

RT-Link: A Time-Synchronized Link Protocol for Energy-Constrained Multi-hop Wireless Networks

Anthony Rowe, Rahul Mangharam and Raj Rajkumar
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
{agr, rahulm, raj}@ece.cmu.edu

Abstract

We propose RT-Link, a time-synchronized link protocol for real-time wireless communication in industrial control, surveillance and inventory tracking. RT-Link provides predictable lifetime for battery-operated embedded nodes, bounded end-to-end delay across multiple hops, and collision-free operation. We investigate the use of hardware-based time-synchronization for infrastructure nodes by using an AM carrier-current radio for indoors and atomic clock receivers for outdoors. Mobile nodes are synchronized via in-band software synchronization within the same framework. We identify three key observations in the design and deployment of RT-Link: (a) Hardware-based global-time synchronization is a robust and scalable option to in-band software-based techniques. (b) Achieving global time-synchronization is both economical and convenient for indoor and outdoor deployments. (c) RT-Link achieves a practical lifetime of over 2 years. Through analysis and simulation, we show that RT-Link outperforms energy-efficient link protocols such as B-MAC in terms of node lifetime and end-to-end latency. The protocol supports flexible services such as on-demand end-to-end rate control and logical topology control. We implemented RT-Link on the CMU FireFly sensor platform and have integrated it within the nano-RK real-time sensor OS. A 42-node network with sub-20us synchronization accuracy has been deployed for 3 weeks in the NIOSH Mining Research Laboratory and within two 5-story campus buildings.

1. Introduction

Networks of embedded wireless nodes provide a versatile platform for applications in industrial control, surveillance and inventory tracking. For cost-effective operation, such nodes feature low-power radios requiring data to be delivered across multiple hops over the air-interface to at least one gateway. For scalable deployment, nodes must be battery-powered and hence require largely collision-free communication. Our focus is on the provision of deterministic node lifetime of several years and delivery of data with bounded end-to-end delay of a few milliseconds. These two properties enable the construction of energy-efficient and robust mesh networks for a large class of applications ranging from observation of sporadic events within sensor networks to real-time

communication within tightly-coupled control loops. An effective approach to energy-efficient service for applications with either periodic or aperiodic flows is to operate all nodes at low duty cycles so as to maximize the shutdown intervals between packet exchanges. The two fundamental challenges in delivering delay-bounded service in such networks are (a) coordinating transmissions so that all active nodes communicate in a tightly synchronized manner and (b) ensuring all transmissions are collision-free. Time synchronization is important because it tightly packs the activity of all nodes so that they may maximize a common sleep interval between activities. Furthermore, it provides guarantees on timeliness, throughput and network lifetime for end-to-end communication. Such assurances are only possible when the link is reliable and collision-free. It is therefore the responsibility of the link layer protocol to provide exclusive and interference-free access to the shared wireless channel and a mechanism to coordinate sleep intervals of all nodes.

We achieve our lifetime and latency goals through the design of a TDMA-based link layer protocol, RT-Link. Tight time-synchronized operation is facilitated through the implementation of our hardware platform, FireFly. Each FireFly node features an IEEE 802.15.4 [1] transceiver, a microcontroller, multiple sensors and several pluggable time synchronization modules. For indoors, an Amplitude Modulation (AM) carrier-current transmitter periodically broadcasts a pulse for global time synchronization. All indoor nodes employ an add-on low-power AM receiver module which detects the pulse and synchronizes the node. For outdoors, each node uses an atomic clock receiver for global time synchronization. We have successfully deployed a 42-node network for 3 weeks with sub-20us synchronization accuracy in the NIOSH Mining Research Laboratory [2] and also within an 8-story campus building with a single source for global time synchronization. RT-Link has been integrated within the nano-RK real-time sensor operating system [3] and is suitable for a wide range of sensor networking applications. Through the design and deployment of RT-Link, we identify the following four observations:

1. RT-Link offers predictable network lifetime with bounded end-to-end delay for packets between the gateway and every node.
2. Provision of global time synchronization for embedded multi-hop wireless networks is both economical and con-

venient.

3. Hardware-based time synchronization offers a robust and scalable alternative to in-band software-based schemes.
4. TDMA-based link protocols offer several flexible options over random access protocols such as logical topology control, end-to-end delay which is independent of the sampling rate and on-demand multi-rate support.

Through analysis and simulation we show that RT-Link outperforms energy-efficient protocols such as B-MAC [4] and S-MAC [5] in terms of operational lifetime, throughput and end-to-end latency. In this paper, we describe the design concepts employed within the RT-Link protocol in Section 3, its capabilities for flexible operation in Section 4, our hardware platform to achieve global time synchronization in Section 5, and analytical and experimental results in Section 6.

2. Related Work

Several MAC protocols have been proposed for low-power and distributed operation for single and multi-hop wireless mesh networks. Such protocols may be categorized by their use of time synchronization as asynchronous, loosely synchronous and fully synchronized protocols. In general, with a greater degree of synchronization between nodes, packet delivery is more energy-efficient due to the minimization of idle listening when there is no communication, better collision avoidance and elimination of overhearing of neighbor conversations. We briefly review key low-power link protocols based on their support for low-power listen, multi-hop operation with hidden terminal avoidance, scalability with node degree and offered load.

2.1. Asynchronous Link Protocols

The Berkeley MAC (B-MAC) [4] protocol performs the best in terms of energy conservation and simplicity in design. B-MAC supports carrier sense multiple access (CSMA) with low power listening (LPL) where each node periodically wakes up after a sample interval and checks the channel for activity for a short duration of 2.5ms. If the channel is found to be active, the node stays awake to receive the payload following an extended preamble. Using this scheme, nodes may efficiently check for neighbor activity. For each transmission instance, the transmitter must remain active for the duration of the receiver's channel check interval. This creates a major drawback since it forces the receiver to check the channel very often (in milliseconds) even when the event sample interval spans several seconds or minutes. For example, if an event occurs ever 20 minutes, all B-MAC receivers check the channel for activity approximately every 80ms to limit the transmitter's burst duration to 80ms [4]. This coupling of the receiver's sampling interval and the duration of the transmitter's preamble severely restricts the scalability of B-MAC when operating in dense networks and across multiple hops. B-MAC does not inherently support collision avoidance due to the hidden terminal problem and the use of RTS-CTS handshaking with LPL is inefficient because the RTS must use the extended preamble. In a multi-hop network, it is necessary to use topology-aware packet scheduling for collision avoidance. Furthermore, upon wake up, B-MAC employs CSMA

which is prone to wasting energy and adds non-deterministic latency due to packet collisions.

