Technical Disclosure Commons

Defensive Publications Series

May 23, 2018

ENABLING DETERMINISM IN A BEST EFFORT HOPPING SEQUENCE TIME SLOTTED CHANNEL HOPPING MESH

Pascal Thubert

Huimin She

Charlie Chen

ENABLING DETERMINISM IN A BEST EFFORT HOPPING SEQUENCE TIME SLOTTED CHANNEL HOPPING MESH

AUTHORS:
Pascal Thubert
Huimin She
Charlie Chen

ABSTRACT

Techniques are described for sharing a network between flows that follow a deterministic hopping sequence (same sequence for all) and flows that follows a best effort hopping sequence (which depends on the receiver Media Access Control (MAC) address for unicast). This is accomplished by moving the best effort channel in case of a collision in a manner that can be predicted by the sender and that maintains the pseudo-randomness of the selection.

DETAILED DESCRIPTION

Currently, despite Internet Protocol version 6 (IPv6) over the Time Slotted Channel Hopping (TSCH) mode of IEEE 802.15.4e (6TiSCH), the deterministic wireless world is still fragmented between the deterministic (e.g., Wireless Highway Addressable Remote Transducer Protocol (WirelessHART) and the best effort schools. Both converge on the need to provide TSCH, one way or another. The difference resides in the way the hopping sequence is computed.

Channel hopping requires that the next transmission be on a different channel. Time slotting enables this at the scale of a network in a slotted "ALOHA" fashion by synchronizing when a device jumps to a different channel. The hopping sequence is the logic that decides which channel is used subsequently. Channel hopping mitigates the unpredictable effects of co-channel interference and multipath fading. Therefore, it is critical that as many channels as possible are used. Blacklisting or whitelisting channels is not a good option.

When determinism is required (e.g., WirelessHART, ISA100.11a, 6TiSCH, etc.), the hopping sequence must be the same for all nodes. If A->B does not interfere with C->D at the beginning of the relevant time, they jump in parallel and never collide. Thus, a time slot can be guaranteed for an undisturbed communication forever (at least by a device that participates in this network).

IEEE 802.15.4 TSCH (as used in 6TiSCH) forces all nodes to follow the same hop sequence regardless of the Media Access Control (MAC) address. For 16 channels, the sequence may be as follows:

5 6 12 7 15 4 14 11 8 0 1 2 13 3 9 10

This sequence constitutes a mapping table. The computation of frequency may be freq = map ([ASN + channel offset] mod nbChannels), where the absolute slot number (ASN) is the number of timeslots since the epoch when the network started. In the 2.4GHz band with channels from 11 to 26, this yields:

```
hopsq=[5,6,12,7,15,4,14,11,8,0,1,2,13,3,9,10]
freq=hopsq[(asn+choff)%16]+11
```

This means that if at timeslot N a node transmits on channel 14, then at time N+1 the node transmits on channel 11, at N+2 on channel 8, etc.

Figure 1 below illustrates a deterministic schedule from the perspective of a controller that is aware of the entire network and schedules all the transmissions. Time is divided into slotframes that are played repeatedly, and the schedule represents the initial allocations for the duration of a slotframe (represented at time t=0, also referred to as “epoch”).

