Competition: CRYSTAL Clear: Making Interference Transparent

Matteo Trobinger[1], Timofei Istomin[1,2], Amy L. Murphy[2], Gian Pietro Picco[1]
[1] University of Trento, Italy {matteo.trobinger, timofei.istomin, gianpietro.picco}@unitn.it
[2] Bruno Kessler Foundation, Italy murphy@fbk.eu

Abstract

To address the challenges of the EWSN dependability competition we extended our synchronous transmission protocol, CRYSTAL, with techniques to mitigate interference—channel hopping and noise detection—and with the capability to deliver aperiodic events to multiple actuators.

1 Core Approach: The CRYSTAL System

Synchronous transmissions have recently gained popularity for wireless sensor networks due to their ability to provide fast, energy-efficient, reliable network-wide dissemination.

Synchronous transmission protocols, pioneered by Glossy [2], build on two properties of the IEEE 802.15.4 PHY: constructive interference and the capture effect. These occur when packet transmissions by neighboring nodes are initiated within a tiny temporal interval (0.5μs and 160μs, respectively) and yield a more reliable transmission instead of a collision. Constructive interference works when the packet is the same, yielding high reliability due to the combination of the identical signals; the capture effect, instead, works with different packets, one of which is received with a probability depending on neighbor density and signal strength.

These properties are exploited in Glossy to construct network-wide floods that are extremely *i) fast*, as each node receiving a packet immediately rebroadcasts it, preserving the required tight timing; *ii) reliable*, due to the aforementioned PHY-level properties, and the inherent spatial and temporal redundancy of flooding. To further increase reliability, packets are retransmitted N times by each node; the value of N is the main knob to control the trade-off between reliability and energy consumption.

The CRYSTAL [3] system builds a schedule atop Glossy to support *aperiodic* and *sparse* data collection from the nodes to the sink, with near-perfect reliability, low latency, and ultra-low power consumption.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2018
14–16 February, Madrid, Spain

ISBN: 978-0-9949886-2-1

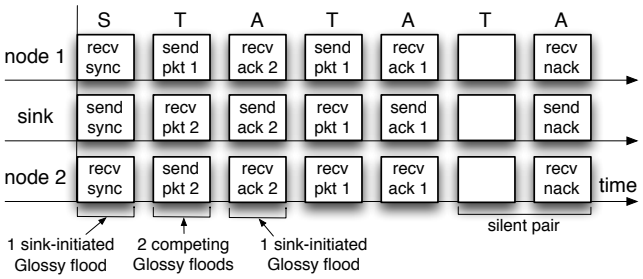

Figure 1. CRYSTAL in a nutshell.

CRYSTAL builds a network-wide transport protocol, in which *i)* a transmission (T) slot is used by U concurrent senders to disseminate their packet; these floods "compete" until, thanks to the capture effect and Glossy redundancy, one reaches the sink with high probability; *ii)* a subsequent acknowledgment (A) slot is used by the sink to flood the identifier of the sender whose packet it received, informing other senders whether re-transmission is needed because their packet was "overcome" by another or no packet was received at the sink.

Figure 1 illustrates the concept in a simplified setting with only 2 nodes and the sink. A synchronization (S) phase is performed at the beginning of each epoch to ensure time synchronization. Communication occurs via the aforementioned TA pairs, which are repeated with fewer and fewer senders until all have successfully transmitted their packet and the network can sleep for the rest of the epoch. This termination condition is in principle easily identified by the first *silent pair*, i.e., one without transmissions in T and whose A contains a negative acknowledgment. In practice, matters are complicated by packet losses in either T or A, which may cause a node or the sink to become prematurely inactive. Therefore, CRYSTAL detects termination after R *consecutive* silent pairs; larger values improve reliability but with higher energy consumption.

2 CRYSTAL for the competition

The competition scenario requires detecting events on several sensor nodes and quickly informing the actuators interested in these events under strong external interference. CRYSTAL efficiently and reliably supports data collection from multiple sources to a single destination, however it must be extended to deliver the events to multiple actuators. Moreover, to address the extreme interference scenarios of

the competition, we enhanced CRYSTAL with interference resilience techniques.

2.1 Multiple Destinations

To deliver events to multiple actuators we extend CRYSTAL as follows. At the beginning of each epoch, the sensor nodes measure the input GPIOs. If the state of the GPIOs has changed since the last epoch, the sensor node generates a data packet and starts transmitting in T slots, until acknowledged by the sink, following the normal CRYSTAL procedure. Additionally, however, the acknowledgement packets will piggyback the sensor data (GPIO state).

