Approaching the Maximum 802.15.4 Multi-hop Throughput

Fredrik Österlind and Adam Dunkels
Swedish Institute of Computer Science
{fros,adam}@sics.se

Abstract

Recent work in sensor network energy optimization has shown that batch-and-send networks can significantly reduce network energy consumption. Batch-and-send networks rely on effective batch data transport protocols, but the throughput of state-of-the-art protocols is low. We present conditional immediate transmission, a novel packet forwarding mechanism, with which we achieve a 109 kbit/s raw data throughput over a 6-hop multi-channel 250 kbit/s 802.15.4 network; 97% of the theoretical upper bound. We show that packet copying is the bottleneck in high-throughput packet forwarding and that by moving packet copying off the critical path, we nearly double the end-to-end throughput. Our results can be seen as an upper bound on the achievable throughput over a single-route, multi-channel, multi-hop 802.15.4 network. While it might be possible to slightly improve our performance, we are sufficiently close to the theoretical upper bound for such work to be of limited value.

Categories and Subject Descriptors

C.2.4 [**Computer Communication Networks**]: Distributed Systems—*Network Operating Systems*

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Wireless sensor networks, 802.15.4, Multi-hop performance

1 Introduction

Recent work on energy optimization for sense-and-send sensor networks have found that data batching improve energy efficiency [6, 12]. By batching the sensed data instead of immediately sending it, the radio duty cycle can be significantly reduced thus leading to better energy efficiency. Examples of recent sensor network deployments that use the batch-and-sense approach include volcano monitoring [17] and bridge health monitoring [9].

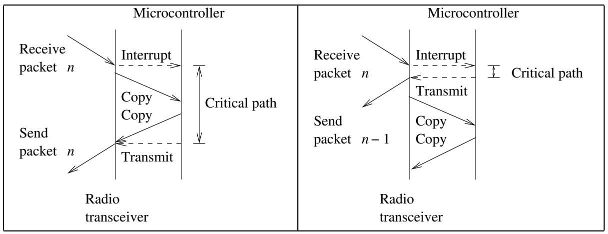

Figure 1. In packet forwarding with packet copying (left), copying and processing of packet n occurs before forwarding packet n. With conditional immediate transmission (right), packet $n-1$ is forwarded immediately after packet n is received but before packet n is copied and processed.

Batch-and-send depends on effective protocols for multi-hop download of the batched data. The performance of existing protocols is, however, much lower than the nominal radio capacity [8]. Flush [8], the current state-of-the-art batch download protocol, reports a multi-hop data throughput of approximately 10 kbit/s over a 250 kbit/s 802.15.4 radio. The Flush protocol cannot, however, be blamed for this performance discrepancy. Rather, it is the underlying layers of the system that limit the throughput. With this paper, we take the logical next step and address the underlying layers that limit the throughput in multi-hop 802.15.4 networks. We show that packet copying between the radio transceiver and the microcontroller is the bottleneck in multi-hop 802.15.4 transport.

We present *conditional immediate transmission*, a packet forwarding abstraction that achieves a data throughput of 97% of the theoretical upper bound and reaches a raw multi-hop throughput of 109 kbit/s over a 6-hop network with a per-hop radio capacity of 250 kbit/s. We identify packet copying as the bottleneck in the critical path of packet forwarding and show that by moving packet copying off the critical path, conditional immediate transmission nearly doubles the multi-hop 802.15.4 throughput. For conditional immediate transmission to reach its full potential, the radio hardware driver must implement pre-copying, but we have designed conditional immediate transmission to be incrementally applicable: unmodified radio drivers will still work correctly,

HotEmNets'08, June 2-3, 2008, Charlottesville, Virginia, USA

```
condition(protocol_id, sender_address):
  if protocol_id = our_protocol_id and
    sender_address = prev_hop_address
    return true
  else
    return false

forward_packet(packet):
  conditional_send(packet, condition)
```

Figure 2. Example packet forwarding with conditional immediate transmission, in pseudo-code. The condition function is invoked when new packets arrive, but only until the packet has been sent. If the condition is true, the packet is immediately sent. By keeping the packet in the memory buffer of the radio transceiver, packet copying is moved off the critical path.

but may not achieve the high throughput provided by conditional immediate transmission.

