Flip-MAC: A Density-Adaptive Contention-Reduction Protocol for Efficient Any-to-One Communication

Doug Carlson
Department of Computer Science
Johns Hopkins University
Baltimore, Maryland 21218
Email: carlson@cs.jhu.edu

Andreas Terzis
Department of Computer Science
Johns Hopkins University
Baltimore, Maryland 21218
Email: terzis@cs.jhu.edu

***Abstract*—As Wireless Sensor Network (WSN) deployments increase in density, we foresee a need to divide the problem of Medium Access Control into two steps: the first quickly reduces the level of contention from hundreds of nodes to a handful, while the second relies on existing techniques to select a single node for communication.**

In this paper, we propose the Flip-MAC mechanism, which takes advantage of non-destructive interference from multiple transmitters to reduce contention to manageable levels in logarithmic time with respect to the initial contention level. We present a description of this protocol and its intended usage, a theoretical description of its operation, simulation results, and an evaluation based on a testbed deployment with actual mote hardware.

I. Introduction

Many wireless sensor networks (WSNs) are deployed in dense patterns, where a single node can have dozens of radio neighbors [1]. The advent of long-range radios is likely to create even denser deployments, with hundreds of nodes within radio range of each other.

In this paper, we consider the case of a dense network of sensing nodes that continuously record high-resolution data, most of which is "uninteresting." Periodically, some event occurs which is detected by multiple nodes. When this happens, an infrastructure node wants to quickly become aware of it and retrieve the high-resolution data from *any* of the detectors which experienced the event. Namboodiri et al. described a similar use case in the context of a home security monitoring application, while we add the assumption that it may not be necessary to service all of the nodes with data to send [2]. Due to the relative rarity of events, we would prefer to use a non-scheduled MAC protocol to handle the channel negotiation process, which generally will have lower maintenance costs than a scheduled protocol.

Rather than proposing a complete MAC protocol to fit this use case, we propose dividing the problem into two steps: the first quickly reduces contention to a manageable level, and the second uses an existing non-scheduled MAC protocol to select a single sender.

In Flip-MAC, our goal is to quickly reduce the level of contention from many senders to a handful. The process we follow is analogous to all nodes performing a series of coin-flips, where the senders whose coins fail to match the receiver's simultaneously exit the competition. The competition ends when no senders match the receiver (as this indicates that relatively few senders remain).

This process is realized through a series of probe-acknowledgement cycles. During the main negotiation process, the receiver sends a probe to one of two possible 802.15.4 addresses, while each eligible sender temporarily sets their ID to one of these two addresses. These choices are made randomly at each node. The eligible senders which "guessed correctly" (by setting their ID to the destination of the probe) send simultaneous acknowledgements and make their selection for the next cycle. When one of these probes goes unacknowledged, the receiver assumes that none of the remaining senders made a matching selection, which indicates that only a few senders remain in negotiation. At this point, we revert to a simple CSMA/backoff scheme and initiate the main data transfer with *any* one of the remaining nodes.

Most Medium Access Control (MAC) protocols make the assumption that collisions between transmitters are solely destructive. It has been demonstrated that this is not always the case [3], [4]. In Flip-MAC, the concurrent transmissions are limited to hardware-generated acknowledgements provided by the CC2420 [5] transciever. These are implicitly synchronized, and produced in a highly deterministic fashion, so they interfere constructively. While Flip-MAC is designed with the CC2420 in mind, it makes use of principles which should apply to any 802.15.4-compliant transciever.

Flip-MAC has three main benefits. First, negotiation takes logarithmic time with respect to density, allowing the protocol to scale to extremely high levels of initial contention. Second, final contention levels are low and largely independent of the initial contention level. Finally, the incorporation of several two-way communications in negotiation helps to bias sender selection in favor of good links: in our intended use case, any sender is equally valid, so this is a positive outcome.

Our testbed evaluation demonstrates that Flip-MAC works effectively with up to 44 contending senders, completing negotiation successfully more than 95% of the time. These experiments also show an increase in median negotiation time from one round to five rounds as density increases from one sender to 44, which agrees well with the expected ideal logarithmic performance. Simulation results show that Flip-MAC should perform well in large networks, effectively reducing the median final contention by an order of magnitude even with low-PRR links (35%).

The rest of this paper is organized as follows. In Section II, we give a brief survey of related systems and MAC protocols, highlighting the important differences of Flip-MAC from them. In Section III, we provide a detailed description of the operation and expected behavior of Flip-MAC. We evaluate Flip-MAC's performance in Section IV. In Section V we propose some extensions to this research, and we offer concluding remarks in Section VI.

II. Related Work

There have been a staggering number of papers written about MAC protocols. Rather than attempt to describe all of them, we wish to point out three protocols which are evocative of Flip-MAC: for a comprehensive overview, see [6].

Receiver-initiated MAC protocols were perhaps most famously explored in RI-MAC [7]. Nodes with data to send wait until the intended recipient sends a beacon frame. All waiting senders respond with data, and if a collision occurs, the receiver sends another beacon with a duration specified. Senders attempt to retransmit their data at some random point in this window.

