# Strawman: Resolving Collisions in Bursty Low-Power Wireless Networks

Fredrik Österlind, Luca Mottola, Thiemo Voigt, Nicolas Tsiftes, Adam Dunkels
{fros,luca,thiemo,nvt,adam}@sics.se
Swedish Institute of Computer Science
Box 1263, SE-164 29 Kista, Sweden

## ABSTRACT

Low-power wireless networks must leverage radio duty cycling to reduce energy consumption, but duty cycling drastically increases the risk of radio collisions, resulting in power-expensive retransmissions or data loss. We present Strawman, a contention resolution mechanism designed for low-power duty-cycled networks that experience traffic bursts. Strawman efficiently resolves network contention, mitigates the hidden terminal problem, and has zero overhead unless activated to resolve data collisions. Our testbed experiments show that Strawman instantaneously provides increased network capacity when needed, allocates the available bandwidth evenly among contenders, and increases energy efficiency in multi-hop collection networks compared to the traditionally used random backoff.

## Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

## Keywords

Low-power wireless, traffic bursts, sensor networks, duty cycling

## 1. INTRODUCTION

Low-power wireless networks employ radio duty cycling to reduce energy consumption. Duty cycling decreases the opportunities to receive data, since the radio is mostly turned off. As a result, the risk of data collisions increase, and is further aggravated in networks that experience traffic bursts. Event-driven networks, such as alarm applications [2, 17], remain quiescent for the majority of the time. When an event is detected, a surge of traffic must be handled by the network before it can return to quiescence. Such sudden bursts of traffic cause radio collisions and power-expensive retransmissions in the network, and motivate the use of complex traffic-adaptive duty-cycling mechanisms or statically over-provisioned networks.

We address the problem of data collisions in duty-cycled networks and present Strawman, a contention resolution mechanism that copes with hidden terminals and is designed for receiver-initiated duty-cycled protocols. In Section 3 we illustrate the design of Strawman and how it instantaneously and dynamically provides increased network capacity when needed. Strawman is activated upon detecting data collisions. In absence of them, Strawman has zero runtime overhead. The Strawman approach to resolving collisions resembles the practice of drawing straws. Strawman uses radio transmissions to implement straws.

We implement Strawman on top of the Contiki operating system, targeting the TMote Sky [28] platform as described in Section 4. We leverage this implementation in Section 5, demonstrating that a Strawman-enabled MAC protocol is able to sustain a range of traffic loads, achieving a goodput increase of up to 77% compared to a scalable random backoff-based contention resolution mechanism. We achieve this result while evenly allocating the available bandwidth among the transmitters involved. Our testbed experiments further show that Strawman improves energy efficiency in multi-hop collection networks that experience bursts. We survey related work in Section 6, and end the paper in Section 7 with brief concluding remarks.

This work builds upon two previous papers that propose the basic Strawman protocol [25], and that theoretically study and derive Strawman's random distribution for generating packets of random lengths [12]. Compared to previous work, we improve Strawman along several dimensions and embed it within RI-MAC, a state-of-the-art, low-power MAC protocol rather than leveraging a simple proof-of-concept implementation. This allows us to test Strawman in a large-scale testbed where we verify Strawman's ability to cope with hidden terminals and a large number of contenders.

## 2. BACKGROUND

Radio communication is one of the most power-expensive activities in low-power networks. Radio duty-cycling mechanisms are employed to preserve power. Radio duty cycling, however, aggravates the risk of data collisions, especially in networks that experience traffic bursts.

### 2.1 Radio Duty Cycling

Modern low-power networks maintain communication with a radio duty cycle of only a few percent, where network nodes wake up regularly to receive transmissions from neighbors according to a pre-configured wake-up interval. Contention-based low-power MAC protocols belong to either of two classes: sender-initiated [6, 10, 27] or receiver-initiated [9, 35]. In sender-initiated protocols, the sender keeps track of past neighbor wake up times, and wakes up to start a data transmission just as it expects the receiver to wake up [10]. Sender-initiated protocols in lossy networks may cause congestion: the sender is unaware of whether an unsuccessful data


*IPSN'12,* April 16–20, 2012, Beijing, China.

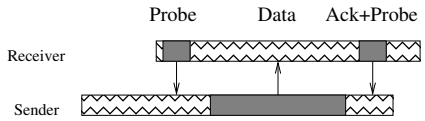



Figure 1: Receiver-initiated radio duty cycling. The sender wakes up and awaits a data probe. Upon receiving a probe, the data packet is sent. Both nodes turn off their radios after the acknowledgement.

transmission is due to link fluctuations, packet collisions, or bad wake up synchronization. The sender therefore repeatedly transmits the same data packet until it is acknowledged, or until it times out after a full wake-up interval.

Receiver-initiated duty-cycling protocols use data probes; nodes wake up periodically and probe for incoming data with a probe transmission. Neighbors that want to send data wake up just before they expect the data probe, and immediately transmit their data upon receiving such a probe, see Figure 1. The subsequent acknowledgement also serves as another data probe, enabling several data packets to be transmitted in a single wake-up. In contrast to sender-initiated protocols, receiver-initiated protocols do not repeatedly transmit radio packets if the data packet is lost. Therefore, compared to sender-initiated protocols, receiver-initiated protocols may offer lower congestion and higher throughput [9].

Duty-cycling mechanisms such as WiseMAC [10] and X-MAC [6] configure their wake-up intervals high enough to avoid collisions, but as low as possible not to waste energy on waking up when there is no data to be received. Another set of protocols additionally adapt their wake-up intervals throughout network operation to accommodate for varying traffic loads [1, 15].

## 2.2 Traffic Peaks

Traffic load variations are common in sensor networks. If all traffic flows in a network are static and known a priori, for instance by having fixed packet transmission schedules and time synchronization in a network, radio duty-cycling overhead can be minimized to a great extent [7]. Such networks are, however, uncommon. By contrast, many networks inherently induce traffic peaks. Consider an event-driven network, such as an alarm network, that lays dormant for an extended period of time until an event occurs. Upon detecting the event, several nodes simultaneously report it, causing a sudden burst of network traffic [17]. Other common reasons for temporarily increased network traffic include network code updates [21], and bulk downloads of sensor data [18].

Traffic peaks occur also in periodic data collection networks. The introduction of a new node causes neighbor discovery services to temporarily generate more network traffic [3]. Moreover, the network topology can change rapidly due to bursty links, generating further traffic [34]. Even in stable collection networks, a router that forwards data from other nodes will experience traffic peaks, due to randomness in data generation and forwarding times.

## 2.3 Collisions in Duty-Cycled Networks

Traffic peaks increase the risk of radio collisions in a duty-cycled network. Data collisions occur when multiple transmissions arrive at a receiver simultaneously, causing data loss and retransmissions. The risk of data collisions is aggravated in duty-cycled networks, since a receiver is awake less, and thus has fewer opportunities to receive data. Data collisions do not necessarily cause data loss; if one transmission is stronger than all others the receiver may still successfully decode it. This phenomenon is called capture effect [20]. Several protocols have exploited it, e.g., for fast flooding [22].

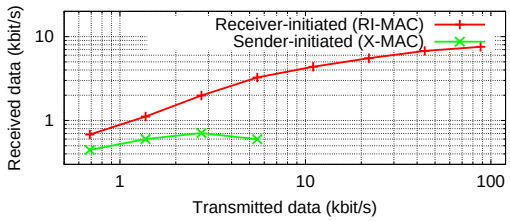



Figure 2: Receiver-initiated MAC protocols outperform sender-initiated in networks with hidden terminals and high traffic, since the sender-initiated network is flooded with colliding data packets.