The contributions of this paper are threefold. First, we present conditional immediate transmission, a packet forwarding mechanism that reaches 97% of the theoretical upper bound of multi-hop 802.15.4 batch transport. Second, we experimentally quantify the performance improvement of using multiple channels for multi-hop transport protocols. Third, we quantify the packet forwarding latency for multi-hop 802.15.4 networks and show that conditional immediate transmission also can reduce the information forwarding latency.

Conditional immediate transmission uses packet precopying to avoid packet copying on the critical path as shown in Figure 1. In a multi-hop forwarding protocol, packet $n-1$ is copied into the memory of the radio transceiver before packet n arrives. When packet n arrives, packet $n-1$ can be immediately forwarded, without copying. To determine if packet $n-1$ should be sent, a few bytes need to be copied from the incoming packet; typically the link-layer source address and the protocol ID must be known to make the forwarding decision. We show, however, that the performance impact of this copying is small.

The conditional immediate transmission abstraction requires very little additional code in a protocol implementation, as shown in Figure 2. To send a packet with conditional immediate transmission, a condition function must be provided. The condition function is evaluated for every incoming packet as long as the packet is pending transmission. Conditional immediate transmission does not replace the normal packet transmission abstraction. How the addition of the conditional immediate transmission abstraction might affect future protocols and their implementations is the subject of future work.

We have implemented conditional immediate transmission in the Contiki operating system, but the implementation uses no Contiki-specific code. The mechanism is generic enough to be implemented in any operating system.

The rest of this paper is structured as follows. In Section 2 we analyze the theoretical limits of 802.15.4 multi-hop throughput for both single-channel and multi-channel protocols. In Section 3 we show that packet copying is the bottleneck of packet forwarding on state-of-the-art hardware platforms; packet copying nearly halves the end-to-end throughput. In Section 4 we present the conditional immediate transmission mechanism that moves packet copying off the critical path and in Section 5 we show that, by using conditional immediate transmission, we achieve a multi-hop throughput that reaches 97% of the theoretical upper bound. We review related work in Section 7. In Section 8 we conclude the paper and discuss the implications of our results on the directions of future sensor network protocol research.

2 Theoretical Performance Limits of 802.15.4

IEEE 802.15.4 is a widely used radio standard for low-power radios. The 802.15.4 standard is defined for two frequency bands: 868 MHz and 2.4 GHz. The 2.4 GHz version of the protocol defines 16 non-overlapping radio channels and has a peak bit rate of 250 kbit/s. In this paper, we use the Chipcon CC2420 radio transceiver [2] and the 2.4 GHz band.

We use an idealized model of a 802.15.4 node to analyze the theoretical performance limits of 802.15.4. Based on our model, we present analytical upper bounds on the throughput of single-hop and multi-hop 802.15.4 throughput. The model does not take practical limiting factors such as radio chip communication or data processing into account. In Section 3, we turn our attention to the practical factors and show that they have a profound impact on the practically achievable 802.15.4 throughput.

2.1 Single-hop Upper Bound

Single-hop 802.15.4 throughput is limited by the serialization delay: it is not possible to send more data than the bit rate allows. The bit rate is, however, not the only limiting factor. The 802.15.4 physical layer frame format consists of a 4-byte preamble, a 1-byte Start of Frame delimiter (SFD), and a 1-byte frame length field. The maximum physical layer payload size is 127 bytes. For the purpose of our measurements, we do not use the full 802.15.4 MAC packet format. Instead, our MAC Protocol Data Unit (MPDU) only consists of application payload data in addition to the 2-byte Frame Check Sequence (FCS) field with CRC information. Our maximum data payload size is hence 125 bytes. In addition to the transmitted overhead bytes, the throughput is further limited by a 192 microseconds turnaround time, equivalent to 6 additional overhead bytes. We can now calculate a theoretical upper bound on the single-hop throughput T_s,

$$T_s = \frac{125}{4+1+1+125+2+6} \times 250 \approx 225,$$

where T_s is measured in kbit/s.

