NOVEL TIME SYNCHRONIZATION MECHANISM FOR LARGE-SCALE AND ULTRA-LOW DUTY CYCLE WIRELESS SENSOR NETWORKS

Lele Zhang
Yajun Xia
Chuanwei Li
Ling Wei

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
Zhang, Lele; Xia, Yajun; Li, Chuanwei; and Wei, Ling, "NOVEL TIME SYNCHRONIZATION MECHANISM FOR LARGE-SCALE AND ULTRA-LOW DUTY CYCLE WIRELESS SENSOR NETWORKS", Technical Disclosure Commons, (October 01, 2020)
https://www.tdcommons.org/dpubs_series/3652
NOVEL TIME SYNCHRONIZATION MECHANISM FOR LARGE-SCALE AND ULTRA-LOW DUTY CYCLE WIRELESS SENSOR NETWORKS

AUTHORS:
Lele Zhang
Yajun Xia
Chuanwei Li
Ling Wei

ABSTRACT

As an important new feature in the wireless smart utility network (Wi-SUN) 2.0 protocol, low-power functionality has a range of implementation problems. One of those problems concerns time synchronization (TS) between an ultra-low duty cycle limited function device (LFD) and a central full function device (FFD) using a channel hopping mechanism. Most of the existing TS proposals focus on reducing the total overhead of all of the nodes in a Wireless Sensor Network (WSN) by improving or enhancing a Reference-Broadcast Infrastructure Synchronization (RBIS) or Timing-sync Protocol for Sensor Networks (TPSN) algorithm, but such algorithms do not consider the ultra-low duty cycle WSN case. To address these sorts of challenges, techniques are presented herein that support a novel TS mechanism whereby, rather than waking up an LFD just for a sync time, the LFD may ‘sleep’ for as long as desired while incurring just a very small energy cost as the LFD awaits a synchronization message.

DETAILED DESCRIPTION

A connected-grid mesh (CG-Mesh) network may contain millions of nodes in a range of different environments such as, for example, a smart grid, street lighting, building automation, etc. A Wi-SUN environment focuses on the construction and maintenance of large-scale outdoor Internet of Things (IoT) wireless networks. In such networking environments many endpoint devices (EDs) operate with limited energy resources (e.g., battery-based nodes). Accordingly, Wi-SUN defines two kinds of devices, a central full function device (FFD) and a limited function device (LFD). FFDs are typically supplied by main power, which may support all of their functionalities without consideration of
energy consumption. LFDs are often based on battery power, and thus are constrained by energy usage. See, for example, Figure 1, below.

![Illustrative Network](image)

Figure 1: Illustrative Network

In order to support the long term service use of LFDs, most solutions adopt an ultra-low duty cycle sleep schedule for energy-limited terminals. More particularly, LFDs may enter a deep-sleep mode for as long a period of time as possible unless the devices have to connect to a network for activities such as sending reports/alarms or getting updates. The devices may wake up, for example, once per week or just once per month. Thus, timely synchronization is a big problem between LFDs and FFDs for the following reasons:

1. When LFDs enter a sleep mode, most of their peripheral devices are shut down to save energy, including their main clock crystal. But, the clock drift (CD) rate of alternative internal RC clocks is not sufficient for synchronization between LFDs and FFDs – e.g., such clocks may develop a one second difference after several hours.

2. Both CG-Mesh and Wi-SUN are based on frequency hopping technology. Frequency hopping, also known as channel hopping, is a method of transmitting radio signals by rapidly switching a carrier among multiple frequency channels, using a pseudo-random sequence known to both a transmitter and a receiver. Compared with fixed
frequency transmissions, frequency hopped transmissions have advantages such as resistance to interference and interception. CG-Mesh and Wi-SUN divide time into countless small slots, and each slot uses a channel which is determined by a pseudo-random algorithm (e.g., DH1CF, TR51, etc.). Therefore, an LFD or an FFD needs to know the right time slot and the number of a target node, otherwise the devices can't know the correct channel for that point in time. If a CD problem is serious, communication attempts will fail. See, for example, Figure 2, below.

![Diagram of communication attempt failure](image)

**Figure 2: Communication Attempt Failure**

Most solutions define the frequency of waking up based on the accuracy of the CD. For example, if the CD is one second in six hours, and the time slot interval is one second, then one can say LFDs need to wake up every three hours to get sync information from an FFD. The clock offset could be less or more, so that the sync up period will be half of a CD interval. If the LFD just needs to wake up monthly and it just sends several hundred bytes of content to an FFD, that sync mechanism unnecessarily wastes significant energy because of waking up so frequently.
To address these sorts of challenges techniques are presented herein that support a novel TS mechanism in such application scenarios.

