SYNCHRONIZATION OF FREQUENCY-HOPPING SCHEDULING IN A FREQUENCY-HOPPING SPREAD SPECTRUM (FHSS) SYSTEM

Wenjia Wu
Ling Wei
Chuanwei Li
Li Zhao

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

Recommended Citation
Wu, Wenjia; Wei, Ling; Li, Chuanwei; and Zhao, Li, "SYNCHRONIZATION OF FREQUENCY-HOPPING SCHEDULING IN A FREQUENCY-HOPPING SPREAD SPECTRUM (FHSS) SYSTEM", Technical Disclosure Commons, (August 27, 2019)
https://www.tdcommons.org/dpubs_series/2440

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
SYNCHRONIZATION OF FREQUENCY-HOPPING SCHEDULING IN A FREQUENCY-HOPPING SPREAD SPECTRUM (FHSS) SYSTEM

AUTHORS:

Wenjia Wu  
Ling Wei  
Chuanwei Li  
Li Zhao

ABSTRACT

Presented herein are device synchronization techniques in a frequency-hopping spread spectrum (FHSS) system without time synchronization when a neighbor's channel hopping table is known. These techniques presented herein reduce the cost and improve performance of synchronization in FHSS, particularly in a low power or battery-powered system.

DETAILED DESCRIPTION

In many cases, Internet of Things (IOT) devices must communicate using a channel-hopping link layer (channel-hopping network). This communication requirement is driven both by regulatory compliance and the fact that channel-hopping networks offer better spectral efficiency. The primary challenges experienced by a device operating on a channel-hopping network are discovery and synchronization with neighboring devices. In general, although inefficient, devices should transmit across all channels to synchronize with peer nodes.

In addition, nodes in a Time Slotted Channel Hopping or Time Synchronized Channel Hopping (TSCH) network must also be time synchronized. A device could, for example, passively wait for EBs or a device could send valid frames to parent devices and use acknowledgements to maintain synchronization. However, if the synchronization is lost due to frequency hopping, the devices need a faster and more efficient way to resynchronize with a target node. The existing method in which the device will scan all available channels one by one, or scan all channels randomly, cannot assure 100% success rate, even in the absence of lossy network issues. The reason for this is that when the device scans the neighbor's channel slot, the neighbor itself is periodically channel hopping.
FIG. 1, below, illustrates an example in which a device could scan three (3) channels in one channel slot time and there are nine (9) total channels. In this example, all scans failed with an unacceptably high cost.

FIG. 1

In many situations, our target is to resynchronize with a certain peer. For example:

- Low power mode- a leaf node should resynchronize with its parent.
- Fast reformation- after reboot, a node should resynchronize with its default parent.
- Wireless Console- a node should initially synchronize with a target node to enable further communication.

For many channel hopping schemes, the channel hopping sequence could be derived from the device's information. For example, with TR51 the channel hopping sequence is calculated by EUI64 and the max channel number. However, under this circumstance, existing methods still may still experience an unacceptable success rate because the scan sequence is blind. In general, the goal is to enable a device to synchronize with a target device when the peer's channel hopping table is known.
The digit channel numbers shown in FIG. 2, below, indicate the "channel index." Because the channel hopping table is known, a real channel number could be mapped from the table and it is simpler to analyze than channel hopping sequenced by a channel index. In particular, in this example it is assumed that “m” = All channel numbers and “n “= scans per channel. In FIG. 2, m = 9  n = 3.

![FIG. 2](image)

Synchronization in FIG. 2 will fail for several reasons. For example, in the first slot, channels [1, 2, 3] are scanned and could be excluded if there are misses. In the next slot, because the peer node will hop to the next channel slot, the excluded list is changed to [1+1, 2+1, 3+1] = [2, 3, 4]. However, from FIG. 2, it can be seen that channel 4 is scanned in the second slot, which is redundant. After all tries have been completed, only channels [3 ~ 9] are excluded rather than [1-9]. Unfortunately, the neighbor's last channel is channel 2, which is outside of the excluded list.

