Wadatkar, P.; Zennaro, M.; Manzoni, P. (2020). On Time Synchronization of LoRaWAN Based IoT Devices for Enhanced Event Correlation. Association for Computing Machinery (ACM). 1-4. https://doi.org/10.1145/3411170.3411264

On Time Synchronization of LoRaWAN Based IoT Devices for Enhanced Event Correlation

Prachi Wadatkar
International Center of Theoretical Physics, Trieste, Italy.
pwadatka@ictp.it

Marco Zennaro
International Center of Theoretical Physics, Trieste, Italy.
mzennaro@ictp.it

Pietro Manzoni
Universitat Politècnica de València, Spain
pmanzoni@disca.upv.es

ABSTRACT

Low power wide area network (LPWAN) technologies are becoming prominent in the era of the Internet of Things. In this context, LoRa long range technology emerges as a good solution thanks to its property of providing long-range along with low power although with limited bandwidth. In this paper, we focus on the use of a LoRa network for time synchronization among various nodes, to support message transmission and critical data sharing in a correct real-time manner. When providing a reliable collaborative service, clock synchronization is required, for example to apply AI solutions in IoT. In this paper, we evaluate different methodologies based on LoRa network time synchronization using commodity low cost hardware so that our solution could be adopted even in low operating expenditures or rural contexts.

CCS CONCEPTS

• **Networks** → *Peer-to-peer protocols*; **Network architectures**; • **Human-centered computing → Accessibility systems and tools;** • **Information systems** → Collaborative and social com- puting systems and tools

KEYWORDS

Industrial Internet of Things (IIOT), Long Range (LoRa), Industry 4.0, Synchronization, Low power wireless area network (LPWAN).

KEYWORDS- Industrial Internet of Things (IIOT), Long Range (LoRa), Industry 4.0, Synchronization, Low power wireless area network (LPWAN).

1 INTRODUCTION

Most IoT applications require a high number of nodes that send data in a coordinated manner. This has an impact on wireless communications due to collisions, on power consumption due to retransmissions and on overall cost in general [1].

Given the requirements of most IoT applications, low power wide area network (LPWAN) is the most promising technology to fulfill the requirement of billions of devices online. This technology has been developed with a focus on cost efficiency, power consumption, and coverage area. Typically, they operate in the unlicensed Industrial, Scientific and Medical (ISM) bands. LoRa modulation has been developed by Semtech and is based on chirp signals which are defined by the time profile having an instantaneous frequency that changes throughout the time interval. LoRaWAN is defined as the communication protocol and network system while the LoRa physical layer enables the long-range communication link. LoRaWAN media access control (MAC) layer enables the communication between the network and multiple devices due to its MAC mechanism [2]. A LoRaWAN network has a star topology where the nodes can only communicate with gateways and not directly with each other. A single gateway can cover a large area and can support thousands of end nodes.

Gateways converge to so called *Network Servers* that enable low power devices to exchange data with Applications. More specifically, in this work we used "*The Things Networks*" (TTN), that provide a set of open tools and a global, open network to build IoT applications at low cost.

The overall aim of this paper is to present an overview of three possible setups for the challenging task of time synchronization of nodes in a LoRaWAN. Time synchronization in wireless communication networks provides a common time frame along with other functions such as message transmission, channel scheduling et al, that should place in the correct order. Often, IoT applications require that all the nodes are synchronized with the network and with each other to exchange time-critical information and warning messages to provide an enhanced event-based correlation. In this paper, we analyze the key factors in LoRa network time synchronization such as precision and accuracy using different methodologies and highlight the advantage of GPS in LoRa network time synchronization [3].

This paper is organized as follows. Section 2 describes the clock synchronization in a wireless sensor network and the methodologies of time synchronization in the LoRa network. Section 3 proposes the implementation of the time synchronization in the LoRa network using commodity hardware. Section 4 evaluates the time synchronization performance, including the error duration. Finally, Section 5 presents the conclusions.

2 CLOCK SYNCHRONIZATION

In wireless communication time synchronization is a challenging task. It helps the end devices to adjust the drifts of the clock within each other and with the network. In this way, every end node in the network can operate with the same view of time. For enhanced event correlation the timely delivery of the message is crucial and effective. However, real end-devices can present time drifts as shown in the Fig. 2. Time synchronization helps to adjust the drift that occurred in the time of end nodes concerning standard time and with each other. In other way synchronizing the clock means setting the end node time to the particular epoch time with standard format of time such as UTC [4].

