October 09, 2019

RE-SYNCHRONIZATION OF BATTERY-POWERED EDGE DEVICES IN CHANNEL HOPPING LOW POWER AND LOSSY NETWORKS

Zaochun Zhang

Rui Huang

Xiao Xu

Xiaopu Zhang

RE-SYNCHRONIZATION OF BATTERY-POWERED EDGE DEVICES IN CHANNEL HOPPING LOW POWER AND LOSSY NETWORKS

AUTHORS:

Zaochun Zhang
Rui Huang
Xiao Xu
Xiaopu Zhang

ABSTRACT

The techniques presented herein propose to overlay a virtual channel-hopping schedule on a mains-powered devices' unicast (Ucast) and broadcast (Bcast) schedule that allows for a relatively large synchronization error for devices that cannot afford to resynchronize often. The virtual channel-hopping schedule does not occupy actual slot time of the mains-powered devices, does not reduce the overall robustness of a Low power and Lossy Network (LLN) to interference and channel agility, but may reduce the re-synchronization time between mains-powered devices and battery-powered edge devices. These techniques may be helpful to extend the lifetime of energy constrained devices in a channel-hopping network (e.g., for gas meters or water meter that can't afford to resynchronize often with mains-powered electric meters).

DETAILED DESCRIPTION

Gas and water meters are important components of a Smart Grid Advanced metering infrastructure (AMI) network, but have much more limited resources than electric meters. For example, gas and water meters often do not have mains power provided at the point of service, requiring them to be battery powered. The expected battery life is often 20 years or more since a single service call can often exceed the cost of the meter itself. A typical Smart Grid AMI architecture limits gas and water meters to "edge device" functionality that does not involve routing or forwarding packets generated by other devices. As such, gas and water meters spend most of their time in a sleep state/mode.

In many cases, LLN devices must communicate using a channel-hopping link layer. This requirement is driven both by regulatory compliance and the fact that channel-hopping systems offer better spectral efficiency. In certain arrangements, each interface determines its own unicast receive schedule and neighboring devices must synchronize with this unicast schedule to properly communicate a unicast frame. By having each device determine their own schedule independently, neighboring transmitter-receiver pairs can communicate simultaneously on different channels.

In addition, certain arrangements overlay a network-wide broadcast schedule where all devices are synchronized to the same channel-hopping schedule. The broadcast schedule is only active for a fraction of the time (e.g., 25%), but allows efficient broadcasts since a single transmission can reach an arbitrary number of neighbors. This hybrid approach allows these arrangements to maximize spectral efficiency for unicast communication while also allowing efficient broadcast communication. Figure 1, below, illustrates unicast (Ucast) and broadcast (Bcast) schedules for electric meters.