As such, presented herein is a new scan rule in which each scan is effective and could exclude one channel. Thus the synchronization will succeed after m' tries, and m' <= m.
For simplicity, it is assumed below in FIG. 3 that the slot offset is the same as in FIG. 2. Proposed is a scan sequence to synchronization with a neighbor in a deterministic manner.

![FIG. 3](image)

In FIG. 3, the scan index in the same slot is the same as in FIG. 2. After hopping to the next slot, the scan index should add one to avoid the useless scan. Then, from FIG. 3, it can be seen that each time the system could exclude 3 channels. And in the worst case, the channel could be found by the last scan.

As such, presented herein is a new scan sequence using a cyclic prefix method to synchronize with a neighbor deterministically, even the slot offset is not zero. In conventional arrangements, the slot offset won't be zero. As a result, the former scan sequence will fail in some cases.

FIG. 4, below, illustrates an example in which all scans will fail.
The reasons that the scans in FIG. 4 fail is that three (3) tries (3,7,2) are useless because the slot offset is not zero. In this example, the devices have no idea when the neighbors will hop to the next channel. As noted above, the new scan rule presented herein can make sure every try is effective.

In Table 1, below, the rows represent the scan number and the columns represent the time slot number of the target. Additionally, “A” is the beginning scan index, “k” is the scan interval.

Assuming there are “m” channels, “n” scan times for per channel, and “A” equals 0, then the (i, j) element in Table 1 can be presented as:
As noted above, every try should be effective, thus:

\[ F_{i,j} = A + i - \left( \frac{j}{n} \right) + jk \]

which is equivalent to the equation:

\[ F_{i,j_1} \ mod \ m \neq F_{i,j_2} \ mod \ m \]
\[ \forall i < \left( \frac{m}{n} \right), \ 0 \leq j_1 < j_2 < m \]

First, it is assumed that \( (m \ mod \ n) \) is equal to 0, thus it is proven that \( k=(m/n) \) can meet the conditions by contradiction. The result reveals that it is possible to set some fixed scan points by even distribution along the scan channels: \( A, A+m/n, A+2m/n, A+3m/n, \ldots \) and then it can be assured that the target's channel will hop to one of them during the \( (m/n) \) time slots.

To solve the problem that the slot offset is not zero, proposed herein is a new cyclic prefix method, which means the last scan channel can be repeated at the start of the scan sequence. The cyclic prefix method could determine a whole scan sequence no matter how many slot offsets are present.

Secondly, if \( (m \ mod \ n) \) is not equal to 0, it is also possible to set \( (m/n)+1 \) fixed scan points by almost even distribution along the scan channels. The even distribution method could determine that the maximum interval between two neighbor scan points as one more than the minimum interval. Similarly, it is possible to make sure that the target's channel will hop to one of them during the \( (m/n)+1 \) time slots.

In summary, presented above is a novel scan rule for each time slot to make sure every scan is effective. Then, a new scan sequence and cyclic prefix method are proposed.
to solve the slot offsets problem and to synchronization with the target by 100% success rate.

The techniques presented herein have several benefits. For example, it is possible to derive that the expectation of scan tries to find synchronization with the traditional method as:

\[
\frac{1}{2} \cdot \frac{n \cdot m}{n-1}
\]

Additionally, the techniques presented herein include a novel method to build a specific scan sequence to synchronization with the target by 100% success rate. The channel number is set as \( m \) and the expectation of scan tries is:

\[
\frac{1}{2} \cdot m
\]

As result, the number of tries that are saved can be given as:

\[
\frac{1}{2} \cdot \left( \frac{n \cdot m}{n-1} \right) - \frac{1}{2} \cdot m = \frac{1}{2} \cdot \left( \frac{m}{n-1} \right)
\]

For instance, \( m = 120, n = 3 \), leading to 30 tries being saved (i.e., \( 90 - 60 = 30 \)). The promotion rate is then 33%.

In summary, the techniques presented herein include a novel method to synchronize frequency-hopping spread spectrum without time synchronization when the neighbor's channel hopping table is known. The techniques presented herein can reduce the cost and improve performance of synchronization FHSS, particularly in low power or battery-powered systems.