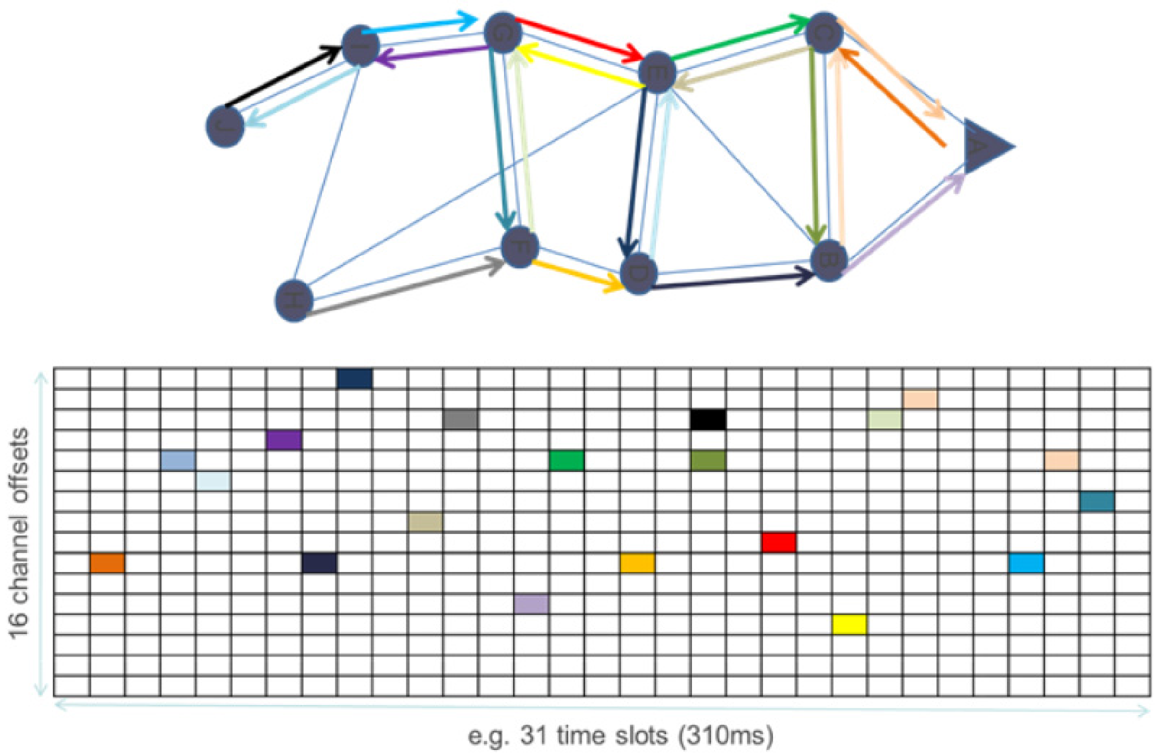

Figure 1

In other words, the time and frequency of a transmission as represented in the schedule determines the time and channel that the subsequent instances of the same transmission will use at any time in the future. The unit of time is a time slot duration, and time, if recorded as an ASN, counts the timeslots since the epoch.

When best effort and large scale are required, standards favor an automatic channel selection whereby the channel is a function of time and destination MAC address. This randomizes the channel that various listeners can select and statistically flattens the chances of collision. In contrast with a deterministic hopping sequence, a best effort spreads the receivers randomly over the possible channels in a fashion that can be generalized as freq = map (hash [receiver's MAC address, ASN] mod nbChannels), where ASN is the number of timeslots since the epoch when the network started.

Certain standards may use a direct hash of time and the node MAC address to select the next channel from the list of channels in use. The pseudo-random hopping sequence consists of 2^{16} slots. The major disadvantage of this method is inefficient band utilization, since the output of the direct hash can contain repeated channels. For example, assuming there are eight channels (0 to 7), the computed channel hop sequence may look like:

3 2 5 2 6 1 7 6.

Channels 2 and 6 are hopped twice and channels 0 and 4 are skipped during the slot window 0 to 7. Therefore, the frequency band is not fully utilized, which may lead to throughput reduction.

TR51CF is introduced in ANSI/TIA-4957.200. The channel hop sequence computed by TR51CF contains non-repeated channels. For example, assuming there are eight channels (0 to 7), the computed channel hop sequence may look like:

3 2 5 0 6 1 7 4 3 2 5 0 6 1 7 4.

Every channel is walked through exactly once in a slot window (0 to 7). However, the same hop sequence is walked through again in the second slot window (8 to 15). This may cause overlapping of the frequency hopping sequences of the two neighboring nodes, which could cause consistent interference.

Both techniques attempt to statistically use all the channels in a balanced fashion. This means that each device pseudorandomly jumps from channel to channel in a fashion that is not related to the way other devices jump, such that all channels are pseudorandomly used. The best effort sequences have a statistical chance to hit any time and channel, and with the current technology it is not possible to protect/blacklist individual cells in the TSCH matrix. In other words, a best effort sequence is bound to collide with a deterministic sequence if they are deployed together, which defeats the purpose of the deterministic hopping sequence (i.e., to maintain isolation at all times).

Accordingly, a mechanism is presented herein to modify the best effort hopping sequence to make it aware of the deterministic sequence. This protects cells in the deterministic schedule matrix while enabling a best effort hopping sequence in all the other cells. This extends an arbitrary automatic best effort hopping sequence to make it compatible with a coexisting deterministic hopping sequence. It is compatible with the current hopping sequences, as well as a depth matrix and associated distribution schemes.