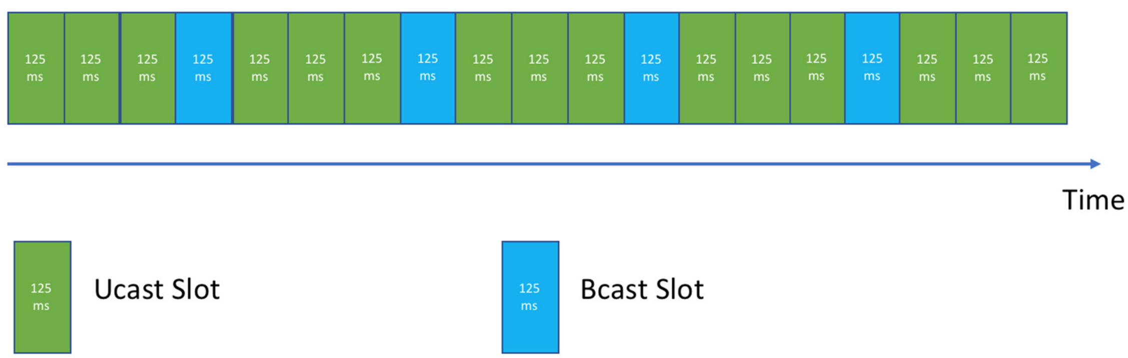

Figure 1

A primary challenge in channel-hopping systems is maintaining synchronization between transmitter-receiver pairs, especially for the energy constrained devices. These devices need to spend most of their time in a sleep mode and cannot afford to resynchronize often. When transmitting, a device must know which channel the receiving device is listening to at the time of transmission and devices synchronize with each other by exchanging messages with schedule information. Immediately after exchanging information, devices are synchronized to within tens or hundreds of microseconds. Existing systems require devices to periodically exchange messages to maintain synchronization. The maximum allowable error is driven by the channel-hopping slot size and guard windows on each slot boundary.

However, for the energy constrained devices, due to the long sleep time the synchronization error may grow to a large value that exceeds the maximum allowable error. As such, these devices may lose synchronization with the network and need to spend time to perform resynchronization, which in turn consumes needed energy.

One method to address such issues is to have the battery-powered devices operate on a fixed channel. In such examples, when the devices wake up from the sleep mode, they do not need to re-synchronize with their parents and only need to stay at this fixed channel to receive message from parents or transmit data to parents. However, the drawback of arrangements is that the devices may not work properly if interference is present at the fixed channel.

Another method to address such issues is to have the mains-powered devices periodically transmit synchronization (sync) message in all supported channels back-to-back. Transmitting the sync message on all channels allows battery-powered devices that have awoken from a sleep mode to receive the information needed to maintain synchronization, regardless of what channel they listen on. However, sync message transmissions are extremely expensive operations.

Yet another alternative is to increase the slot size and guard window. However, doing so would require increasing those parameters by 1-2 orders of magnitude, which reduces the overall LLN's robustness to interference and channel diversity.

The techniques presented herein propose a virtual channel-hopping schedule that allows for relatively large synchronization errors for devices that cannot afford to resynchronize often between parents (e.g., electric meters, mains-powered, *etc*.) and children (e.g., gas and water meters, battery-powered devices, *etc*.). In particular, the techniques presented herein overlay a virtual channel-hopping schedule on a normal Ucast and Bcast schedule, without reducing the overall robustness of an LLN to interference and channel agility.

In accordance with examples presented herein, the virtual channel-hopping schedule does not occupy actual slot time. The mains-powered parents initiates the virtual channel-hopping schedule to allow for the relatively large synchronization errors for battery-powered devices that need to spend most of time in a sleep mode and that cannot afford to resynchronize often. The parent provides the children with the schedule information, such as time slot duration, common time base, current time within the hopping schedule, *etc*. when the children join the network for first time.

As noted, the virtual channel-hopping schedule does not occupy the parents' actual slot time and parents use this schedule to transmit synchronization messages. However, children will use this schedule as their main schedule before they re-synchronize with their parents after waking up from a sleep mode (e.g., since parents' virtual channel-hopping schedule has long slot time to allow a large time deviation error). Figure 2, below, illustrates the overlay of a virtual schedule on a normal Ucast and Bcast schedule.

