# Wireless ACK Collisions Not Considered Harmful

Prabal Dutta[†] Răzvan Musăloiu-E.[‡] Ion Stoica[†] Andreas Terzis[‡]

[†]*Computer Science Division*
*University of California, Berkeley*
*Berkeley, California 94720*

[‡]*Computer Science Department*
*Johns Hopkins University*
*Baltimore, Maryland 21218*

## ABSTRACT

We present an acknowledged anycast primitive that allows a node to wirelessly transmit a packet and efficiently determine that at least one neighbor successfully received it. The initiator transmits a single packet to a unicast, multicast, or broadcast address and all nodes that match the destination respond with identical acknowledgment packets automatically generated by the hardware. Although these acknowledgments interfere, they usually do so non-destructively, so the initiator can decode their superposition. We call such an exchange a *Backcast* and show that this operation is feasible using a commodity radio, general because it enables multiple network services, efficient because it is independent of the neighborhood size and runs in constant time, and scalable because it works with no fewer than a dozen interfering acknowledgments.

## 1 INTRODUCTION

Anycast is a fundamental and widely used communications primitive that allows a node to send data to any one of several potential recipients. One challenge with providing an *acknowledged anycast* service efficiently is that the initiator may not know *a priori* which neighbors, if any, would acknowledge a transmitted packet. The initiator generally has two options in this case. One option is to contact neighbors sequentially, assuming that they are even known in advance. Unfortunately, this approach is inefficient since it scales poorly with node density. The other option is to contact all neighbors at once, perhaps using a link layer multicast or broadcast address. This approach is confronted with the well-known ACK implosion problem in which a potentially arbitrary number of neighbors can result in an arbitrary number of replies. Wireless networks further exacerbate this problem because hidden terminals can lead to collisions that corrupt packets, reduce bandwidth, and waste energy.

Imagine, however, if acknowledged anycast could be implemented efficiently: an initiator would transmit a single packet to a multicast or broadcast address, all nodes that match the destination address would acknowledge the packet concurrently, and the initiator would correctly decode the superposition of multiple acknowledgments to learn that at least one node received the packet despite the obvious ACK collisions. We term such an exchange a *backcast* and suggest that it could offer a wireless Boolean OR service abstraction: a node could pose a *true* or *false* question to its neighbors and each neighbor would vote *false* by ignoring the packet or *true* by acknowledging it. Section 2 hypothesizes how such a service could work.

Furthermore, a reliable, efficient, and scalable acknowledged anycast service would enable or improve multiple applications. For example, a low-power, network wakeup service would be possible [7]. A low-power, receiver-initiated unicast service that eliminates the long preambles common in today's low-power listening protocols would also be feasible [8]. Finally, single-hop collaborative feedback [3] would benefit from the OR semantics of acknowledged anycast. Section 3 discusses these and other backcast applications.

Section 4 explores the veracity of our thesis – that an acknowledged anycast service can be implemented efficiently – via a range of experiments based on the IEEE 802.15.4-compliant CC2420 radio [11]. The results show that a commodity radio can decode the superposition of at least a dozen identical acknowledgments with greater than 97% probability. These results suggest that an efficient and robust one-hop anycast service that does not suffer from ACK implosion is possible with at least the O-QPSK modulation scheme used in 802.15.4.

Our results suggest some important relationships between the signal strength and quality of acknowledgments, number of responders, and delay variation. In a controlled experiment with equal path loss and round trip times between the initiator and responders, we find that the two-responder case exhibits slightly worst signal quality and reception rates than all other cases. Section 5 discusses these results in greater details and argues that the well-known capture effect does not explain the surprisingly robust performance of backcast.

## 2 BACKCAST

A backcast is a link-layer frame exchange in which a single radio frame transmission triggers zero or more acknowledgment frames that interfere non-destructively at the initiator. Figure 1 illustrates a backcast exchange involving three nodes. The two responders have their radios configured to automatically acknowledge any received frames. The backcast exchange begins with the initiator transmitting a probe frame to the hardware broadcast address. Both responders receive the probe and they both transmit *identical* acknowledgments. Although these two acknowledgments collide at the initiator, as long as certain conditions are met, this collision is non-destructive, allowing the initiator to correctly decode the acknowledgment frame and conclude that at least one of its neighbors responded.

In addition to the broadcast address, a backcast probe can be sent to a multicast or unicast address, to which only a subset of the initiator's neighbors might respond. The choice of the destination address of a backcast probe depends on the radio's capabilities as well as the needs of the communications service using backcast. For example, the hardware broadcast address might be appropriate when waking up an sleeping network while a unicast address would be appropriate for communications with a single node.