The base design for deterministic time slots is well known per TSCH as used in 6TiSCH. The idea of building shared network with deterministic and non-deterministic flows is also used. The time slots for deterministic flows are computed by a controller and taken from a reserved time/frequency matrix, as illustrated in Figure 2 below.

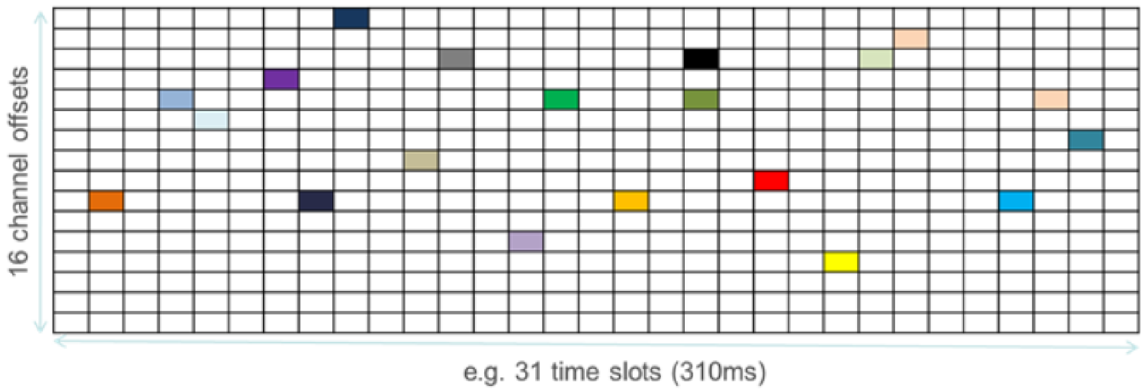

Figure 2

These techniques focus on solving the Gordian knot, which causes a best effort schedule to statistically collide with a deterministic schedule when they are used in the same network. To achieve this, the deterministic schedule is distributed to all nodes that may participate to non-deterministic flows, and changes the best effort scheduling technique to skip the forthcoming collisions when they are bound to occur.

When a node computes its received slots for the forthcoming time slots, it also computes which channels will be used by the cells in the deterministic schedule. If there is no collision, it uses the computed channel. If there is a collision, the node selects an alternate time and channel so as to avoid the collision. Multiple algorithms (basic or complex) may be used.

Several naive approaches to solve the aforementioned problem are described herein. A first naive approach is to pick the next channel in the hopping sequence of the receiver if a collision is detected in the next time slot, and then the next if that time slot also collides, etc. until a channel is found that does not collide. Each time, this is similar to advancing time an additional tick (e.g., ASN in TSCH) for the receiving node. However, this causes the time to virtually advance faster for receiving nodes that experience more collisions. This advance must be communicated to the sending nodes so as to compute the right channel, which can cause desynchronization and prevent communication.

In a second naive approach, if channel p collides with a deterministic schedule in the next time slot, then p+1 may be used, and if p+1 collides, p-1 may be used, and then p+2, etc. This uses the nearest collision free channel. However, at some time slot the deterministic channels may be packed in adjacent channels, which will cause the displaced best effort channel to pack on the channels next to the blacklisted block and collide with one another.

Therefore, the channel for a best effort transmission must be dynamically selected in a fashion that (1) depends on parameters that are always available to both the receiver and the sender to avoid issues of the first naive approach, and (2) is randomized between the channels that are not used by deterministic flows to avoid issues of the second naive approach.

Accordingly, a first method is presented to add an incremental parameter to the hash computation of the channel. As mentioned, the typical computation for the receive channel is in the general form of freq = map (hash [receiver's MAC address, ASN] mod nbChannels). As described herein, this is transformed to freq = map (hash [receiver's MAC address, ASN, increment] mod nbChannels), where the increment can be a simple integer 0, 1, 2, etc., or the nth entry in a well-known list. In case of a collision, the next increment is taken and a new hash is computed until the frequency does not collide with that used by a deterministic flow.

In a second method presented herein, the time frequency matrix is considered, and the nearest feasible timeslot is selected. The distance is counted as d = aT + bF, where T is the number of timeslots that the transmission is advanced or delayed, and F is the number of channels that are added or subtracted modulo nbChannels. Regarding distance in the time domain using the depth matrix (or another filter), the depth matrix is a filter that expresses timeslots when a node can send, receive, or stay idle. This eliminates some timeslots from being selectable as alternate receive since they are transmitted or idle. In Case of Emergency (ICE) slots have a higher risk of collision. As illustrated in Figure 3 below, they are used only in case of emergency, long delayed packets, and high priority packets.