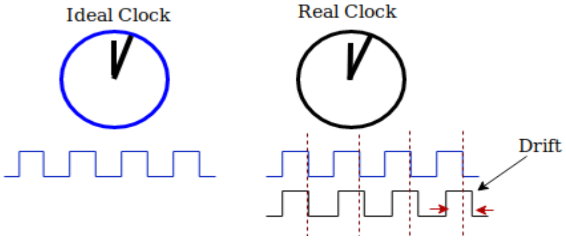

Figure 1: Clock Communication Network.

2.1 Clock synchronization in LoRa and LoRaWAN

IoT applications based on LoRa nodes assume an RTC (Real Time Clock) at each node that needs to be synchronized with a common reference time. Since LoRa nodes operate independently, their RTC may not be synchronized with each other. This can lead to difficulties when trying to interpret information sensed at different nodes. In the area of IoT and wireless sensor networks, the clock synchronization problem has been already studied. Many synchronization algorithms rely on the clock information acquired using GPS [5]. However, GPS has its well-known drawbacks, mainly derived from the high-power consumption and cost-efficiency. LoRaWAN does not incorporate a time synchronization protocol nor includes time related information in message headers, but only slack timing requirements because of long transmission duration [6]. Class A end devices allow bidirectional communication based on Aloha type of protocol. These devices are application-specific and keep transmitting until there is a message from the gateway; to get this downlink message from the gateway, it should wait till uplink transmission. Fig. 3 shows how the Class A end device receives slot timing. With Class B devices, all gateways must transmit synchronously beacon of time reference which is called ping slots while the network infrastructure to initiate a downlink communication. The application layer requests the network layer to switch to Class B mode from Class A mode. The beacon must be transmitted and if no beacon has been received, the synchronization with the network will be lost. Then the MAC layer must inform the application layer that the device switched to Class A. Due to this, Class B consumes more power than Class A and C. Finally, Class C devices can listen all the time except in transmit mode. They are ideal for applications requiring more downlink transmissions but will utilize more power compare to Class A and Class B counterparts. [7].

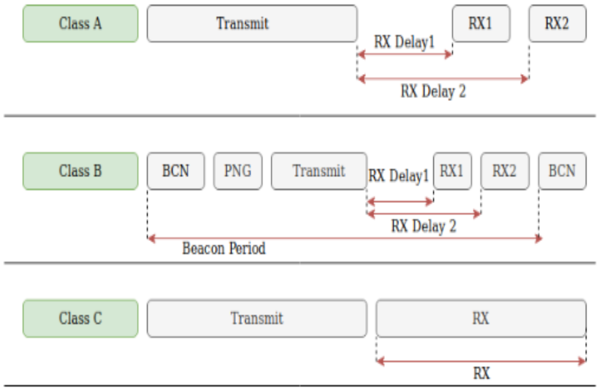

Figure 2: End Device Receive Slot Timing.

3 THE EVALUATED METHODOLOGIES

In this section, we describe the three methodologies for time synchronization among nodes using a LoRa network that we will evaluate. For these experiments, we have used a Pycom LoPy4 node with a Pycom Expansion Board. Nodes under test (NUT) were configured in the 868MHz European frequency band, the spreading factor was set to 7 to minimize the time on air and obey to the duty cycle regulation.

3.1 Time synchronization using WiFi

The first methodology is the time synchronization using WiFi. In this approach the NTP (Network Time Protocol) is used to synchronize the internal RTC (Real Time Clock) of the devices.

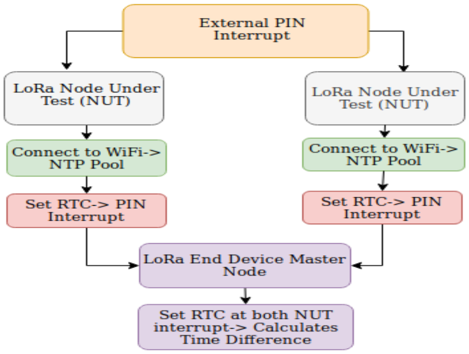

Figure 3: System Architecture of Time Synch using WiFi.

We performed the experiments using three nodes. One of them was used as a master node to compare the time difference between the other two nodes under test (NUTs). The NUTs were set-up with the same configuration of frequency, power and spreading factor. To activate our measurements, we used an external interrupt sent through to the general-purpose input/output (GPIO) pins of the nodes under test. When the external interrupt was triggered, the NUTs obtained the time value by requesting it to the NTP (Network Time Protocol) server. On receiving the value, they write it to the corresponding RTC register. Once the RTC is set on the nodes under test, the master node executes a callback function triggered when the input level of the output pin changes from both nodes and calculates the time difference between them. Finally, the master node sends the time difference between the nodes under test to the TTN server. To optimize the process, we have set the system to get the external interrupt every hour.