The key to a successful backcast is that ACK collisions are non-destructive. This condition can hold due to power capture if one ACK frame has a higher power than the sum of the remaining ACK frames [1], or delay capture if one ACK frame arrives some period of time before the rest [2], or message retraining capture – a "message in message" model – where the radio attempts to resynchronize mid-packet if it detects a suddenly elevated energy level [6], or trivially if the radio uses an on-off keying (OOK) modulation scheme [10].

The central hypothesis of this paper is that backcast is possible under a much wider range of conditions than what capture would predict. In particular, we hypothesize that backcast is possible using minimum shift keying (MSK) and orthogonal quadrature phase shift keying (O-QPSK) modulation schemes for certain radio designs provided that: (i) inter-symbol interference resulting from different path lengths is limited, (ii) concurrent ACK frames do not cancel each other at the physical layer, (iii) the radio can automatically generate an ACK frame with an accurate and precise turnaround time, and (iv) the superposition of multiple ACKs is semantically meaningful (e.g., the ACKs are identical). Despite this list of constraints, Section 4 shows that backcast works in practice under a range of both controlled and realistic conditions using a commodity radio.

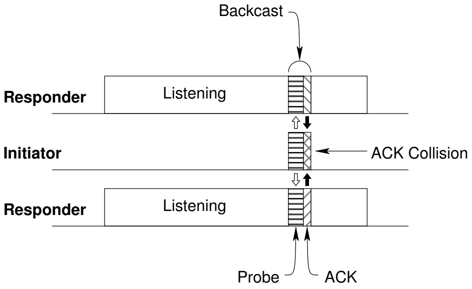

**Figure 1**: A backcast exchange involving three nodes. The backcast initiator transmits a probe frame that two responders acknowledge. Although their acknowledgments collide, they do so non-destructively, so the initiator can decode the resulting frame.

## 3 BACKCAST APPLICATIONS

In this section, we demonstrate the generality of backcast by applying it to some important network services.

### 3.1 Low-Power Asynchronous Wakeup

Waking up a multi-hop network of duty-cycled nodes is a fundamental problem in sensor networks. Applications as diverse as interactive data collection, exceptional event detection, and target tracking require nodes to wake up neighbors or even the entire network.

Dutta et al. proposed one approach to this problem [4]. In their scheme, every node periodically transmits a beacon and then briefly listens for channel activity (either a packet or increased energy). If any channel activity is detected, the node remains awake, but if no activity is detected, the node goes back to sleep. To wake up the network, the initiator listens for a time equal to the beacon period to identify all one-hop nodes. Then, during the next such period, the initiator contacts each of its one-hop neighbors in turn. These neighbors then repeat this process for the two-hop neighbors, and so on. If two or more nodes attempt to contact the same node in a lower tier, the paper conjectured that the concurrent transmissions may collide, but that the receiver would detect channel energy, remain awake, and give the transmitters a chance to enter backoff and compete.

Musăloiu-E. et al. proposed *low power probing* (LPP) as another solution to the wakeup problem [7]. According to the LPP protocol, nodes periodically broadcast short probes requesting acknowledgments. If such an acknowledgment arrives, the node wakes up and starts acknowledging other nodes' probes; otherwise it goes back to sleep. The key difference between the two approaches is that the responses in the first approach are software-generated, while LPP uses hardware acknowledgments (HACKs). What is surprising is that LPP works even if a node has many neighbors, a case in which multiple acknowledgments would collide. In fact,

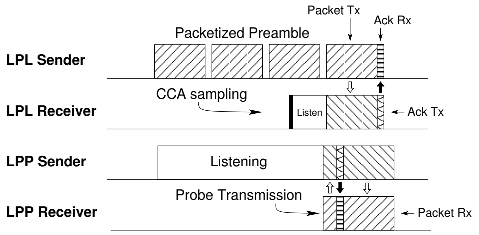

**Figure 2**: A side-by-side comparison of LPL and LPP operations. LPP replaces LPL's long preamble with listening and LPL's clear channel assessment with a backcast exchange.

LPP implicitly uses backcast to sidestep ACK implosions but the paper does not recognize this fact – something that this paper identifies.