2.2. Loosely Synchronous Link Protocols

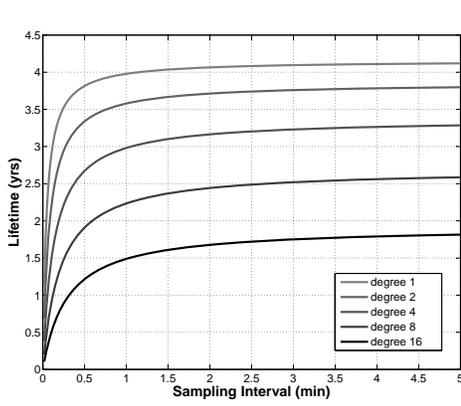
Protocols such as S-MAC [5] and T-MAC [6] employ local sleep-wake schedules known as *virtual clustering* between node pairs to coordinate packet exchanges while reducing idle operation. Both schemes exchange synchronizing packets to inform their neighbors of the interval until their next activity and use CSMA prior to transmissions. As all the neighbors of a node cannot hear each other, each node must set multiple wakeup schedules for different groups of neighbors. The use of CSMA and loose synchronization trades energy consumption for simplicity. WiseMAC [7], is an iteration on Aloha designed for downlink communication from infrastructure nodes and has been shown to outperform 802.15.4 for low traffic loads. WiseMAC, however, does not support multiple hop communication. Both T-MAC and WiseMAC use preamble sampling to minimize receive energy consumption during channel sampling. The use of CSMA in each scheme degrades performance severely with increasing node degree and traffic.

2.3. Fully Synchronized Link Protocols

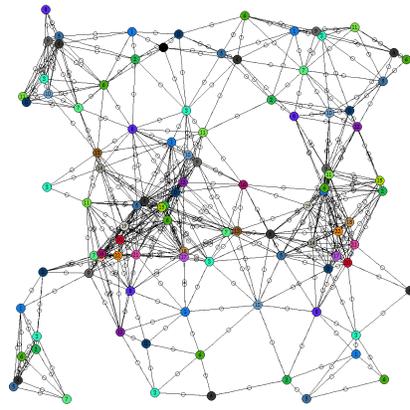
TDMA protocols such as TRAMA [8] and LMAC [9] are able to communicate between node pairs in dedicated time slots. TRAMA supports both scheduled slots and CSMA-based contention slots for node admission and network management. LMAC describes a light-weight bit-mask schedule reservation scheme and establishes collision-free operation by negotiating non-overlapping slot across all nodes within the 2-hop radius. Both protocols assume the provision of global time synchronization but consider it an orthogonal problem. RT-Link has similar support for contention slots but employs Slotted-ALOHA [10] rather than CSMA as it is more energy efficient with LPL. Furthermore, RT-Link integrates time synchronization within the protocol and also in the hardware specification. RT-Link has been inspired by dual-radio systems such as [11, 12] used for low-power wake-up. However neither system has been used for time synchronized operation. Several in-band software-based time synchronization schemes such as RBS [13], TPSN [14] and FTSP [15] have been proposed and provide good accuracy. In [16], Zhao provides experimental evidence showing that over one-third of the population of immobile nodes in an indoor environment routinely suffer a link error rate over 50% even when the receive signal strength is above the sensitivity threshold. This severely limits the diffusion of in-band time sync updates and hence reduces the network performance. RT-Link employs an out-of-band time synchronization mechanism which also globally synchronizes all nodes and is less vulnerable than the above schemes. We believe that hardware-based time sync adds new properties to wireless sensor networks and warrants exploration in a practical environment.

3. RT-Link Protocol Design

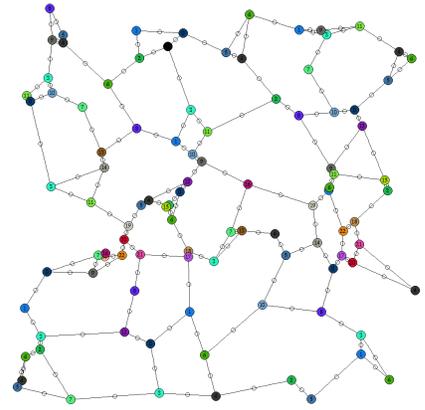
RT-Link is a TDMA-based link layer protocol designed for networks that require predictability in throughput, latency



(a) Life given increasing node degree between 1 and 16. Node degree one yields the longest life.



(b) Physical Connectivity Graph



(c) Pruned Logical Connectivity

Figure 1. Relationship between topology degree and energy necessitates the logical pruning of the network graph.

and energy consumption. All packet exchanges occur in well-defined time slots. Global time sync is provided to all fixed nodes by a robust and low-cost out-of-band channel. We now describe in detail the RT-Link protocol, packet types, supported node types and the protocol operation modes.

3.1. Protocol Overview

RT-Link supports two node types: fixed and mobile. Both node types include a microcontroller, 802.15.4 transceiver and multiple sensors and are described in detail in Section 5. The fixed nodes have an add-on time sync module which is normally a low-power radio receiver designed to detect a periodic out-of-band global signal. In our implementation, we designed an AM/FM time sync module for indoor operation and an atomic clock receiver for outdoors. For indoors, we use a carrier-current AM transmitter [17] which plugs into the power outlet in a building and uses the building’s power grid as an AM antenna to radiate the time sync pulse. We feed an atomic clock pulse as the input to the AM transmitter to provide the same synchronization regime for both indoors and outdoors. The time sync module detects the periodic sync pulse and triggers an input pin in the microcontroller which updates the local time. As shown in Figure 2, this marks the beginning a finely slotted data communication period. The communication period is defined as a fixed-length cycle and is composed of multiple frames. The sync pulse serves as an indicator of the beginning of the cycle and the first frame. Each frame is divided into multiple slots, where a slot duration is the time required to transmit a maximum sized packet. RT-Link supports two kinds of slots: Scheduled Slots (SS) within which nodes are assigned specific transmit and receive time slots and (b) a series of unscheduled or Contention Slots (CS) where nodes, which are not assigned slots in the SS, select a transmit slot at random as in slotted-Aloha. Nodes operating in SS are provided timeliness guarantees as they are granted exclusive access of the shared channel and hence enjoy the privilege of interference-free and hence collision-free communication. While the support of SS and CS are similar to 802.15.4, RT-Link is designed for operation across synchronized multi-hop networks. After an active slot is complete, the node schedules its timer to wake up just before the ex-

pected time of next active slot and promptly switches to sleep mode. In our default implementation, each cycle consists of 32 frames and each frame consists of 32 5ms slots. Thus, the cycle duration is 5.12sec and nodes can choose one or more slots per frame up to a maximum of 1024 slots every cycle. The common packet header includes a 32-bit transmit and 32-bit receive bit-mask to indicate during which slots of a node is active. RT-Link supports 5 packet types including HELLO, SCHEDULE, DATA, ROUTE and ERROR. The packet types and their formats are described in detail in [18].