Thanks to the broadcast nature of both T and A floods, ideally all actuators will have an updated view of the status of the sensor nodes before the end of each epoch. However, due to interference, some of the packets may not be delivered to all actuators, keeping some nodes unaware of some events. Moreover, since the acknowledgements contain only a partial knowledge of the system state (the GPIO state of a single sensor), the loss of this packet will cause inconsistency in the actuators' view. Therefore, to compensate for the losses and eventually bring all the nodes to a consistent global view on the system state, the sink will also flood the current state of the GPIO of *all* the sensor nodes in the synchronisation (S) phase and/or in additional dedicated slots.

2.2 Interference Resilience

We also extended CRYSTAL to include two interference resilience techniques that aim to *escape* interference and to *fight* it after detecting its presence. Differently from the three winning solutions of the 2017 competition [4, 1, 5], our techniques have been designed *on top* of Glossy.

Escaping Interference: Channel Hopping. Exploiting frequency diversity is a well-known technique for interference resilience. Interference usually affects only some of the 16 channels available. Therefore, a network-wide channel-hopping sequence can be exploited to enable subsequent TA pairs to move to different channels, reducing the probability that consecutive ones execute on noisy channels. This simple modification does not affect any CRYSTAL parameters.

Channel hopping is driven by the S phase; the channels of TA pairs in the epoch depend on the S channel, itself based on a predefined sequence. This aligns all nodes to the same, rotating channel at the beginning of each epoch, independent of the number of TA phases they executed in the previous one.

A key decision is which channel to use next. We expect to optimise our hopping technique taking advantages of information acquired during the preparation phase.

Fighting Interference: Noise Detection. Our next technique relies on the ability to detect abnormally high noise levels. Originally in CRYSTAL, the distributed termination condition relied on counting *silent pairs* and missed acknowledgements. Under high noise, these *missing-packet* conditions occur often due to interference. If noise strikes during the T phase close to the sink, the sender will retransmit the packet in the next T slot. If the sink still does not receive the packet in R consecutive T slots, it mistakenly detects termination and puts the whole network to sleep. Instead, noise in the network periphery may cause a node to timeout and go to sleep before the rest of the network, when several data and acknowledgement packets in a row are lost. In both cases, data may remain un-delivered because termination was falsely detected.

Adding noise detection and changing termination conditions fights these cases. Intuitively, in the presence of *high-noise* missing packets do not count towards termination, keeping the network awake and allowing more opportunities for data and acknowledgments to escape the interference. More precisely, we made the following modifications to the termination rule in [3]:

- define R_{noise} as the maximum number of *consecutive* slots *i)* without a packet *and ii)* with high noise.
- change the termination rule at the sink; the network goes to sleep when *either i)* R non-noisy no-data T slots occur since the last received data, *or ii)* $max(R, R_{noise})$ consecutive noisy no-data T slots occur.
- change the termination rule elsewhere; a node goes to sleep when *either i)* it receives a sleep command from the sink, *or ii)* it detects Z non-noisy no-data slots since the last packet received in T or A, *or iii)* $max(Z, R_{noise})$ consecutive noisy, silent A slots occur.

Fighting *and* Escaping Interference. Although each noise-resilience technique improves performance along some dimension, it is only through their combination that we expect strong interference to be effectively overcome with very low energy consumption. Indeed, frequency diversity reduces the probability of the sink to be exposed, in consecutive TA pairs, to high noise levels, mitigating the main drawback of noise detection, namely keeping the network awake also when it is not strictly needed. On the other hand, the ability to detect and react to noise is helpful in reducing packet loss when hopping from one bad channel to another one.

3 Summary

With extensions for interference resilience and delivery to multiple destination, CRYSTAL offers a general purpose transport protocol suitable to the aperiodic transmission scenarios expected at the 2018 EWSN competition.

4 Acknowledgements

This work is partially funded by CrystalClear, a project funded through the H2020 project WiSHFUL (Grant agreement n645274).

5 References

[1] A. Escobar et al. Competition: RedFixHop with Channel Hopping. In *Proc. of EWSN*, 2017.

[2] F. Ferrari et al. Efficient Network Flooding and Time Synchronization with Glossy. In *Proc. of IPSN*, 2011.

[3] T. Istomin et al. Data prediction + synchronous transmissions = ultra-low power wireless sensor networks. In *Proc. of SenSys*, 2016.

[4] R. Lim et al. Competition: Robust Flooding using Back-to-Back Synchronous Transmissions with Channel-Hopping. In *Proc. of EWSN*, 2017.

[5] B. A. Nahas and O. Landsiedel. Competition: Towards Low-Power Wireless Networking that Survives Interference with Minimal Latency. In *Proc. of EWSN*, 2017.