Most low-power protocols are designed to cope with data collisions to some extent; typically using random backoff times before attempting to retransmit a packet to a busy receiver. However, in networks with bursty traffic, backoff mechanisms result in high latency and energy costs, since data packets may again collide at the receiver when they are retransmitted.

We perform a simple experiment that demonstrates how random backoff behaves in a congested network, using both a receiver-initiated and a sender-initiated protocol. The experiment is performed on the TWIST testbed [14], where a large set of neighbors send data to a single node, causing severe network congestion. Figure 2 shows that, as expected, the receiver-initiated network (RI-MAC) achieves a significantly higher receiver goodput than the sender-initiated network (X-MAC). This is due to packet floods in the sender-initiated network: when a lost data packet is not acknowledged by the receiver, the sender floods the network for a full wake-up interval (1 second), causing further data collisions. This experiment shows that receiver-initiated protocols have better performance than sender-initiated protocols in severely congested networks, and that random backoff does not fully avoid data collisions. Indeed, although the sender-initiated protocol should refrain from transmitting when it detects ongoing transmissions, data collisions still occur due to hidden terminals.

## 2.4 Hidden Terminals

The hidden terminal problem occurs when two or more nodes that are outside each others' communication ranges send data to the same receiver. Data transmissions may therefore collide at the receiver without the senders noticing; the nodes are hidden to each other. RTS/CTS schemes have long been used to mitigate the hidden terminal problem [36]. Data transmissions are preceded by a transmission request message (RTS). If the medium is available and the transmission is granted (CTS), any potentially interfering neighbor refrains from accessing the medium for the duration of the data transmission.

In the context of low-power wireless, however, traditional RTS/CTS protocols have been shown to induce a high overhead. Polastre et. al. show that an RTS/CTS mechanism can have an overhead of several hundred percent in low-power networks with small data payloads [27]. In receiver-initiated protocols, the problems of traditional RTS/CTS-based protocols are further aggravated: due to the implicit sender-synchronization by data probes, the RTS messages themselves collide at the receiver.

# 3. STRAWMAN

Strawman is a contention resolution mechanism designed for receiver-initiated radio duty-cycling protocols. Upon detecting data packet collisions, Strawman dynamically and instantaneously enables increased network capacity by quickly receiving data from several neighbors, and has otherwise zero overhead.

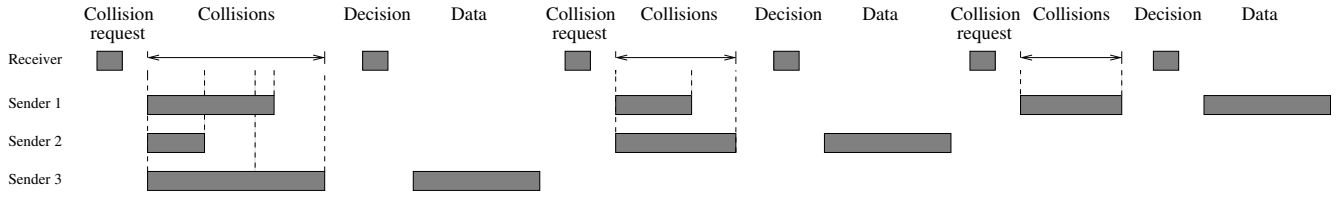



Figure 3: Strawman senders "draw straws" to gain channel access by simultaneously transmitting COLLISION packets with randomly picked lengths, resulting in a deliberate collision. The receiver does not have to correctly decode any of the COLLISION packets, but only needs to measure the duration of the collision. The sender with the longest COLLISION gains channel access and sends its packet. The process is repeated until all packets are sent.

## 3.1 All Transmit Simultaneously

We have designed Strawman for receiver-initiated MAC protocols, leveraging the implicit sender synchronization due to receiver-initiated operation. The receiver probes the channel for incoming data by transmitting a data probe packet. Next, neighboring devices with receiver-destined data transmit their DATA packets. In presence of multiple transmitters, the packets may collide at the receiver. Strawman intervenes at this stage *only if* a collision actually occurs. Note that the receiver samples the channel while waiting for a data packet, and regards channel activity without successfully receiving a packet as an indication of radio collisions.

Upon detecting a data collision, indicated by radio activity that exceeds the Clear Channel Assessment (CCA) threshold, the receiver sends a COLLISION REQUEST packet. The senders interpret this packet as the beginning of a Strawman *round*, and contend for the channel by sending a COLLISION packet of random length. The receiver estimates the length of the longest COLLISION packet by sampling the received signal strength. The receiver then broadcasts a DECISION packet containing the longest measured length, implicitly informing the corresponding transmitter that it is now granted access to the channel. While the selected transmitter transfers the DATA packet, the other contenders remain silent, as they also recognized *not* to be given access to the channel based on the information in the DECISION packet. The subsequent COLLISION REQUEST broadcasted by the receiver both initiates a new Strawman round and acknowledges the previously received data packet. In the case that several contenders have chosen the same random length and their DATA packets have collided, the receiver nevertheless sends another COLLISION REQUEST since it has detected the contenders' transmissions by sampling the channel. This process repeats until all contenders have successfully sent their DATA packets. Figure 3 depicts an example execution with Strawman that schedules and transfers three data packets.

Strawman's COLLISION and DECISION packets provide a functionality similar to RTS/CTS handshakes [36], but allow *multiple* transmitters to request access to the channel *simultaneously*.

## 3.2 Collisions of Random Length

The random lengths of COLLISION packets effectively determine which transmitter is granted access to the channel. In a sense, this resembles random back-off techniques, as it is still a random choice at the transmitter side that regulates channel access. However, in Strawman the contenders *actively* compete for the channel, using the COLLISION packet to inform the receiver on their random choice. Unlike random back-off techniques, this entails that the other contenders also know that they are *not* given access to the channel, based on the DECISION packet.

We use a truncated decreasing geometric distribution to draw the random lengths of COLLISION packets. Compared to the more common uniform distribution, a truncated geometric distribution provides higher variance within a bounded interval for random samples. In Strawman, this translates into better scalability [12].

We use a granularity of 7 bytes for the COLLISION packet length. 7 bytes correspond to a transmission time of 224$\mu$s at 250 kbits/s—the bandwidth of our target radios—enabling an accurate estimation of the COLLISION packet length, as we report in Section 5.1.

## 3.3 Multi-channel Operation

Strawman reduces contention by multi-channel operation; the receiver randomly selects which radio channel the senders should use. Like A-MAC [9], we allocate a pre-determined channel for the transmission of the initial data probe packets and then, for the DATA transfer and any subsequent Strawman rounds, all communication takes place on another radio channel. Particularly, the initial data probe contains an entry indicating the radio channel to use next. Upon receiving the data probe, every contender immediately switches to the indicated channel prior to sending the DATA packet. When the execution completes, all involved nodes return to the initial channel.

# 4. IMPLEMENTATION

We have implemented Strawman on Contiki, targeting the TMote Sky [28] platform equipped with 802.15.4-compliant CC2420 radios. As experimental and evaluation platform for Strawman, we implement our own version of RI-MAC [35], and extend it with multi-channel operation. Our implementation of RI-MAC uses wake-up schedule synchronization [10], and hop-by-hop acknowledgments.