3.2. Network Operation Procedures

RT-Link operates on a simple 3-state state machine as shown in Figure 3. In general, nodes operating in the CS are considered Guests, while nodes with scheduled slots are considered Members of the network. When a fixed node is powered on, it is first initialized as a Guest and operates in the CS. It initially keeps its sync radio receiver on until it receives a sync pulse. Following this, it waits for a set number of slots (spanning the SS) and then randomly selects a slot among the CS to send a HELLO message with its node ID. This message is then forwarded (via flooding if explicit routes are not present) to the gateway and the node is eventually scheduled a slot in the SS. On the other hand, when a mobile node needs to transmit, it first stays on until it overhears a neighbor operate in an SS. The mobile node achieves synchronization by observing the Member’s slot number and computes the time until the start of the CS. Mobile nodes are never made members because their neighborhood changes more frequently and

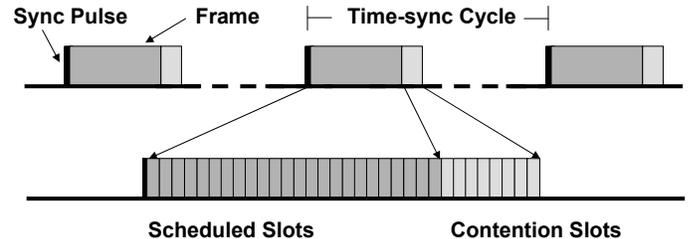


Figure 2. RT-Link time slot allocation with out-of-band synchronization pulses

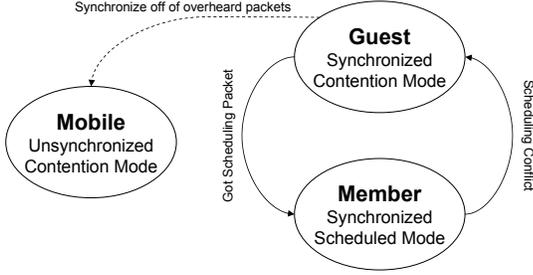


Figure 3. RT-Link node state machine.

hence remain silent until a Member provides it a time reference. All nodes with scheduled slots listen on every slot in the CS using LPL. When a node chooses to leave the network, it ceases broadcasting HELLO packets and is gracefully evicted from the neighbor list from each of its neighbors. The gateway eventually detects the absence of the departed node from each of the neighbors' HELLO updates and may reschedule the network if necessary.

For fixed nodes that are unable to receive the global time beacon and for mobile nodes, RT-Link provides software-based in-band time sync. Nodes can implicitly pass time synchronization onto another node using the current slot in the packet header. This implicit time synchronization can cascade across multiple hops.

4. RT-Link Protocol Enhancements

In this section we briefly discuss enhancements that complement the basic RT-Link protocol. These may be executed prior to or during network deployment to improve the overall throughput, end-to-end latency and network lifetime.

4.1. Logical Topology Control

RT-Link schedules communication based on the global network topology. This requires a topology-gathering phase followed by a scheduling phase. In order to acquire the network connectivity graph, we aggregate the neighbor lists from each node at the gateway. We then construct connectivity and interference graphs and schedule nodes based on k -hop coloring, such that two nodes with the same slot schedule are mutually separated by at least $k+1$ hops. Figure 1(a) shows the impact of node degree on lifetime. As the number of neighbors a node communicates with increases, the number of transmit and receive slots correspondingly increases consuming more energy. In Figure 1(b) we show the connectivity graph of a randomly generated topology with 100 nodes. The graph was colored based on the connectivity to ensure that it is free of collisions. Links can then be removed by instructing an adjacent node to no longer wakeup to listen on that particular timeslot. Using this principle we can reduce the degree of nodes while checking to maintain network connectivity. The reduced degree topology shown in Figure 1(c) reduces the average network energy by more than a factor of 3. Such logical topology control is not possible with random access protocols.

4.2. Interference-free Node Scheduling

In order to achieve high throughput in a multi-hop wireless network, it is necessary to minimize the number of collisions along each transmission hop. This problem has traditionally

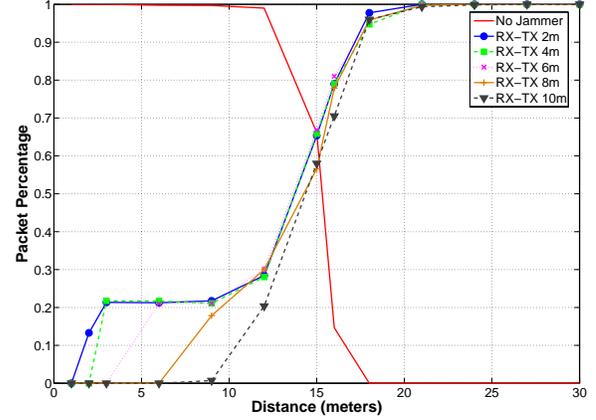


Figure 4. Packet success rate while transmitting in a collision domain. The decreasing line shows the packet success rate with respect to RX-TX distance with no jamming. The increasing lines show the packet success rate between RX-TX pairs at fixed distances as the distance between the jammer and the receiver changes.

been solved as a distance- k node coloring (slot scheduling). To determine the interference range of a node, we placed a set of nodes along a line in an open field and first measured the packet loss between a transmitter and receiver as the transmitter's distance was varied. Once the stable communication distance between a transmitter and receiver was determined, we evaluated the effect of a constantly transmitting node (i.e. a jammer) on the receiver. Our experimental results for stable transmit power level 8 are shown in Figure 4. We notice that 100% or the packets are received up to a transmitter-receiver distance of 10m. Following this, we placed the transmitter at a distance of 2, 4, 6, 8, 10 and 12 meters and for each transmitter position, a jammer was placed at various distances. At each point, the transmitter sent one packet every cycle to the receiver for 2000 cycles. We measure the impact of the jammer by observing the percentage of successfully received packets. We observe two effects of the jammer: First, the effect of the jammer is largely a function of the distance of the jammer from the receiver and not of the transmitter from the receiver. Between 12-18 meters, the impact of the jammer is similar across all transmitter distances. Second, when the transmitter and jammer are close to the receiver, (i.e. under 9m), the transmitter demonstrates a capture effect and maintains an approximately 20% packet reception rate. The above results show that the jammer has no impact beyond twice the stable reception distance (i.e. 20m) and a concurrent transmitter may be placed at thrice the stable reception distance (i.e. 30m). Such parameters are incorporated by the node coloring algorithm in the gateway to determine collision-free slot schedules. Results for a scheduled multi-hop network are presented in Section 6.

