

Competition: Adaptive Software Defined Scheduling of Low Power Wireless Networks

Michael Baddeley, Aleksandar Stanoev, Usman Raza, Yichao Jin, Mahesh Sooriyabandara
Toshiba Research Europe Ltd.

michael.baddeley, aleksandar.stanoev, usman.raza, yachao.jin, mahesh@toshiba-trel.com

Abstract

We address the considerable and varied challenges of the EWSN 2019 dependability competition through a software-defined approach to synchronous flooding, whereby a flexible synchronous flooding architecture can instantiate, tailor, and schedule multiple different protocols based on information gathered in an initial configuration phase. In this manner, a single framework can adapt protocol behaviour by building high-level logic on top of lower layer synchronous flooding primitives, in order to meet specific requirements arising from the periodic and aperiodic nature of transmissions from multiple data sources. The adaptability afforded by this approach allows an optimal synchronous flooding protocol solution to be configured and run whatever the application requirements.

1 Introduction and Motivation

Flooding protocols based on concurrent transmissions present a novel approach to networking in low-power wireless nodes. Since the proposal of Glossy [4], multiple protocols have exploited so-called constructive interference and the capture effect to transmit periodic [3] and aperiodic [5] data over the multi-hop mesh networks. Our solution provides an highly-optimized synchronous flooding layer, whilst separating this from the higher-level protocol and configuration layers. This allows both shared (multi initiator) and dedicated (single initiator) synchronous flooding primitives to be instantiated and scheduled by higher layers. By linking these primitives with higher-level logic, guards, and offsets, the framework is able to easily configure synchronous flooding protocols to meet application requirements and fulfill the competition scenarios Figure 1:

- *Collection scenario*: Up to eight source nodes communicating to a single destination node over a multi-hop network (*multicast-to-point* (MP2P) traffic).

- *Dissemination scenario*: Up to eight source nodes disseminating actuation commands to a unique set of destinations nodes in the network (*point-to-multipoint* (P2MP) traffic).

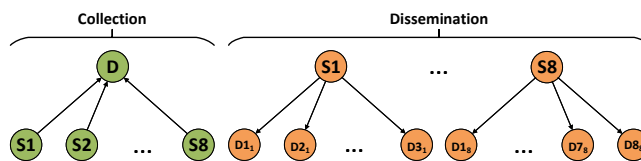


Figure 1. Collection and dissemination scenarios

2 High-Level Framework

The adaptive high-level framework underpinning this competition entry is based on the flexible synchronous flooding solution provided by Atomic-SDN [1], a Software Defined Networking SDN architecture for low-power wireless based on synchronous flooding control. Figure 2 shows the separation of configuration, protocol, and synchronous flooding layers.

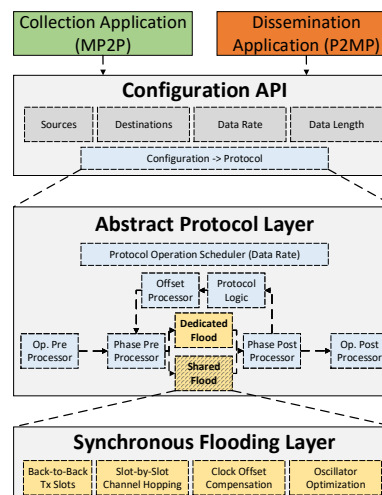


Figure 2. Adaptive synchronous flooding stack.

Starting from the low-level synchronous flooding layer and through to the higher-layers, the framework works in the following manner:

1. Scheduled flooding *phases* are designated as either dedicated (single source) or shared (multi source).
2. These *phases* can be linked together to form a protocol *operation*.
3. A protocol *operation* is defined by the offsets, guards, and logic blocks applied in the Abstract Protocol Layer.
4. The behaviour of the Abstract Protocol Layer is defined by the information passed from the Configuration API.

3 Low-Level Optimization

We perform extensive optimization of the synchronous flooding layer based on the particular idiosyncrasies of the MSP430 and CC2420 hardware used within the competition.

