Improving Sensor Network Robustness with Multi-Channel Convergecast

Thiemo Voigt, Fredrik Österlind, Adam Dunkels

Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden
{thiemo,fros,adam}@sics.se

Abstract. Most of the existing sensor network deployments are convergecast applications that transmit data from multiple sources to one or more sinks. In this paper, we present the design of a self-organizing, collision-free multi-channel convergecast protocol. We present experiments that demonstrate our protocol's energy-efficiency for low duty cycle applications by comparing it to X-MAC. Our experiments also demonstrate that our protocol's ability to switch channels dynamically increases robustness against interference.

1 Introduction

The number of sensor network deployments is increasing rapidly. Many of the current deployments gather environmental data and send them to one or a few sinks. This paradigm is often called convergecast and a number of convergecast protocols have been developed [2, 8, 9, 11, 16].

Most sensor networks operate in license-free bands such as 868 MHz or 2.4 GHz. In the 2.4 GHz band, sensor networks need to co-exist with IEEE 802.11 (WLAN), Bluetooth and other networking technologies which can have a serious impact on the IEEE 802.15.4 network performance if the channel allocation is not carefully taken into account [14]. Modern low-power radios such as the IEEE 802.15.4-compliant CC2420 offer multiple channels which makes it possible to switch channels in order to avoid interference. So far, however, there are only a few attempts to make use of multiple channels. One of the major reasons for this is that algorithms and techniques developed for general wireless networks are not appropriate for wireless sensor networks. Many protocols are designed for more powerful radio hardware and require frequency hopping spread spectrum wireless cards [15]. Another frequent assumption made by existing general approaches is that the hardware is capable of sensing on multiple channels at the same time. However, radios typically found on sensor boards are single radio transceivers that cannot simultaneously transmit and receive. On the other hand, they can operate on different channels at different times.

We present an energy-efficient convergecast protocol that improves robustness by using multiple channels available on modern low-power radios such as the CC2420. To the best of our knowledge, this is the first convergecast protocol that uses multiple channels for this purpose. Our protocol builds on the notion staggered slots introduced by DMAC [11]. We deploy distributed slot assignment to achieve a collision free convergecast protocol without network-wide time synchronization, as well as channel-switching in case a currently used channel is interfered. We use acknowledgements for slot assignment and inter-node synchronized wake-up scheduling. WiseMAC has exploited a similar idea by including the sampling schedule offset into acknowledgements [7]. In addition, by adding control information in the acknowledgement, we provide synchronized channel switching to increase robustness against interference. We present experiments on real hardware that demonstrate the energy-efficiency of our protocol. Simulations with the COOJA simulator [12] as well as experiments on real hardware demonstrate that our scheme increases robustness against interference. The protocol's ability to synchronize node wake-up without explicit time synchronization is especially useful for low duty cycle data collection applications that demand long lifetime.

The rest of paper is outlined as follows: In the next section we present the design of our convergecast protocol. Section 3 discusses the implementation of our multi-channel algorithm in more detail and presents simulation results. Experimental results are shown in Section 4. Before concluding, we discuss related work in Section 5.

2 Convergecast Protocol Design

Our convergecast protocol uses the notion of staggered slots introduced by DMAC [11].

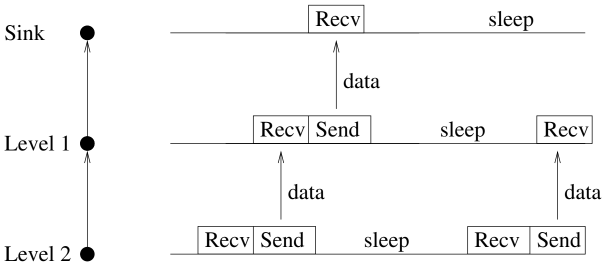

Fig. 1. Staggered wake-up a la DMAC

Figure 1 presents the staggered wake-up scheme employed by DMAC. As shown in the figure, different levels of the tree send at different times in order to reduce delay and contention. However, in DMAC there is still contention between nodes on the same level.

Figure 2 shows our basic idea. The receiver of a message sends an ACK that besides acknowledging a packet also states in how many seconds it will turn on its radio again and is ready to receive packets from its children. This way, no

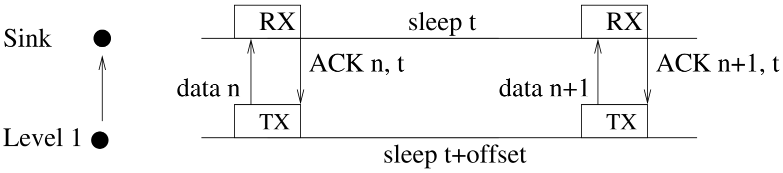

Fig. 2. Basic synchronization scheme

explicit time synchronization needs to be performed since only relative time is of importance. Note that nodes do not need to have their radio turned on during the whole duration of the TX slots.