A-MAC [3] is another receiver-initiated MAC protocol, but it takes advantage of the fact that hardware-generated 802.15.4 acknowledgement frames can be sent and processed much more quickly than full 802.15.4 data frames to reduce wasted idle-listening time. A-MAC also takes advantage of the fact that hardware acknowledgments collide constructively to "robustly distinguish the case of zero replies (indicating no pending traffic) from one or more replies (indicating pending traffic)." The authors demonstrate impressive gains in current consumption over RI-MAC and Low-Power Listening (LPL) [8], but point out that limitations remain, particularly in the realm of dense networks. Additionally, their method is only evaluated up to densities of four contending senders (though they show that acknowledgements are still readily decodable with 94 simultaneous senders). Like any protocol which relies on detecting collisions and backing off, as contention increases, backoff times suffer. We see A-MAC and Flip-MAC coexisting well together, as they leverage the same communication primitives.

The StrawMAN [9] system is reminiscent of Flip-MAC, in that they both allow multiple transmitters to coexist and make use of randomized negotiation. In StrawMAN, all pending senders transmit a packet of variable length, while the receiver reads the Received Signal Strength Indicator (RSSI) to determine the length of the longest transmission. The receiver then broadcasts a message containing the length of the longest sender-request packet, and that sender responds by transmitting its data. However, the authors of [3] point out that pollcast [10], implemented with a similar technique can be prone to false positives when multiple nodes perform this activity or external interference is present.

The authors of Alert [2] attempt to solve a similar problem as we do (selecting a single sender from a pool of many eligible senders), but they approach it through a randomized channel/time slot assignment that is optimized to minimize the delay of the first message and the overall delay to collect all messages. Our approach does not require any synchronization between nodes, has channel utilization dictated primarily by the receiver's latency requirements, and is simpler to analyze.

Finally, while not a MAC protocol, Glossy [4] was recently introduced to simultaneously solve the problems of efficient network flooding and time synchronization. Like Flip-MAC, Glossy relies on non-destructive concurrent packet transmissions to achieve high performance. The authors of Glossy take considerable care to transmit data packets (not acknowledgements) with very precise timing, while we use the basically "free" precision of hardware-generated acknowledgement. Their work is indicative of the nascent trend in WSN research of exploiting concurrent radio transmissions.

III. Protocol Description

Most non-scheduled MAC protocols rely on collision detection and backoffs at some point: in both A-MAC and RI-MAC, senders back off in response to collision-detection messages from the receiver, for example. When contention is very high, these backoffs can hurt the MAC protocol's performance. In the extreme case, hard-coded maximum backoff limits will cause receivers to abort when contention is too high. Flip-MAC is used to quickly reduce contention to low levels, at which point traditional MAC protocols can work effectively.

The high-level operation of Flip-MAC uses a series of probes to randomly select nodes from a pool of eligible senders. Each probe is sent to a random ID from a small range of choices, and any senders[1] which correctly guess the ID selected (by setting their ID to it temporarily) remain in negotiation. The selected senders respond with acknowledgements, which collide non-destructively and allow the receiver to tell whether nodes are still participating in negotiation. This removes some fraction of the remaining senders on each negotiation round. In this manner, we quickly reduce the number of eligible senders to the point where a basic MAC protocol will perform well. We reserve a few bits from the 802.15.4 address to allow the negotiation to take place solely through transmissions from the recipient and fast, non-interfering hardware acknowledgements from the senders.

In this section, we describe the protocol in detail and characterize its behavior under loss-free conditions.

[1] Unless otherwise noted "sender" refers to a node which ultimately wishes to send data to the "receiver" (not to be confused with the node which is transmitting a Flip-MAC control packet).

Abbreviation	0	1	2	Meaning
Not Used	0	0	0	Reserved for normal data traffic
DP	0	0	1	Is Data Pending?
NC_0	0	1	0	Negotiation choice: 0 selected
NC_1	0	1	1	Negotiation choice: 1 selected
RC_0	1	0	0	Resolution confirmation: 0 selected
RC_1	1	0	1	Resolution confirmation: 1 selected
RC_x	1	1	0	Resolution confirmation: none selected

TABLE I
PREDICATE ADDRESS PREFIXES. THESE ARE PREPENDED TO THE UNIQUE 13-BIT ID OF A NODE TO FORM A PREDICATE ADDRESS.

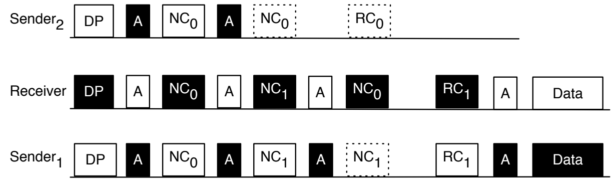


Fig. 1. Negotiation sequence. Black boxes indicate transmissions, with text indicating the PA to which a probe was sent. White boxes indicate receptions, the text within indicates the PA at which the receiver was listening. Broken lines indicate that nothing was received due to a PA mismatch.

