in half or a fixed number, we split it by the number of falling edges to speed up the process.

Furthermore, the length of sub-ranges is flexibly determined so that each of them cover the corresponding node(s) of a falling edge. The output of Algorithm 2 (i.e. *splitList*) is a list of *edgeCount* subranges where the i^{th} subrange is [*splitList*[i], *splitList*[i+1]-1]. This process will continue until all collisions are resolved. That way, we conceptually create a *"dynamic contention tree"* where each tree node represents a range of sensor node IDs, and apparently the tree nodes having the smallest possible range will contain a single node ID. The contention tree is traversed in depth-first-search order and a probe is issued for each node encountered.

The maximum number of contenders that can be identified per probe depends on two key factors namely, (1) the maximum number of detectable simultaneous transmitters, and (2) the maximum payload length divided by delta (granularity of contention packet length). In practice, the *minimum* of the above two factors determines at most how many subranges can be generated in response to the same probe.

It is worth noting that the BS splits a range such that the resulting subranges do not overlap, and their union covers the entire range of the original probe. This mechanism provides fault tolerance against undetectable (false-negative) edges. Moreover, it enables new contenders to compete for the medium even in the middle of the resolution process, which is another aspect of dynamicity of this work.

The first step in designing an efficient CR scheme is to quickly reduce contention to a manageable level. Using the proposed structure, the problem can be broken into several smaller sub-problems more quickly than current variants of tree splitting. The main difference between this scheme and tree splitting hereof is that it does not split the probing range by a fixed number. Each tree node can have as many sub-trees as the maximum number of detectable falling edges. Moreover, the length of the sub-slots are not fixed either. On the other hand, leveraging partially resolved contention in a recursive fashion called *backtracking* enables SLSRQ to outperform length-based approaches under high contention. To our best knowledge, this is the first time that such a mechanism as dynamic contention tree is applied to WSNs.

Algorithm 2 Pseudo-code of SLSRQ's slot splitting for the coordinator

splitCollisionalSlot(fromID, toID, edges)

1: $idRange \leftarrow toID - fromID + 1$
2: $edgeCount \leftarrow$ getCount($edges$)
3: $delta \leftarrow 10$ {granularity of signal length (bytes)}
4: $maxLength \leftarrow 11$ {max payload length (110) ÷ delta }
5: **if** $edgeCount < 2$ **then**
6: $\quad splitList[1] \leftarrow fromID$
7: $\quad splitList[2] \leftarrow fromID + (idRange \div 2)$
8: **else**
9: $\quad step \leftarrow 1$
10: $\quad$ **if** $idRange > maxLength$ **then**
11: $\quad\quad step \leftarrow idRange \div maxLength$
12: $\quad$ **end if**
13: $\quad$ **for** $i = 1$ **to** $edgeCount$ **do**
14: $\quad\quad splitList[i] \leftarrow fromID + \lfloor edges[i] \div \text{delta} \times \text{step} \rfloor - (\text{step} \div 2)$
15: $\quad$ **end for**
16: **end if**
17: **return** *splitList*

F. Handling Exceptional Situations

Under the condition of CI, at least two edges should be successfully detected so that the slot is not considered collisional. Any other case than idle or packet reception will be treated as a destructive collision. One such exceptional case happens when there are more concurrent signals than the edge detection threshold [8], [11], [12]. In this case, where STAIRS does not provide a workaround and thus fails, no falling edge is detected, and only a destructive collision can be identified. Similar case happens when concurrent transmissions interfere destructively due to clock drifts (in which case a node replies with a larger delay or earlier than expected) or in the presence of excessive external interference. In such situations, the BS simply splits the collisional slot in half.

There is another uncommon condition in which the slot range is split in half as well. That is when two or more nodes with adjacent or close IDs pick the same signal length because their corresponding probe covers a relatively long range while there is no other contender in that range. In this case, only one edge is detected which would normally (that is, in the absence of a collision) be treated as a successful transmission. In the above two cases in which zero and one falling edges are detected respectively, the only information inferred is the presence of a collision. These situations are handled as in steps 5 to 7 of Algorithm 2. Most of the above cases have been identified during the implementation and testing of the proposed protocol, showing that the employed simulator is precise enough and close to reality.