3.2 Time synchronization using TTN Downlink

The second methodology is based on the use of the information embedded in the TTN downlink message from the gateway. The nodes under test on receiving a synchronization downlink packet, write the time value to the corresponding RTC register. In "*The Things Network*", the downlink messages are allowed but limited to 10 per day.

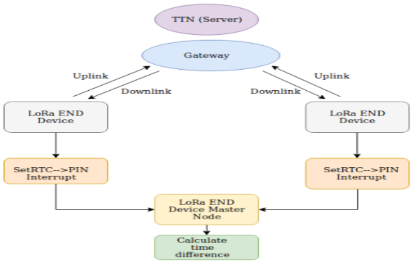

Figure 5: System Architecture of Time Synch using TTN downlink.

The raw downlink packet format is as follows:

```
{
  "dev_id": "my-dev-id", // The device ID
  "port": 1,             // LoRaWAN FPort
  "confirmed": false,    // Whether the downlink
                            should be confirmed
                            by the device
  "payload_raw": "AQIDBA==" // Base64 encoded
                              payload: [0x01,
                              0x02, 0x03, 0x04]
}
```

The Unix Epoch Time Stamp is encoded into binary format in the `payload_raw` field by the gateway. It consists of 16 bytes which represent time values of: milliseconds, seconds, minutes, hours, date, month and year. The confirmed field was left unchecked; setting it to `True` is necessary only if an acknowledgment from the node is required, `port` was set to 1 in all cases.

Once the downlink message is received at the nodes under test, they extract the time information and correspondingly set the RTC register time information. The NUTs raise an interrupt and the master node calculates the time difference between the nodes under test. The frequency of the downlink message was set to one every 4 hours to comply with the TTN limitations.

3.3 Time synchronization using GPS

The third methodology is based on the use of the GPS timing information. For this experiment, we have used the Pytrack board from Pycom that has a GPS + Glonass GNSS chipset. This board provides the precise time in addition to longitude, latitude, altitude and speed. The acquired timestamp through GPS consists of milliseconds, seconds, minutes, hour, date, month and year and refers to the UTC time zone. The nodes under test on receiving the GPS information write the time value to the appropriate register in the RTC. After stamping RTC, NUTs raise an interrupt at the master node where it writes the time value to the appropriate register in RTC at both interrupts. After that, the master node calculates the time difference between the nodes under test.

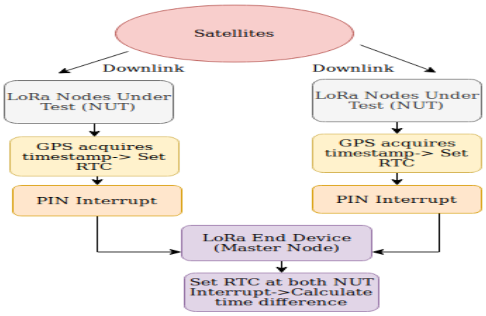

Figure 6: System Architecture of Time Synch using GPS.

4 EVALUATION

In this section, we present the results of the experiments we performed to determine the best time synchronization methodology. The configuration of the LoRa end nodes was as follows: frequency is set at 868MHz, spreading factor is 7 and payload is 16 bytes long. During the experiments no packet loss was observed.

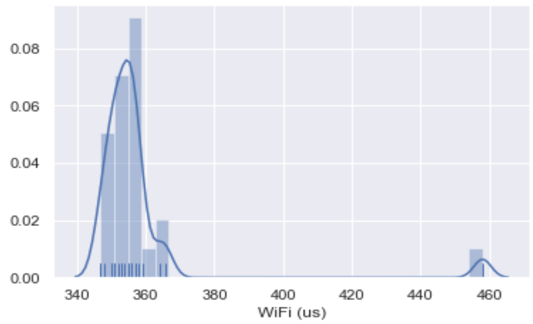

Figure 7: Time Synchronization using WiFi.

Fig. 7 shows the results obtained with the WiFi based methodology. Time difference is indicated in microseconds and a total of 25 values are plotted. The minimum time difference obtained was of 347 µsec and the maximum time difference was 458 µsec. The mean value 358 µsec.

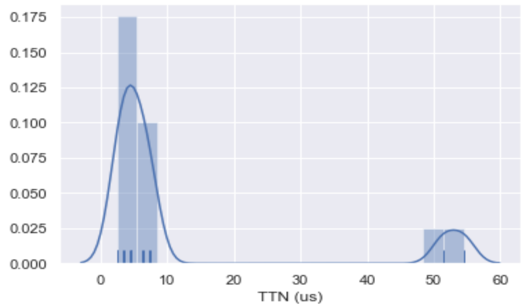

Figure 8: Time Synchronization using TTN Downlink.