A. Operation and Implementation

In our scheme, nodes may only use the lower 13 bits of their 16-bit 802.15.4 short ID for unique identification. The three highest-order bits are reserved for encoding "predicate addresses" (PA's) used to control Flip-MAC and select from eligible senders. Table I describes the format used.

Figure 1 demonstrates the negotiation process in detail. Nodes with data to send to a specific recipient change their 802.15.4 ID to match the "data-pending" (DP) PA of the recipient. Nodes periodically probe to their own DP PA. If a sender receives a probe on this address, they acknowledge it and randomly select one of the "negotiation-choice" (NC) PA's for the recipient and set their 802.15.4 address to this. Likewise, if a receiver gets an acknowledgement to a DP probe, they randomly select one of their NC PA's and send a probe to it. This process continues: every time that a sender receives a probe, they assign themselves to a new NC PA. Every time that the receiver gets acknowledgements, they probe to a new NC PA. In the basic case, we consider a set of two NC PA's (requiring a single bit), but a larger number could be used at the expense of consuming more ID's.

If a sender doesn't get a probe within the alloted time, they assume that they picked incorrectly, and change their address to the "resolution-confirmation" (RC) PA corresponding to their last correct choice. The sender "hopes" that it was in the last batch of "winners" and waits for immediate confirmation. If this confirmation doesn't arrive, it concludes that some other node has been selected and stops participating[2].

If the receiver gets no acknowledgements, it assumes that no nodes matched its last selection (implying that relatively few senders remain). Once this state is reached, the receiver sends a probe to the RC PA corresponding to the last acknowledged NC probe that it sent.[3] When this is acknowledged, the behavior reverts to the low-contention MAC protocol in use.

Flip-MAC was implemented in nesC [11] for TinyOS on the TelosB platform [12]. Since it's ultimately intended to coexist with another MAC protocol, great care was taken to make minimal component wiring changes: the only major logical change in the CC2420 radio stack is the replacement of the `CC2420CsmaC` component with a `CC2420FlipMacC` component.

We used a simple CSMA + exponential backoff scheme for the low-contention MAC protocol. This code, including full instrumentation for debugging and data collection, consumes approximately 24 KB of ROM and 3 KB of RAM.

B. Convergence Behavior: Loss-Free Setting

In the absence of packet loss, Flip-MAC can be characterized by answering two key questions. First, how many rounds of negotiation are required before the receiver ends the process? Second, how many senders will remain at the end of the negotiation process?

1) Negotiation Length: We start investigating negotiation length by analyzing how many senders will remain after k rounds.

For generality, we let p_{sel} equal the probability that a sender selects the same NC PA as the receiver, and we assume that the selection of NC's is independent for each node. The probability, then, of a sender selecting the same NC as the receiver k times is simply p_{sel}^k. We can therefore define the number of nodes remaining at the k^{th} round, given an initial number of senders n, with a simple binomial distribution:

$$R(n, k, p_{\text{sel}}) = B(n, p_{\text{sel}}^k)$$

The negotiation process ends when a probe is sent, but no acknowledgements are received. If m senders remain, this is equal to the probability that all m senders fail to pick the same NC PA as the receiver, and given by $P_{\text{na}}(m)$ (the probability of no-acknowledgements received from m actively-negotiating nodes):

$$P_{\text{na}}(m) = (1 - p_{\text{sel}})^m$$

Combining these, we get $P_{\text{s}}(k, n)$, the probability of stopping on the k^{th} round, when starting with an initial density of n:

$$P_{\text{s}}(k, n) = \sum_{j=0}^{n} r(j, n, k-1, p_{\text{sel}}) P_{\text{na}}(j)$$

where $r(j, n, k, p)$ is the probability mass function of $R(n, k, p)$. In plain English, the probability of stopping at the k^{th} round when starting from an initial density of n nodes is given by summing the probability of having exactly j nodes at

[2]We consider the problem of notifying senders that their data is no longer desired to be orthogonal to this work. One 3-bit PA prefix is not in use, so this could be added as a "cancel pending transmission" message in the future

[3]In the case where the DP probe was acknowledged, but not the first NC probe, a special RC_x PA is used.

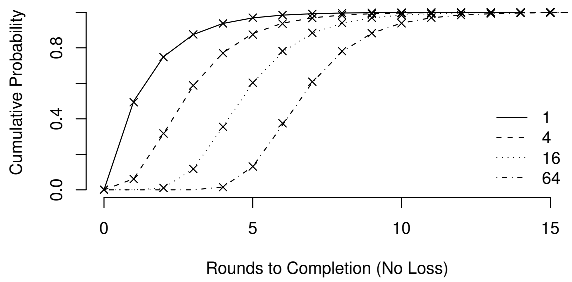


Fig. 2. CDF of rounds to completion for several initial sender densities, from simulation and definition of $P_s(k,n)$ The curves are from the analytical results, the points are from the corresponding results of 10,000 simulation trials.

the preceding round, and having all j nodes pick incorrectly, for all values of j up to the original density.

Figure 2 shows the cumulative density function that this produces for several densities, as well as the results from simulating this process directly with $p_{sel} = 0.5$. There is very good agreement between the analytical result and the simulation. Note that, as one would expect, doubling the sender density adds a single round to the negotiation length.