G. Termination

As mentioned earlier, SLSRQ terminates when every collision in the original ID range is resolved. In other words, the termination condition is that the BS detects an idle slot or receives a successful reception in response to a probe range containing the maximum ID of the original range. Having said that, SLSRQ identifies the contenders one by one until the termination condition is satisfied. Depending on the usecase, it may be stopped at any point as long as the minimum required set of contenders are obtained.

V. Performance Evaluation

The performance of SLSRQ is compared with that of BSTCR and STAIRS as two state of the art centralized CR protocols in the tree-splitting and length-based classes, respectively. The selected schemes are representative of the state of the art in the field of WSNs because the results presented in [8] suggest that STAIRS outperforms Strawman which in turn performs better than many other CR schemes.We chose not to include s^2TDMA in the evaluations (despite the results presented in [5]) because its performance is generally not better than that of BSTCR as mentioned in Section II. For instance, given the two contenders 2 and 18 in the range [1, 20], by splitting into 3 or more sub-ranges the s^2TDMA would require more probes than BSTCR to resolve the contention. Other schemes reviewed in Section II are designed to identify a subset of contenders (and not all of them), and thus they are excluded from our evaluations. CI and the cascading RSSI sequence phenomenon are already shown to be practical on real platforms. As such, the aforementioned aspects have been incorporated in the simulator according to the results obtained in empirical studies.

A. Simulation Setup

The employed simulator is already shown to generate valid results [4], [17], and is close enough to reality to validate the presented ideas. It implements the MAC layer of the Tmote Sky (also known as TelosB) platform which incorporates a TI/Chipcon CC2420 radio transceiver. CC2420 is one of the most popular chips for use on wireless sensor nodes. Moreover, the CI phenomenon used in STAIRS works only with Direct Sequence Spread Spectrum (DSSS)-transceivers, and CC2420 is one such radio transceiver.

Both SLSRQ and STAIRS rely on contenders alignment, and online detection of RSSI change points. It is assumed that concurrent transmissions are perfectly aligned by immediately responding to the BS probe, even though we have modeled synchronization inaccuracies as well.

TABLE I: Simulation Radio Parameters

Parameter	Value	
Channel bit rate	250	kbps
Symbol period time	16	μs
RX/TX switching delay	192	μs
Radio on/off transition time	1.792	ms
Frame length	127	bytes
Maximum payload length	110	bytes
Contention packet length granularity	10	bytes
Maximum number of detectable edges	10	

Moreover, RSSI fluctuations are ignored, assuming perfect falling edge detection unless mentioned otherwise. How to handle RSSI instability for high-accurate edge detection is already a solved problem [8]. Furthermore, CI-based protocols are shown to be able to operates reliably even in challenging environments [18].

The network consists of a BS and a number of IoT nodes placed in a similar distance from the coordinator and communicating with it directly. Note that our proposed CR protocol also supports multihop communication. However, for simplicity and lack of space, a single hop and single sink setting is used as in [5], [17]. The maximum number of detectable edges (simultaneous senders) is considered to be 10, and thus changes in the quantity of more than 10 concurrent senders are not detected unless stated otherwise. Based on the discussion presented in Section IV-D, we set the length of contention signals on the granularity of 10 bytes, which corresponds to a transmission time of 320 μs at 250 kbits/s. To minimize latency as well as synchronization errors, we preload packets in the radio's TX buffer [7]. Table I shows the TI/Chipcon CC2420 radio model parameters used in our simulations. We refer the reader to [17] for more details on the simulator internals.

