Poster: Accurate Cross-Technology Clock Synchronization Among Off-the-Shelf Wireless Devices

David Grubmair, Rainer Hofmann, Carlo Alberto Boano, and Kay Römer
Institute of Technical Informatics, Graz University of Technology, Austria
grubmair@student.tugraz.at; {rainer.hofmann, cboano, roemer}@tugraz.at

Abstract

Clock synchronization in distributed IoT systems is a necessary feature to allow a coherent data collection and event detection. This task is challenging as today's IoT systems often consist of heterogeneous wireless devices using incompatible technologies. Because of this, existing solutions often resort to the use of multi-radio gateways, which allow an indirect synchronization, but increase end-to-end delays. In this work, we present X-Sync, a novel approach allowing a *direct*, *bidirectional* clock synchronization across wireless devices with incompatible physical layer. X-Sync leverages cross-technology communication in combination with MAC timestamping and performs a repetitive sampling of the received signal strength to accurately determine the beginning of a cross-technology frame. Preliminary results on off-the-shelf devices show that X-Sync can reach an accuracy $< 1\mu$s.

1 Introduction

With the growing adoption of wireless data acquisition systems in industrial settings, the number of measurement devices on the market has steadily increased. Even if the latter use different wireless standards, it is crucial to ensure a coherent data collection and event detection, which requires an accurate clock synchronization despite the incompatible physical layer. Achieving this, however, is challenging, as different standards employ diverse modulation schemes, data rates, and channel bandwidths, which makes an exchange of timing information via the usual communication channel infeasible. Because of this, state-of-the-art solutions rely on multi-radio gateways to forward timestamps among heterogeneous devices [1]. The use of gateways, however, introduces extra costs, increases traffic, and leads to variable end-to-end delays lowering the synchronization accuracy.

Cross-technology communication (CTC) using packet-level modulation allows to *broadcast* timing information to multiple heterogeneous devices [2]. In packet-level modulation, data is often encoded in the duration of legitimate packets and can be decoded using energy detection, i.e., by sampling the received signal strength (RSS). Due to the limited RSS sampling frequency of commercial wireless devices, the beginning of a CTC frame cannot be detected accurately on the receiving side. This introduces a *sampling delay* that increases the end-to-end latency by a non-deterministic and highly varying value (corresponding to $l_2 - l_3$ in Fig. 1). Moreover, the time needed to exchange information between two devices is notably higher than in traditional approaches, as a cross-technology data exchange using packet-level modulation involves the transmission of many legitimate packets.

To date, Crocs [3] is the only work that uses CTC to enable a unidirectional clock synchronization from Wi-Fi to IEEE 802.15.4 devices. To remove the delay introduced by the long transmission time, Crocs splits the synchronization process into two parts: a mutually-detectable event (a well-known sequence of packets) is used as synchronization point, and the device timestamp is sent separately. To accurately detect the beginning of an event, the IEEE 802.15.4 receiver performs an autocorrelation of the measured RSS values, which results in a ms-level synchronization accuracy.

We present *X-Sync*, a novel scheme that enables a *bidirectional* clock synchronization with *sub-µs-level accuracy* among *several* wireless technologies. X-Sync does not rely on multi-radio gateways or commonly-detectable events. Instead, it uses packet-level modulation CTC, and employs MAC timestamping to remove non-deterministic delays. To reduce the delay introduced by the limited RSS sampling frequency, X-Sync performs a repetitive sampling of the RSS to accurately determine the beginning of a CTC frame.

2 X-Sync Principle

To allow the synchronization of multiple devices simultaneously and independently of the employed technology, a generic way of exchanging timestamps among heterogeneous devices is needed. To this end, we have extended *X-Burst* [2] – a generic and portable CTC framework allowing constrained IoT platforms with incompatible physical layer to seamlessly interact using packet-level modulation – to enable the transmission and reception of timestamps. X-Burst traditionally encodes information in the duration of legitimate packets by transmitting a CTC frame embedding a preamble (used by receivers to detect the presence of a CTC transmission) and a payload containing the actual data to be exchanged. X-Sync adds an additional preamble

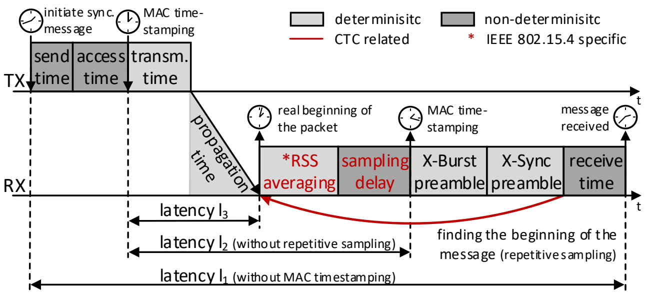

Figure 1: Our approach for cross-technology synchronization.

after the original X-Burst preamble to accurately determine the beginning of the CTC frame, as explained below, and transmits a device timestamp as part of the X-Burst payload.