4.3. Coloring and Ordering

In multi-hop wireless networks, the goal for higher throughput has traditionally been approached from the perspective of maximizing the set of concurrent transmitters in the network [19]. This is achieved either by scheduling nodes

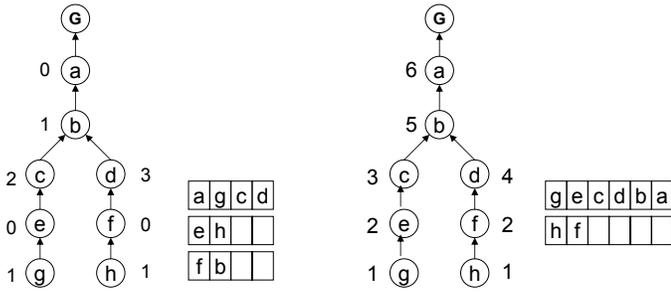


Figure 5. Maximal concurrency schedule (left) compared to a delay sensitive schedule (right). Note that the maximal concurrency schedule needs two frames to deliver all data.

or links such that they operate without any collisions. In the networks considered here, the applications generate steady or low data rate flows but require low end-to-end delay. In Figure 5 we see two schedules, one with the minimal number of timeslots, the other containing extra slots but provisioned such that leaf nodes deliver data to the gateway in a single TDMA cycle. The minimal timeslot schedule maximizes concurrent transmissions, but causes queuing delays and hence does not minimize the upstream latency of all nodes. By assigning the time slots appropriately in preference to faster uplink and downlink routes, we note that for networks with delay-sensitive data, ordering of slots should take priority over maximizing spatial reuse.

The generation of minimum delay schedules is similar to the distance-two graph coloring problem that is known to be NP-complete [20]. In practice, many heuristics can work well and result in a small constant deviation from the optimal [20]. To illustrate the minimum-delay capability of RT-Link, we briefly discuss one such heuristic to schedule a network where the traffic consists of small packets being routed up a tree to a single gateway. The heuristic consists of four steps. The first step builds a spanning tree over the network rooted at the gateway. Using Dijkstra’s shortest path algorithm any connected graph can be converted into a spanning tree. As can be seen in Figure 6(b), the spanning tree must maintain "hidden" links that are not used when iterating through the tree to ensure the 2-hop constraint is still satisfied in the original graph. Once a spanning tree is constructed, a breadth first search is performed starting from root of the tree. The heuristic begins with an initially empty set of colors. As each node is traversed by the breadth first search, it is assigned the lowest value in the color set that is unique from any 1 or 2-hop neighbors. If there are no free colors, a new color must be added into the current set. The next step in the heuristic tries to eliminate redundant slots that lie deeper in the tree by replacing

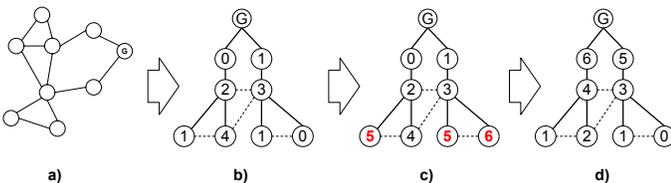


Figure 6. Ordered Coloring to minimize upstream end-to-end delay.

them with larger valued slots. As will become apparent in the next step, this manipulation allows data from the leaves of the tree to move as far as possible towards the gateway in a single TDMA cycle. Figure 6(c) shows how the previous three nodes are given larger values in order to minimize packet latencies. The final step in the heuristic inverts all of the slot assignments such that lower slot values are towards the edge of the tree allowing information to be propagated and aggregated in a cascading manner towards the gateway.

4.4. Explicit Rate Control

RT-Link allows explicit rate control by specifying a 4-bit rate index r , in the schedule assigned to each node. A flow’s rate is defined by the number of active frames that it transmits specified by 2^{r-1} . For example, rate 1 transmits on every frame while rate 3 transmits on every 4th frame. Using this scheme we can vary a flow’s rate by control the number of slots and the rate index assigned to it.

5. RT-Link Implementation

In the following section, we describe our hardware platform as well as two different hardware-aided out-of-band time synchronization solutions. First, we introduce FireFly, a custom 802.15.4 wireless sensor node. Following this, we describe an add-on board for receiving the atomic clock broadcast for outdoor synchronization and a board for receiving an AM broadcast synchronization pulse for indoors. We then evaluate the timing and energy impact of our synchronization hardware on the MAC protocol.

5.1. Hardware

We developed a low-cost low-power hardware platform called FireFly as shown in Figure 7. The board uses an Atmel Atmega32L [21] 8-bit microcontroller and a Chipcon CC2420 [22] IEEE 802.15.4 wireless transceiver. The microcontroller operates at 8Mhz and has 32KB of ROM and 2KB of RAM. The FireFly board includes light, temperature, audio, dual-axis acceleration and passive infrared motion sensors. We have also developed a lower-cost version of the board called the FireFly Jr. that does not include sensors, and is used to forward packets in the network. The FireFly board interfaces with a computer using an external USB dongle.

5.2. Time Synchronization

In order to achieve the highly accurate time synchronization required for TDMA at a packet level granularity, we use two out-of-band time synchronization sources. One uses the WWVB atomic clock broadcast, and the other relies on a carrier-current AM transmitter. In general, the synchronization device should be low power, inexpensive, and consist of a simple receiver. The time synchronization transmitter must be capable of covering a large area.

5.2.1. Implementation

The WWVB atomic broadcast is a pulse width modulated signal with a bit starting each second. Our system uses an off-the-shelf WWVB receiver (Figure 8) to detect these rising edges, and does not need to decode the entire time string. When active, the board draws 0.6mA at 3 volts and requires

less than 5uA when powered off. Inside buildings, atomic clock receivers are typically unable to receive any signal, so we use a carrier-current AM broadcast. Carrier-current uses a building's power infrastructure as an antenna to radiate the time synchronization pulse. We used an off-the-shelf low-power AM transmitter and power coupler [17] that adhere to the FCC Part 15 regulations without requiring a license. The transmitter provides time synchronization to two 5-story campus buildings which operate on 2 AC phases. Figure 8 shows an add-on AM receiver module capable of decoding our AM time sync pulse. We use a commercial AM receiver module and then designed a custom supporting-board which thresholds the demodulated signal to decode the pulse. The supporting AM board is capable of controlling the power to the AM receiver.