We simulate a simple MAC that runs iterations of collision resolution one by one to identify all contenders in the network. Each contender tries to transmit one message to the BS. Having successfully delivered its message inferred by an ACK from the BS, the winning node quits the competition. In case of SLSRQ and BSTCR, all contenders can be recursively identified in the same iteration (in the absence of interference). As for STAIRS, a new round is started immediately after completion of the current round until all contenders have successfully transmitted their messages. Since the compared algorithms are based on receiver-side collision detection, it is assumed that the receiver can correctly detect all collisions unless mentioned otherwise. Also, we assume perfect signal length estimation.

The main evaluation criterion is *resolution time* which is the difference between the time at which the BS transmits its first probe and the time at which the last contender is identified by successfully delivering its message. All results are averaged over at least 10 simulation trials per experiment. Even though the original ID range is the same in each experiment, SLSRQ generates different ID ranges

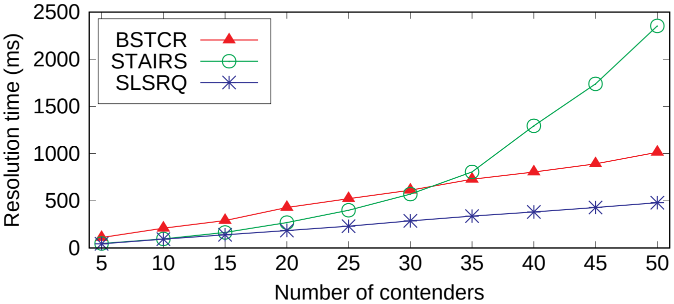

Fig. 2: Impact of the number of contenders ($N = 20{\times}K$)

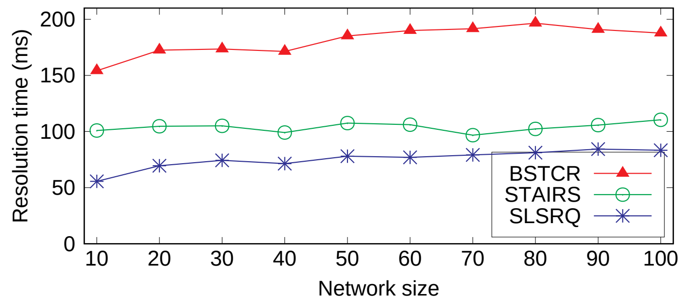

Fig. 3: Impact of network size (K=10)

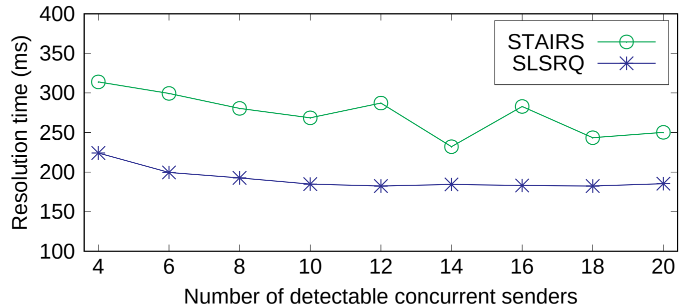

Fig. 4: Impact of the number of detectable concurrent senders (K=20, N=400)

owing to its recursive and adaptive nature, and thus its performance is observed under a variety of conditions. We compare the protocols in ideal conditions (with no packet loss, extra collisions, or undetectable edges) as well as in a more realistic setting (in the presence of external interference). In the presented figures' caption, N denotes the total number of nodes (i.e. network size) while K refers to the number of contenders. Therefore, not every node in the network is necessarily a contender. Similar to [5], we use a maximum of 50 contenders (i.e. $K \leq 50$).

B. *Interference-free Experiments*

In this section, the three algorithms are compared in ideal conditions with perfect collision/edge detection and no packet loss/corruption. Performance assesment in the absence of interference enables these schemes to be examined in the best case.

1) Number of Contenders: In the first experiment we study the performance of the three schemes while varying the number of contenders from 5 to 50. In this case the network size (N) is 20 times the size of active contenders (K) to conduct a fair evaluation by avoiding the best case situation ($N = K$) for BSTCR and SLSRQ. Note that N has no impact on the level of contention.