We use this implementation of RI-MAC to evaluate Strawman's performance. For comparison, we build another version of RI-MAC with a random backoff-based contention resolution mechanism. In addition, we extend it with the geometric distribution proposed by SIFT [17] to increase contender scalability. We use a delay granularity of 320$\mu$s—a minimum slot size enforced by the radio's turnaround time, also used by the original implementation of RI-MAC [35].

**Length estimation.** We implement the transmission of COLLISION packets as 802.15.4 frame transmissions. On every node, we preload a COLLISION packet in the radio's outgoing buffer, similarly to existing work [29, 24]. This improves the overall latency and allows for synchronized transmissions of COLLISION packets from multiple transmitters—the key to correctly determine the length of the longest COLLISION packet at the receiver.

We implement COLLISION packet length estimation by subsequent Clear Channel Assessment (CCA) checks at the receiver. We experimentally calibrate the number of CCA checks that are to return an indication of "busy channel" to determine a correct COLLISION length estimate. Unless otherwise specified, we always use the CC2420's default CCA threshold of -77 dBm.

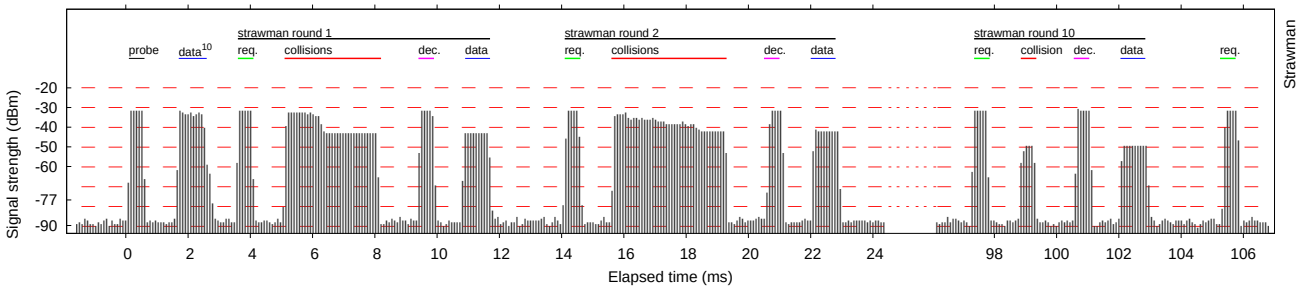


Figure 4: Signal strength profile of Strawman resolving collisions from 10 simultaneous contenders.

**Alleviating channel noise.** By relying on subsequent CCA checks to estimate packet lengths, we risk confusing channel noise with COLLISION transmissions. We leverage two simple techniques to alleviate the problem. First, as the transmissions of COLLISION packets from the contenders are synchronized, the receiver knows exactly when they occur, and starts sampling the channel immediately *before* this time. If the channel is busy, Strawman aborts its operation, as this condition indicates channel noise. Similarly, if a receiver estimates a longer COLLISION packet length than possible, it assumes channel noise and aborts its operation. If two consecutive Strawman rounds experience either of these conditions and also fail to receive DATA packets, the receiver returns to sleep and will operate on a different radio channel the next wake-up as described in Section 3.3.

**Example run.** Figure 4 shows a signal strength profile of a Strawman-enabled network operating on a single radio channel. These patterns correspond to a concrete execution of the processing intuitively described in Figure 3.

The setup in Figure 4 includes one receiver and ten contenders in the same collision domain. The individual Strawman rounds can be identified by the signal strength patterns. Starting from the leftmost side of the picture, a RI-MAC data probe is sent at time 0 ms, resulting in simultaneous data transmissions between 2 ms and 3 ms from all 10 contenders. These data packet transmissions collide at the receiver.

The collision causes the activation of Strawman, with the COLLISION REQUEST packet being sent out by the receiver at about 4 ms. In the first Strawman round, all 10 contenders transmit COLLISION packets, starting at about 5 ms. The signal strength profile of this phase indeed suggests that multiple COLLISION packets of different lengths collide. The receiver measures the length of the longest COLLISION packet, and sends out the DECISION packet at about 9.5 ms. The contender granted channel access transmits the data packet at 11 ms and drops out the following Strawman round, where the remaining contenders will repeat the same procedure.

At the rightmost side of the picture, only one contender is left. As a result, the COLLISION phase is shorter: with fewer contenders the probability to be granted channel access with a smaller COLLISION length increases. The 10th DATA packet is finally acknowledged at time 105 ms.

# 5. EVALUATION

We evaluate Strawman's performance along several dimensions. Our results reveal several key findings:

- Our technique for estimating the length of COLLISION packets, which determines which node is granted channel access, is accurate in a range of different situations, as illustrated in Section 5.1.
- Strawman has no overhead when data collisions do not occur, and a limited energy cost when data collisions are resolved, as we illustrate in Section 5.2.
- A Strawman-enabled MAC protocol can sustain a range of different traffic loads, quickly reacting to changing conditions, and does so by evenly allocating the available bandwidth, as we show in Section 5.3 and 5.4.
- Strawman's performance is a result of its ability to cope with hidden terminals efficiently: we investigate the presence of hidden terminals in our experimental setup and how Strawman reacts to them in Section 5.5, comparing its performance against that of Black Burst [33].
- In a realistic scenario using standard tree routing protocols, Strawman makes the network much more robust to sudden traffic bursts and significantly reduces the corresponding energy overhead, as we show in Section 5.6.

Based on these results, we argue that Strawman is a welcome addition to receiver-initiated low-power MAC protocols. By not imposing any additional overhead in absence of collisions, it allows the MAC protocol to run without unnecessary performance penalties. Should collisions occur, Strawman quickly intervenes to resolve them efficiently.

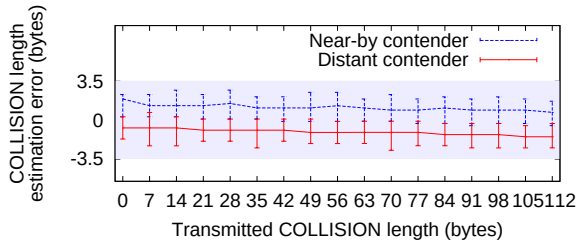
## 5.1 Collision Length Estimation

Strawman relies on accurately estimating COLLISION lengths. A COLLISION length estimation determines who wins channel access via the subsequent DECISION transmission. If the COLLISION length is underestimated, multiple contenders may transmit DATA packets simultaneously, causing collisions. On the other hand, if it is overestimated, no contender will send its DATA. We now perform a set of micro-benchmarks to assess how effective our channel sampling technique is for estimating COLLISION lengths.

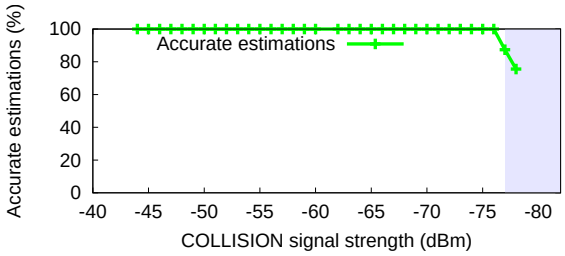
### *5.1.1 Collision Lengths*

We study how the COLLISION length affects the accuracy of packet length estimation. We use two TMote Sky nodes configured as receiver and contender. The receiver periodically probes the channel for incoming data. The contender replies with a COLLISION packet of varying length. For every possible COLLISION length, we run at least 350 repetitions of the experiment. We furthermore use two different distances between the nodes: a *near-by* contender is placed 0.5 m. from the receiver, a *distant* contender is 10 m. from the receiver. We decrease the transmission power of the distant contender, so that it can barely communicate with the receiver. The receiver uses the CC2420's default CCA threshold of -77 dBm.