2.2 Multi-hop Upper Bound

Multi-hop forwarding incurs additional limitations on throughput compared to the single-hop case. The number of nodes within interference range that have to transmit the same radio packet defines our upper bound on the multi-hop throughput. An upper bound on multi-hop throughput when all nodes in a n-hop route interfere with each other is

$$T_m(n) = \frac{1}{n} \times T_s,$$

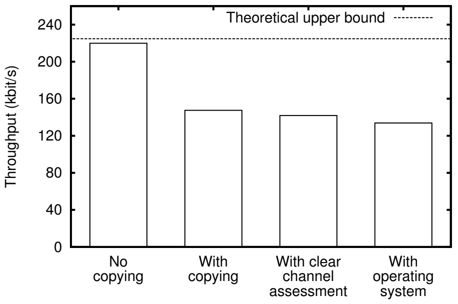

Figure 3. The bottleneck in single-hop transfer is neither the operating system nor clear-channel assessment, but packet copying. Packet copying reduces the single-hop throughput with one third: from 220 kbit/s to 148 kbit/s. The theoretical upper bound is 224.8 kbit/s.

where $T_m(n)$ is measured in kbit/s.

2.3 Multi-channel Upper Bound

The negative effect of contention and interference on multi-hop data throughput can be reduced by using multiple radio channels; several communication protocols use multiple radio frequencies to achieve higher multi-hop throughput [11, 18]. With the number of radio channels larger than the number of transmitting nodes, each unicast transmission can be performed on a dedicated radio channel. With multiple radio channels our multi-hop upper bound on throughput is half the single-hop upper bound: 112.5 kbit/s.

3 The Practical 802.15.4 Bottleneck: Packet Copying

In addition to the theoretical upper bound, there are several practical aspects that limit achievable multi-hop throughput with 802.15.4. These aspects include packet copying betwen the microcontroller and the radio transceiver, Clear-Channel Assessment (CCA), and the additional overhead of operating system and communication stack.

To quantify the effect of the practical aspects on 802.15.4 throughput, we circumvent the aspects, one by one, and measure the resulting throughput. We find that the bottleneck is packet copying and that the performance impact both of the operating system and the CCA is small.

Our hardware platform is the Tmote Sky [14], which is equipped with a Chipcon CC2420 802.15.4-compatible packet-based radio transceiver [2] and a TI MSP430 microcontroller. As in other state-of-the-art 802.15.4 hardware platforms, communication between the microcontroller and the radio chip is via a Serial Peripherals Interface (SPI) bus. We do not use any CC2420-specific features such as automatic address recognition, encryption, or automatic acknowledgments.

We use the Contiki operating system [4, 5] as our experimental platform but do not use any Contiki-specific features. We do not use any power-saving MAC protocol, but directly use the radio transceiver.

The CC2420 has two separate on-chip 128-byte memory buffers: one receive buffer and one transmit buffer. Before sending a radio packet, the MSP430 microcontroller copies the packet data into the transmit buffer over the SPI bus. To transmit the packet, the microcontroller sends a separate transmit command to the radio transceiver, which then transmits the packet over the radio.

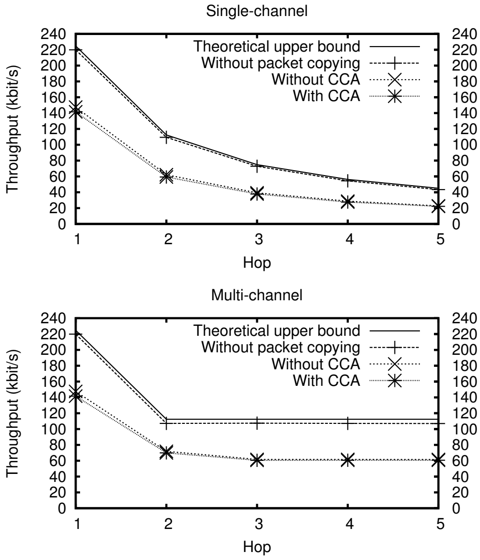

Figure 4. Experimental results of multi-hop 802.15.4 throughput with a single channel (top) and multiple channels (bottom). Packet copying nearly halves the throughput both in the single-channel case and in the multi-channel case. Clear-channel assessment (CCA) has little impact on throughput.