As depicted in Figure 2, the resolution time of all three schemes are increased as the number of contenders goes up. While STAIRS performs well for small number of contenders, it suffers considerable performance degradation as the number of contenders goes beyond 15. The reason is that at that point, only the falling edges involving 10 or less signals are correctly detected. SLSRQ shows considerable performance improvement compared to both BSTCR and STAIRS, suggesting that our approach is more efficient than the state of the art.

2) Network Size: Both BSTCR and SLSRQ need to know the network size for optimal performance, and for a fixed number of contenders their best case happens when all nodes in the network contend for the medium (i.e. $N = K$). However, the exact knowledge of N or K may not always be available to the BS. In this experiment, the number of contenders is fixed at 10, and the network size is varied from 10 to 100. It is observed from Figure 3 that both BSTCR and SLSRQ are slightly downgraded as the network size increases (i.e. contention level goes down) due to extra splitting. However, STAIRS is the least affected by variations in network size because its performance is affected by the number of contenders only.

3) Number of Detectable Concurrent Transmitters: CI is unreliable when the number of concurrent transmitters is large [11], [13]. Empirical studies show that up to a maximum of 20 [8] to 24 [12] simultaneous transmitters can be detected under the condition of CI. Therefore, this parameter is varied between 4 and 20 to simulate the conditions where several edges are not detectable as well as the situation in which all edges can be successfully detected. It is observed from Figure 4 that as more edges are successfully identified, performance of both algorithms improve. However, SLSRQ is more resilient to changes in the parameter under test because the number of detectable simultaneous transmitters has less influence on its performance than that of STAIRS. Moreover, this variable has negligible impact on SLSRQ for values greater than 10.

4) Energy Efficiency: Power consumption is a primary concern in many WSN applications. However, explicit CR consumes high power as contenders keep their radio on so as not miss out on any probes from the coordinator. Figure 5 shows the radio on times (a.k.a duty cycle) for all nodes until the last contender transmits successfully. It includes the time the radio is sending, receiving, or listening, all of which require roughly the same energy.

Contrary to STAIRS, both BSTCR and SLSRQ schemes show a decreasing energy requirement as the number of contenders goes up. The proposed algorithm is slightly

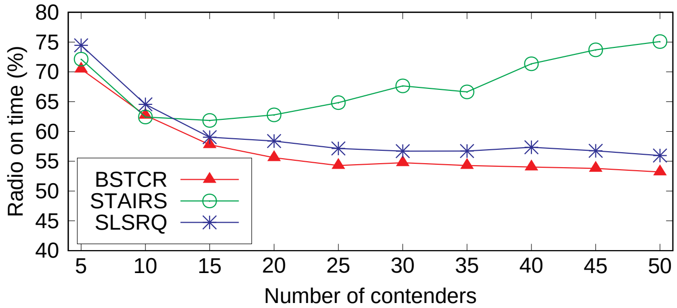

Fig. 5: Energy consumption in terms of radio duty cycle ($N = 20{\times}K$)

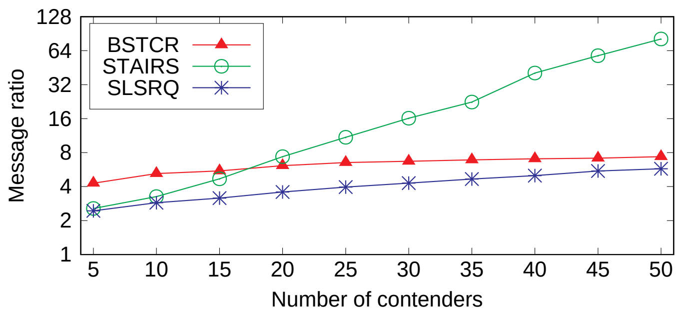

Fig. 6: Average number of messages transmitted by a contender ($N = 20{\times}K$)

more power consuming than BSTCR, yet it is more energy efficient due to its considerably shorter resolution time. Receiving concurrent contention packets in CI would take no longer than receiving a single packet of maximum length. Moreover, since the coordinator is typically a base station, which may be line-powered, its power consumption is less of a concern.