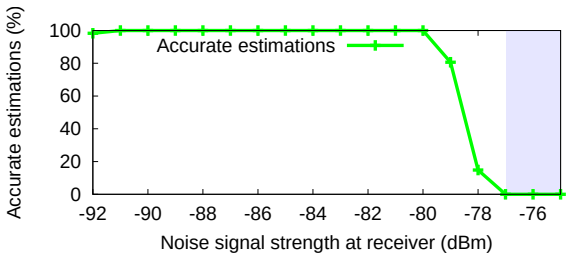
**Results.** Figure 5a shows the median error in estimating the COLLISION length, against the actual transmitted length. The error bars reflect the minimum and the 98th percentile of the estimated lengths. We observe that 98% of the estimations are within the



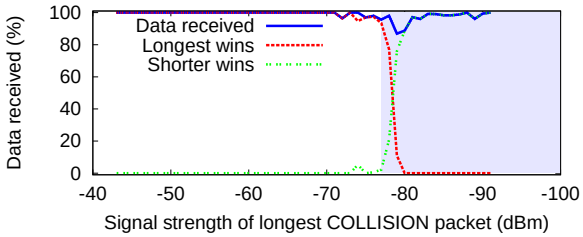

(a) The COLLISION length estimation error remains within the 7-byte granularity for both near-by and distant contenders.




(b) The COLLISION length is accurately estimated when received with signal strength above the CCA threshold.




(c) The COLLISION length is accurately estimated when the external interference is below the CCA threshold.




(d) The contender with the longest COLLISION packet wins channel access and sends data (up to -77dBm) unless its COLLISION is too weak to be detected, in which case the contender with the shorter COLLISION packet wins.

Figure 5: Micro-benchmarks: Strawman accurately estimates the lengths of COLLISION packets (a) of different lengths, (b) received with different signal strengths, (c) under external interference, and (d) under interference from an out-of-range contender.

7-byte level of granularity used in our implementation, indicated by the shaded area in the chart. This shows that in almost all cases a Strawman-enabled receiver accurately estimates the length of the COLLISION packet of a single contender, in absence of interference.

Figure 5a also illustrates the effect of distance, and therefore signal strength, on length estimation accuracy: packets from a distant contender are underestimated and packets from a near-by contender are overestimated. This phenomenon is an artifact of CC2420's CCA that is calculated from a moving average of the last 8 received signal strength values. Nevertheless, 98% of length estimations of both distant and near-by contenders' packets are still within the 7-byte level of granularity.

### 5.1.2 Collision Signal Strengths

We study how COLLISION packets' received signal strengths affect the accuracy of packet length estimation. We use the same experimental setup as above but also vary the receiver-contender distance to generate different signal strengths at the receiver. The receiver logs the signal strength of each received COLLISION packet along with the corresponding length estimate.

**Results.** Figure 5b shows the ratio of COLLISION length estimations inside the 7-byte level of granularity. In a real network, bad length estimations decrease network performance and cause data collisions. This experiment shows that bad length estimations are uncommon unless the COLLISION packet's signal strength is close to the CCA threshold.

### 5.1.3 Interference from External Noise

We study how external radio interference affects the accuracy of packet length estimation. We use two TMote Sky nodes: one receiver and one contender set 3 m. apart. In addition, to obtain repeatable experiments, we leverage the method by Boano et al. [5] to generate a constant and controllable interference, using a third TMote Sky node as interferer. We control the signal strength of external radio interference by moving the interferer closer to the receiver. We ensure that the single contender always receives the COLLISION REQUEST packet and sends the corresponding COLLISION. The receiver logs the noise level immediately before sending the COLLISION REQUEST packet as well as the COLLISION length estimation.

**Results.** As expected, Figure 5c shows that the correctness of our COLLISION length estimation starts falling outside the 7-byte granularity level only as the interference level approaches the CCA threshold. Under these conditions, the receiver is unable to discern the transmission of a COLLISION packet from noise. Similar situations, however, would break most traditional transmission schemes based on CCA checks. Indeed, the CCA check would always indicate the channel as busy. The transmission scheme would react first by deferring the transmission, and then ultimately dropping the packet upon expiration of a timeout or after a maximum number of CCA checks.

### 5.1.4 Interference from Out-of-range Contenders

A distant Strawman contender that receives a COLLISION REQUEST and sends back COLLISION packet may be unable to reach the receiver due to asymmetric or fluctuating radio links. In our final micro-benchmark, we study how such out-of-range contenders affect the outcome of Strawman rounds with multiple contenders.

We use three TMote Sky nodes: one receiver and two contenders. One contender is kept *near-by* the receiver at 0.5 m. whereas we vary the distance of the second contender from the receiver, from 0.5 to 20 m. We configure the output power so that the receiver cannot hear the moving contender at 20 m. distance. To obtain a controlled setting, we configure the two contenders to use fixed COLLISION lengths, rather than the previous random lengths. Particularly, the near-by contender always competes with the shortest possible COLLISION length (0 bytes payload), whereas the moving contender uses the longest possible (112 bytes payload). Therefore,

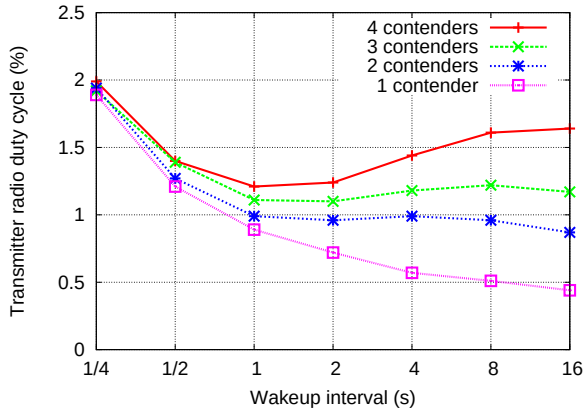



Figure 6: Contender radio duty cycle against wake-up interval. Strawman intervenes only when DATA collisions occur, and has no overhead otherwise. The $x$ axis uses a logarithmic scale.

the moving contender should always be the one granted access to the channel, as long as the receiver hears its COLLISION packet. To analyze this aspect, we log the signal strength of the COLLISION packet coming from the moving node.

**Results.** Figure 5d shows the total delivery ratio for DATA packets sent by either contender, against the signal strength of the COLLISION packet coming from the moving node. As long as the signal strength of the moving contender is sufficiently high to be perceived by the receiver, the moving contender is scheduled to transmit the DATA packet, corresponding to almost 100% data delivery from this node.

The situation progressively reverses as the COLLISION signal strength of the moving contender becomes weaker, until the receiver hears only the short COLLISION packet from the near-by fixed node. Under these conditions, only the near-by node is allowed to transmit the DATA packet. Nevertheless, the receiver almost always successfully receives a DATA packet from either of the two contenders.

## 5.2 Energy Cost of Resolving Collisions

Strawman makes networks robust against traffic bursts, but has an energy cost when used. If the network is constantly overloaded with traffic, queues of pending packet form that induce an energy cost in the network. We perform an experiment to demonstrate the relationship between network traffic, number of contenders, and the radio duty cycle.

