# Competition: Low-Power Wireless Bus Baseline

Fabian Mager* Romain Jacob† Reto Da Forno† Marco Zimmerling*
*Networked Embedded Systems Lab, TU Dresden, Germany
†Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland
firstname.lastname@tu-dresden.de firstname.lastname@tik.ee.ethz.ch

## Abstract

The Low-Power Wireless Bus (LWB) is a communication protocol for low-power wireless multi-hop networks that was published in 2012. It provides a shared-bus abstraction for higher layer protocols and hides the complexity of the underlying network. Internally, LWB uses fast and reliable Glossy floods to exchange information within the network, thus supporting arbitrary traffic patterns. The original code base was never officially released, so we are participating in this competition with two goals: 1) to provide a baseline against which other protocols can be compared, and 2) to make a clean and refactored version of the code base publicly available.

## 1 Motivation and Intent

Since its inception in 2016, the International EWSN Dependability Competition has become a respected annual venue where teams from academia and industry compete to find the best multi-hop low-power wireless solution in terms of end-to-end reliability, end-to-end latency, and energy consumption [6]. The competition has spurred highly innovative protocol designs and optimized implementations, which demonstrate the capabilities of low-power wireless technology in relevant scenarios. Over the years, the competition infrastructure has been revised towards a full-fledged benchmarking infrastructure that enables "... automated, seamless, and repeatable execution of experiments on real hardware ..." [5] in the presence of controlled, repeatable Wi-Fi interference [7].

In support of a community effort in developing the methodology for rigorous experimental comparison, reproducibility, and benchmarking of low-power wireless protocols [1], we participate with the Low-Power Wireless Bus (LWB) [2]. The original LWB paper appeared in the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys 2012), but the original code for the TelosB platform was never officially released. We thus pursue two complementary goals:

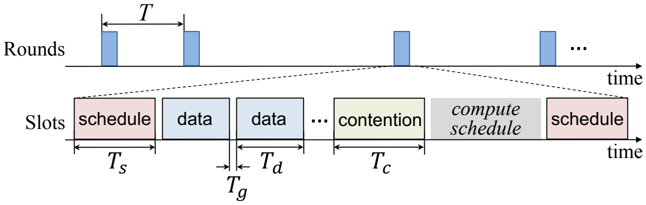

**Figure 1. Operation and timing parameters of LWB.**

1. provide a LWB reference implementation *whose functionality and performance are as equivalent as possible to the original LWB*, serving as a solid baseline for other contestants to quantitatively compare against;
2. make this reference implementation available to the public as open source for others to build upon and compare against in different scenarios and on other testbeds.[1]

To achieve both goals, we base our solution on the original LWB implementation and refrain from any modifications to the protocol logic *other than those absolutely necessary for the competition scenarios*. In particular, unlike the "typical" contestant, we do not consider improvements and optimizations that have been proposed since the original LWB paper, such as mechanisms that would dramatically increase the resilience of LWB to interference (*e.g.*, channel hopping) [4] or different scheduling policies that would result in significantly lower, bounded end-to-end latency [8]. In the following, Section 2 describes the basic operation of LWB and the very few changes we made to comply with the competition scenarios, while Section 3 details all important protocol settings we use.

## 2 LWB Operation and Minor Adjustments

LWB is a communication protocol for low-power wireless multi-hop networks [2]. It provides a higher-layer protocol with a shared-bus abstraction despite complex, dynamic multi-hop network topologies, and supports arbitrary traffic patterns (from point-to-point to all-to-all). Since LWB's protocol logic is independent of the network state (*e.g.*, link qualities, node locations, hop distances), it seamlessly supports high network dynamics because of, for example, mobile nodes.

LWB operates in globally time-triggered communication rounds with a varying round period $T$ depending on the application's traffic demands, as shown in Fig. 1. Each round

---

[1] Code available at `https://github.com/ETHZ-TEC/LWB-Baseline`

**Table 1. LWB parameters.**

| Symbol | Description | Value |
|---|---|---|
| $l$ | Payload size | Max. 64 B |
| $h$ | Network diameter | 7 |
| $T_{min}$ | Min. round period | 4 |
| $T_{max}$ | Max. round period | 30 |
| $d_{max}$ | Max. data slots per round | 50 |
| $n_s$ | Schedule retransmissions | 5 |
| $n_d$ | Data retransmissions | 3 |
| $T_s$ | Duration of a schedule slot | 42 ms |
| $T_d$ | Duration of a data slot | 31 ms |
| $T_c$ | Duration of a contention slot | 8 ms |
| $T_g$ | Gap between consecutive slots | 4 ms |

consists of several slots, where in each slot one node initiates a Glossy flood [3].[2] Glossy provides reliable one-to-all data exchange based on synchronous transmissions and accurate network-wide time synchronization. There are different kinds of slots in a round. A distinct host node uses the first and the last slot in a round to distribute the schedule for the current and the next round, respectively. Nodes use the contention slot to join the network by communicating their traffic demands to the host. Data slots are used to exchange application data. The scheduling policy used by the host determines the round period $T$ and which nodes are allocated a data slot in a round. The maximum number of data slots is $d_{max}$, while the round period $T$ ranges between $T_{min}$ and $T_{max}$.