The energy required to activate the AM receiver module and to receive a pulse is equivalent to sending one and a half 802.15.4 packets. The use of a more advanced single chip AM radio [23] would bring these values lower and allow for a more compact design. We estimate that using a single chip AM radio receiver, the synchronization energy cost would be less than one tenth the energy of sending or receiving a single in band packet.

5.2.2. Scalability and Performance

In order to maintain scalability across multiple buildings, our AM transmitter locally rebroadcasts the atomic clock time signal. The synchronization pulse for the AM transmitter is a line-balanced 50us square wave generated by a modified FireFly node capable of atomic clock synchronization.

In order to evaluate the effectiveness of the synchronization, we placed five nodes at different points inside a five story building. Each node was connected to a data collection board using several hundred feet of cables. The data collection board timed the difference between when the synchronization pulse was generated and when each node acknowledged the pulse. This test was performed while the MAC protocol was active in order to get an accurate idea of the possible jitter including MAC related processing overhead. Figure 9 shows a histogram with the distribution of each node's synchronization time jitter. An AM pulse was sent once per second for 24 hours during normal operation of a classroom building. The graph shows that the jitter is bounded to within 200us. 99.6% of the synchronization pulses were correctly

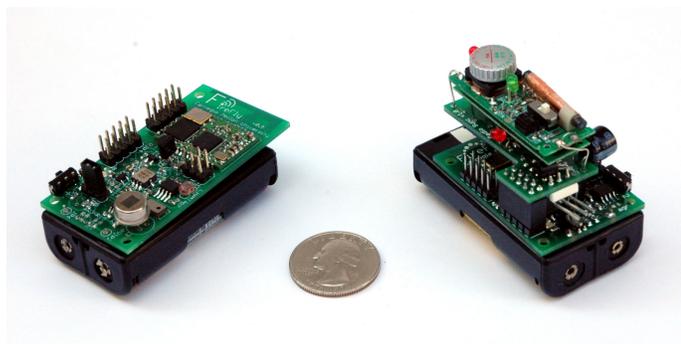


Figure 7. FireFly and FireFly Jr board with AM synchronization module

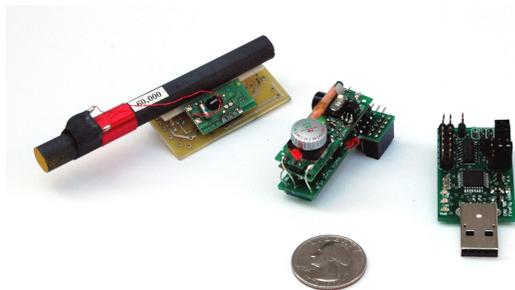


Figure 8. Left to Right: WWVB atomic clock receiver, AM receiver and USB interface board.

detected. We found that with more refined tuning of the AM radios, the jitter could be bounded to well within 50us.

In order to maintain synchronization over an entire TDMA cycle duration, it is necessary to measure the drift associated with the clock crystal on the processor. We observed that the worst of our clocks was drifting by 10us/s giving it a drift rate of $10e-5$. Our previous experiment illustrates that the jitter from AM radio was at worst 100us indicating that the drift would not become a problem for at least 10 seconds. The drift due to the clock crystal was also relatively consistent, and hence could be accounted for in software by timing the difference between synchronization pulses and performing a clock-rate adjustment. In our final implementation we are able to maintain globally synchronization to within 20us.

5.3. TDMA Slot Mechanics

When a node is first powered on, it activates the AM receiver and waits for the first synchronization pulse. Figure 10 shows the actual timing associated with our TDMA frames. Once the node detects a pulse, it resets the TDMA frame counter maintained in the microcontroller which then powers down the AM receiver. When the node receives its synchronization pulse, it begins the active TDMA time cycle. After checking its receive and transmit masks, the node determines which slots it should transmit and receive on. During a receive timeslot, the node immediately turns on the receiver. The receiver will wait for a packet, or if no preamble is detected it will time out.

The received packet is read from the CC2420 chip into a memory address that was allocated to that particular slot. We employ a zero-copy buffer scheme to move packets from the receive to the transmit queue. In the case of automatic packet

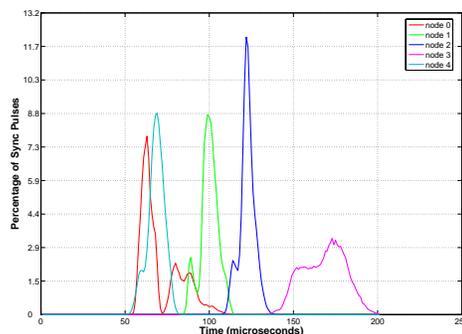


Figure 9. Distributions of AM carrier current time synchronization jitter over a 24 hour period.

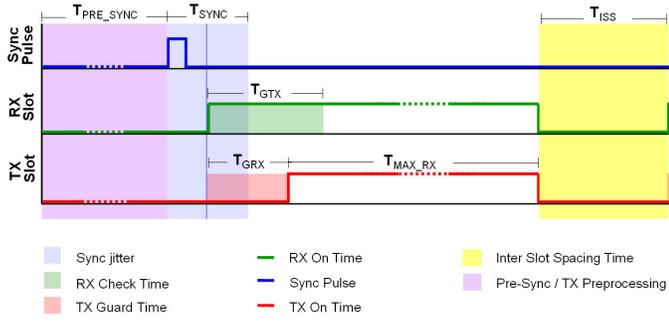


Figure 10. RT-Link operation and timing parameters.

aggregation, the payload information from a packet is explicitly copied to the end of the transmit buffer. When the node reaches a transmit timeslot, it must wait for a guard time to elapse before sending data. Accounting for the possibility that the receiver has drifted ahead or behind the transmitter, the transmitter has a guard time before sending and the receiver preamble-check has a guard time extending beyond the expected packet. Table 2 in the next section shows the different timeout values that work well for our hardware configuration. Once the timeslot is complete, there needs to be an additional guard time before the next slot. We provide this guard time plus a configurable inter-slot processing time that allows the MAC to do the minimal processing required for inter-slot packet forwarding. This feature is motivated by memory limitations and reduction of network queue sizes.