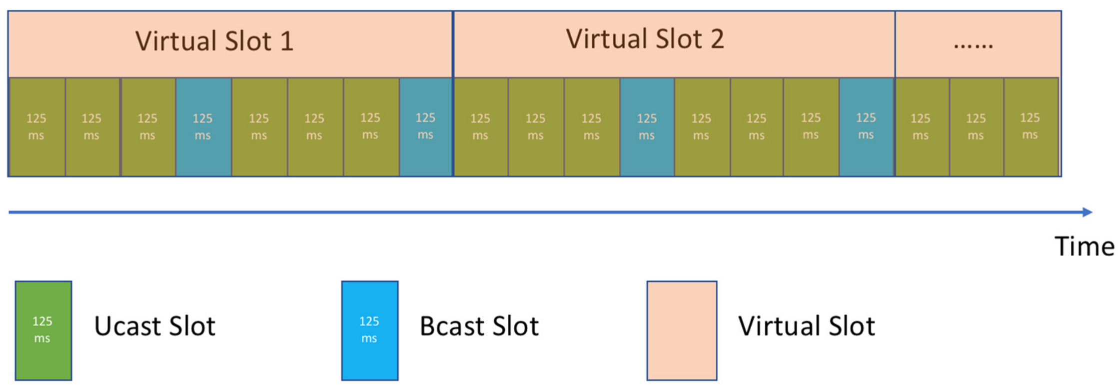

Figure 2

In the techniques presented herein, the parents send synchronization messages (e.g., sync beacons) periodically using the virtual channel-hopping schedule. A sync beacon contains information about the parents' virtual schedule, Ucast schedules, and/or network-wide broadcast schedules. When waken up from a sleep mode, a child stays active for a short period of time at the parents' virtual schedule in order to receive the sync beacon from the parent.

When a child receives a synchronization message, the child will update the parents' virtual schedule, Ucast and Bcast schedule, and correct the time error. The child can then use the parents' Ucast schedule to transmit application message or receive messages from parent, if they have application traffic. Figure 3, below, illustrates a parent's transmit synchronization message using virtual hopping schedule.

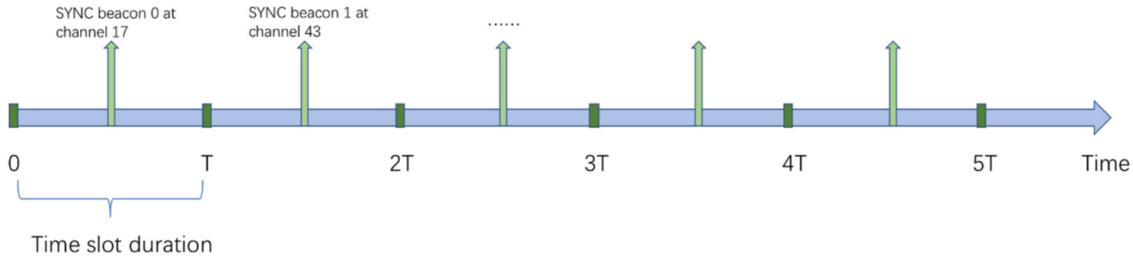

Figure 3

Mains-powered devices that have battery-powered children will periodically unicast/broadcast sync beacons at its virtual channel-hopping schedule. As noted, children stay active for a short time to listen to sync beacons and then go back to sleep if there is not any application traffic. Assuming a virtual schedule time slot duration is T and parents transmit the sync beacon only once at the middle of the time slot, then the maximum tolerance for time deviation would be +-T/2. By increasing the sync beacon transmissions at each time slot, the maximum tolerance for time deviation can theoretically reach to +-T. For simplicity, the techniques are described with reference to only one sync beacon transmission at each time slot. Figure 4, below, illustrates the maximum tolerance of time deviation for one sync beacon transmission.

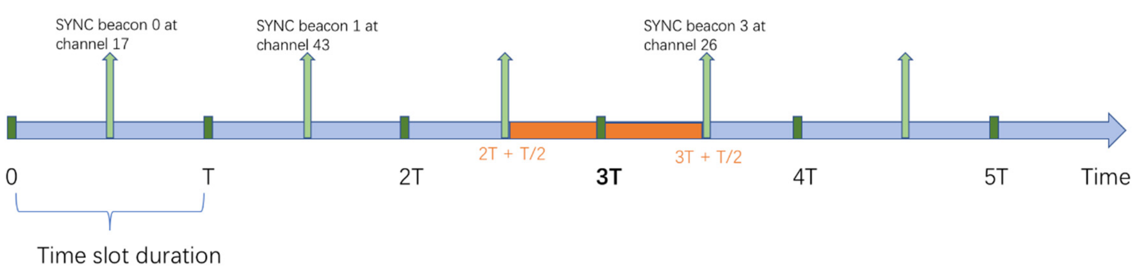

Figure 4

In one example, there may be a gas meter that should wake up at time 3T. As such, the actual wake-up time would be:

- If gas meter has fast clock (positive drift), then it may wake up before Time 3T, at the orange area between Time (2T + T/2) and Time 3T.
- If gas meter has slow clock (negative drift), then it may wake up after Time 3T, at the orange area between Time 3T and Time (3T + T/2).