**Setting.** We simulate a network with a single receiver and four contenders in Cooja, which allows us to have perfect control of the system execution. All contenders are hidden to each other but have a perfect and static link to the receiver. Every contender generates a DATA packet once every 4 seconds. We vary the nodes' wake-up intervals from four times per second to once every 32 seconds. By increasing the wake-up interval, we expect the risk of DATA collisions to increase.

**Results.** Figure 6 shows the average radio duty cycle for contenders against their configured wake-up interval. With short wake-up intervals, the collision risk is small. As the wake-up interval increases, collisions occur more often and Strawman intervenes to reschedule DATA packets. According to this chart, this generates a limited energy overhead in the configurations we tested. Strawman's cost of rescheduling colliding DATA packets is indeed the difference between the single and the multi-contender curves in Figure 6.

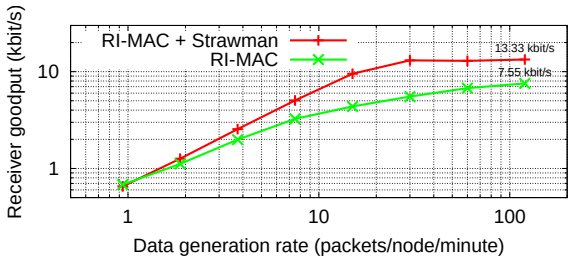



Figure 7: Goodput in RI-MAC using Strawman and random backoff. Strawman achieves up to 77% higher goodput than random backoff for high data rates.

This experiment also demonstrates that Strawman networks are robust with regards to the configuration of the wake-up interval. Strawman's efficient contention resolution allows all packets destined to a given receiver to be delivered within the (few) wake-up intervals available. This is possible because Strawman quickly provides increased network capacity when needed, which in this case is precisely at the time of waking the receiver up.

## 5.3 Different Traffic Loads

We evaluate the performance of Strawman in sustaining a range of different traffic loads, especially in terms of the network capacity provided against different network demands. We also study the fairness properties of Strawman in allocating the available bandwidth among multiple contenders, and how the CCA threshold affects the performance we observe in this setting. We describe next the settings common to all experiments in this section.

**Setting.** We use TWIST [14], a testbed with 100 Tmote Sky sensor nodes that provides a particularly dense network: a single node transmission can be received by up to 65 other nodes. A dense network has a potentially large number of contenders, which is beneficial to study the performance of Strawman. We find that the TWIST topology results in a number of hidden terminals, an aspect that we investigate more deeply in Section 5.5. All nodes operate at maximum transmission power. We compare Strawman with our implementation of random backoff-based RI-MAC, as described in Section 4.

Our setup includes a single receiver node probing the channel for data once per second. All other nodes in the testbed act as contenders. The payload size of the DATA packets is 110 bytes: including the overhead of the network stack, this corresponds to a maximum sized 802.15.4 frame. We repeat the experiment using a wide range of data generation rates: from roughly one DATA packet per minute, up to 2 DATA packets per second. We expect the network to reach its maximum capacity within this interval. To measure the sink goodput, we log on the nodes all DATA packets transmitted and received. We exclude from the statistics duplicate DATA packets, which may occur in case of lost acknowledgements.

### *5.3.1 Goodput and Fairness*

We start by measuring the receiver goodput against varying traffic loads, and by investigating the fairness properties of Strawman.

**Results.** Figure 7 shows the receiver goodput against varying traffic loads for both Strawman and random backoff, in logarithmic scale. Strawman is able to receive all generated DATA packets up until a data generation rate of one packet per node every 4 seconds. Random backoff, in contrast, loses packets already at lower data generation rates, resulting in reduced goodput. The maximum goodput achieved for Strawman is 13.33 kbit/s. For random backoff, the maximum goodput is 7.55 kbit/s. The experiment logs show that

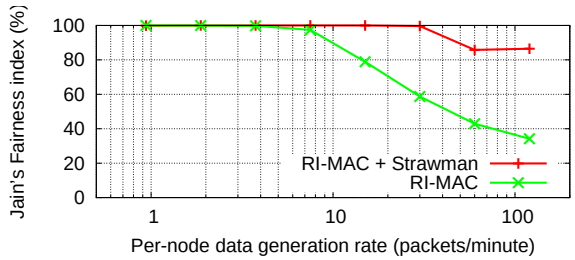



Figure 8: Fairness properties of Strawman and random backoff. Random backoff is driven by capture effect, and is thus unfair with regards to different contenders.

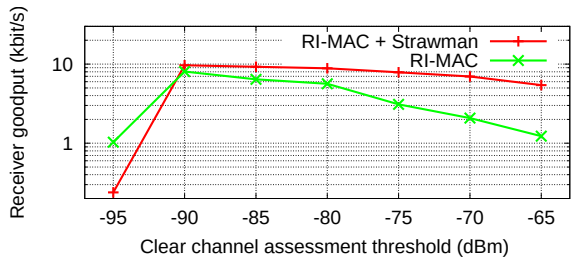



Figure 9: Influence of CCA threshold on Strawman and random backoff. In the best configuration, Strawman still performs better than random backoff.

Strawman successfully funnels over 15 DATA packets each wake-up, in comparison with random backoff's 8.6 packets.

We also investigate how Strawman divides the available bandwidth among contenders. To study this aspect, we use Jain's Index as a fairness measure calculated over the 30 most active contenders, as some nodes are in a grey-zone and do not participate in every round. Figure 8 shows the corresponding results in the same range of data generation rates of Figure 7 in logarithmic scale. The plot indicates that Strawman is generally more fair in scheduling contenders compared to random backoff. Indeed, the latter inherently relies on capture effect to decide which node, among multiple senders, ultimately delivers a packet. This entails that the choice is implicitly driven by the physical topology, and therefore likely to be biased towards near-by contenders. On the contrary, in Strawman the choice of which node is granted access to the channel is completely random.

### 5.3.2 *Clear Channel Assessment Sensitivity*

The detection of neighbors' ongoing transmissions is strongly influenced by the CCA threshold. In addition, it also affects the occurrence of hidden terminals, since contenders become more or less sensitive to hearing each other. Existing work postulate that, in principle, all hidden terminals conditions may be removed in a star network simply by increasing the sensitivity [39].

To investigate how Strawman is affected by the CCA threshold configuration, we repeat the experiments previously discussed using a fixed data generation rate: each node generates 15 DATA packets per minute, and we vary the CCA threshold across different repetitions.

**Results.** Figure 9 shows the goodput for different values of CCA threshold, ranging from -95 dBm to -65 dBm. Note that the $y$ axis uses logarithmic scale. When the CCA threshold is set to low values, Strawman cannot distinguish COLLISION packets from background noise and does not schedule any DATA packets. The goodput performance consequently suffers. When the CCA threshold is set to a high value, Strawman cannot detect weak COLLISION transmissions and ignores contenders far from the receiver. On the other hand, in such situations random backoff suffers even more, since the hidden terminal problem is severely aggravated, and random backoff cannot deal efficiently with it.