Figure 11 shows a sample trace of two nodes communicating with each other. The rapid receiver checks at the end of the cycle show the contention period with low-power listening for the duration of a preamble.

5.4. Integration With Nano-RK RTOS

We implement RT-link as a network task using the Nano-RK [3] real-time operating system. Nano-RK is a fixed priority preemptive reservation based real-time operating system with support for virtual energy budgets. Individual slots inside each TDMA frame is represented as periodic tasks with execution times equal to a single slot size and a period equal to the frame interval. There is a single link layer task running at the highest priority. This task is given a worst case execution time of the sum of its active slots and a period equal to the frame size. So while the TDMA communication is modeled as multiple fixed period tasks, in actuality it executes as a single task with all of the periods composed together.

6. Performance Evaluation

In this section, we compare the multi-hop performance of RT-Link with that of Low Power Listen (LPL) CSMA protocols such as B-MAC. We first validated our implementation of RT-Link in a 10-node test-bed. Following this, we use simulation to compare latency and lifetime.

6.1. Energy Model

To calculate the node duty cycle and lifetime we sum the node's energy consumptions over a TDMA frame. Table 1 shows the power consumed by each component assuming operation at 3 volts. Table 2 and Table 3 show the timing parameters and the energy of each operation during the TDMA

frame. The active time of each TDMA slot, T_{active} , is dependent on the total number of slots, N_{slots} , the maximum slots transmit time $T_{max_payload}$, the AM synchronization setup T_{sync_setup} and capture T_{sync} as well as inter slot processing time T_{ISS} . The number of slots and the length of the TDMA frame are dependent on the desired application sampling rate and throughput configured by the developer.

$$T_{active} = T_{sync_setup} + T_{sync} + N_{slots} * (T_{max_payload} + T_{ISS}) \quad (1)$$

The idle time, T_{idle} , between slots is the difference between the active time and the total frame time, T_{frame} . This is typically customized for the specific application since it has impact on both battery life and latency. For long sampling intervals, idle time can be added at the end of the active TDMA slots.

$$T_{idle} = T_{frame} - T_{active} \quad (2)$$

The three customizable parameters that define the lifetime of a node are the TDMA frame time, the number of TDMA slots (including the number of contention slots $N_{contention}$) and the degree d of the node. As the degree increases, the node must check the start of additional time slots and may potentially have to receive packets from its neighbors. The minimum energy that the node will require during a single TDMA frame E_{min} is the sum of the different possible energy consumers assuming no packets are received and the node does not transmit packets:

$$E_{min} = E_{sync} + (d + N_{contention}) * E_{GRX} + E_{CPU_active} + E_{CPU_sleep} + E_{radio_idle} + E_{radio_sleep} \quad (3)$$

The maximum energy the node can consume during a single TDMA frame is the minimal energy consumed during that frame summed with the possible radio transmissions that can occur during a TDMA frame.

$$E_{max} = E_{min} + (d + N_{contention}) * E_{RX} + N_{TX_slots} * E_{TX} \quad (4)$$

The maximum power consumed by a node over a TDMA frame can be computed as follows:

$$P_{avg} = E_{max} / T_{frame} \quad (5)$$



Figure 11. Channels 1 and 2 show transmit and receiver activity for one node. Channels 3 and 4 show radio activity for a second node that receives a packet from the first node and transmits a response a few slots later. The small pulses represent RX checks that timed out. Longer pulses show packets of data being transmitted. The group of pulses towards the right side show the contention slots.

Power Parameters	Symbol	I(ma)	Power(mW)
Radio Transmitter	P_{radio_TX}	17.4	52.2
Radio Receiver	P_{radio_RX}	19.7	59.1
Radio Idle	P_{radio_idle}	0.426	1.28
Radio Sleep	P_{radio_sleep}	$1e^{-3}$	$3e^{-3}$
CPU Active	P_{CPU_active}	1.1	3.3
CPU Sleep	P_{CPU_sleep}	$1e^{-3}$	$3e^{-3}$
AM Sync Active	P_{sync}	5	15

Table 1. Power Consumption of the main components.

Timing Parameters	Symbol	Time (ms)
Max Packet Transfer	$T_{max_payload}$	4
Sync Pulse Jitter	T_{sync}	$100e^{-3}$
Sync Pulse Setup	T_{sync_setup}	$20 + (\rho * T_{frame})$
RX Timeout	T_{GRX}	$300e^{-3}$
TX Guard Time	T_{GTX}	$100e^{-3}$
Inter Slot Spacing	T_{ISS}	$500e^{-3}$
Clock Drift Rate	ρ	$10e^{-2} s/s$

Table 2. Timing Parameters for main components.

The lifetime of the node can be computed as follows:

$$Lifetime = (E_{capacity}/E_{max}) * T_{Frame} \quad (6)$$

Figure 1(a) shows the lifetime of a single node with respect to the sample interval and the number of neighbors. The node in this example is set to operate at the lowest rate that matches the sampling rate interval with no contention slots. As the degree of the node increases, the number of receive checks increase hence decreasing the lifetime. As mentioned before, logical pruning of the topology by selective listening can have a large impact on system lifetime.

6.2. Lifetime

Two major factors control node lifetime in sensor networks are the topology and event sampling rate. We have already shown how RT-Link allows for logical pruning of topology to conserve energy. We will now investigate the lifetime with respect to event sampling rate. A typical LPL-CSMA approach must balance long preamble transmit times with the frequency of channel activity checks. As described in [4] we observe a curve similar to Figure 12(a) where at a given sampling rate, there is an optimal lifetime produced by a particular check interval. The authors in [4] neglected to include the voltage when calculating power and hence their lifetimes where ex-

Energy Parameters	Symbol	Energy (mW)
Synchronization	E_{sync}	$P_{sync} * (T_{sync} + T_{sync_setup})$
Active CPU	E_{CPU_active}	$P_{CPU_active} * T_{active}$
Sleep CPU	E_{CPU_sleep}	$P_{CPU_sleep} * T_{idle}$
TX Radio	E_{radio_tx}	$P_{radio_tx} * (T_{max_payload} + T_{GTX})$
RX Radio	E_{radio_rx}	$P_{radio_rx} * T_{max_payload}$
Idle Radio	E_{radio_idle}	$P_{radio_idle} * T_{active}$
Sleep Radio	E_{radio_sleep}	$P_{radio_sleep} * T_{idle}$
RX Radio Check	E_{GRX}	$P_{radio_rx} * T_{GRX}$

Table 3. Energy of components with respect to power and time.