Generally, it is known that for an LFD the energy consumption of transmission is far more than receiving, so LFDs need to complete a transmission in as short a period of time as possible. If one FFD and some surrounding LFDs are selected as an object, the topology is a star architecture. Thus, RBIS is a good option, however a wireless mesh network (WMN) uses channel hopping communication rather than propagating synchronization beacon information periodically. Even if an FFD is allowed to broadcast a beacon periodically, it will still be difficult for the children LFDs to match the right channel due to CD. One solution is using a fixed channel, but the link quality will become worse and unstable.

Aspects of the techniques that are presented herein do not give up a frequency hopping mechanism, but integrate key benefits from the RBIS method. Elements of particular interest and note within the techniques that are presented herein are discussed below.

A first element comprises each FFD broadcasting timing synchronization information periodically in a neighborhood by using frequency hopping, with the dwell interval of each channel being sufficiently long enough for LFDs to match. See, for example, Figure 3, below.

Under this approach, similar to the RBIS algorithm, an FFD takes charge of periodically broadcasting a synchronization message. The particulars regarding the frequency/interval of such broadcasting are discussed below in connection with a second element of the techniques presented herein.

Also under this approach an FFD uses the same channel for a long period to propagate a synchronization message, such as 12 hours or 24 hours per round. During this dwell window an FFD broadcasts a synchronization messages numerous times, using the same channel each time.
As mentioned previously, a Wi-SUN-based WMN uses frequency hopping technology that is composed of two kinds of channel hopping sequence - unicast (Ucast) and broadcast (Bcast) – as shown in Figure 4, below.
Each node has its own unicast channel hopping sequence and the whole network (e.g., a personal area network (PAN)) has one common broadcast channel hopping sequence. In practice the broadcast schedule time slots overlap the unicast schedule. When a broadcast slot is approaching, all of the nodes switch to the broadcast channel for transmitting/receiving multicast/broadcast traffic. The nodes turn to their respective unicast schedule as soon as the broadcast schedule is over. Because both Ucast/Bcast slots are transitory (e.g., 125 milliseconds (ms) or even shorter) it is difficult for an LFD to properly align if their CD has not yet been eliminated.

Consequently, under this approach one injects many small timing-sync slots overlapping both channel hopping schedules on FFDs with very slow channel switching (e.g., they may jump to a next hop every 24 hours). More particularly:

1. Only FFDs have timing-sync slots, LFDs do not have such slots.
2. FFDs spread synchronization messages across timing-sync slots.
3. These slots overlap both Bcast and Ucast schedules.
4. Each FFD has its own channel hopping sequence for timing-sync, which is also produced by using a pseudo-random algorithm. The channel will be updated per round.
5. Each FFD may have its own channel switching dwell interval according to a practical requirement (also referred to as a virtual channel hopping dwell). For example (and as illustrated in Figure 5, below) FFD A may switch its timing-sync channel every 18 hours, FFD B may change its every 24 hours, and FFD C could do so every 30 hours.
6. It is important to note that such a schedule does not affect existing Ucast/Bcast channel hopping schedules, it just adds a new schedule overlapping the existing two schedules.
As noted previously, it is difficult for LFDs to hit the right channel because of CD. Accordingly, under this approach a long duration is employed for the channel hopping dwell interval of multiple synchronization messages.

For example, one LFD may drift +/- 1 second in 10 minutes. Therefore, it could have +/- 108 seconds of difference to an FFD every 18 hours. If it is assumed that an LFD’s targeting FFD changes its timing-sync channel every 18 hours, the LFD will know the total offset as long as it figures out how many rounds have progressed since receiving the last synchronization message. So, the time difference (i.e., CD) of the LFD is 144 minutes for every waking-up, which is minor to 18 hours and only occupies 13.3% of virtual channel duration. When this LFD wakes-up, it could calculate the time difference based on virtual channel schedule information which had been synced at the last time (e.g., 30 days ago). There are three possible cases for this in practice. In a first case, the time duration of +/- 72 minutes could cross a previous channel and current channel. In this case, the LFD could set a timer to wake up again, which may help to ensure that the whole time duration could all be in the current channel. In a second case, the time duration could be +/- 72 minutes in the middle of the current channel. In this case, the LFD may do nothing but use the current channel to the receive sync message. In a third case, the time duration of +/- 72 minutes could cross a current channel and a next channel. In this case, the LFD could sets a timer to wake up again, which could help to ensure that the whole time duration could all be in the next channel. Thus, if the LFD wakes up after 30 days, its time offset will be +/- 72
minutes. So, the LFD could know its possible time interval in an FFD's timeline (as illustrated in Figure 6, below).