2) Sender Density Post-Negotiation: The negotiation process ends when none of the eligible senders select the same NC PA as the receiver. Aside from the time required to reach this point, we are also concerned with the final contention level.

The expected value of the number of senders participating in a no-acknowledgement round can be taken from our definition of $P_{\text{na}}(m)$, with initial sender density n.

$$E(n_{\text{na}}) = \sum_{i=1}^{n} i \cdot P_{\text{na}}(i)$$

This examines the probability of stopping with i nodes in contention for every i up to the maximum possible, n. The expected value converges to two as n increases with $p_{\text{sel}} = 0.5$. Simulation results bear this out.

IV. Evaluation

In this section, we seek to characterize Flip-MAC's performance under a variety of conditions. First, we evaluate Flip-MAC's behavior in absolute terms on a moderately-sized testbed. We then turn to a Python simulation in order to evaluate how correlated and uncorrelated packet losses affect Flip-MAC. Finally, we explore the beneficial positive bias which Flip-MAC exhibits in link selection at large scales.

Unless otherwise noted, all figures referred to in this section show the 10^{th}, 25^{th}, 50^{th}, 75^{th}, and 90^{th} percentiles of measurements.

A. Testbed Evaluation

1) Procedure: In order to evaluate Flip-MAC, we ran a series of testbed experiments over the course of 40 hours. One node was selected to be the receiver, and a variable number of nodes were selected to act as senders. Every two seconds, the receiver sent a DP probe, and all senders attempted to send a data packet to it. When each round was finished, every node logged the results to the serial port: their final status, timestamps from each packet reception/transmission, and the RSSI/LQI of the last packet which they received in the round.

The receiver was set to a centrally-located node[4] on a two-floor testbed inside of a university academic building, consisting of 60 nodes total. We ran multiple 15-minute experiments with progressively larger sections of the testbed: only the closest node, the nodes in the same room, the nodes on the same hallway, the nodes on the same floor, and finally the nodes on the entire testbed. The maximum possible contention was 59. However, since not all nodes were in communication range, we will refer to the sets of senders by the median initial contention level observed over the experiments rather than the number of nodes which were designated as senders. The median initial contention level was 44 for the full testbed. Approximately 28,000 individual probes were sent over approximately 60 test batches.

2) Proof-of-Concept: The first question we need to answer is "Does Flip-MAC work in practice?" Figure 3 gives a high-level answer. When we aggregate results based on the portion of the testbed in use, we can see that while the initial contention rises from one to 44, we see the negotiation failure rate go from close to zero to almost 5%. These failures can be categorized as "RC Failures," where the final RC probe was not acknowledged, or "DP Failures," where the initial DP is not acknowledged.[5] We suspect that the loss increases because as we expand the pool of senders, we add progressively worse and worse links (farther away from the receiver). We note that prior work has shown that ACKs can be decoded robustly under good link conditions for up to 94 nodes [3]. By using no routing protocol and attempting to use every possible link on the testbed, we are working in a worst-case environment, but we still see more than 95% of negotiations end successfully.

Figure 4 further suggests that link quality variations contribute to negotiation failures. The negotiation failure rates of the multi-room experiments (13, 28, and 44) trace roughly similar shapes, though higher contention levels are impacted more heavily. The worst performance occurred during daytime hours on the second day of testing.

3) Contention Reduction: Flip-MAC is primarily intended to reduce contention from high levels to manageable ones. Figure 5 demonstrates its effectiveness in this task. The median final contention remains at or below two. While we see fairly long tails at the highest contention levels, we note that the 75^{th}percentile remains below seven, which is easily manageable by many MAC protocols.

Flip-MAC's expected logarithmic running time is one of its primary benefits, and Figure 6 shows rough agreement with

[4]We had intended to repeat this with different receivers, but the other nodes we tried experienced serial communication issues which made the data unusable. We discarded test results where we detected that the receiver had experienced poor serial communication.

[5]In cases where negotiations ended successfully, the simplistic CSMA protocol we used achieved a 97% delivery rate. A more sophisticated protocol, or one that included retransmissions, would likely do better.

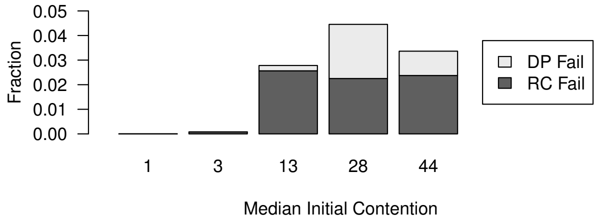


Fig. 3. As the initial contention increases, we see an increase in negotiation failures. However, non-acknowledged DPs (which are most impacted by initial contention) account for only 2.2% of outcomes in the worst case.