Parameter	Symbol	Value
Sleep Power	P_{sleep}	90mW
Sample Time	T_s	
Check Interval	T_c	
Channel Check Time	T_{cca}	2.5ms
Sample Energy	E_{sample}	150mJ
Battery Capacity	C_{bat}	2500mAh
Voltage	V	3.0

Table 4. LPL-CSMA parameters.

aggerated. We show the corrected graph using power values based on our hardware. The lifetime can be computed in (8).

$$E_{idle} = P_{sleep} * (T_s - (T_{cca} * \frac{T_s}{T_c})) \quad (7)$$

$$L = C_{bat} / \frac{(\frac{T_s}{T_c} * E_{sample}) + (T_c * P_{tx}) + E_{rx} + E_{cpu} + E_{idle}}{T_s * V * 24 * 365} \quad (8)$$

Table 4 describes the above values where L is the node lifetime in years. For a given sampling rate, checking the channel more or less frequently can be quite inefficient. In a multi-hop environment, this means that for a particular event rate of interest, the end-to-end latency is a function of the system check interval which must be fixed in order to achieve the optimal lifetime. This implies that without time synchronization, large sampling intervals will lead to longer latencies. Figure 12(b) shows the optimal check interval as a function of the sampling rate. This is determined by taking the zero of the derivative of equation 8 for every sampling rate. The dot represents the optimal check interval at the 30 minute sampling rate from the previous graph. Here we see that even as the event rate approaches 100 minutes, the check interval must still be less than 4 seconds to achieve the best lifetime. In that period of time with a single neighbor, approximately 1500 checks would have gone wasted. Figure 12(c) shows sampling rate with respect to lifetime for RT-link (with and without hardware time synchronization), the optimal node lifetime and the optimal LPL-CSMA lifetime. The overall optimal lifetime assumes perfect node synchronization meaning that the only energy to be consumed is the minimum number of perfectly coordinated packet transmit and receives and the system idle energy. The LPL-CSMA line represents the lifetime given the optimal check interval. We see that for fast sampling rates, hardware time synchronization makes less of a difference. This is because synchronization can be achieved by timing the arrival of normal data messages that already contain slot information. As the sampling rate increases, extra messages must be sent to maintain in-band time synchronization. We see that across the range of a few seconds to nearly two hours, RT-Link with hardware synchronization is quite close to the optimal lifetime and out performs the LPL-CSMA mac protocol by a significant margin.

6.3. End-to-end Latency

In order to investigate the performance of RT-Link, we simulated its operation to compare the end-to-end latency with asynchronous and loosely synchronized protocols across various topologies. To study the latency in a multi-hop scenario we focused on the impact of the hidden terminal problem on the performance of B-MAC and S-MAC. All the tests

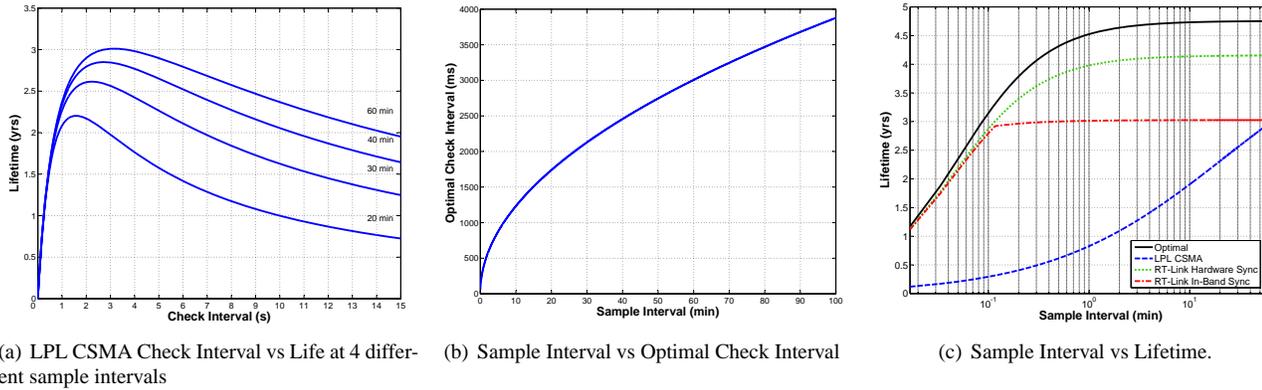


Figure 12. Effect of Sample Interval on LPL CSMA Check Interval as well as Lifetime for both CSMA and TDMA

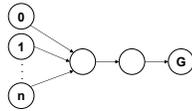


Figure 13. Multi-hop network topology with hidden terminal problem.

in [4] were designed to avoid the hidden terminal problem and essentially focused on extremely low-load and one-hop scenarios. We simulated a network topology of two "backbone" nodes connected to a gateway. One or more leaf nodes were connected to the lower backbone node as shown in Figure 13. Only the leaf nodes generated traffic to the gateway. The total traffic issued by all nodes was fixed to 1000 1-byte packets. At each hop, if a node received multiple packets before its next transmission, it was able to aggregate them up to 100-byte fragments. The tested topology is the base case for the hidden terminal problem as the transmission opportunity of the backbone nodes is directly affected by the degree of the lower backbone node.

We compare the performance of RT-Link with a 100ms and 300ms cycle duration with RTS-CTS enabled B-MAC operating with 25ms and 100ms check times. The RTS-CTS capability was implemented as outlined in [4]. When a node wakes up and detects the channel to be clear, RTS and CTS with long preambles are exchanged followed by a data packet with a short preamble. We assume B-MAC is capable of perfect clear channel assessment, zero packet loss transmissions and zero cost acknowledgement of packet reception. We observe that as the node degree increases (Figure 14), B-MAC suffers a linear increase in collisions, leading to an exponential increase in latency. With a check time of 100ms, B-MAC saturates at a degree of 4. Increasing the check time to 25ms, pushes the saturation point out to a degree of 8. Using the schedule generate by the heuristic in Section 4, RT-Link demonstrates a flat end-to-end latency.

The clear drawback to a basic B-MAC with RTS-CTS is that upon hidden terminal collisions, the nodes immediately retry after a small random backoff. To alleviate problem, we provided nodes with topology information such that a node's contention window size is proportion to the product of the degree and the time to transmit a packet. As can be seen in Figure 15, this allows for a relatively constant number of col-

lisions since each node shares the channel more efficiently. This extra backoff, in turn increases latency linearly with the node degree. We see that RT-Link suffers zero collisions and maintains a constant latency.