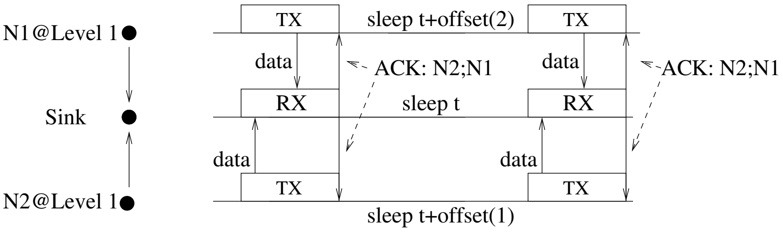

Fig. 3. On-demand slot assignment

By adding offsets for different nodes into the ACK packet, we can extend the basic scheme to let a parent such as the sink node assign slots to its children. Figure 3 demonstrates how the sink assigns different offsets to its children. Each acknowledgement packet contains ACK-fields denoting the positive and negative acknowledgement of the children's data packets in always the same order. This way, the position of the ACK-field can be used by the children to compute their offset into the parent's RX slot. The parent's RX slot must be long enough to allow a maximum number of children to transmit. In the scenario in Figure 3, node $N2$ may send before $N1$.

The scheme can be applied recursively to extend it towards a whole tree. However, when extending the scheme to several levels we need to take care to avoid collisions between nodes on different levels. Towards this end, we introduce a maximum number of children per node. Based on the maximum and its position in the tree, a node can compute its wake-up time and hence offsets for its children. This way, we build a collision-free tree without explicit time synchronization. An example is shown in Figure 4. In this figure, $N2$ is the parent of $N3$ and $N4$. However, the send and receive offsets may no longer be aligned, i.e. a node's RX slot is not immediately followed by its TX slot. Hence, a node needs to turn its

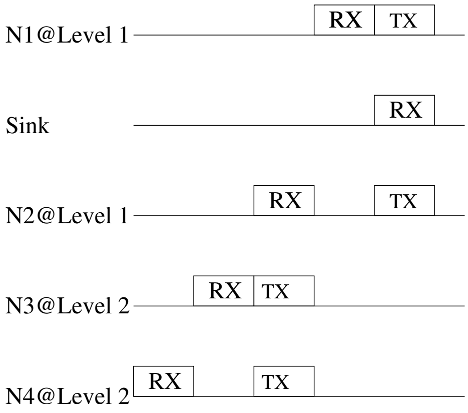

Fig. 4. Slot assignment for larger tree

radio on and off more often which implies a little energy overhead that should be well compensated for by avoiding collisions.

2.1 Multiple Channels

Current standards for low power networking allow the usage of multiple channels. For example, in IEEE 802.15.4, there are 16 channels in the 2.4 GHz band. However, only a limited number of protocols leverage the possibility of channel switching in case a channel becomes unusable due to interference. We add channel switching in our design by using the same idea of adding additional control information into the acknowledgement. In our current design, we add information about the next two channels to use into each acknowledgement. If a node does not receive any data message from its children, it blacklists the corresponding channel [1]. More details can be found in the next section.

3 Simulating Channel Switching

We have implemented a channel switching algorithm that switches between two channels. In each acknowledgement, a node announces to its children on which channels the next two ACKs will be sent. For example, the first ACK in Figure 5 announces that the sink will send the next ACK on Channel 6 and the successive one on Channel 1. In order to receive the ACK on the correct channel, the level 1 node always performs channel switching after having sent its payload packet.

Figure 5 is produced from the log of a simulation in the COOJA [12] simulator. COOJA supports radio traffic on multiple channels and it is possible to add disturber nodes that interfere with transmissions on a certain channel. The acknowledgements sent by the level 1 node are not shown.

[1] We might also choose a new, not blacklisted channel.