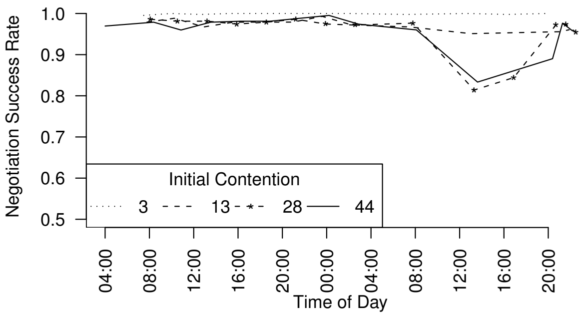


Fig. 4. Testbed delivery rate as a function of time, for the initial contention levels tested. Note Y-axis scale.

the analytical results. There aren't measurements at very large scales to definitively bear out the analysis, but we can see that the median number of rounds required only rises from one to five as we increase the contention from 1 to 44 nodes.

4) Time Overhead: In order to determine the overhead required in negotiation, we measured the difference between when a receiver sent their first DP probe to the point where negotiation completes and nodes begin to send their data with the 32khz on-board clock. We tuned the inter-round interval to 16 milliseconds, which allows enough time for senders to consistently receive a probe, send their acknowledgement, and switch to a new predicate address in time to receive the next probe. In addition to this, approximately 1.6 milliseconds of overhead are involved in sending the initial DP probe, and the final RC probe consumes another one-round interval. From the results in Figure 6, we can expect less than 66 milliseconds of overhead in 90% of the cases where no contention is present. Our implementation attempted to keep the CC2420 radio stack as clean as possible. It's certainly possible that if the lower layers of the stack were to be re-written with Flip-MAC in mind, it would be possible to reduce the period between probes. However, this optimization runs counter to the idea that Flip-MAC could supplement low-contention MAC protocols.

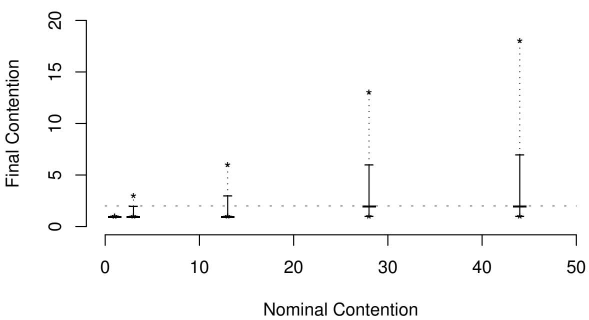


Fig. 5. Results from testbed experiments demonstrating Flip-MAC's ability to reduce contention. The median final contention remains at or below two (marked with horizontal line)

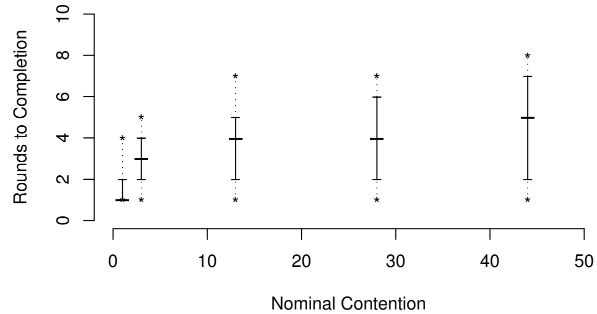


Fig. 6. Results from testbed experiments showing relationship between rounds-to-completion and initial sender density. The median rounds-to-completion only rose from 1 to 5 as we increased contention from 1 to 44 nodes.

B. Simulation

We wrote a simulator in Python to help us assess Flip-MAC's performance in a more controlled environment than the testbed provides. In the remainder of this section, we present descriptions of how the simulation indicates that Flip-MAC will behave when subjected to different types of packet loss and how this impacts sender selection bias. Unless otherwise noted, each experiment simulated 128 senders, and each set of error-bars depicted is derived from the results of 1,000 trials.

1) Correlated Loss: The first type of loss we are concerned with is correlated loss, where a probe is lost to all senders or an entire set of acknowledgements is lost to the receiver. This could occur, for example, if a burst of WiFi traffic or some other interference is present during negotiation.

If a probe is lost to all senders, no acknowledgements will be sent in response. The remaining senders from the previous step and the receiver will therefore both move on to the resolution-confirmation phase. This has the effect of shortening the negotiation process, but potentially leaves many nodes in contention when it's complete.

If all acknowledgements are lost, the situation is slightly more complicated. When this occurs, some senders receive a probe, send their acknowledgements, and move on to the next round of negotiation. However, the receiver assumes that no nodes responded to its last probe, and moves on to the resolution-confirmation stage. At this point, it's possible that nodes which *failed* at the last negotiation round are currently expecting an RC message, and nodes which succeeded are waiting for an NC message. It's also possible that no nodes are waiting for an RC message at this point. We accept that such failures can occur and will simply retry the negotiation when they do.

We investigated the impact of correlated loss by simulating the negotiation process with 128 initial senders and a correlated acknowledgement PRR which varied from 0 to 100%. Varying the correlated probe loss rate elicits a similar response, so we omit those results for brevity.

Figure 7 confirms our expectations: a low acknowledgement PRR leads to very short negotiations, but can potentially leave very high levels of contention. We leave it to future work to make Flip-MAC more resilient to correlated losses. This could be achieved in practice by incorporating retries into the negotiation process, or allowing the low-contention MAC protocol to coordinate quickly restarting negotiations with Flip-MAC if contention is too high to work effectively.

