Practical Network Coding in Sensor Networks: Quo Vadis?

Thiemo Voigt[1], Utz Roedig[2], Olaf Landsiedel[3,4], Kasun Samarasinghe[1], Mahesh Bogadi Shankar Prasad[1]

[1] Swedish Institute of Computer Science (SICS), Kista, Sweden
[2] Lancaster University, Great Britain,
[3] Chalmers University of Technology, Sweden
[4] KTH Royal Institute of Technology, Sweden

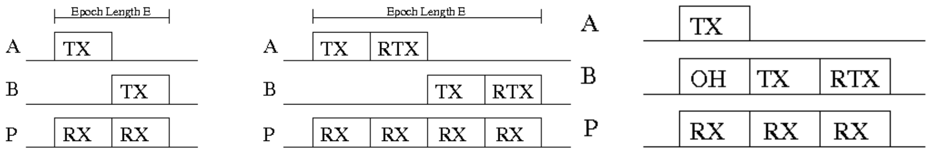
Abstract. Network coding is a novel concept for improving network capacity. This additional capacity may be used to increase throughput or reliability. Also in wireless networks, network coding has been proposed as a method for improving communication. We present our experience from two studies of applying network coding in realistic wireless sensor networks scenarios. As we show, network coding is not as useful in practical deployments as earlier theoretical work suggested. We discuss limitations and future opportunities for network coding in sensor networks.

1 Network Coding in Wireless Sensor Networks

Network Coding was introduced by Ahlswede et al. [1], proving that it can increase multicast capacity. Since then, it has been investigated in several different networked scenarios which demand different traffic characteristics. Most previous research has focused on theoretical aspects of applying network coding to sensor networks. There are, however, also more practical examples of applying network coding in wireless networks. Network coding has, for example, demonstrated its usefulness in the general networking domain (e.g., Katti et al. [2]). Hou et al. have proposed AdapCode [3], a network-coded variant of the Deluge protocol [4]. Aoun et al. specifically target real-time wireless sensor networks [5]. They use network coding together with packet skipping to improve on-time goodput in real-time WSNs. Their topology, however, is not applicable to data collection.

While network coding for data dissemination can increase reliability, it is harder to apply network coding for convergecast, probably the most important traffic paradigm in wireless sensor networks where data is collected from multiple sources and transported to one or more data sinks.

We have worked on two efforts to apply network coding in wireless sensor networks. In the first study, we have applied network coding to extend GinMAC, a state-of-the-art MAC protocol [6]. In the second, we have tried to apply network coding for reliable and low delay communication in network with very high packet loss rates [7]. In the second case it has turned out that network coding cannot live up to the expectations while in the first case in all scenarios we studied, network



(a) Schedules without NC (0 and 1 retransmissions) (b) Schedule with NC

Fig. 1: Schedules with and without Network Coding (NC). NC compresses the schedule while still enabling retransmissions.

coding can be replaced with simple overhearing and the retransmission of the lost packets which avoids the complexity of network coding. In the remainder of this paper, we detail our studies and discuss limitations and usefulness of network coding in sensor networks.

2 Network Coding in GinMAC

We have applied network coding in GinMAC [6], the MAC layer of the GINSENG system designed for industrial monitoring and control. In industrial process automation and control networks it is common that many nodes monitoring the same production process and are hence deployed in close proximity to each other. Thus, most nodes are in interference range of all other nodes in the network and many nodes are also capable of overhearing packets. GinMAC is a single channel state of the art TDMA-based MAC layer for performance-controlled sensor networks [8]. GinMAC uses an offline process to dimension a network before deployment. The heart of the dimensioned network is a TDMA schedule with an epoch length consisting of E slots. There are three types of slots, namely *basic* slots for transmission (TX) and reception (RX), *additional* slots for retransmissions (RTX) to increase reliability and *unused* slots to decrease the duty cycle. The schedule determines latency, power consumption and reliability. The latter is increased by adding more retransmission slots.

We employ overhearing as a mechanism for tuning network reliability and latency. Due to external interference a node may not be able to deliver a packet to a parent node. However, a neighbouring node may be able to overhear the packet and then try to deliver the packet on its behalf. Thus, instead of the sender retransmitting packets, neighbouring nodes that overhear the initial transmission take care of the retransmission. Obviously, a node that overhears packets will now need capacity to transmit these. In order to avoid allocation of additional transmission slots we propose network coding to combine several overheard packets into a single retransmission. Thus, we use overhearing and network coding for retransmissions reducing the overall number of transmission slots.

Figure 1 shows the resulting schedules with and without network coding for a binary tree with the leave nodes A and B. In the example node A has the first transmission slot. With network coding node B overhears A's transmission in this transmission slot. Thereafter, node B transmits its own packet in its allocated slot. Finally, node B uses the retransmission slot in which it transmits a network coded packet that contains the information of the packet transmitted by A and B. For example, packets from A and B can be combined using simple binary XOR coding. The root node will receive A and B's packets if one original transmission is received and the network coded retransmission is received (both packets are also received if both original packet transmissions are received).