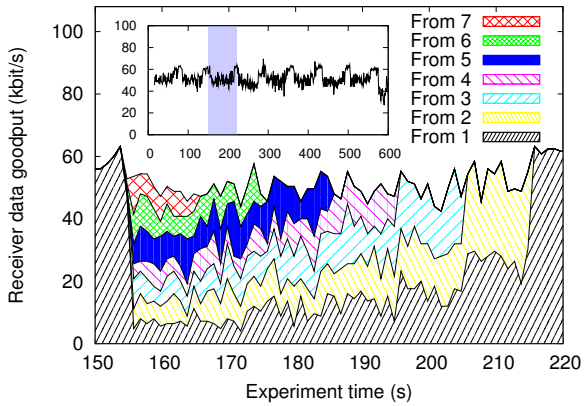



Figure 10: Saturated link: goodput over time. Every node quickly obtains its fair share of the medium while the total goodput remains high.

The best overall performance we obtain in these experiments corresponds to a CCA threshold of -90 dBm. With this configuration, Strawman still performs better than random backoff, yielding a goodput of 9.7 kbit/s against 7.3 kbit/s. However, we are not aware of real deployments using CCA threshold settings different from the default one. Indeed, it is very difficult, let apart tedious, to run a statistically significant set of experiments to determine the best CCA threshold in a given environment.

## 5.4 Reacting to Sudden Traffic Bursts

We now study how Strawman handles intense traffic surges in which multiple contenders attempt to transmit at full speed to a single receiver, and in particular how Strawman allocates the bandwidth among contenders when new bursts are introduced into the network. Ideally, we expect Strawman to provide each active contender with a fair share of the medium while maximizing the overall throughput.

**Setting.** We use a 1-hop network with 8 TMote Sky nodes running RI-MAC with Strawman, measuring the resulting goodput. Seven of the nodes are configured to *always* contend for permission to transmit data to the single receiver. In contrast to the experiments in Section 5.3, this network is both smaller and has reduced logging, resulting in higher total goodput. The number of active contenders during the experiment varies at intervals of 10 seconds. Each data packet has a payload of 100 bytes.

**Results.** Figure 10 shows an excerpt of the goodput measurements over time. In the beginning, only node #1 is active. After 155 seconds, all nodes (#1-#7) become active for 10 seconds. Node #7 is then deactivated at time 165 seconds, leaving 6 contenders active. The remaining contenders are then progressively deactivated, one every 10 seconds, until the system is back to a condition with only node #1 is active. The chart brings two fundamental insights:

1. Strawman *instantaneously* matches changed traffic conditions; when the number of contenders suddenly increases from 1 to 7, Strawman quickly reacts without a significant reduction in the total goodput.
2. As the number of contenders varies, Strawman *evenly* divides the available bandwidth among the contenders in the system;

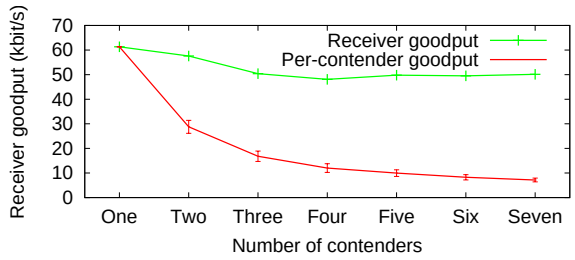



Figure 11: Saturated link: overall goodput performance. With Strawman, the receiver goodput remains high for up to 7 contenders, with no noticeable reduction already from the case with 3 contenders.

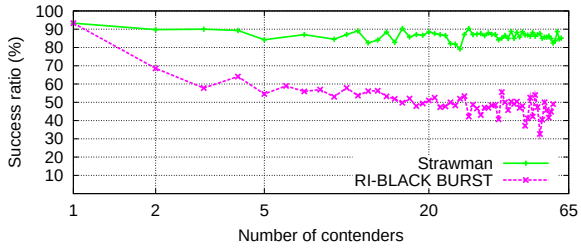



Figure 12: Data delivery using Strawman and RI-BLACK BURST. Strawman successfully mitigates hidden terminals up to 65 contenders. The $x$ axis uses a logarithmic scale.

in the long run, this results in a fair allocation of bandwidth resources.

To provide a quantitative assessment on the overall goodput performance in this experiment, Figure 11 shows the average goodput depending on the number of active contenders. As expected, the total goodput is highest with only one contender: 61.3 kbit/s. At the opposite end of the spectrum, the total goodput with 7 active contenders is 50.1 kbit/s, yet there is no significant reduction already starting from the case with 3 active contenders. The chart therefore shows that Strawman successfully keeps the link almost saturated independently of the number of contenders.

## 5.5 Coping with Hidden Terminals

We aim at identifying the presence of hidden terminals in our setup, and their effect on Strawman's performance.

**RI-BLACK BURST.** To quantify the presence and effects of hidden terminals, we develop a variant of the Black Burst protocol proposed by Sobrinho et al. [33]. The Black Burst protocol, like Strawman, resolves contention by measuring the longest of several colliding transmissions. By contrast, Black Burst does not employ DECISION packets to inform the contenders who gains channel access, but instead relies on contenders' clear channel assessments: if the channel is clear, a contender concludes that its COLLISION packet was the longest and accesses the channel. The Black Burst protocol cannot cope with hidden terminals since it lacks the DECISION packet. We develop a receiver-initiated Black Burst variant that we call RI-BLACK BURST. The Black Burst protocol is designed for CSMA-based WiFi networks, and does not synchronize contenders with an initial COLLISION REQUEST transmission. To isolate the effects of hidden terminals, we therefore compare Strawman with RI-BLACK BURST.

**Setting.** We use the same TWIST testbed setup as in Section 5.3. In absence of hidden terminals, both Strawman and RI-BLACK BURST grant channel access to the same contenders. Indeed, as all contenders can hear each other, there will be only one of them that find the channel clear after transmitting of the COLLISION packet. However, in presence of hidden terminals, Strawman and RI-BLACK BURST behave differently. With RI-BLACK BURST multiple contenders that are hidden from each other may access the channel simultaneously. On the other hand, RI-BLACK BURST is slightly faster than Strawman, as it does not need to transmit the DECISION packet.

**Results.** Figure 12 shows the overall data delivery ratio of Strawman and RI-BLACK BURST against a varying number of contenders. We were unable to find more than 65 one-hop transmitters in our testbed. Based on these results, we conclude that hidden terminals do exist, but they affect the performance of RI-BLACK BURST only. Using Strawman, the overall data delivery remains high up to 60 contenders: more than 85% of all Strawman rounds with 60 contenders successfully deliver the DATA packet. By contrast, RI-BLACK BURST delivers less than 60% of the DATA packets already with 3 contenders, and becomes drastically worse than Strawman as the number of contenders increases.

## 5.6 Multi-hop Data Collection

We assess the impact of Strawman in a realistic network scenario by setting up a data collection network over a multi-hop topology. In such a scenario, the network operation is subject to issues such as concurrent control and application traffic, inter-node wireless interference, and packet losses and retransmissions.

**Setting.** We use 82 nodes in the TWIST testbed, this time by configuring the CC2420 radio chip to use a lower transmission power mode to promote multi-hop topologies. The resulting network setup stretches across at least 4 hops.

To establish multi-hop routes, we use the Contiki Collect protocol, which establishes a tree-shaped routing topology from all nodes to a sink using a routing metric based on ETX. Contiki Collect and TinyOS CTP have been shown to achieve similar performance in low-power data collection [19]. To study how our network is affected by sudden traffic bursts, we also instrument the protocol with the ability to temporarily disable all route maintenance. By doing so, we ensure that multiple traffic bursts are forwarded over the same routes, thus factoring out the influence of route maintenance on our study. Nevertheless, the setup still includes data collisions, retransmissions, and acknowledgements.