When the CC2420 has received a radio packet it flags an interrupt in the MSP430 microcontroller. The interrupt handler typically notifies a process in the operating system that fetches the incoming radio data over the SPI bus.

The CC2420 can communicate at any of the 16 channels within the 2.4 GHz band specified by the IEEE 802.15.4 standard. To switch channel, the microcontroller sends a command to the CC2420 over the SPI bus. The CC2420 has calibrated itself to the new channel within 48 μs [2].

To measure the overhead of packet copying, we must be able to forward packets but avoid copying data to and from the radio transceiver. To achieve this, we create a mock-up experimental setup where we never copy incoming packets from the radio but simply discard the contents of incoming packets. Likewise, we do not copy any data into outgoing packets but simply transmit what happens to be in the output buffer of the CC2420. Such a system can never transport any useful data—a functional system must always copy all data to and from the radio transceiver—but the mock-up setup fulfills our purpose: we can measure the overhead introduced by packet copying. In Section 4 we show that by rearranging the order of packet transmission and packet copying, we avoid copying data on the critical path, thereby significantly improving throughput.

We conduct single-hop experiments with two Tmote Sky nodes and multi-hop experiments with six nodes. To avoid underestimating the effect of inter-path interference, we ensure that all nodes are in communication range of all other nodes. To measure throughput, we send 1000 maximum size packets from one node and measure the time it takes for the packets to be forwarded to the final receiver. We increase the data rate until we start dropping packets due to collisions. We define the highest data rate at which we received all packets as the maximum achievable throughput.

3.1 The Impact on Single-hop Throughput

Figure 3 shows that the throughput impact of packet copying is high and that the impact of the CCA and the operating system is low.

3.2 The Impact on Multi-hop Throughput

Figure 4 show that packet copying has an even larger impact on multi-hop throughput than on single-hop throughput. The reason for the larger impact is that packet copying is performed twice at every hop: from the radio transceiver to the microcontroller, and back again. By temporarily disabling packet copying we nearly double the throughput. Our hypothesis is that we can achieve such high throughput by moving packet copying off the critical path.

4 Conditional Immediate Transmission: Moving Copying off the Critical Path

Conditional immediate transmission moves packet copying off the critical path by separating the copying of packets, and the initiation of a radio transmission. By copying the outbound packet into the memory of the radio transceiver prior to receiving the next packet, the outbound packet can be immediately transmitted.

Initiating a transmission with conditional immediate transmission is conditional: the transport protocol uses a small amount of information in incoming packets to decide whether or not to immediately send the pending packet in response to the incoming data. The mechanism trades end-to-end latency for throughput, and allows a forwarding node to begin transmitting directly from the radio interrupt handler.

When sending a packet with conditional immediate transmission, the packet is not sent immediately but is pending for transmission. The pending packet data is buffered in the radio driver, but is also immediately written to the memory of the radio transceiver. A condition function is registered with the outgoing packet.

The operation of our mechanism is correct even if other packets are sent when a pending packet is already buffered. Such packets may for example originate in a process unknowingly of the ongoing bulk transfer. If another packet is to be sent when the pending packet is in the radio buffer, the pending packet is overwritten, and rewritten again once the transmission of the other packet is complete.

Like state-of-the-art bulk transfer protocols, conditional immediate transmission assumes at maximum one concurrent bulk transfer. With multiple concurrent bulk transfers, the bulk transfer ID of an incoming packet may not match the outgoing pending packet. How multiple concurrent bulk transfers affect the performance of conditional immediate transmission is the subject of future work.

When the radio transceiver has received a new radio packet in its receive buffer, the radio driver checks for any registered outgoing radio packet in its interrupt handler. If there is a pending packet, the transport protocol is directly invoked by calling the packet's condition function. The incoming radio packet still resides in the radio's receive buffer, whereas the registered pending packet resides in the radio's transmit buffer.