As a result we have now constructed a schedule in which both nodes A and B have a way to retransmit a packet despite that we provision only one retransmission slot. Thus, with network coding less slots are provisioned while we still provide a retransmission option for all nodes. The slight drop in reliability compared to the original scheme with one retransmission may be well acceptable for some application scenarios since a gain in transmission latency is achieved as the schedule is compressed. Figure 1 shows that the schedule compression for the binary tree is 3 : 4 (0.75). In our previous work we have shown that we achieve approximately similar error packet reception rates for GinMAC with and without network coding and that for high packet loss rates we can also achieve a reduction in power consumption [6].

3 Network Coding for Reliable, Low-delay Communication

Many sensor network applications, in particular sensor actuator networks, require a certain delay and reliability to enable control loops on top of a wireless sensor network. Network coding promises highly reliable communication. We discuss our lessons learned from integrating it into a collection tree application [7]. While network coding increased reliability in some scenarios, we noticed the following two key limitations: (1) strongly increased delay, and (2) high overhead due to limited lack of adaptability:

Delay: Commonly, WSN applications operate at low data rates: often, a node generates packets in the order minutes and the whole network traffic is in the order of a couple of packets per second. However, to achieve a robust interleaving and mixing of packets for network coding, one commonly applies a coding scheme across multiple packets. Thus, nodes have to wait for these packets to accumulate, which significantly increases delay. For example, in today's collection protocols such as CTP [9] or Contiki Collect [**?**], one sees typical network delays in the order of milliseconds. However, waiting for enough packets for coding to accumulate takes a couple of seconds. Thus, the delay added by network coding is several order of magnitude higher than the network delay. Hence, network coding seems only beneficial when the applications do not have delay constraints.

Overhead and lack of adaptability: As network coding is an end-to-end scheme, its feedback is limited when compared to per-hop acknowledgments. Thus, it is difficult for the source to adjust its coding rate to changing network conditions. Reflecting this, we had to employ a redundancy level in our coding scheme that matches the maximum expected loss rate. As a result, this approach adds a large overhead compared to the average scenario. Especially, when keeping in mind, that links in WSNs are often bursty [10], leading to high, short-term packet losses.

4 Limitations and Applicability of Network Coding

In the previous section we have already outlined some limitations of applying network coding for data collection. While our results with GinMAC (see Section 2) are encouraging, we later realized that we do not need network coding in this scenario. If either packet A or packet B is lost, we can simply retransmit the lost packet. If both packets A and B are lost, then network coding does not help either and it is better to retransmit only one of the lost packets. We have not found larger topologies where we could use network coding while it was not possible to do the same with overhearing and standard packet retransmissions only. For the single-sink scenario, we have found one use of network coding: when we do not apply acknowledgements which might be useful when the schedule needs to be compressed to a minimum but still some retransmissions should be enabled. In particular for trees where each node can have more than one sibling (e.g., ternary trees), network coding may be useful when acknowledgements are not enabled. Furthermore, in multi-sink scenarios, similar to the original network coding scenarios, we expect network coding to be beneficial.

5 Conclusions

In this paper, we have demonstrated some drawbacks of applying network coding in real-world sensor network scenarios. While we have pointed out some limitations we have also shown some potential use cases.

6 Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 224282, SSF and VR. This work has been partially supported by CONET, the Cooperating Objects Network of Excellence.

References

1. R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 2000.
2. S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: practical wireless network coding. In *SIGCOMM*, 2006.
3. I. H. Hou, Y. E. Tsai, T. F. Abdelzaher, and I. Gupta. AdapCode: Adaptive Network Coding for Code Updates in Wireless Sensor Networks. In *IEEE INFOCOM*, 2008.

4. J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *SenSys'04*, November 2004.
5. M. Aoun, A. Argyriou, and P. van der Stok. Performance evaluation of network coding and packet skipping in ieee 802.15.4-based real-time wireless sensor networks. In *EWSN*, 2011.
6. K. Samarasinghe, T. Voigt, L. Mottola, and U. Roedig. Poster abstract: Network coding with limited overhearing. In *EWSN*, 2011.
7. M. Bogadi SP. Reliable, low delay communication in wireless sensor networks. Master's thesis, KTH, Stockholm, 2011.
8. J. Brown, B. McCarthy, U. Roedig, T. Voigt, and C. Sreenan. Burstprobe: Debugging time-critical data delivery in wireless sensor networks. In *European Conference on Wireless Sensor Networks (EWSN)*, Bonn, Germany, February 2011.
9. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection Tree Protocol. In *ACM SenSys*, 2009.
10. J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, M. Durvy, JP Vasseur, A. Terzis, A. Dunkels, and D. Culler. Beyond Interoperability: Pushing the Performance of Sensornet IP Stacks. In *ACM SenSys*, Seattle, USA, November 2011.
11. K. Srinivasan, M. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *ACM SenSys*, 2008.