For the competition, we use the minimum energy scheduler as described in the original LWB paper [2]. Despite a major overhaul of the code base, our implementation is *functionally equivalent* to the one used in [2], which includes the compression of schedules sent by the host that was not described in the original paper due to space restrictions. In comparison to the old code base, we integrated LWB into the latest Contiki-NG tree and ensured a cleaner hardware/software separation. Further, the CC2420 radio is put into "power down" mode whenever possible to save significant amounts of energy during idle periods. To ensure meaningful latency measurements, we deliver packets only at the end of a round (rather than between slots) when the application has enough time to acquire the MCU for processing the packets and other activities, as done in the original code. The next section describes all key parameter settings we use; we had to adjust some of them compared to the settings used in [2] to meet the requirements of the competition scenarios (*e.g.*, in terms of payload size and network diameter).

## 3 Parameter Settings

Table 1 gives an overview of the important parameter settings. Most time-dependent parameters are a consequence of the maximum payload size of 64 B, prescribed by the competition scenarios. The duration of a single transmission in Glossy over one hop ($T_{hop}$) for $l$ payload bytes is determined using:

$$T_{hop}(l) = T_{rd} + T_{sw} + T_{cal} + ((l + L_{hdr} + L_{ovh}) * R)\ [\text{us}] \quad (1)$$

This includes hardware-specific values such as $T_{rd} = 3\,\text{us}$ processing delay of the radio, $T_{cal} = 192\,\text{us}$ to calibrate the radio's internal oscillator and $R = 32\,{}^{us}\!/_{B}$ derived from the radio's data rate of 250 kbps. Additionally, there is a Glossy-specific software delay $T_{sw} = 24\,\text{us}$ after receiving a packet. The number of bytes traveling through the air consists of the actual number of payload bytes $l$, the synchronization and PHY header $L_{hdr} = 6\,\text{B}$ and $L_{ovh} = 6\,\text{B}$ for the LWB and Glossy headers.

The minimum duration of a slot during a LWB round depends on the network diameter $h$, the number of retransmissions $n$ and the payload size $l$.

$$T_{slot}(l) = (h + 2 * n - 2) * T_{hop}(l)\ [\text{us}] \quad (2)$$

We can calculate the duration of a schedule slot ($T_s$), respectively a data slot ($T_d$) by inserting the corresponding parameters into (2) with the maximum payload size $l = 64\,\text{B}$. With $T_{max} = 30$ as in the original paper and the minimum packet generation interval as well as the maximum number of sources given by the competition scenarios, we set $d_{max} = 50$. To deliver 50 messages from the destination nodes to the observer nodes of the testbed, we must reserve some time after the communication round, which affects the choice of $T_{min}$. Since we have not added a channel hopping mechanism, we remain on channel 26 throughout the experiments. The experiment setup, including the traffic patterns, is patched into the binary file by the competition infrastructure before the experiment begins. Taking this into account, we choose the first destination node in the first communication pattern as our LWB host node.

## 4 Acknowledgments

This work has been supported by the German Research Foundation (DFG) within the Cluster of Excellence cfaed, the EcoCPS and the NextIoT projects.

## 5 References

[1] C. A. Boano, S. Duquennoy, A. Förster, O. Gnawali, R. Jacob, H.-S. Kim, O. Landsiedel, R. Marfievici, L. Mottola, G. P. Picco, X. Vilajosana, T. Watteyne, and M. Zimmerling. Towards a Benchmark for Low-power Wireless Networking. In *IEEE Int. Workshop on Benchmarking Cyber-Physical Networks and Systems*, 2018.

[2] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. Low-power wireless bus. In *ACM Conference on Embedded Networked Sensor Systems*, 2012.

[3] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. In *ACM/IEEE Int. Conf. on Information Processing in Sensor Networks*, 2011.

[4] R. Lim, R. Da Forno, F. Sutton, and L. Thiele. Competition: Robust flooding using back-to-back synchronous transmissions with channel-hopping. In *Int. Conf. on Embedded Wireless Sys. and Networks*, 2017.

[5] M. Schuß, C. A. Boano, and K. Röemer. Moving Beyond Competitions: Extending D-Cube to Seamlessly Benchmark Low-Power Wireless Systems. In *IEEE Int. Workshop on Benchmarking Cyber-Physical Networks and Systems*, 2018.

[6] M. Schuß, C. A. Boano, M. Weber, and K. Römer. A Competition to Push the Dependability of Low-Power Wireless Protocols to the Edge. In *Int. Conf. on Embedded Wireless Systems and Networks*, 2017.

[7] M. Schuß, C. A. Boano, M. Weber, M. Schulz, et al. JamLab-NG: Benchmarking Low-Power Wireless Protocols under Controlable and Repeatable Wi-Fi Interference. In *Int. Conf. on Embedded Wireless Systems and Networks*, 2019.

[8] M. Zimmerling, L. Mottola, P. Kumar, F. Ferrari, and L. Thiele. Adaptive real-time communication for wireless cyber-physical systems. *ACM Trans. Cyber-Physical Systems*, 2017.

---

[2] In the contention slot, more than one node may initate a flood.