The condition function allows a transport protocol to access one or a few bytes, for example the source address header field, from the packet stored in the radio buffer. The data may for example be represented as a Chameleon packet attribute [5]. Using the address, the protocol decides whether the pending packet should be sent and on which radio channel it should be sent.

Regardless of whether an outgoing packet was transmitted or not, the operating system handles the new incoming packets as usual. The full packet data is copied from the radio chip, and is forwarded via the communication stack to the transport protocol. The protocol does, however, not forward the new packet directly, but sends it using conditional immediate transmission.

Note that with conditional immediate transmission, the transport protocol still has access to the full packet data before it is forwarded. The transport protocol may hence alter or piggyback additional data to the packet.

5 Evaluation

We evaluate conditional immediate transmission in terms of both throughput and latency. Our evaluation confirms our hypothesis that conditional immediate transmission significantly increases throughput, but also that conditional immediate transmission can reduce multi-hop information forwarding latency.

We have implemented a simple Flush-like [8] multi-hop bulk transfer protocol using conditional immediate transmission and the Rime protocol stack [5]. Since we are not interested in measuring the impact of packet loss, our protocol does not resend lost packets. To handle lost packets, a future packet loss-sensitive implementation of the protocol might use the packet retransmission mechanisms from Flush.

We measure throughput by sending a 1000 packets large data batch over the network, where each packet has 125 bytes payload data. We vary the number of hops between each experiment and perform experiments with both a single 802.15.4 channel and with multiple 802.15.4 channels. Our results show that multi-channel forwarding more than doubles the throughput.

5.1 Throughput

To evaluate the throughput improvement of conditional immediate transmission, we measure the multi-hop throughput for both a single-channel multi-hop network and a multi-channel multi-hop network. We use the same experiment setup as in Section 3.

Our results, as shown in Figure 5, show that conditional immediate transmission significantly increases throughput over copy-based forwarding. In the single-channel case, throughput reaches 97% of the theoretical upper bound, as calculated in Section 2. The raw data throughput over a 6-hops multi-channel network is 109 kbit/s.

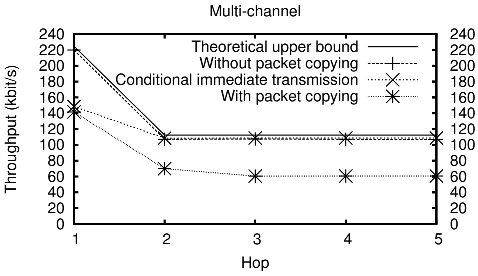

Figure 5. Multi-hop throughput with conditional immediate transmission is 97% of the theoretical upper bound on 802.15.4 throughput. This does not take packet headers into account; an 8 byte packet header limits the application data throughput to 102 kbit/s for the 6-hop multi-channel case.

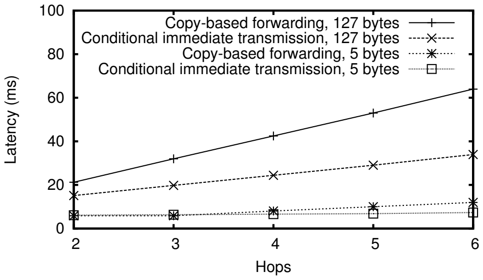

Figure 6. Conditional immediate transmission halves 6-hop information transmission latency for full size (127 byte) packets and reduces it with 40% for small (5 byte) packets.

5.2 Latency

Conditional immediate transmission increases the end-to-end data latency because packets are stored at each forwarding node, but the mechanism can also be used to reduce multi-hop latency. By exploiting the reduced critical path, a packet can be setup with conditional immediate transmission to be sent immediately when a notification packet arrives. While batch data protocols typically are not latency-sensitive, low-latency forwarding could be used for time-critical information. Figure 6 shows latency measurements from a 6-hop network. We measure the latency by sending a single packet in a 6-hop loop back to the sender and measure the time from the transmission to the reception of the packet, including copying at the sending and receiving node.