LPP, which uses backcast, is more efficient than Dutta's proposal. The reason is that LPP does not suffer from (destructive) collisions and thus does not enter a contention phase. Furthermore, since distinguishing between collisions and other sources of noise or interference is difficult, such an approach could exhibit high false positives in practice. This observation suggests that Dutta's approach might perform poorly in dense networks deployed in interference-rich environments.

## 3.2 Low-Power Unicast

Polastre et al. proposed *low power listening* (LPL), a widely-adopted technique for low duty-cycle communications. An LPL receiver periodically checks for channel energy and stays awake upon detecting activity, while a transmitter prepends a packet with a preamble that is at least as long as the receiver's check period [8]. While LPL was designed to wake up individual nodes, LPP was designed to wake up the whole network [7]. We now come full circle by describing how LPP can be modified to wake up individual nodes and thus provide the same unicast service abstraction as LPL, while using a receiver-initiated protocol.

Directly replacing LPL with LPP, as Figure 2 illustrates, is possible yet inefficient. In the LPP protocol, a receiver transmits a probe packet to the hardware *broadcast* address and the sender responds with a HACK, causing the receiver to stay awake to receive a data packet. The problem with this approach is the sender's radio will acknowledge every probe it receives since they are sent to the broadcast address. In turn, this causes all but one of the sender's neighbors to wake up unnecessarily. Let us call this the *overreacting problem.*

LPP can be modified to avoid the overreacting problem as follows. When a sender $X$ has pending traffic for a receiver $Y$, $X$ enables its radio's hardware address recognition and sets its radio's hardware address to $Y + k$ (where $k$ is 0x8000 or 0x800000000000).

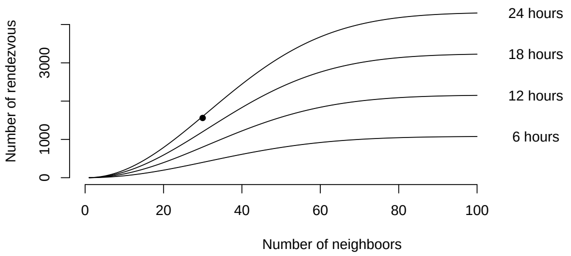

**Figure 3**: Total number of successful rendezvous, predicted by the birthday paradox, over different time intervals as a function of neighborhood size. The black dot represents the daily average number of rendezvous recorded on a 30-node testbed. In all cases the probing interval is 20 seconds and a single backcast lasts ∼20 msec.

Now, instead of broadcasting a probe, receiver $Y$ sends a probe to destination address $Y + k$, requesting a HACK. Sender $X$ (as well as any other nodes with pending traffic to $Y$) respond to the probe (multiple HACKs interfere non-destructively). If its probe is acknowledged, $Y$ remains awake to receive a packet while sender $X$ does not succumb to the overreacting problem.

## 3.3 Opportunistic Rendezvous

In the services outlined so far, backcasts are used as purely control traffic: signals to wake up nodes or alerts for inbound traffic. In this respect, backcast messages carry no application-level information. This observation raises the following question: *are there advantages for the probes to carry an application payload?* Note that acknowledgments cannot carry node-specific payloads as this would violate the requirement posited in Section 2 that acknowledgments be identical . We attempt to answer this question in two steps. First, we show that carrying a payload does not compromise backcast's feasibility or performance. We then sketch one service enabled by this extension.

To explore the first question we varied the probe's payload, from one byte up to its maximum size of 116 bytes for the CC2420 radio we use [11]. As expected the time necessary for a complete backcast operation increases linearly with the size of the probe. More importantly, a backcast carrying the maximum payload requires only ∼50% more time (31.27 msec) than one with a one byte payload (20.77 msec). The reason is that actual probe transmission corresponds to only a subset of the total time the radio is active, the rest devoted to turning the radio on and waiting for acknowledgments.

Since including application payloads generates only moderate overhead, we explore the original question through an extension to the primitive described in Section 3.1. Specifically, we augment probes to include the initiators' local clock value. Then nodes that overhear these probes can use them to perform network-wide clock synchronization (e.g., through a distributed consensus algorithm). However, since nodes keep their radios mostly off to conserve energy, this mechanism will only work if many probes are actually overheard (we term such an event, an *opportunistic rendezvous*).

Fortunately, even if a node keeps its radio on for only 20 msec during a 20 second interval (i.e., a 0.1% duty cycle), the birthday paradox works to our advantage, as Figure 3 shows. Even with few neighbors, the probability of a rendezvous is non-negligible. Furthermore, because nodes send frequent backcasts, the contact probability accumulates over time, resulting in numerous rendezvous in the span of a few hours.