6.4. Experimental Evaluation

We deployed RT-Link on 42 nodes in the National Institute for Occupational Safety and Health (NIOSH) experimental coal mine facility. The goal of the deployment was to track the location of mobile nodes carried by miners as well as monitor an assortment of sensors. Coal mines can be many miles deep and consist of a grid of passageways cut through the coal seam. Figure 16 shows a map of the coal mine with the overlaid network topology. We see that due to the remaining coal pillars, the degree of the network graph is quite low (at most 5), but the depth is quite large (over 15 hops). Long linear chains can be problematic for in-band time synchronization due to the increasing probability of packet loss across the multiple hops. Since coal miners require power at the face of the mine, there is typically a main power line fed into the mine that is ideal for our AM transmitter. Any nodes located on the main corridor can use the AM time synchronization while nodes on the periphery can use in-band time synchronization. We left the nodes for three weeks logging data every 20 seconds. We found that a few nodes located far away from the AM time synchronized region of the network experienced problems due to dropped packets that lead to higher than normal power consumption. During the network setup we saw that all nodes had reliable links. Narrow passageways, miners and machinery increase packet loss by blocking line of site communication. We gain two lessons from this deployment. First, even in controlled environments link quality can change

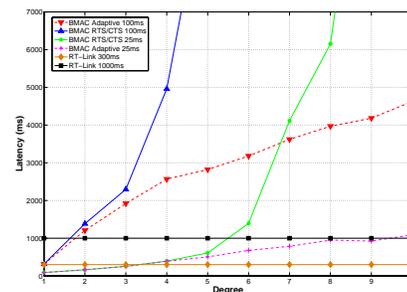


Figure 14. Impact of Latency with node degree

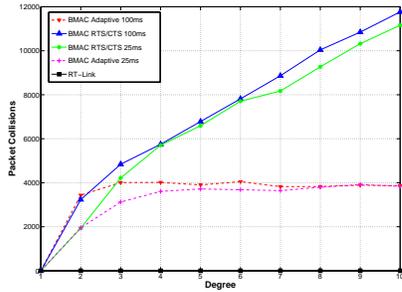


Figure 15. Effect of node degree on collisions for B-MAC

due to motion in the environment and unforeseen perturbations over time however the topology will return to a steady state. Second, as hop length increases, reliability decreases which causes time synchronization degradation and increased energy consumption in the form of extended synchronization wait times. This indicates that we should further explore how to address link faults in an energy efficient manner.

7. Conclusion

In this paper we explore the design, implementation and performance of a link layer protocol for energy-constrained multi-hop wireless networks with end-to-end delay constraints. We introduced RT-Link, a time-synchronized link protocol for fixed and mobile embedded radios. We identify three key observations in the design and deployment of RT-Link: (a) Hardware-based global-time synchronization is a robust and scalable option to in-band software-based techniques. (b) Achieving global time-synchronization is both economical and convenient for indoor and outdoor deployments. (c) RT-Link achieves a practical lifetime of over 2 years. RT-Link has been implemented on FireFly, our sensor network platform, and has been deployed on networks with 42 IEEE 802.14.5 nodes. It outperforms energy-efficient protocols such as B-MAC in energy consumption and end-to-end delay.

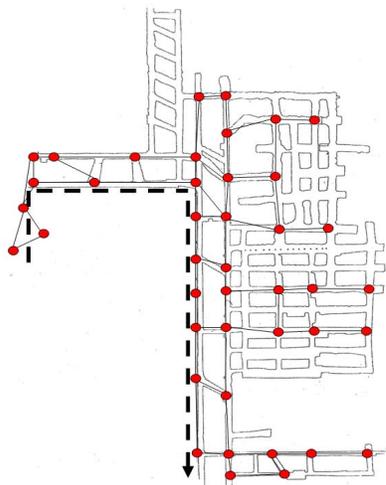


Figure 16. Coal mine map with network topology. Dotted line shows leaky feeder time synchronization cable.

References

- [1] E. Callaway M. Bourgeois C. Mitter J. A. Gutierrez, M. Naeve and B. Heile. IEEE 802.15.4: A developing standard for low-power low-cost wireless personal area networks, 2001.
- [2] R. J. Tuchman and R. F. Brinkley. A History of the Bureau of Mines Pittsburgh Research Center. *US Bureau of Mines*, pages 1–23, 1990.
- [3] Anand Eswaran, Anthony Rowe, and Raj Rajkumar. Nano-RK: an Energy-aware Resource-centric RTOS for Sensor Networks. *IEEE Real-Time Systems Symposium*, 2005.
- [4] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. *SenSys*, November 2005.
- [5] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. *INFOCOM*, June 2002.
- [6] T. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. *SenSys*, November 2003.
- [7] A. El-Hoiydi and J. Decotignie. Wisemac: An ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. *ISCC*, 2004.
- [8] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Sensys*, 2003.
- [9] L.F.W. van Hoesel and P.J.M. Havinga. A lightweight medium access protocol for wireless sensor networks. *INSS*, 2004.
- [10] L. G. Roberts. Aloha packet system with and without slots and capture. *SIGCOMM*, 5(2):28–42, 1975.
- [11] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Topology management for sensor networks: Exploiting latency and density. *MobiHoc*, 2002.
- [12] C. Guo, L. C. Zhong, and J. Rabaey. Low power distributed mac for ad hoc sensor radio networks. *Globecom*, 2001.
- [13] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcast. *USENIX OSDI*, 2002.
- [14] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. *Proc. ACM Sensys*, 2003.
- [15] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The flooding time synchronization protocol. *Proc. ACM Sensys*, 2004.
- [16] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. *Proc. ACM Sensys*, 2003.
- [17] Radio systems 30w tr-6000 am transmitter data sheet, 2001.
- [18] Anthony Rowe, Rahul Mangharam, and Raj Rajkumar. RT-Link: A Time-Synchronized Link Protocol for Energy-Constrained Multi-hop Wireless Networks. *CMU Tech Report TR05-08*, 2005.
- [19] Randolph D. Nelson and Leonard Kleinrock. Maximum probability of successful transmission in a random planar packet radio network. *INFOCOM*, pages 365–370, 1983.
- [20] Hari Balakrishnan et al. The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks. *IEEE Journal on Selected Areas in Comm.*, 22(6):1069–1079, August 2004.
- [21] Atmel corporation, atmega32 data sheet, 2005.
- [22] Chipcon inc., chipcon cc2420 data sheet, 2003.
- [23] Tea5551t 1-chip am radio philips semiconductors, 1990.