6 Discussion

The motivation for our work is the large discrepancy between the throughput of state of the art bulk transport protocols and the nominal data rate of the underlying radio. Armed with our results presented in Sections 3 through 5 we can now provide an answer to this question.

The single-channel Flush protocol [8], the current state of the art in batch transport protocols, reports a multi-hop application throughput of 10 kbit/s. This throughput is about 47% of the 5-hop single-channel throughput with packet copying that we measure in the top graph in Figure 4: 21 kbit/s. Although Flush's interference range may have been less than 5 hops, the higher header-to-data ratio limits the achievable throughput; we believe our results are consistent.

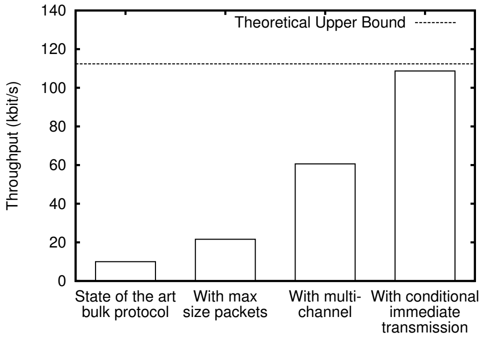

Figure 7. By using maximum sized packets, multiple channels, and conditional immediate forwarding, the resulting throughput is 97% of the upper bound.

After the effect of packet size, we look at the effects of multiple channels and packet copying. The bottom graph in Figure 4 shows that multi-channel forwarding increases throughput with a factor of three to roughly 62 kbit/s. Finally, Figure 5 shows that by avoiding packet copying on the critical path, conditional immediate transmission results in a throughput of 109 kbit/s.

In Figure 7, we provide an answer to the question of the discrepancy between the throughput of bulk transport protocols and the achievable radio throughput: with maximum sized packets, multiple channels, and conditional immediate transmission, the resulting throughput is 97% of the achievable upper bound.

7 Related Work

Our work is orthogonal to recent work on sensor network batch data transport protocols [8, 13, 15] in that conditional immediate transmission can potentially be used to improve throughput for any batch data transport protocol. Our results show significant throughput improvements for a rate-controlled Flush-like protocol.

The throughput we achieve should not, however, be directly compared with that of existing transport protocols. Protocols such as Flush [8], RMST [15], PSFQ [16] and RCRT [13] provide complete data transport solutions, whereas our work focuses on the low-level mechanism that limit the achievable throughput. The throughput of these protocols is limited by non-maximized packet sizes, as well as by not allowing use of multiple radio channels. Our work is orthogonal: we focus on profiling data transfer critical path in order to optimize overall throughput. We do not concern outselves with retransmissions or non-optimal radio environments. We believe that our forwarding mechanism can successfully be incorporated with existing rate control mechanisms, such as the mechanism used by Flush.

To the best of our knowledge, conditional immediate transmission is a unique mechanism for sensor network packet forwarding. Sensor network operating systems such as TinyOS [10] use asynchronous split-phase radio APIs that transmit packets when and if allowed by the MAC layer.

Thread-based systems such as Mantis [1] use blocking wait APIs. In contrast, conditional immediate transmission allows the radio driver to pre-copy the packet to its internal memory, thus speeding up transmission of the packet, when the condition is true.

Our work is inspired by previous work on zero-copy buffer mechanisms in general purpose operating systems [3, 7]. Our work is similar in that we also reduce the amount of data copying in order to improve the system performance. Our problem domains are different, however: we do not need to handle different protection domains, nor are we affected by cache misses.

8 Conclusions

Inspired by the large discrepancy in nominal data rate and actual application throughput for recent batch data transport protocols, we present a packet forwarding mechanism with which we achieve a 109 kbit/s data rate over a 6-hop 250 kbit/s network; 97% of the theoretical upper bound. Our work highlights the importance of avoiding data copying in the critical path to achieve high multi-hop throughput. We also show that multi-channel improves throughput significantly over single-channel forwarding.