The number of exchanged messages by a protocol can provide a rough estimate of energy consumed during the resolution. We define *message ratio* as the total number of messages transmitted by contenders divided by the total number of contenders. In other words, this parameter shows how many messages are required on average for each contender prior to being granted access to the medium. Figure 6 reaffirms that SLSRQ is more energy efficient compared to the other schemes.

5) Scalability: We define *resolution time ratio* as the resolution time divided by the total number of contenders. This criterion is independent of the time and the platform, and can be indicative of an algorithm's scalability. Figure 7 shows the resolution time ratio for the three schemes, revealing the problem that STAIRS encounters at larger scales. This outcome confirms the fact that CR techniques that solely rely on contention packet length can incur serious performance penalty under high contention. In order to handle a large number of simultaneous transmitters, such schemes as STAIRS and Strawman should be used in combination with other techniques, e.g. recursion, to better resolve the tie cases. Both BSTCR and SLSRQ demonstrate good scalability due to the fact that they retain partially resolved contentions.

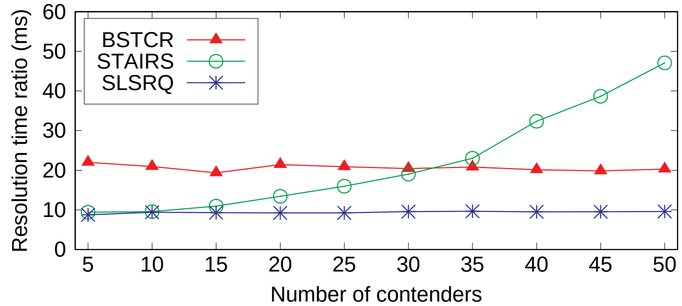

Fig. 7: Scalability of the compared schemes ($N = 20{\times}K$)

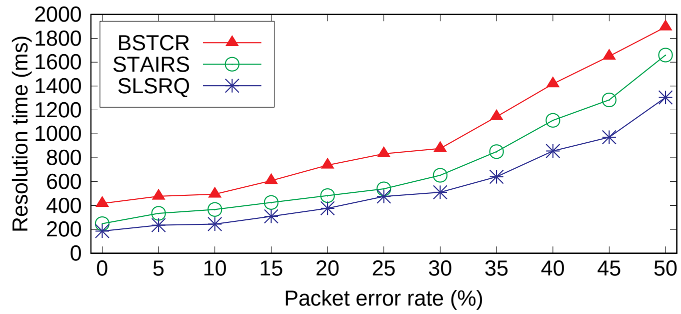

Fig. 8: Impact of packet error rate (K=20, N=400)

C. RF Noise/Interference Experiments

In wireless networks links are not stable due to channel fading and interference. As a result, understanding and modeling interference in networks of this type is important, especially in designing efficient MAC protocols [19]. Certain inaccuracies have been incorporated into the simulator to model external interference and evaluate this work in a more realistic environment.The network is composed of 400 nodes of which 20 nodes content for the medium in all of the experiments.

The following is a set of experiments by changing packet errors, and collision/edge detection accuracy levels. The modeled interference affects all nodes in the same way. The end-to-end interference rates are constant over the entire simulation run. However, the momentary rates may vary at different times during a simulation. To optimize the performance of BSTCR and SLSRQ, we minimize the impact of idle slots by issuing the next probe as soon as the current slot is found to be idle instead of waiting for the current probe period to elapse. This optimization is not applicable to STAIRS as the BS does not generate probes that trigger no transmission except at the end of the resolution process.