Using this setup, we test three different traffic patterns, corresponding to different settings we study to isolate the effect of Strawman in absence or presence of traffic burst. We test every traffic pattern for at least 40 minutes:

**No traffic (NT):** the network generates no radio traffic. This profile serves to demonstrate Strawman's sensitivity to external noise and provides a baseline for the bursty traffic experiment.

**Periodic traffic (PT):** each node generates a DATA packet every 5 minutes, on average. This allows us to study how Strawman handles sporadic collisions, mostly due to hidden terminals.

**Bursty traffic (BT):** after making sure routes are stable, we disable route maintenance and instantaneously generate one DATA packet each on 8 randomly-selected nodes. This generates a sudden surge of traffic that yields intense collisions across multiple hops leading to the sink, which is the scenario we target.

Throughout the study, we compare Strawman against the random backoff-based version of RI-MAC. We draw our conclusions based on data delivery ratio at the sink and system-wide radio energy consumption.

| | RI-MAC + Strawman | RI-MAC |
|---|---|---|
| NT radio duty cycle (%) | 0.34 | 0.40 |
| PT radio duty cycle (%) | 3.94 | 4.40 |
| BT radio on-time (sec) | *4.53* | *8.16* |

Table 1: Strawman improves on RI-MAC's energy consumption both in absence and in presence of collisions.

**Results.** Regardless of the traffic pattern, the data delivery at the sink is always comparable using Strawman or random back-off. Specifically, all 8 packets included in a traffic burst are always delivered to the sink in either configuration.

Table 1 shows energy consumption figures under different traffic profiles. Already with no radio traffic (NT), Strawman slightly reduces the necessary radio duty cycle. We attribute this behavior to its ability to more quickly distinguish channel noise from actual transmissions. In particular, when RI-MAC mistakes channel noise for DATA and sends a COLLISION REQUEST packet, Strawman immediately expects a COLLISION packet in reply. If this does not happen, Strawman immediately turns the radio off. With random backoff, by contrast, RI-MAC must wait for the duration of the backoff window before turning off the radio again.

Under periodic traffic (PT), the Strawman-enabled network has a lower radio duty cycle than the backoff-based RI-MAC network. This improvement is due to Strawman's ability to immediately resolve collisions thus avoiding the need for later retransmissions.

To quantify the net energy overhead due to radio communication under bursty traffic (BT), we subtract the NT radio duty cycle discussed above from the total radio usage during each burst. We report the total radio on-time to funnel the packet burst to the sink on the bottom row of Table 1. Compared to the backoff-based RI-MAC, Strawman halves this figure in our setting. As a result, Strawman makes the network much more robust against sudden traffic bursts, by reducing the energy overhead of contention resolution when collisions occur.

# 6. RELATED WORK

Strawman builds on the body of work in contention resolution schemes, on protocols dealing with traffic bursts, as well as on recent findings about simultaneous wireless transmissions. In the following, we briefly survey the literature on these topics.

**Contention resolution schemes.** Common solutions to channel contention problems are random back-off schemes, even in traditional networks. Such techniques are also applied in the wireless domain, and specifically to sensor networks [17, 27, 38]. In this context, one of the main design choices is the random distribution to sample from. As examples, early solutions use uniform distributions [27], whereas Jamieson et al. propose a geometric distribution [17]. Strawman differentiates from these techniques in the use of *active* contenders, as opposed to the passive behavior of competing nodes when using random back-off. Nevertheless, the work on random distributions carried out in this context has inspired us to use a geometric distribution which provides advantages over uniform distributions [12].

Strawman's core mechanisms bear similarities with bit-dominance protocols. However, Strawman leverages dynamic priorities rather than static, as contenders in Strawman compete with different priorities every time. Solutions inspired by bit-dominance protocols exist also in the wireless domain [26, 30]. However, they require the underlying physical medium to be based on On-Off-Keying (OOK) modulation. By contrast, Strawman does not impose requirements on the underlying modulation mechanism. The Black Burst [33] protocol by Sobrinho et al. and the HIPERLAN protocol [16] are similar to Strawman in that they measure packet lengths to resolve contention; the contender with the longest transmission wins channel access. Like Strawman, they are designed for wireless networks and do not rely on OOK modulation. Unlike Strawman, they do not cope with hidden terminals, and are not designed for duty-cycled low-power networks. We develop a receiver-initiated version of the Black Burst protocol and compare it with Strawman in Section 5.5, to quantify the performance effects of hidden terminals on receiver-initiated low-power protocols.

To avoid collisions due to hidden terminal problems, Request-To-Send/Clear-To-Send (RTS/CTS) protocols are typically used. However, they are shown to exhibit a considerable overhead when used for wireless transmissions [4, 27]. Alternative solutions also exist. For instance, ZigZag decoding [13] exploits the effects due to interference cancellation in 802.11 networks to enable decoding of colliding packets. Strawman uses a form of RTS/CTS mechanism to resolve collisions, yet this is based on multiple simultaneous transmissions, in a sense similarly to ZigZag decoding. We are not aware of other collision resolution mechanisms based on multiple simultaneous transmitters in sensor networks. Strawman leverages this technique to improve on the resulting latency and throughput.

**Dealing with traffic bursts.** Strawman operates at the MAC level. In this context, the predominant approach in sensor networks is CSMA, because of its simplicity and low overhead compared to TDMA [40]. To deal with traffic bursts, adaptive MAC protocols change their operation along different dimensions, e.g., by tuning the wake-up periods [1, 15], by using packet trains, and by alternating between CSMA and TDMA techniques [31]. Namboodiri and Keshavarzian designed Alert, a MAC protocol designed for traffic bursts in mostly idle networks [23]. Their goal is to minimize the delay of the first message. They reduce contention by a combination of time and frequency multiplexing. These MAC-level techniques are complementary to Strawman, as they operate at the protocol level rather than during the actual transmission of the individual packets, as Strawman does.

In a broader perspective, the existing literature includes several mechanisms for handling traffic bursts in sensor networks. These typically entail some form of cross-layer interaction, either by requiring information sharing between layers, or by affecting the performance of upper layers. As a result, these need to be aware of the underlying mechanisms to counteract their influence. Examples include ESRT that requires interaction between the application layer and the MAC layer [32], and the adaptive MAC layer by Woo and Culler [38], which uses random back-off, thus incurring performance penalties for the application layer. Differently, Strawman does not require application awareness or cross-layer interactions.

**Simultaneous wireless transmissions.** A few recent works exploit low-level radio effects [8, 9]. As examples, Demirbas et al. use radio collisions to implement Pollcast and Countcast, network primitives that enable voting among immediate neighbors [8]. Dutta et al. show that collisions of identical 802.15.4 packets do not necessarily lead to data corruption, and implement an anycast-like primitive called Backcast [9]. Lu and Whitehouse exploit the capture effect for rapid flooding of sensor networks [22]. The Glossy protocol relies on what the authors call *constructive interference*, i.e. the superposition of the same RF signals generated by multiple senders, to further improve flooding efficiency [11]. Whitehouse et al. present a mechanism for recovering partial information from semi-collided packets [37]. Unlike these approaches, Strawman does not attempt to extract information from the colliding packets,

but we measure the duration of the longest transmission to infer the length of the longest packet. Once again, Strawman is therefore not bound to any specific radio modulation or encoding technique.

# 7. CONCLUSIONS

We present Strawman, a new contention resolution mechanism for low-power wireless networks. Strawman leverages synchronized packet collisions to implement efficient and fair contention resolution among hidden terminals.