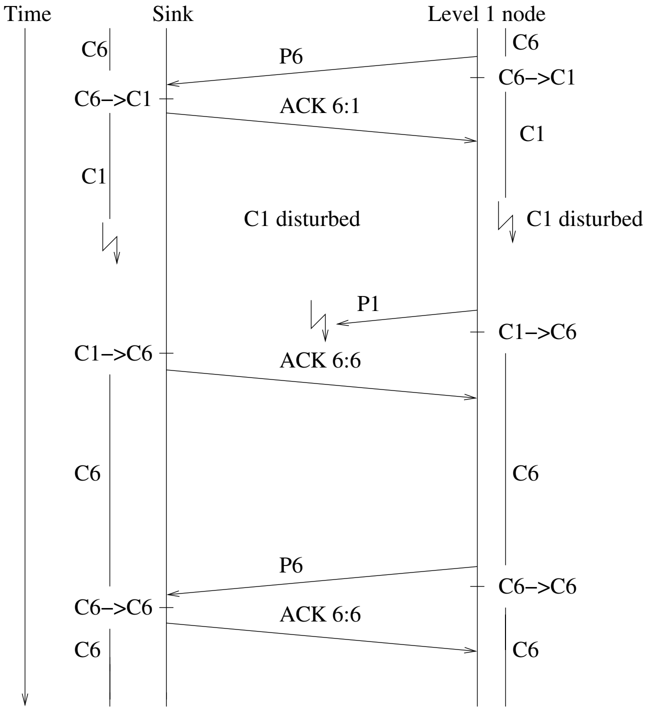

Fig. 5. Channel switching during interference

The figure shows that when the sink notices that it does not receive a message on a certain channel, it announces that it will send future acknowledgements on the channel that is not interfered. This way it is also indicated on which channel the sender should transmit. Note that in the figure only the acknowledgments' fields for channel usage are shown. The acknowledgements also advertise received packets. For example, the second acknowledgement also advertises that the sink has not received the previous packet.

4 Implementation and Results

In this section we present results from experiments with real hardware using the Tmote Sky platform. We have implemented the proposed scheme in the Contiki operating system [3] above the broadcast layer of the Rime protocol stack [4], i.e. we turn the radio on and off and change channels at the application layer. The scheme could also be implemented in the MAC layer below the Rime stack. Our current implementation is not optimized in that it does not try to minimize the guard times of the RX slots.

4.1 Energy-efficiency of the proposed approach

We estimate the energy consumption of four nodes deployed in a chain. With our self-organizing approach, we simply set the maximum number of children per node to one to achieve a four-level network setup. The energy consumption is estimated using Contiki's software-based on-line energy estimation method [5]. We concentrate on the energy for radio listening as radio listening is the dominating factor for power consumption in WSNs [5]. We compare our protocol to X-MAC, a power-saving MAC protocol that is designed to run on top of the 802.15.4 physical layer [1]. X-MAC reduces the power consumption by switching the radio on and off at regular intervals. To send a packet, a node broadcasts a train of short strobe packets. The strobe packet train is long enough to allow all nearby devices to be switched on at least once. When receiving a unicast strobe, a receiver immediately sends a short acknowledgment packet allowing the sender to send the full packet.

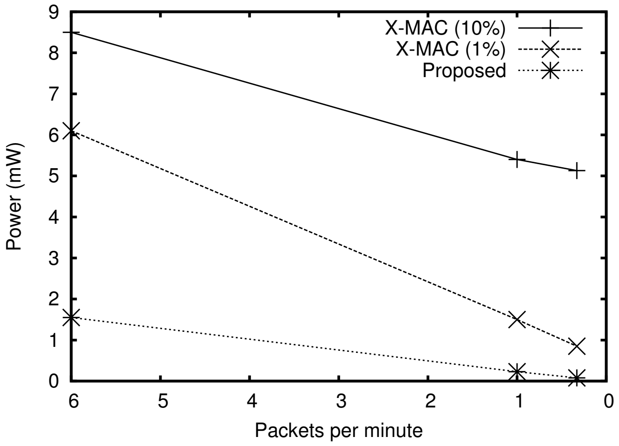

Fig. 6. Comparison of average radio listen power

Figure 6 shows the average power consumption for radio listening comparing X-MAC with the proposed protocol. Since our protocol has a constant radio listening time for each packet, the radio listening time decreases approximately linearly when less packets are sent. In all scenarios, the proposed protocol performs better than X-MAC.