1) Packet Loss: We compare the performance of the three schemes under varying packet loss rates from zero up to 50%. In this experiment, random packet loss is applied to all successfully transmitted packets regardless of their type (probe, contention packet, or ACK). As mentioned in

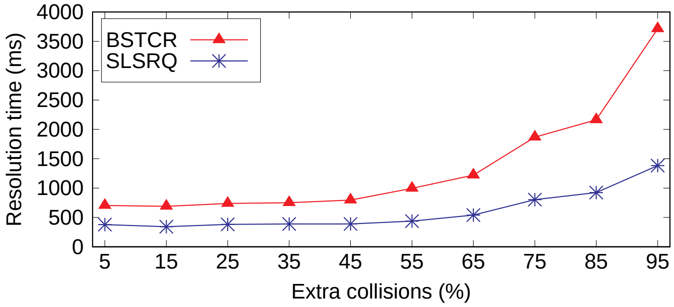

Fig. 9: Impact of extra collisions (K=20, N=400, PER=20%)

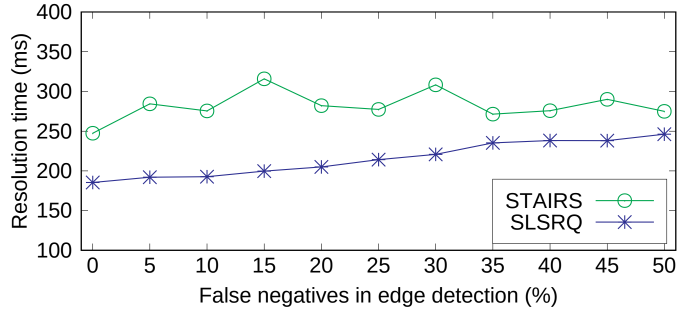

Fig. 10: Impact of undetectable edges (K=20, N=400)

Section V-C4, a corrupt packet can have a similar effect to that of a lost packet. Therefore, we use packet error rate (PER) instead of packet loss rate to encompass both corrupted and lost packets.

As illustrated in Figure 8, the resolution time of all algorithms goes up as more packets are lost (or discarded) due to external interference. The obtained results suggest that packet errors have a similar impact on all of the compared schemes. It is worth mentioning that in tree-based approaches, the original root probe is issued only once in the ideal condition. However, in the presence of external interference these protocols may need to split the original root probe several times and recreate the contention tree as not all contenders may be identified during the first traversal.

2) Collision Detection Accuracy: RF interference often manifests as an *additive* increase [19]–[21] in RSSI. Only in case of Microwave interfered packets does the elevation in signal strength come with minor leading and trailing drops [21]. Therefore, it is a valid assumption that RSSI boosts in the presence of interference, and higher RSSI leads to better collision detection. Strictly speaking, interference helps better detect true collisions by reducing the chance of false negatives. On the other hand, the higher RSSI increases the chance of false positives by detecting energy on the channel. As a result, we study how the performance of BSTCR and SLSRQ are affected by false collisions. Since STAIRS does not depend on collision detection it is excluded from this experiment.

We fix the PER at 20%, and change the ratio of false positives in collision detection (i.e. extra collisions) from 5% to 95% in steps of 10. Figure 9 shows the impact of collision detection accuracy on the protocols' resolution time. False positives lead to unnecessary splitting of the collision domain, thereby lowering the efficiency. As extra collisions become more prevalent, the resolution time increases. At 95% the performance of SLSRQ and BSTCR is degraded by 3.6x and 5.2x, respectively indicating that SLSRQ is less sensitive to collision detection inaccuracies as, contrary to BSTCR, it is not totally dependent on detecting destructive collisions.

3) Edge Detection Accuracy: In addition to collision detection, external interference may impair edge detection as well by causing false negatives (fading, undetectable edges) as well as false positives (extra edges). By using a relatively large delta (10 bytes), the probability of false positives is already reduced to a great extent. As such, the impact of false negatives on the performance of STAIRS and SLSRQ is studied. To do so, the false negative rate is increased from zero to 50% in steps of five.

Figure 10 shows that as more edges are missed, performance of both protocols degrades because less information can be extracted regarding the quantity and length of the contention packets. However, STAIRS' performance is less impacted by false negatives. The reason is that in SLSRQ, undetectable edges may lead to splitting the corresponding range in such a way that some unnecessary probes are issued, which will receive no contention packets, wasting the bandwidth. However, this is not the case with STAIRS.