We have developed our mechanism for 802.15.4 systems, but the mechanism is general enough to be used for any system where packet copying is a bottleneck.

The conditional immediate transmission mechanism show that packet-based radios can achieve high throughput, despite requiring data copying between the radio transceiver and the microcontroller. Our work highlights the need for more flexible radio transceiver interfaces, however, we do not want to go back to the bit- or byte-level radios that were used in the previous generation of sensor network hardware.

Our work can be seen as an upper bound on the achievable throughput over a single-route, multi-channel, multi-hop 802.15.4 network. Although it might be possible to slightly improve our performance, for example by reducing interrupt latency, we are sufficiently close to the theoretical upper bound for such work to be of limited value. Rather, our results suggest that other mechanisms, such as multi-route mechanisms, could be pursued to further improve the end-to-end throughput. With a multi-route, multi-channel protocol, the bottleneck would be the packet copying in the sending and receiving nodes. According to our measurements (Figure 3), it might be possible to achieve a 148 kbit/s end-to-end throughput; a 30% improvement over our results.

Acknowledgments

This work was partly financed by VINNOVA, the Swedish Agency for Innovation Systems.

9 References

[1] H. Abrach, S. Bhatti, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Shucker, J. Deng, and R. Han. MANTIS: system support for MultimodAl NeTworks of In-Situ sensors. In *Proc. WSNA'03*, 2003.

[2] Chipcon AS. CC2420 Datasheet (rev. 1.3), 2005.

[3] P. Druschel and L. Peterson. Fbufs: a high-bandwidth cross-domain transfer facility. In *Proceedings of the fourteenth ACM symposium on Operating systems principles*, pages 189–202, Asheville, North Carolina, United States, 1993.

[4] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the First IEEE Workshop on Embedded Networked Sensors (IEEE Emnets '04)*, Tampa, Florida, USA, November 2004.

[5] A. Dunkels, F. Österlind, and Z. He. An adaptive communication architecture for wireless sensor networks. In *Proceedings of the Fifth ACM Conference on Networked Embedded Sensor Systems (SenSys 2007)*, Sydney, Australia, November 2007.

[6] P. Dutta, D. Culler, and S. Shenker. Procrastination might lead to a longer and more useful life. In *Proceedings of HotNets-VI*, Atlanta, GA, November 2007.

[7] G. Ganger, D. Engler, F. Kaashoek, H. Briceno, R. Hunt, and T. Pinckney. Fast and flexible application-level networking on exokernel systems. *ACM Transactions on Computer Systems*, 20(1):49–83, February 2002.

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

[9] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN*, pages 254–263, 2007.

[10] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler. The Emergence of Networking Abstractions and Techniques in TinyOS. In *Proceedings of ACM/Usenix Networked Systems Design and Implementation*, San Francisco, California, USA, March 2004.

[11] C. Liang, R. Musaloiu-Elefteri, and A. Terzis. Typhoon: A reliable data dissemination protocol for wireless sensor networks. In *EWSN*, pages 268–285, 2008.

[12] G. Mathur, P. Desnoyers, D. Ganesan, and P Shenoy. Ultra-low power data storage for sensor networks. In *Proceedings of the fifth international conference on Information processing in sensor networks*, Nashville, Tennessee, USA, 2006.

[13] J. Paek and R. Govindan. Rcrt: rate-controlled reliable transport for wireless sensor networks. In *Proceedings of SenSys 2007*, 2007.

[14] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Proc. IPSN/SPOTS'05*, Los Angeles, CA, USA, April 2005.

[15] F. Stann and J. Heidemann. RMST: Reliable Data Transport in Sensor Networks. In *Proceedings of the First International Workshop on Sensor Net Protocols and Applications*, pages 102–112, Anchorage, Alaska, USA, April 2003. IEEE.

[16] C.Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: A Reliable Transport Protocol For Wireless Sensor Networks. In *First ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, September 2002.

[17] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation 2006*, Seattle, November 2006.

[18] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. Abdelzaher. Mmsn: Multi-frequency media access control for wireless sensor networks. In *INFOCOM*, 2006.