### 3.4 Robust Pollcast

Demirbas et al. recently proposed *pollcast*, a two-phase primitive in which a node broadcasts a poll about the existence of a node-level predicate $P$ and then all nodes for which $P$ holds reply simultaneously [3]. The poller detects one or more positive answers by reading its radio's Clear Channel Assessment (CCA) signal. While pollcast offers a novel approach for quickly calculating predicates, the proposed mechanism has some drawbacks, as the paper acknowledges: simultaneous pollcasts within a two-hop neighborhood would cause false positives as would interference from other networks.

Backcast provides a more robust primitive for implementing pollcast, which in turn can be used to implement the applications outlined in [3]. To leverage the backcast primitive, pollcast might be modified to first transmit the predicate, then transmit the poll, and finally listen for an acknowledgment. The predicate would be sent to the broadcast address but it would also include an ephemeral identifier chosen by the initiator. Upon receiving the predicate, and evaluating it as true, a responder would enable acknowledgments and temporarily change its hardware address to match the ephemeral identifier in the probe packet. Then, a backcast probe sent to the ephemeral identifier would trigger a response from all the nodes for which the predicate was true. The CC2420 radio supports just two hardware addresses – a 16-bit one and a 64-bit one – allowing just one or two concurrent pollcasts. Future radios could perform address decoding in parallel over dozens of addresses, perhaps using a content addressable memory.

## 4 EVALUATION

This section provides empirical evidence that backcast works with one commodity radio. These observations are based on experiments with very controlled parameters (Section 4.2), to larger, more realistic environments using a sensor network testbed (Section 4.3).

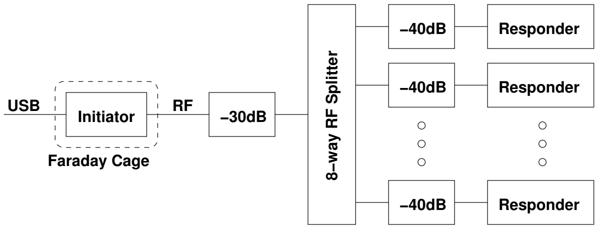

**Figure 4**: Experimental setup for the controlled tests. An initiator is connected via a 30-inch, 50 Ω RF cable and a 30 dB attenuator to the common port of an 8-way RF splitter. The other splitter ports are connected via 6-inch, 50 Ω RF cables and 40 dB attenuators to responders. A Faraday cage around the initiator limits over-the-air RF leakage.

### 4.1 Methodology

We implemented backcast in the TinyOS embedded operating system [5] and our experiments are based on the widely-used Telos mote [9] that includes the Texas Instruments CC2420, an IEEE 802.15.4 radio [11]. The 802.15.4 protocol and the CC2420 radio are ideal for demonstrating backcast because they provide the needed protocol and hardware support.

The 802.15.4 MAC defines a frame control field that includes an acknowledge request flag. If a receiver is configured for automatic acknowledgments, then an acknowledgment frame is transmitted after twelve symbol periods (192 $\mu$sec) for all incoming frames that meet three conditions: they (i) have the acknowledge request flag set, (ii) are accepted by the radio's address recognition hardware, and (iii) contain a valid CRC. Acknowledgments are transmitted without performing clear channel assessment and have the following fields: preamble, start-of-frame delimiter, length, frame control, sequence number, and frame check sequence. Notably absent from this list is a source address, ensuring that all ACKs for a given sequence number are identical.

The experiments that follow show how different responder configurations affect the acknowledgments' signal strength and quality. Signal strength is measured over the first eight symbols and reported as the received signal strength indicator (RSSI) in dBm. Signal quality (LQI) is also measured by the radio over the first eight symbols and is reported as a 7-bit unsigned value that can be viewed as the average correlation value or chip error rate.

### 4.2 Performance in a Controlled Setting

We first explore how the RSSI and LQI of acknowledgment frames are affected as the number of responders increase in a controlled setting. Figure 4 presents the setup for this experiment. Eight nodes are sequentially turned on so that the number of responders monotoni-