4) Packet Corruption: Although RSSI variance of a corrupted frame is often higher than that of a correctly received frame [20], in wireless networks the whole packet is rarely corrupted and typically a small number of bits in a packet may be erroneous. Therefore, packet corruption may lead to minimal edge detection inaccuracies. On the other hand, since the content of contention packets are not used by either STAIRS or SLSRQ, their operation cannot be affected by corrupted content. If a corrupted packet cannot be corrected, it will be regarded as a lost packet, and thus re-transmitted. As such, packet corruption generally cannot have a worse impact than packet loss.

5) Cumulative Effect: We investigate how the compared algorithms work in the presence of moderate interference causing inaccuracies in the form of packet errors, extra collisions, and edge misdetection. It is shown that CI can sustain a reliability of more than 95% under severe interference from co-located networks [22]. In practice, about 90% of edges can be accurately detected using an online change point detection algorithm [8]. Moreover, depending on the interferers' transmission power, a Packet Reception Rates (PRR) of more than 90% is possible [23]. Additionally, the observed false positive/negative collision detection rates in an office room is less than 1% [17]. As a result, we set the packet error, extra collision and edge

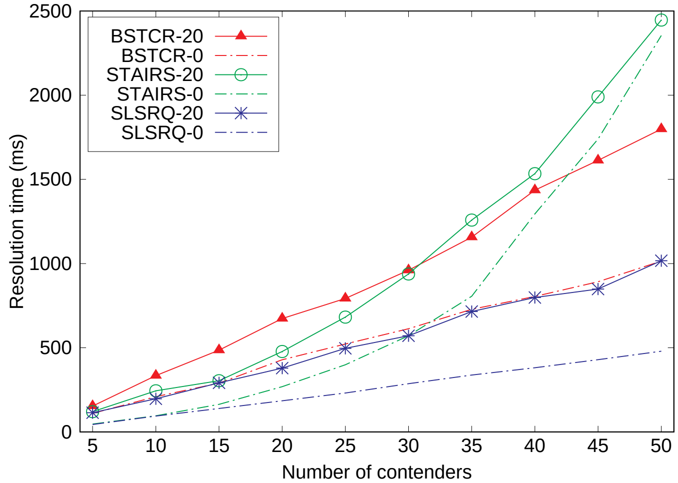

Fig. 11: Impact of external interference (20% packet error, 20% extra collisions and 20% undetectable edges, $N = 20{\times}K$)

misdetection rates, all at 20% to conduct this experiment under a stricter condition that the references mentioned above.

Figure 11 shows the performance of the three schemes in both the presence and the absence of external interference. In case of the tree-based schemes, performance degradation remains almost constant for all number of contenders. The average increase in the resolution time is 1.62 for BSTCR and 2.13 for SLSRQ. Note that BSTCR is not based on edge detection, and thus its performance is not affected by missing edges. In case of STAIRS, the resolution time is delayed by 1.7 times on average, starting from 2.5x to 1.03x in the end. The decreasing performance degradation of STAIRS reveals that as the number of contenders goes up, the scalability issue dominates the impact of interference. Based on this observation, STAIRS could face substantial challenges at higher scales regardless of the severity of interference.

VI. Conclusions

This paper proposes a novel CR technique based on constructive interference as a candidate solution to the open problem of handling various levels of contention. By utilizing the length and strength of concurrently transmitted synchronized signals, this technique performs range splitting adaptively depending on the extent of contention inferred. The presented SLSRQ scheme applies recursion in order to retain and leverage the partially resolved contention while examining the colliding signals to split the collision domain more intelligently. Simulation results confirm that the proposed scheme resolves collisions quickly and more efficiently than the state of the art. Furthermore, we demonstrate that our approach is more energy efficient and scalable than the state of the art. Due to its determinism, SLSRQ makes it easier to guarantee QoS on timely packet delivery.

Acknowledgment

This research was in part supported by a grant from IPM (No. CS1401-4-199).