The accuracy of a synchronization scheme is primarily limited by the delays added during the transmission and reception process [4]. Fig. 1 shows these delays in detail. Traditionally, the end-to-end latency when exchanging a timestamp equals l_1, i.e., it is computed from the instant in which the application issues the message embedding the timestamp to the instant in which the latter is decoded by the receiver.

Removing higher layer delays. To get rid of non-deterministic delays introduced by the OS and network stack (send / receive time in Fig. 1), as well as by the medium access control scheme (access time), X-Sync uses *MAC timestamping* [4], where timestamps are generated by reading the local clocks immediately when a message is sent or received.

Compensating low RSS sampling frequencies. The sampling and processing speed of RSS values is limited by the used hardware and strongly varies among platforms [2]. To compensate the sampling delay and thus, to minimize the variability of the end-to-end latency, each CTC frame contains a well-known sequence of legitimate packets (X-Sync preamble). This sequence is used to accurately determine the beginning of a CTC frame. To this end, X-Sync samples the RSS only at specific points in time and predicts the beginning of the next packet within the X-Sync preamble. Depending on the sampled value, i.e., if a packet was detected (RSS$\geq$threshold) or not (RSS$<$threshold), the instant of time in which the next RSS value is sampled is adjusted accordingly. That is, if no packet was detected, the RSS sampling point was chosen too early and hence the next RSS sampling point will be slightly postponed (and vice-versa). This process, which is called *repetitive sampling*, is repeated until the last packet of the X-Sync preamble is reached. Since the nominal durations of all previously-received packets and gaps are known, the start of the CTC frame can be determined and the sampling delay compensated accordingly.

Compensating RSS averaging. IEEE 802.15.4 devices experience a *constant*, device-dependent offset caused by the averaging of the RSS values [2]. In particular, depending on the used threshold and setup, the detection of packets is delayed up to 128 μs. Therefore, the calculated start of the CTC frame has to be corrected by a constant, measurable factor.

Minimizing the number of synchronization packets. To reduce the number of synchronization messages and thus the energy consumption, X-Sync estimates the clock drift between two devices using linear regression. This way, after

Platform		Sync. Error [μs]		
TX	RX	Min	Median	Max
CC2650 (BLE)	CC2650 (IEEE)	-0.64	-0.20	0.68
	Zolertia Firefly	-0.84	-0.13	0.25
	TelosB mote	-40.32	9.21	197.9
CC2650 (IEEE)	CC2650 (BLE)	-0.83	0.54	1.65

Table 1: An evaluation on IEEE 802.15.4 and BLE devices shows that X-Sync can achieve sub-μs-level accuracy.

sufficient data points are collected, the synchronization interval can be increased. To improve precision, we apply filtering on the received timestamps – which is especially important in the context of CTC due to its susceptibility to external interference – using random sample consensus (RANSAC).

3 Preliminary Evaluation

We have implemented X-Sync using Contiki-NG on three off-the-shelf IoT devices: the TI CC2650 LaunchPad, the Zolertia Firefly, and the TelosB mote. To evaluate the accuracy of X-Sync, we use the CC2650 LaunchPad in BLE and IEEE 802.15.4 mode to transmit its clock to various IEEE 802.15.4 and BLE devices, respectively. We use a synchronization interval of 10 s and measure the error as the maximum deviation over a time of 1000 s at a rate of 1 Hz. Additionally, a constant factor was used to compensate static delays (transm. / propagation time in Fig. 1). Table 1 shows our initial results. The high variance on the TelosB mote is due to its limited RSS sampling rate and processing power.

4 Conclusions

Our preliminary evaluation shows that X-Sync can synchronize devices using heterogeneous technologies with sub-μs-level accuracy, outperforming the state-of-the-art. X-Sync can also be used as a building block for existing high-level synchronization schemes such as FTSP [5].

5 Acknowledgements

This work has been performed within the LEAD project "Dependable Internet of Things in Adverse Environments" funded by Graz University of Technology. This work was also partially funded by the SCOTT project. SCOTT (`http://www.scott-project.eu`) has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737422. This joint undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium, Norway. SCOTT is also funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" between May 2017 and April 2020. More information at `https://iktderzukunft.at/en/`.

6 References

[1] G. Aloi et al. A Mobile Multi-Technology Gateway to Enable IoT Interoperability. In *Proc. of the 1st IoTDI Conf.*, 2016.

[2] R. Hofmann et al. X-Burst: Enabling Multi-Platform CTC between Constrained IoT Devices. In *Proc. of the* 16^{th} *SECON Conf.*, 2019.

[3] Z. Yu et al. Crocs: Cross-Technology Clock Synchronization for WiFi and ZigBee. In *Proc. of the 15th EWSN Conf.*, 2018.

[4] S. Ganeriwal et al. Timing-sync Protocol for Sensor Networks. In *Proc. of the 1st SenSys Conf.*, 2003.

[5] M. Maróti et al. The Flooding Time Synchronization Protocol. In *Proc. of the* 2^{nd} *SenSys Conf.*, 2004.