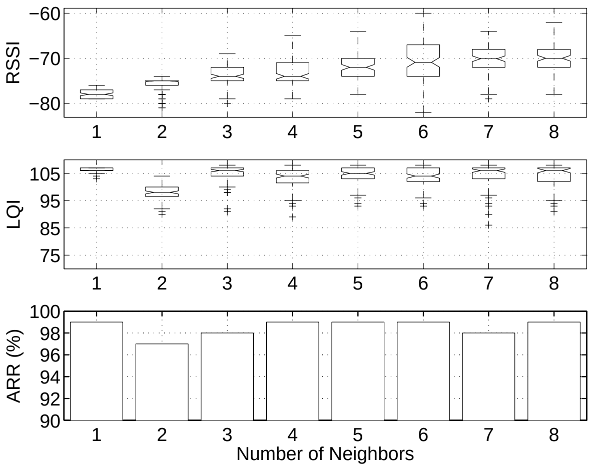

**Figure 5**: Results of the controlled experiments. The received signal strength (RSSI), link quality indicator (LQI), and acknowledgment reception rate (ARR) are shown for each trial.

cally increases from one to eight. In each of the eight trials, the initiator transmits 100 packets to the hardware broadcast address, at 125 msec intervals, and logs the RSSI and LQI values of the resulting acknowledgments. The results are shown in Figure 5 and indicate that median RSSI values increase, median LQI values stay nearly constant, both values show greater variance, and few acknowledgments are lost. Section 5 discusses these results in detail.

### 4.3 Performance in a More Realistic Setting

Next, we explore how backcast performs in a more realistic university testbed setting. The testbed is located in an office building and contains 47 Telos motes. For this experiment however, we used only 12 nodes approximately situated at the same distance from the initiator. These experiments compare the performance of hardware-generated acknowledgments (HACKs), software-generated acknowledgments (SACKs), and HACKs with randomized preamble lengths of between 3 and 16 bytes (VP-HACKs) that start at the same time but may end at different times. HACKs are automatically generated by the radio, while SACKs require the host processor to intervene, introducing considerable delay and jitter. We introduce VP-HACKs to explore how acknowledgments with smaller delay variations than SACKs interfere at the initiator. Note that while SACKs have non-uniform delays due to software processing, the VP-HACKs are all delayed by an integer multiple of the symbol time (composed of 32 chips) and the symbols themselves are orthogonal codes.

In each experiment, 500 packets are transmitted at 125 msec intervals. This procedure generates a gradual increase in the number of colliding ACK frames. The LQI and acknowledgment reception rates are shown in Figure 6. The results show that HACK and SACK LQI values exhibit higher variance and volatility as responders increase. Both HACKs with random preambles and SACKs exhibit quickly decreasing LQI and ARR values, while HACKs incur practically no loss. Section 5 discusses these results in detail.

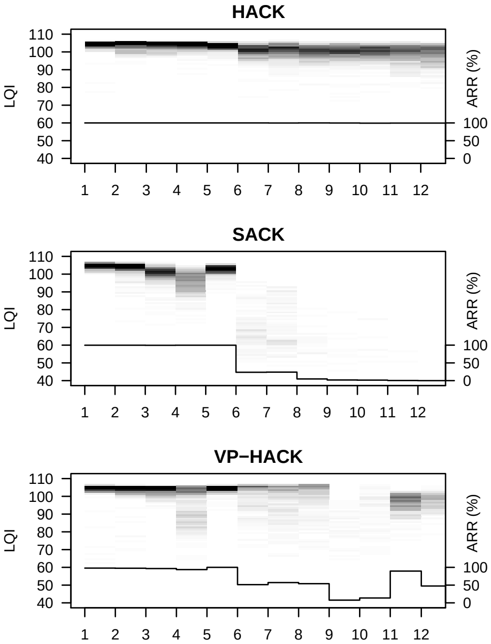

**Figure 6**: Backcast in a realistic environment, using hardware and software acknowledgments. The data are shown as a two-dimensional histogram; darker areas indicate a higher density of LQI samples. The lower line shows the average acknowledgment reception rate (ARR).

## 5 DISCUSSION

The results from Sections 4.2 and 4.3 suggest some important relationships between signal strength and quality of acknowledgments, number of responders, and delay variation. Further analysis, described below, supports our hypothesis that the capture effect alone cannot explain the surprisingly robust performance of backcast.

First, as the number of colliding acknowledgments increases, so does the median RSSI value. This trend is not surprising since for every doubling of nodes, an additional 3 dB of power is injected into the channel (assuming nodes transmit at nearly equal power levels and are equally distanced from the initiator). What is slightly more surprising is that RSSI variance is substantial and spans a range of 10-20 dB, which is both below and above the single node case, and that the distribution of values in the two-node case has many outliers.