2) Uncorrelated Loss: The second type of loss we are concerned with is uncorrelated loss, where each sender's link experiences loss events independent of the others (due to different distances, sources of interference, etc).

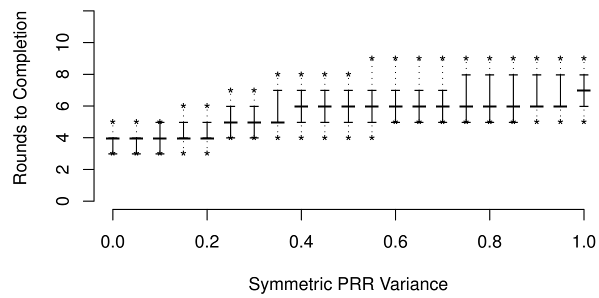
The impact of probe loss is intuitively simple to understand in this setting: in order for a sender to acknowledge an NC probe and continue, it must choose the NC PA correctly *and* receive the NC probe. This effectively changes p_{sel} to $p_{\text{sel}} p_{\text{rs}}$, where p_{rs} is the PRR of the receiver-to-sender link. If all senders experience identical, but independent, loss rates, then this is a simple substitution in the preceding analysis. Incorporating variable loss rates between senders, however, greatly complicates the analysis without providing much new insight, so we examine this through simulation.

The impact of uncorrelated acknowledgement loss is also fairly straightforward: if *any* acknowledgements reach the receiver, *there is no impact at all*. A node which sends a lost acknowledgement proceeds to the next negotiation round, and since the receiver gets *some* acknowledgements, it proceeds as well.

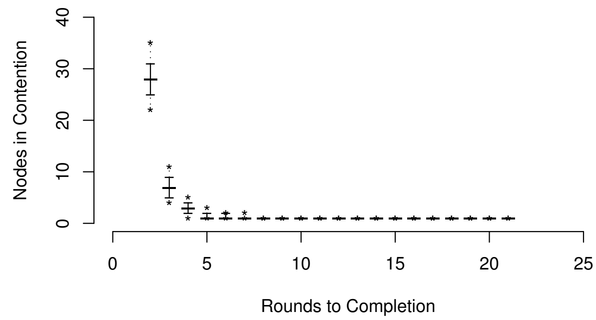
To investigate how uncorrelated losses affect Flip-MAC, we first simulated its operation on 128 senders, each with the same PRR (both for probes and acknowledgements). Figure 8 shows that Flip-MAC is more resilient to uncorrelated loss than it is to correlated loss: negotiations tend to last longer, and as long as PRR is not uniformly terrible, we ultimately end up with very little contention. This makes sense, as a single correlated loss event can stop the entire process, but the probability of many uncorrelated losses occurring simultaneously is much lower.

In practice, it's not likely that each sender will experience the same uncorrelated PRR. In Figure 9, we run the same simulation just described, but we draw each sender's PRR from a Gaussian distribution with mean 0.5 and a variable variance[6]. When variance is low, the process completes quickly and leaves a large number of nodes in contention. This makes sense: in the extreme case, where variance is 0, every node has an effective p_{sel} of 0.25. As we increase the variance, we generate more nodes with "good" PRR: these are able to remain in negotiation longer. When there is a wide enough margin between "good" and "bad" nodes, the "good" nodes can keep negotiation running long enough for the "bad nodes" to all drop out. The next section digs deeper into this effect.

[6]Sample PRR's drawn with value less than 0 or greater than 1.0 were set to 0 and 1.0, respectively.



(a) Nodes with high PRR tend to last longer in negotiations than those with low PRR. Increasing the variance gives us more of these.



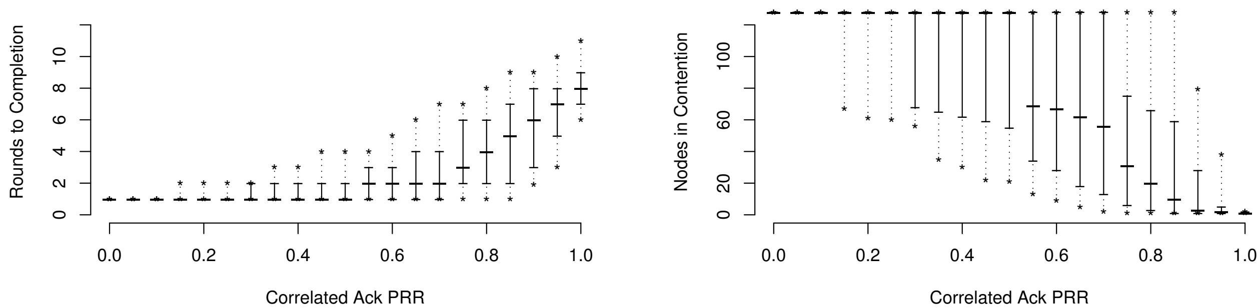
(b) If there are enough "good" nodes to keep negotiations running for a few rounds, the final contention remains low. Short negotiations, caused by poor links, lead to high contention.

