

Poster Abstract: Self-organizing, Collision-free, Multi-channel Convergecast

Thiemo Voigt
 Swedish Institute of Computer Science
 thiemo@sics.se

Abstract—Most of the existing sensor network deployments are convergecast applications where data is transmitted from multiple sources to a sink. In this paper, we present the design of a self-organizing, collision-free multi-channel convergecast protocol. We present experiments that show that our protocol works as expected and demonstrate its energy-efficiency for low duty cycle applications by comparing it to X-MAC.

I. 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. One of the most prominent convergecast protocols is D-MAC that introduced the notion of staggered slots [1]. Building on the notion of staggered slots, we present a set of ideas that enable distributed slot assignment to achieve a collision free convergecast protocol without time synchronization as well as channel-switching in case a currently used channel is interfered. Our basic idea is to use acknowledgements for slot assignment and synchronized node wake-up scheduling. WiseMAC has exploited a similar idea by including the sampling schedule offset into acknowledgements [2]. By adding control information in the acknowledgement, we also provide synchronized channel switching to increase robustness towards interference. Simulations with the COOJA simulator [3] indicate that our scheme handles interfered channels as expected. We present experiments on real hardware that demonstrate the energy-efficiency of our protocol. The protocol's ability to synchronize node wake-up without explicit time synchronization seems very appealing and useful for data collection applications that demand a low duty cycle and long lifetime.

II. KEY DESIGN IDEAS

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 D-MAC 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 contains information about the next scheduled wake-up time. This way, no explicit time synchronization needs to be performed since only relative time is of importance.

By adding offsets for different nodes into the ACK packet, we can extend the basic scheme to let a parent such as the sink

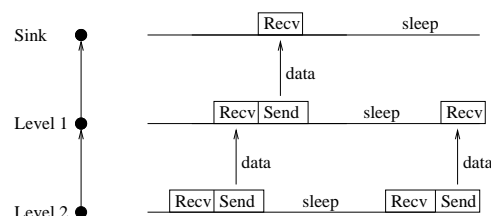


Fig. 1. Staggered wake-up a la DMAC

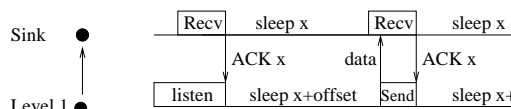


Fig. 2. Basic synchronization scheme

node assign slots to its children. Figure 3 demonstrates how the sink assigns different offsets to its children. 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. However, the send and receive offsets may no longer be aligned. Hence, a node needs to turn its radio on and off more often which implies a little energy overhead that should be well compensated for by avoiding collisions.

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.

III. IMPLEMENTATION AND RESULTS

In this section we present results from experiments with real hardware using the Tmote Sky platform. We have implemented

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

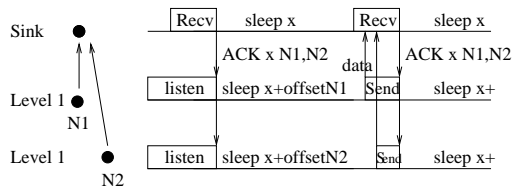


Fig. 3. On-demand slot assignment

the proposed scheme in the Contiki operating. Our current implementation is not optimized in that it does not try to minimize the radio on-times but uses rather large slots.

Energy-efficiency We estimate the energy consumption of four nodes deployed in a chain. For our self-organizing approach, we simply set the maximum number of children per node to one. The energy consumption is estimated using Contiki's software-based on-line energy estimation method [4]. We concentrate on the energy for radio listening as radio listening is the dominating factor for power consumption in WSNs [4]. We compare our protocol to X-MAC, a popular energy-efficient MAC protocol [5].

Packet freq.	6 packets/min	1 packet/min	0.33 packet/min
X-MAC (10%)	8.5 mW	5.4 mW	5.13 mW
X-MAC (1%)	6.1 mW	1.5 mW	0.85 mW
proposed	10.8 mW	2.0 mW	0.65 mW

TABLE I
COMPARISON OF AVERAGE RADIO LISTEN POWER

Table 1 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. When the number of packets sent is very low the proposed protocol performs better than X-MAC with a low duty cycle. The experiments also show that with the same duty cycle X-MAC consumes less energy when there is less traffic. 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 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.

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 4 announces that the sink will send the next ACK on Channel 6 and the successive one on Channel 1. Figure 4 is produced from the log of a simulation in the COOJA [3] 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 announcements sent by the level 1 node are not shown.

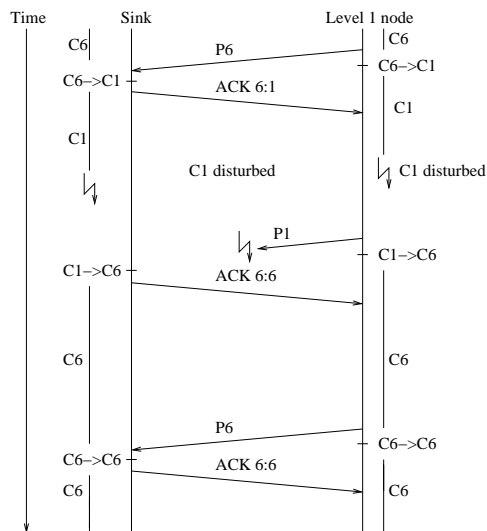


Fig. 4. Channel switching during interference

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 acknowledgements' fields for channel usage are shown. The acknowledgements also advertise received packets. E.g. the second acknowledgement also advertises that the sink has not received the previous packet.

IV. CONCLUSIONS

In this paper, we have presented a convergecast protocol that dynamically builds a collision-free tree based on information in the acknowledgements and performs channel switching to reduce interference problems.

ACKNOWLEDGMENTS

This work was funded by the Swedish Energy Authority.

REFERENCES

- [1] 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)*, 2004.
- [2] A. El-Hoiydi, J.-D. Decotignie, C. C. Enz, and E. L. Roux, "wisemac, an ultra low power mac protocol for the wisenet wireless sensor network," in *SenSys*, 2003, pp. 302–303.
- [3] 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.
- [4] A. Dunkels, F. Österlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors*, 2007, pp. 28–32.
- [5] M. Buetner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *ACM SenSys*, Nov. 2006.