Fig. 8 shows the results obtained with the TTN downlink-based methodology. Time difference is indicated in milliseconds and spans between 2 msec and 55 msec, a total of 13 values are plotted. Extra latency is introduced since the downlink channel transmits only when the receiving window is open. The minimum time difference obtained was of 2,438 msec and the maximum time difference was 54,532 msec. The mean value is 12,249 msec.

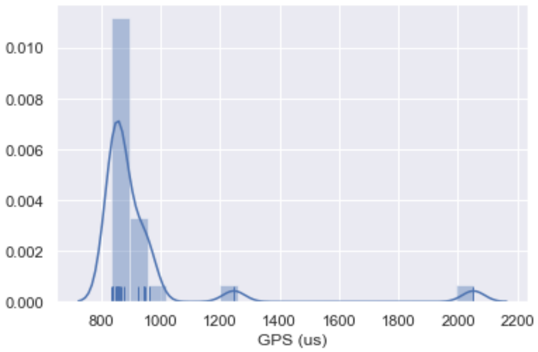

Figure 9: Time Synchronization using GPS.

Finally, Fig. 9 shows the results obtained with the GPS based methodology. Time difference is indicated in microseconds and lies between the 840 microseconds to 2 milliseconds; a total of 25 values are plotted. The mean value is 939 µsec. For this experiment we have used PyTrack boards which have a high power consumption.

CONCLUSION

In this paper we focused on the use of a LoRa network for time synchronization among various nodes, to support message transmission and critical data sharing in a correct real-time manner. When providing a reliable collaborative service, clock synchronization is necessary, for example to apply AI solutions in IoT. We evaluated different methodologies using commodity low cost hardware so that our solution could be adopted even in low operating expenditures or rural contexts.

We observed that, whereas the TTN downlink method can help to synchronize the device remotely with the help of gateways the time difference between the nodes achieved is in milliseconds, an order of magnitude too high for some applications. Using GPS, time synchronization can achieve better results, but it requires extra power for the HW required.

Finally, when Internet connectivity is available, through WiFi for example, time synchronization using NTP pooling can give accuracy in microseconds, thus making it the best choice.

REFERENCES

[1] Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., Watteyne, T.: Understanding the limits of LoRaWAN. IEEE Commun. (2017)

[2] LoRa Alliance Inc.: LoRaWANTM specification. LoRa Alliance, Inc., San Ramon (2015)

[3] Singh R.K., Berkvens R., Weyn M. (2020) Time Synchronization with Channel Hopping Scheme for LoRa Networks. In: Barolli L., Hellinckx P., Natwichai J. (eds) Advances on P2P, Parallel, Grid, Cloud and Internet Computing. 3PGCIC 2019. Lecture Notes in Networks and Systems, vol 96. Springer, Cham.

[4] Hasan, Khondokar Fida, Wang, Charles, Feng, Yanming, & Tian, Glen (2018) Time synchronization in vehicular ad-hoc networks: A survey on theory and practice. Vehicular Communications, 14, pp. 39-51.

[5] Gu, Chaojie & Tan, Rui & Huang, Jun. (2019). Secure Data Timestamping in Synchronization-Free LoRaWAN.

[6] Ebi, Christian & Schaltegger, Fabian & Rust, Andreas & Blumensaat, Frank. (2019). Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure. IEEE Access. PP. 1-1.

[7] Rhee, Ill-Keun & Lee, Jaehan & Kim, Jangsub & Serpedin, Erchin & Wu, Yik-Chung. (2009). Clock Synchronization in Wireless Sensor Networks: An Overview. Sensors (Basel, Switzerland). 9. 56-85.

[8] L. Tessaro, C. Raffaldi, M. Rossi and D. Brunelli, "Lightweight Synchronization Algorithm with Self-Calibration for Industrial LORA Sensor Networks," 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, 2018, pp. 259-263.

[9] T. Polonelli, D. Brunelli and L. Benini, "Slotted ALOHA Overlay on LoRaWAN - A Distributed Synchronization Approach," 2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC), Bucharest, 2018, pp. 129-132.

[10] Polonelli, Tommaso & Brunelli, Davide & Marzocchi, Achille & Benini, Luca. (2019). Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment. Sensors. 19.

[11] Hong Y.S., No J.H. (2007) Time Synchronization in Wireless Sensor Network Applications. In: Obermaisser R., Nah Y., Puschner P., Rammig F.J. (eds) Software Technologies for Embedded and Ubiquitous Systems. SEUS 2007. Lecture Notes in Computer Science, vol 4761. Springer, Berlin, Heidelberg.

[12] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni and M. Gidlund, "Using LoRa for industrial wireless networks," 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), Trondheim, 2017, pp. 1-4.