Fig. 9. Probe and ack PRR are equal for each sender, drawn from a Gaussian distribution with mean 0.5.

3) Sender Selection Bias: We suggested above that variable PRRs lead to different outcomes for senders with good links from senders with bad links. Our evaluation on the testbed showed a slight preference for nodes with higher RSSIs, but without instantaneous ground-truth PRR measurements, the connection to actual link quality is a little tenuous. However, we can simulate this effect with higher numbers of nodes and finely controlled PRRs.

In Figure 10, we can observe a positive selection bias for nodes with good links. We use the same dataset used in Figure 9, but we look at the rank of the selected sender when the nodes are ordered by PRR. A value of 128 indicates that the selected node had the best PRR, and 1 indicates it had the worst.

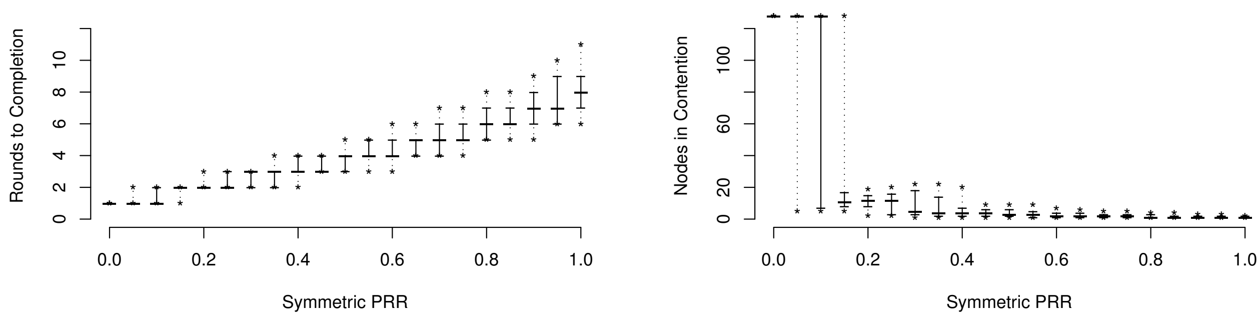
The effect is striking: as long as there is a modest degree of variability in links, Flip-MAC is strongly biased in favor of selecting good links. The accumulated effect of several loss events is sufficient to weed out the worst links. Keep in mind that these effects manifest over the span of less than 12 negotiation rounds for the most part (Figure 9): if sender density is higher, there will be more time for this effect to



(a) Losing all acks stops negotiation immediately. When ack PRR is low, most negotiations do not progress very far.

(b) If acks are not reliable (e.g. due to persistent interference at the receiver), Flip-MAC is not able to effectively reduce the level of contention.

Fig. 7. Impact of correlated acknowledgement loss on performance.



(a) When link quality is poor, negotiation time is low: the receiver stops the process at the first round where no acks are received.

(b) "Early stopping" leaves many nodes in contention, as few eliminations occur.

Fig. 8. Impact of symmetric packet loss on duration and final contention. All senders experience the same bi-directional PRR.

work. Since this effect will be stronger as negotiation length increases, it's not terribly surprising that the (relatively-short) negotiations observed on our testbed didn't turn up a dramatic bias.

The effect of a single lost acknowledgement is very different from the effect of a single lost probe from the sender's perspective (the former may have no effect at all, while the latter instantly disqualifies them). This naturally raises the concern that when links are not symmetric, Flip-MAC will be biased in favor of good receiver-to-sender links, but not necessarily good sender-to-receiver links. Figure 11 explores this. For this simulation, we selected probe PRR and ack PRR independently from a Gaussian distribution with mean 0.5 and variable variance. This shows that while the process is somewhat biased towards links with better receiver-to-sender connectivity than sender-to-receiver connectivity, the median asymmetry is well below the variance of the underlying distribution. The simulation picks the "winner" at random from the pool of nodes which remained in contention at the end of negotiation, so in practice this effect will likely be mitigated by the fact the receiver is more likely to get a data packet from a node with a good sender-to-receiver link than from a node with a poor one.

V. Future Work

There are several promising avenues for expansion in Flip-MAC.

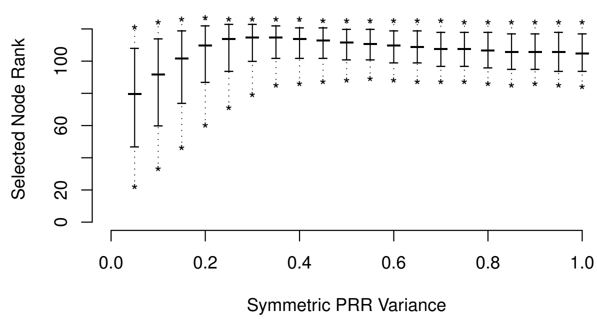


Fig. 10. Simulation of 128 senders with mean symmetric PRR of 0.5. A modest degree of variance in PRRs is sufficient to consistently select one of the best nodes in this case. The Y-value indicates the rank of the selected sender's probe PRR (128 is the best, 1 is the worst).