Figure 6 also shows that with the same duty cycle, X-MAC consumes less energy when there is less traffic, i.e. a duty 10% duty cycle will not per se extend the lifetime of the network with a factor of 10 but that the lifetime extension depends on the traffic volume. The reason for this is that after the intended receiver of a packet has indicated that it is ready to receive a packet, the receiver must have its radio turned on until it has received the packet. This

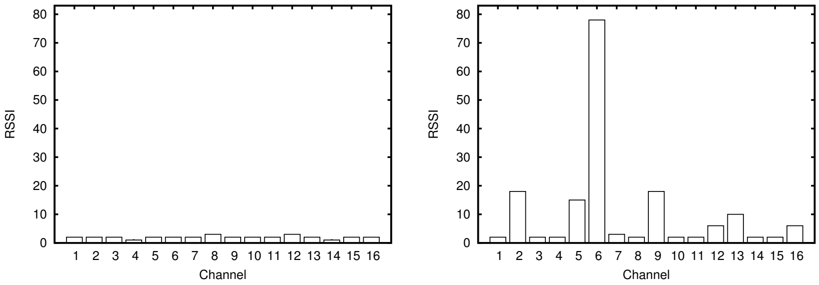

Fig. 7. RSSI levels without an interfering node (left) and RSSI levels with a nearby node interfering on channel 6 (right).

task consumes much more energy than listening for the short X-MAC strobe packets.

The results indicate that our proposed protocol is suitable for data collection applications with very low duty cycles, for example applications collecting temperature values in buildings.

4.2 Channel Switching

Channel Switching Overhead Using a microcontroller hardware timer, we measure the duration of our channel switching radio driver function call. The function waits until any pending transmission is finished, and then commands the radio chip to change operating frequency. Finally, the function activates the new frequency by resetting the radio to receive mode. Without any pending transmissions, our results show that the duration is approximately 131 μseconds. These experiments demonstrate that the additional delay introduced by switching radio channels is not significant.

Measuring Channel Quality We have implemented a small procedure to measure channel quality based on the received signal strength indicator (RSSI). Figure 7 shows the RSSI levels of the 16 channels with and without a disturber node on channel 6. The increase of the RSSI on channel 6 when the disturber is turned on is clearly visible.

Increased Robustness with Multiple Channels In the next experiment, we use a sink node, a sender and a disturber node. The latter is programmed to cause interference by sending packets back-to-back on a predefined channel. The sink sends acknowledgements every 10 seconds. Therefore, the sender transmits 6 packets per minute. The algorithm is the same as described in Section 3. Five minutes and 25 seconds after the beginning of the experiment, the disturber interferes with the packet transmission on channel 6.

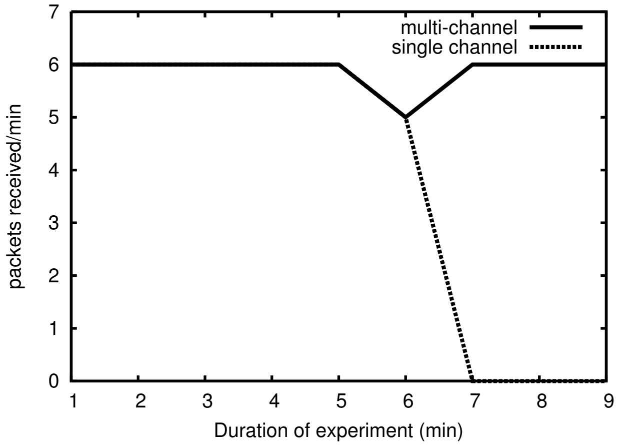

Fig. 8. Using multiple channels, the communication between sender and sink can be sustained despite interference on one of the channels.

Figure 8 shows that an application that uses a single channel only is not robust against channel interference. As expected, when multiple channels are used, our channel switching algorithm takes the loss of one packet as an indication to move future communication from the corresponding channel. Hence, after one packet is lost, all packets arrive reliably at the sink again. Our scheme expects that retransmissions of lost packets are performed during the node's next TX slot. For example, the automatic retransmissions in 802.15.4 cannot be used, since our acknowledgements are not unicast packets.

We have not yet integrated the channel quality measurement procedure into our protocol. This would allow us to proactively stop using a channel when it is interfered instead of using packet loss as the indication for interference. Since the quality measurement procedure is very fast, its energy consumption is almost negligible.

5 Related Work

One of the reasons for reduced robustness and reliability in sensor networks are temporal disturbances/uncertainties in the radio medium. In 2003, experiments by Zhao et al. have demonstrated the existence of temporal disturbances, i.e. they have shown that packet reception rates of sensor nodes vary significantly over time even in quite static environments [17]. Petrova et al. have measured that different 802.11 channels interfere with a set of different 802.15.4 channels [13].