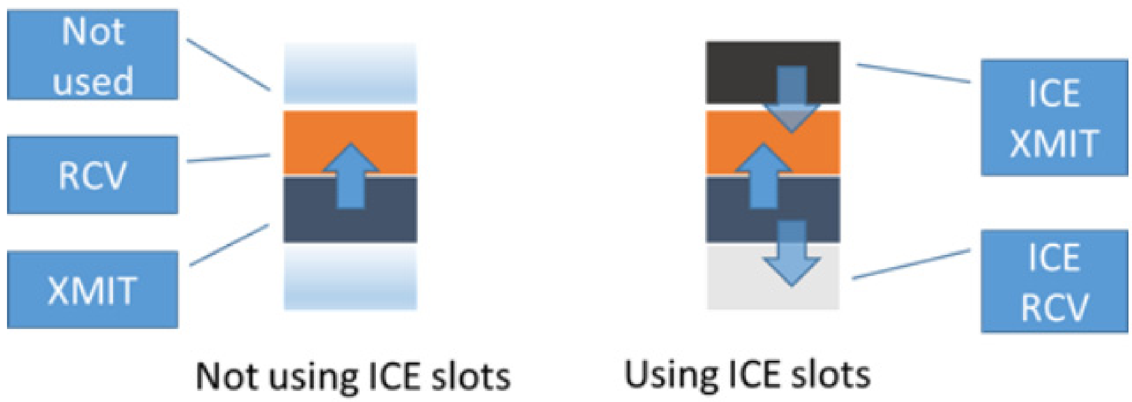

Figure 3

Figure 4 below illustrates a depth matrix that represents a collision with a deterministic flow computed in the next slotframe at timeslot 6 by a receiver A located at depth 4.

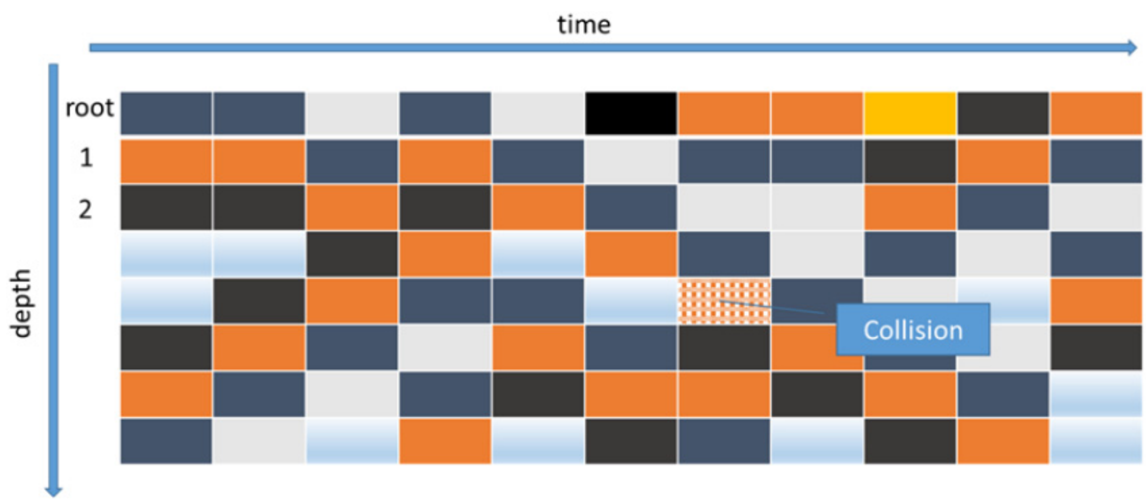

Figure 4

As illustrated in Figure 5 below, the depth matrix shows which other times are feasible for a receive operation. Some possible alternates may be available for normal traffic and others in case of emergency. Optionally, a displaced time slot may be an emergency, and therefore ICE slots may be used so as to avoid displacing too far in time. Using an ICE slot moves the reception two timeslots away in time, and using a normal reception slot moves the reception at least four timeslots away, and may collide with another scheduled receive, which may push it farther out. This is why the option of using ICE slots generally makes the computation simpler.