In real deployment scenarios, we may potentially need to deal with concurrent negotiations. In order to reduce conflict between these, it would be good to include a channel-switching mechanism. In this manner, the DP probes would occupy a single control channel, while the negotiation could take place on a separate node-specific channel. This takes the scheme proposed in A-MAC one step further. By moving both data transfer and the lion's share of control traffic away from the most-frequently used channel, we aim to isolate receivers from each other.

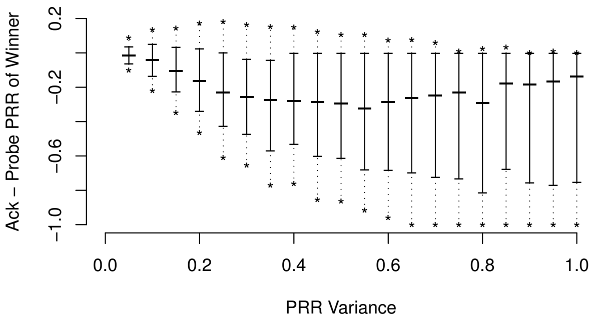


Fig. 11. Simulation of 128 senders with independently chosen probe and ack PRRs (mean 0.5). The Y-value indicates Ack PRR - probe PRR: a value of 0 is a symmetric link, a negative value indicates that the probes have a higher PRR than the acks.

Currently, we use a fixed p_{sel} during the negotiation. However, if the initial level of contention is known ahead of time to be very high, it could be advantageous to start with a lower p_{sel} (say, 0.25 instead of 0.5 on the first round) in order to reduce contention more quickly. While this carries a risk of ending with higher final contention if it is carried through the entire negotiation, it could be a good tradeoff in the early rounds. We don't currently make use of the RSSI measurements which the CC2420 provides us for received acknowledgements, but these could potentially be used to estimate the level of contention (more senders leads to more acks, which will have higher RSSI).

Perhaps the most exciting idea which Flip-MAC inspires is the concept of structuring MAC protocols into multiple phases. It would be an interesting engineering problem to design a new and more modular radio stack which promotes this viewpoint. One could imagine adaptively modifying the DP probe frequency in response to usage, adjusting initial back-offs in the low-contention MAC protocols, and other interactions between modules.

VI. CONCLUSION

In this paper, we have demonstrated the promise of Flip-MAC through analysis, simulation, and testbed experiments. Our testbed results suggest that Flip-MAC can work in practice, while simulation and analysis lead us to expect good performance as network density increases. The simulation indicates that not only is Flip-MAC robust to the introduction of poor links in the mix of initial senders, it compensates for this by favoring good links.

In the future, we see an exciting line of new research in combining well-structured MAC protocol modules to suit the needs of different deployment scenarios.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their help in improving the quality of this paper. This work is partially supported by the National Science Foundation under grants #0754782 and #0546648.

REFERENCES

[1] J. Liu, B. Priyantha, F. Zhao, C.-J. M. Liang, Q. Wang, and S. James, “Towards Fine-Grained Data Center Cooling Monitoring using RACNet,” in *HotEmNets*, 2008.

[2] V. Namboodiri and A. Keshavarzian, “Alert: An adaptive low-latency event-driven mac protocol for wireless sensor networks,” in *Proceedings of the 7th international conference on Information processing in sensor networks*, ser. IPSN ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 159–170. [Online]. Available: http://dx.doi.org/10.1109/IPSN.2008.23

[3] 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 *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Nov 2010, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1869983.1869985

[4] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, “Efficient network flooding and time synchronization with glossy,” in *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN)*. Chicago, IL, USA: ACM/IEEE, 2011.

[5] Texas Instruments, “CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver,” Available at Available at http://www.ti.com/lit/gpn/cc2420, 2006.

[6] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, “Mac essentials for wireless sensor networks,” *Communications Surveys Tutorials, IEEE*, vol. 12, no. 2, pp. 222 –248, 2010.

[7] Y. Sun, O. Gurewitz, and D. B. Johnson, “RI-MAC: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks,” in *Proceedings of the 6^{th} International Conference on Embedded Networked Sensor Systems (SenSys)*, Nov 2008, pp. 1–14.

[8] M. Buettner, G. Yee, E. Anderson, and R. Han, “X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks,” in *Proceedings of the 4^{th} ACM SenSys Conference*, 2006.

[9] F. Osterlind, N. Wirstrom, N. F. Nicolas Tsiftes, T. Voigt, and A. Dunkels, “Strawman: Making sudden traffic surges graceful in low-power wireless networks,” in *HotEmNets*, 2010.

[10] M. Demirbas, O. Soysal, and M. Hussain, “A singlehop collaborative feedback primitive for wireless sensor networks,” in *Proceedings of the 27^{th} IEEE Conference on Computer Communications (INFOCOM)*, 2008.

[11] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, “The nesC Language: A Holistic Approach to Networked Embedded Systems,” in *Proceedings of Programming Language Design and Implementation (PLDI) 2003*, Jun. 2003.

[12] MoteIV Corporation, “Tmote Sky,” Available at: http://www.sentilla.com/moteiv-endoflife.html.