For both of the above cases, if the time deviation less than T/2, then the gas meter can stay active at channel 26 for a short time (0~T) to receive the SYNC beacon 3.

Children can obtain their parent's clock drift and slot time duration in the joining process stage. Using these parameters, the children can know the maximum time they can sleep by calculating the time deviation based on parents' clock drift and their own clock drift. Assuming the time slot duration T=2000ms, +-25ppm crystal oscillators error for both parents, and children (25us deviation for 1 seconds approximately), the approximate maximum time ($T_{max\text{-}sleep}$) the children can sleep will be:

$T_{max\text{-}sleep}$ x (25 + 25) < T/2, clock drift at two opposite direction in worst case.

$T_{max\text{-}sleep}$ = 5.5 hours

If children sleep less than $T_{max\text{-}sleep}$, then children can keep the same virtual channel hopping with parents and only need to stay active for a short time (0~T) to receive sync beacon from the parents when awoken up from the sleep mode.

The time slot duration, T, is configurable and a bigger time slot duration will allow for a larger synchronization error for energy constrained edge devices. However, the active time to wait for the sync beacon will be increased. The resynchronization burden can be left to the mains-powered devices to transmit more sync beacons in each time slot duration to decrease the active time of energy constrained edge devices to wait sync beacon.

In the techniques presented herein, it is possible to properly increase the sync beacon transmit times at the children expected wake-up window. The energy constrained edge devices try not to transmit data unless required. If there is some application data that needs to be transmitted to a parent when the child wakes up, the child can tell its parent how long they will sleep next time in the same message. As such, the parent can properly increase the sync beacon transmit times at the end of the next wake up window of the child. Figure 5, below, illustrates multiple synchronization message transmissions at a child's expected wake-up window.

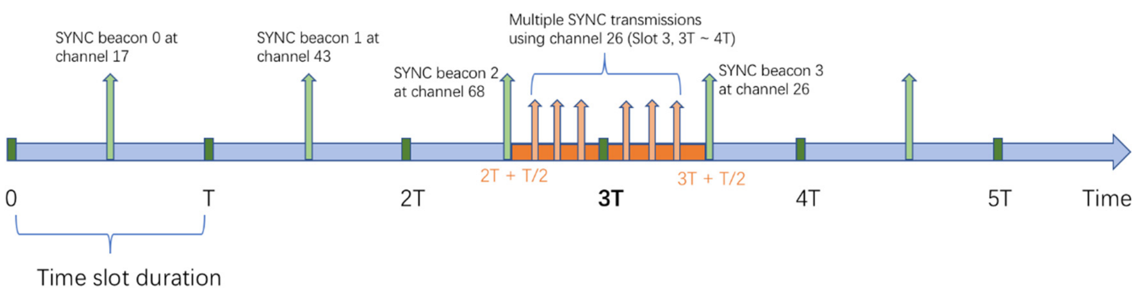

Figure 5

When children set the wake-up time to, for example, 3T, the actual wake-up time might be 2T+T/2 ~ 3T+T/2 (sleep time less than $T_{max\text{-}sleep}$). No matter when the children wake up at this window, they all would stay at channel 26 (shown in above figure) to listen to the sync beacon from the parents.

If parents know there are children that will wake up at time 3T, they will properly increase the sync beacon transmission times at the time window (2T+T/2 ~ 3T+T/2). This will decrease the children's active time to further to save energy.