References

[1] P. Du and G. Roussos, "Adaptive communication techniques for the internet of things," *Journal of Sensor and Actuator Networks*, vol. 2, no. 1, pp. 122–155, 2013.

[2] D. Xu and Y. Yao, "Splitting tree algorithm for decentralized detection in sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 12, pp. 6024–6033, 2013.

[3] C. Law *et al.*, "Efficient memoryless protocol for tag identification," in *ACM DIAL-M'00*, 2000, pp. 75–84.

[4] V. Salmani *et al.*, "Bin-MAC: A hybrid MAC for ultra-compact wireless sensor nodes," in *IEEE DCOSS'12*, 2012, pp. 158–165.

[5] P. Parsch and A. Masrur, "A slot sharing TDMA scheme for reliable and efficient collision resolution in WSNs," in *ACM MSWiM'16*, 2016, pp. 213–220.

[6] S. Laitrakun and E. J. Coyle, "Reliability-based splitting algorithms for time-constrained distributed detection in WSNs," in *IEEE DCOSS'13*, 2013, pp. 207–214.

[7] F. Österlind *et al.*, "Strawman: Resolving collisions in bursty low-power wireless networks," in *ACM IPSN'12*, 2012, pp. 161–172.

[8] X. Ji *et al.*, "Walking down the STAIRS: Efficient collision resolution for wireless sensor networks," in *IEEE INFOCOM'14*, 2014, pp. 961–969.

[9] T. Chang *et al.*, "Constructive interference in 802.15.4: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 217–237, 2019.

[10] W. Zeng *et al.*, "Low power counting via collaborative wireless communications," in *ACM IPSN'13*, 2013, pp. 43–54.

[11] C. Noda *et al.*, "On the scalability of constructive interference in low-power wireless networks," in *EWSN'15*. Springer, 2015, pp. 250–257.

[12] D. Wu *et al.*, "Simultaneous query for wireless sensor networks: A power based solution," *IEEE Transactions on Mobile Computing*, vol. 15, no. 2, pp. 475–488, 2016.

[13] W. Du, J. C. Liando, H. Zhang, and M. Li, "When pipelines meet fountain: Fast data dissemination in wireless sensor networks," in *ACM SenSys'15*, 2015, pp. 365–378.

[14] D. Liu *et al.*, "Contention-detectable mechanism for receiver-initiated MAC," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 4, pp. 31:1–31:27, 2019.

[15] M. Kodialam *et al.*, "Fast and reliable estimation schemes in RFID systems," in *ACM MobiCom'06*, 2006, pp. 322–333.

[16] M. Miskowicz, "Average channel utilization of CSMA with geometric distribution under varying workload," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 2, pp. 123–131, 2009.

[17] V. Salmani and P. H. Chou, "Resilient round robin: A lightweight deterministic mac primitive," *ACM Transactions on Sensor Networks*, vol. 11, no. 2, pp. 31:1–31:38, 2014.

[18] M. Brachmann *et al.*, "Keep the beat: On-the-fly clock offset compensation for synchronous transmissions in low-power networks," in *IEEE LCN'17*, 2017, pp. 303–311.

[19] R. Maheshwari *et al.*, "On estimating joint interference for concurrent packet transmissions in low power wireless networks," in *ACM WINTECH'08*, 2008, pp. 89–94.

[20] J. Hauer, A. Willig, and A. Wolisz, "Mitigating the effects of RF interference through RSSI-based error recovery," in *EWSN'10*, ser. LNCS, vol. 5970. Springer, 2010, pp. 224–239.

[21] F. Hermans *et al.*, "SoNIC: Classifying interference in 802.15.4 sensor networks," in *ACM IPSN'13*, 2013, pp. 55–66.

[22] M. Schuß *et al.*, "A competition to push the dependability of low-power wireless protocols to the edge," in *ACM EWSN'17*, 2017, pp. 54–65.

[23] D. Son, B. Krishnamachari, and J. S. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," in *ACM SenSys'06*, 2006, pp. 237–250.