While modern low-power radios such as the IEEE 802.15.4-compliant CC2420 are available, so far there are only a few attempts to use the available channels. Many of these have leveraged multiple channels to increase throughput and increase performance. Zhou et al. have focused in their simulation results on throughput, energy efficiency and channel access delay [18]. Durmaz Incel et al. have shown that a multi-channel version of LMAC increases performance proportional to the number of available frequencies compared to the single-channel version [6]. Liang et al. have reduced the dissemination time of large objects in wireless sensor networks by utilizing multiple channels [10]. In contrast to these efforts, we use multiple channels to increase robustness and reliability by switching to a different channel if interference makes it impossible to use the selected channels.

One of the most energy-efficient convergecast protocols is Dozer [2]. In contrast to our approach, Dozer does not build a collision-free delivery tree. The same is true for Twinkle [9], DMAC [11] and the approach proposed by Gandham et al. [8]. The latter tries to reduce latency by minimizing the number of required timeslots. Zhang et al. focus on bursty convergecast where large bursts of packets are transmitted [16]. None of these protocols uses multiple channels.

6 Conclusions

In this paper, we have presented a convergecast protocol that dynamically builds a collision-free tree based on information in the acknowledgements. The protocol also performs channel switching to reduce interference problems. Our simulations and experiments with real hardware have demonstrated the effectiveness of the proposed approach.

Acknowledgments

This work was funded by the Swedish Energy Authority.

References

1. M. Buettner, G. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *ACM SenSys*, Boulder, USA, November 2006.
2. N. Burri and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *6th international conference on Information processing in sensor networks (IPSN 2007)*, Cambridge, USA, April 2007.
3. A. Dunkels, B. Gönvall, 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*, Tampa, Florida, USA, November 2004.
4. 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.
5. A. Dunkels, F. Österlind, N. Tsiftes, and Z. He. Software-based on-line energy estimation for sensor nodes. In *4th workshop on Embedded networked sensors (EmNets'07)*, pages 28–32, 2007.

6. O. Durmaz Incel, S. Dulman, and P. Jansen. Multi-channel Support for Dense Wireless Sensor Networking. In *European Conference on Smart Sensing and Context (EuroSCC)*, October 2006.
7. Amre El-Hoiydi, Jean-Dominique Decotignie, Christian C. Enz, and E. Le Roux. wisemac, an ultra low power mac protocol for the wisenet wireless sensor network. In *ACM Conference on Networked Embedded Sensor Systems (SenSys 2003)*, pages 302–303, 2003.
8. S. Gandham, Y. Zhang, and Q. Huang. Distributed Minimal Time Convergecast Scheduling in Wireless Sensor Networks. *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS)*, July 2006.
9. B. Hohlt and E. Brewer. Network Power Scheduling for TinyOS Applications. *IEEE Int. Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2006.
10. C.J.M. Liang, R. Musaloiu-E, and A. Terzis. Typhoon: A Reliable Data Dissemination Protocol for Wireless Sensor Networks. In *Proceedings of the Fifth European Conference on Wireless Sensor Networks (EWSN2008)*, Bologna, Italy, January 2008.
11. G. Lu, B. Krishnamachari, and C. S. Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In *International Parallel and Distributed Processing Symposium (IPDPS)*, April 2004.
12. F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In *Proceedings of Proceedings of the First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006)*, Tampa, Florida, USA, 2006.
13. M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella. Performance study of IEEE 802.15. 4 using measurements and simulations. In *Proceedings of IEEE WCNC*, April 2006.
14. S.Y. Shin, S. Choi, H.S. Park, and W.H. Kwon. Packet Error Rate Analysis of IEEE 802.15. 4 under IEEE 802.11 b Interference. In *International Conference on Wired/Wireless Internet Communications (WWIC 2005)*, May 2005.
15. A. Tyamaloukas and J. J. Garcia-Luna-Aceves. Channel-hopping multiple access. In *IEEE International Conference on Communications (ICC)*, June 2000.
16. H. Zhang, A. Arora, Y. Choi, and M.G. Gouda. Reliable bursty convergecast in wireless sensor networks. *Computer Communications*, 30(13):2560–2576, 2007.
17. J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *The First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, California, November 2003.
18. G. Zhou, C. Huang, T. Yan, T. He, J.A. Stankovic, and T.F. Abdelzaher. MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks. In *Proceedings of IEEE Infocom*, Barcelona, Spain, April 2006.