![Figure 6: Illustrative Interval Calculation](image)

Generally, the above effects may be summarized using the following formula:

\[ T_0 = R_{cd} \times (T_{wake} - T_{sync}) \]  

(1)

In this formula, \( T_0 \) denotes a time offset of an LFD between a last synchronization time and a wake up time. \( R_{cd} \) denotes the CD rate of an LFD. \( T_{wake} \) denotes the wake up time. And \( T_{sync} \) denotes the time of a last synchronization. So, the possible time interval is \([T_{wake} - T_0, T_{wake} + T_0]\).

It is important to note that a calculated dwell time does not necessarily need to be uniform across all of the elements (e.g., LFD) of an environment (e.g., PAN).

It is also important to note that the possible time interval may not be larger than the round duration. For example, if there are 18 hours in a round the possible span could not be larger than 18 hours. Therefore, there are two situations that may arise when an LFD wakes up (illustrated in Figure 7, below):

1. The possible time interval is in one channel. If so, the LFD just needs to use this channel for receiving a synchronization message.

2. The possible time interval crosses two adjacent channels – e.g., a previous channel and the current channel or the current channel and a next channel. If so, the LFD
needs to wait a moment (e.g., delta t) to ensure that it is in one channel for receiving a synchronization message.

Figure 7: LFD Wake Up Situations

A second element of the techniques that are presented herein comprises a method for calculating how frequently timing-sync messages are sent by each FFD.

It is important to note that an LFD’s energy consumption varies depending upon, for example, its radio state (as shown in Figures 8 and 9, below).

Figure 8: RF215 (Ateml RF Chip) Power Consumption

| Low current consumption (incl. baseband processing / without I/Q interface) |
|-----------------|----------------|
| • Deep sleep    | 30nA           |
| • RX listen     | 6.28mA (RPC mode dependent) |
| • RX active     | 28mA           |
| • TX            | 62mA @14dBm output power |
Accordingly, LFD energy consumption could be simplified as the following rule:

$$P_{\text{sleep}} < P_{\text{listen}} \leq P_{\text{rx}} < P_{\text{tx}}$$  \hspace{1cm} (2)

While the power consumption of FFDs may not be considered because most of them have a main power supply, power consumption for LFDs should be considered. Based on the above, it is desirable for LFDs to avoid frequently entering Rx/Tx states.

Consequently under this approach the frequency of broadcasting timing-sync messages is based on energy consumption and a lifetime requirement.

Assume that $E_{\text{max}}$ denotes the maximum energy used for sending a timing-sync to an LFD, $C_r$ denotes receiving current, $T_{\text{ndi}}$ denotes the dwell interval between two timing-sync messages.
Sync messages, \( f \) denotes the waking frequency of LFD (e.g., weekly or monthly), and \( T_{lt} \) denotes the required lifetime (e.g., 10 years). The following formula illustrates that \( T_{tdi} \) may be computed as:

\[
T_{tdi} = \frac{E_{max}}{C_r \cdot T_{lt} \cdot f}
\]  

(3)

For example, if one LFD could use 1000 milliamp hour (mAh) at most for receiving timing-sync messages for ten years, and it is expected to wake up every ten days, the Rx current consumption is 30 milliamps (mA), the timing-sync dwell interval is recommended as less than 329 seconds per round. As shown in Figure 10, above, the worst situation is when an LFD always wakes up at the beginning of a TSDI window, thus it has to wait the entire dwell interval to catch the next synchronization message.

This approach also recognizes that there are multiple LFDs around one FFD, so each LFD has its own recommended TSDI value. For example, as illustrated in Figure 11, below, LFDs X, Y, and Z have different maximum TSDI – node X is 5 minutes, node Y is 10 minutes, and node Z is 1 hour. In order to cover all of the children LFDs, FFD P is required to have its broadcast frequency threshold align with the smallest child, which in this example is node X. This threshold could be adjusted dynamically when a network’s topology is changed. Furthermore, if node X detaches from node P, then the FFD's threshold will be node Y's TSDI value (10 minutes). If a new node were to join this example, it would be recommended that it’s TSDI be 3 minutes and that node P's threshold be adjusted to be 3 minutes as well.
In summary, TS can be challenging between FFDs and LLDs within, for example, a Wi-SUN environment. To address those challenges techniques have been presented that support a novel TS mechanism whereby an LFD may ‘sleep’ for as long as desired, while incurring just a very small energy cost as the LFD awaits a synchronization message, comprising among other things extended dwell intervals and efficient synchronization message generation timing.