These results suggest that elements of both constructive and destructive interference of the carrier signal may be at play. When three or more acknowledgments collide, both the outliers and RSSI variance *decrease*, suggesting that the statistical superposition of an increasing number of signals diminishes destructive interference, possibly due to the central limit theorem.

Second, the median LQI value is largely independent of the number of nodes in the controlled setting (except for the two node case) and it shows a slight decrease in the more realistic setting (computed, but not shown). Since LQI is inversely correlated with chip error rate, the data show that most acknowledgments are decoded with relatively few chip errors, even when a dozen acknowledgments collide. The data suggest that acknowledgment collisions are rarely destructive and in most cases not particularly corrupting either. LQI values show a lower median value for two responders than they do for either one or more than two responders, suggesting once again that elements of both constructive and destructive interference of the carrier signal may be at play. The RSSI distributions are largely symmetric with few outliers but the LQI distributions are left-tailed. This observation suggests that although collisions rarely improve the chip error rate, they can make it worse.

Finally, the data show that hardware acknowledgments exhibit negligible loss rates with no fewer than twelve concurrent packets, while software acknowledgments approach very high loss rates with just six or seven concurrent acknowledgments, as well as a substantial decline in link quality with just three or four acknowledgments. In between these two extremes are the variable-length preamble HACKs (VP-HACKs). The two distinctions between SACKs and VP-HACKs are in timing and composition. First, SACKs are delayed by multiples of the CPU clock cycle since a SACK requires software processing, but a VP-HACKs are delayed by an integer multiple of the symbol time. Since the symbols are chosen from an orthogonal set, this may explain the better performance of VP-HACKs compared with SACKs, despite the fact that VP-HACKs collide more frequently and are not even identical. Since these three types of acknowledgments differ in the delay and jitter of their transmissions, we argue the capture effect alone cannot explain the surprisingly robust performance of HACK-based backcasts.

## 6 Summary

This paper shows that a standards-based commodity radio can correctly decode the superposition of up to a dozen identical acknowledgment frames. This observation suggests that an efficient and robust acknowledged anycast service that does not suffer from ACK implosions may be feasible. The ability to transmit a multicast or broadcast packet and receive an acknowledgment in constant time independent of the number of responding nodes, an exchange we call *backcast*, enables or improves a range of useful communication services.

## 7 Acknowledgments

Special thanks to David Culler, Rodrigo Fonseca, Jonathan Hui, Mike Liang, and the anonymous reviewers for their substantial and insightful feedback. This material is based upon work partially supported by the National Science Foundation under grants #0435454 (“NeTS-NR”), #0454432 (“CNS-CRI”), and #0627611 (“CAREER”), an NSF Graduate Fellowship, a Microsoft Graduate Fellowship, and generous gifts from HP, Intel, Microsoft, and Sharp.

## References

[1] J. Arnbak and W. van Blitterswijk. Capacity of slotted aloha in rayleigh-fading channels. *IEEE Journal on Selected Areas in Communications*, 5(2):261–269, Feb 1987.

[2] D. Davis and S. Gronemeyer. Performance of slotted aloha random access with delay capture and randomized time of arrival. *IEEE Transactions on Communicationss*, 28(5):703–710, May 1980.

[3] 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.

[4] P. Dutta, D. Culler, and S. Shenker. Procrastination Might Lead to a Longer and More Useful Life. In *Proceedings of HotNets-VI*, 2007.

[5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *Proceedings of ASPLOS 2000*, 2000.

[6] R. Mud, J. Boer, A. Kamerman, H. Van Driest, W. Diepenstraten, R Kopmeiners, and H. Von Bokhorst. Wireless LAN with enchanced capture provision. US Patent No. US5987033, 19919.

[7] R. Musaloiu-E., C.-J. Liang, and A. Terzis. Koala: Ultra-low power data retrieval in wireless sensor networks. In *Proceedings of the 7$^{th}$ International Conference on Information Processing in Sensor Networks (IPSN)*, 2008.

[8] J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2$^{nd}$ ACM Sensys Confence*, 2004.

[9] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. In *Proceedings of the 4$^{th}$ International Conference on Information Processing in Sensor Networks (IPSN/SPOTS)*, 2005.

[10] M. Ringwald and K. Romer. BitMAC: a deterministic, collision-free, and robust MAC protocol for sensor networks. In *Proceedings of the 2$^{nd}$ European Workshop on Wireless Sensor Networks*, 2005.

[11] Texas Instruments. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. Available at `http://www.chipcon.com/files/CC2420_Data_Sheet_1_3.pdf`, 2006.