Strawman is designed for low-power networks that experience traffic bursts, but is activated only upon detecting data collisions and has zero overhead unless needed. We have implemented and evaluated Strawman in a receiver-initiated protocol, where it replaces the traditional backoff-based mechanism. Our testbed experiments show that without Strawman, a small number of hidden terminals may drastically degrade performance during traffic bursts whereas Strawman enables high throughput even in the presence of a large number of hidden contenders. Lastly, we show that Strawman can reduce the radio duty cycles in data collection networks, both when the traffic is regular and when sudden traffic bursts occur.

# 8. ACKNOWLEDGEMENTS

This work was partially supported by VINNOVA, the Swedish Agency for Innovation Systems, by the Swedish Foundation for Strategic Research, by the EU Commission and by CONET, the Cooperating Objects Network of Excellence.

# 9. REFERENCES

[1] M. Anwander, G. Wagenknecht, T. Braun, and K. Dolfus. Beam: A burst-aware energy-efficient adaptive mac protocol for wireless sensor networks. In *International Conference on Networked Sensing Systems (INSS)*, 2010.

[2] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5), Dec. 2004.

[3] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. Sensorscope: Out-of-the-box environmental monitoring. In *ACM/IEEE IPSN*, 2008.

[4] V. Bharghavan, A. Demers, S. Schenker, and L. Zhang. MACAW: a Media Access Protocol for Wireless LANs. In *ACM SIGCOMM*, London, UK, 1994.

[5] C. A. Boano, Z. He, Y. Li, T. Voigt, M. Zuniga, and A. Willig. Controllable Radio Interference for Experimental and Testing Purposes in Wireless Sensor Networks. In *IEEE SenseApp*, Zurich, Switzerland, Oct. 2009.

[6] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *ACM SenSys*, Boulder, Colorado, USA, 2006.

[7] N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *ACM/IEEE IPSN*, Cambridge, Massachusetts, USA, 2007.

[8] M. Demirbas, O. Soysal, and M. Hussain. Singlehop Collaborative Feedback Primitive for Wireless Sensor Networks. In *IEEE INFOCOM*, 2008.

[9] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis. Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless. In *ACM SenSys*, Zurich, Switzerland, Nov. 2010.

[10] A. El-Hoiydi, J.-D. Decotignie, C. C. Enz, and E. L. Roux. Poster Abstract: WiseMAC, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Network. In *ACM SenSys*, 2003.

[11] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. In *Proceedings of the International Conference on Information Processing in Sensor Networks (ACM/IEEE IPSN)*, Chicago, IL, USA, April 2011.

[12] E. Ghadimi, P. Soldati, F. Österlind, H. Zhang, and M. Johansson. Hidden terminal-aware contention resolution with an optimal distribution. In *The Eighth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2011.

[13] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. *SIGCOMM Comput. Commun. Rev.*, 38(4), 2008.

[14] V. Handziski, A. Köpke, A. Willig, and A. Wolisz. TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality (REALMAN'06)*, 2006.

[15] P. Hurni and T. Braun. Maxmac: A maximally traffic-adaptive mac protocol for wireless sensor networks. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN)*, Coimbra, Portugal, Feb. 2010.

[16] P. Jacquet, P. Minet, P. Mühlethaler, and N. Rivierre. Priority and collision detection with active signaling - the channel access mechanism of hiperlan. *Wireless Personal Communications*, 1997.

[17] K. Jamieson, H. Balakrishnan, and Y. C. Tay. Sift: a MAC Protocol for Event-Driven Wireless Sensor Networks. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN)*, Zurich, Switzerland, Feb. 2006.

[18] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. Flush: A reliable bulk transport protocol for multihop wireless networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Sydney, Australia, Nov. 2007.

[19] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, M. Durvy, J. Vasseur, A. Terzis, A. Dunkels, and D. Culler. Beyond Interoperability: Pushing the Performance of Sensornet IP Stacks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Seattle, WA, USA, 2011.

[20] K. Leentvaar and J. Flint. The capture effect in FM receivers. *Communications, IEEE Transactions on*, 24(5):531–539, 1976.

[21] 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 USENIX Symposium on Networked Systems Design & Implementation (NSDI)*, Mar. 2004.

[22] J. Lu and K. Whitehouse. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009.

[23] B. Namboodiri and A. Keshavarzian. Alert: An adaptive low-latency event-driven mac protocol for wireless sensor networks. In *ACM/IEEE IPSN*, St. Louis, USA, Apr. 2008.

[24] F. Österlind and A. Dunkels. Approaching the maximum 802.15.4 multi-hop throughput. In *HotEmnets*, June 2008.

[25] F. Österlind, N. Wirström, N. Tsiftes, N. Finne, T. Voigt, and A. Dunkels. StrawMAN: Making Sudden Traffic Surges Graceful in Low-Power Wireless Networks. In *Proceedings of the Workshop on Hot Topics in Embedded Networked Sensor Systems (HotEmnets)*, Killarney, Ireland, June 2010.

[26] N. Pereira, B. Andersson, and E. Tovar. WiDom: A dominance protocol for wireless medium access. *IEEE Transactions on Industrial Informatics*, 3(2):120, 2007.

[27] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Baltimore, MD, USA, 2004.

[28] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Proceedings of the International Conference on Information Processing in Sensor Networks (ACM/IEEE IPSN)*, Los Angeles, CA, USA, Apr. 2005.

[29] B. Raman, K. Chebrolu, S. Bijwe, and V. Gabale. PIP: A Connection-Oriented, Multi-Hop, Multi-Channel TDMA-based MAC for High Throughput Bulk Transfer. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Zürich, Switzerland, 2010.

[30] M. Ringwald and K. Römer. BitMAC: A Deterministic, Collision-Free, and Robust MAC Protocol for Sensor Networks. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN)*, Istanbul, Turkey, Jan. 2005.

[31] M. Ringwald and K. Römer. Burstmac - an efficient mac protocol for correlated traffic bursts. In *International Conference on Networked Sensing Systems (INSS)*, 2009.

[32] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz. ESRT : Event-to-Sink Reliable Transport in Wireless Sensor Networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing (MobiHOC 2003)*, 2003.

[33] J. L. S. Sobrinho and A. S. Krishnakumar. Real-Time Traffic over the IEEE 802.11 Medium Access Control Layer. In *Bell Labs Technical Journal*, 1996.

[34] K. Srinivasan, M. Kazandjieva, S. Agarwal, and P. Levis. The $\beta$-factor: measuring wireless link burstiness. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Raleigh, NC, USA, 2008.

[35] Y. Sun, O. Gurewitz, and D. Johnson. RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Raleigh, NC, USA, 2008.

[36] F. A. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sensing Multiple Access and Busy Tone Solution. In *IEEE Trans. on Commun.*, volume 23. IEEE, 1975.

[37] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting The Capture Effect For Collision Detection And Recovery. In *Proceedings of the IEEE Workshop on Embedded Networked Sensor Systems (IEEE Emnets)*, Sydney, Australia, May 2005.

[38] A. Woo and D. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the International Conference on Mobile Computing and Networking (ACM MobiCom)*, Rome, Italy, 2001.

[39] X. Yang and N. Vaidya. On physical carrier sensing in wireless ad hoc networks. In *IEEE INFOCOM*. IEEE, 2005.

[40] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, New York, NY, USA, June 2002.