- *Back-to-back transmissions*: First proposed in the 2017 competition [6], we perform back-to-back transmissions after a node first successfully receives. This allows a packet to rapidly propagate across the network, as opposed to an interleaving Rx-Tx model.
- *Slot-by-slot random channel hopping*: A pseudo-random array is generated before every protocol operation, allowing the synchronous flooding layer to hop to a different channel in every slot. Due to the spatially diverse and high levels of interference experienced in the competition, this rapid channel hopping allows floods to survive propagation across the entire network.
- *Clock-offset estimation*: We employ on-the-fly clock offset compensation [2] on the unstable MSP430 oscillator in order to mitigate the clock frequency deviations across the multi-hop network. This helps to maintain the $0.5 \mu\text{s}$ synchronization between nodes that is necessary to take advantage of concurrent transmissions.
- *Radio oscillator optimization*: If the period between finishing a flooding phase and starting the next allows time to turn off and then turn on the oscillator, then the radio is turned off in order to save energy.
- *Number of transmissions*: As the radio has already ramped up for transmission by the time a CRC check fails, it is necessary to skip the next Tx slot if the packet was corrupted. This, alongside a high network hop count

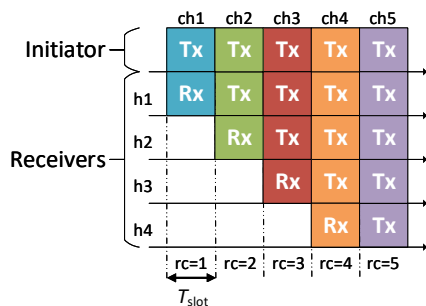


Figure 3. Low-level synchronous flooding phase (dedicated) with back-to-back transmissions and slot-by-slot hopping.

and completely jammed channels, necessitates a high number of transmission slots in each phase.

4 Protocol Operation

The high-level framework combined alongside the optimized synchronized flooding layer is used to construct optimized protocols (Figure 4) capable of handling both periodic and aperiodic data within each of the competition scenarios. In both protocols, the periodicity between each round of the protocol operation is calculated as accumulative duration of each phase, plus the maximum guard time needed to either read or write a packet from the EEPROM.

Collection Protocol: A *two-phase* network wide flooding operation is used in the collection protocol, consisting of a dedicated acknowledgement (ACK) phase and a shared transmission (TX) phase. In each round of the protocol operation, sources will compete in the TX phase to try and send their data to the destination, while the destination will use the ACK phase to acknowledge any sources heard in the previous round. For purposes of maintaining network synchronization, the ACK also serves as the synchronization phase, meaning the destination must act as the network timesync.

Dissemination Protocol: A *N-phase* network wide flooding operation is used in the dissemination protocol. An initial synchronization phase allows any node to serve as the network timesync, while all sources are scheduled a dedicated phase in which to act as the flood initiator. Sources will repeatedly transmit their last packet until the arrival of new data.

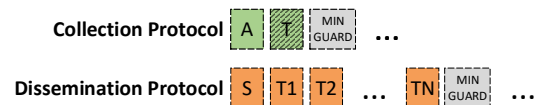


Figure 4. Phase-level depiction of optimized collection and dissemination protocols.

5 References

- [1] M. Baddeley, U. Raza, R. Nejabati, G. Oikonomou, M. Sooriyabandara, and D. Simeonidou. Utilizing Synchronous Flooding for Reliable and Scalable SDN in Low-Power Wireless Networks. *arXiv preprint arXiv:1812.08057*, 2018.
- [2] M. Brachmann, O. Landsiedel, and S. Santini. Keep the Beat: On-The-Fly Clock Offset Compensation for Synchronous Transmissions in Low-Power Networks. In *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pages 303–311, Oct 2017.
- [3] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. Low-power wireless bus. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, SenSys '12*, pages 1–14. ACM, 2012.
- [4] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 73–84, April 2011.
- [5] T. Istomin, A. L. Murphy, G. P. Picco, and U. Raza. Data prediction + synchronous transmissions = ultra-low power wireless sensor networks. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, SenSys '16*, pages 83–95. ACM, 2016.
- [6] R. Lim, R. Da Forno, F. Sutton, and L. Thiele. Competition: Robust flooding using back-to-back synchronous transmissions with channel-hopping. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN)*, 2017.