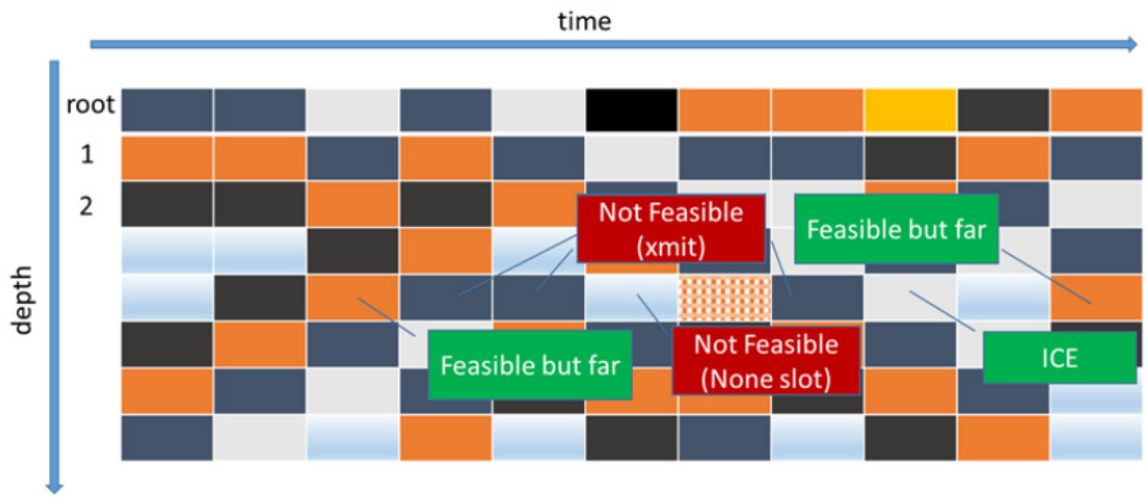

Figure 5

Figure 6 below illustrates an example with respect to distance in the frequency domain. In this example, the matrix depicts what happens in the next slot frame, with seven whitelisted channels for this network. As shown, node A's best effort schedule would collide with a deterministic reservation (shown in red) on time slot 6. Red blocks denote the time and frequency location of deterministic transmissions, which are to be avoided/blacklisted by this invention. Optionally, node A considers its nearest siblings (nodes B, C, D, E and F), and also computes their receive schedule. This results in the matrix illustrated below.

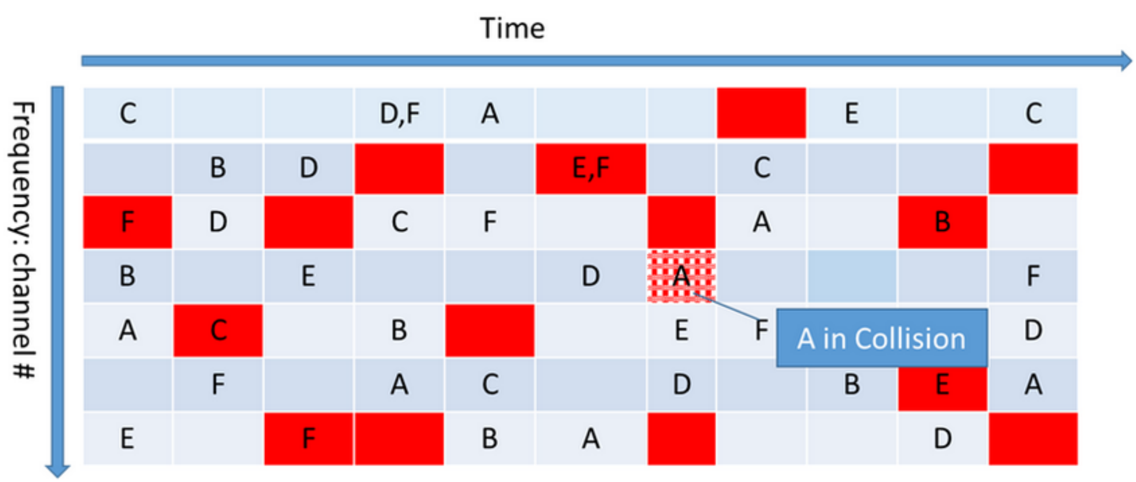

Figure 6

As node A computes its schedule for the next slot frame, it finds the upcoming collision with the deterministic transmission. Searching for the nearest possibilities in the time and frequency matrix, it finds the matrix illustrated in Figure 7 below.

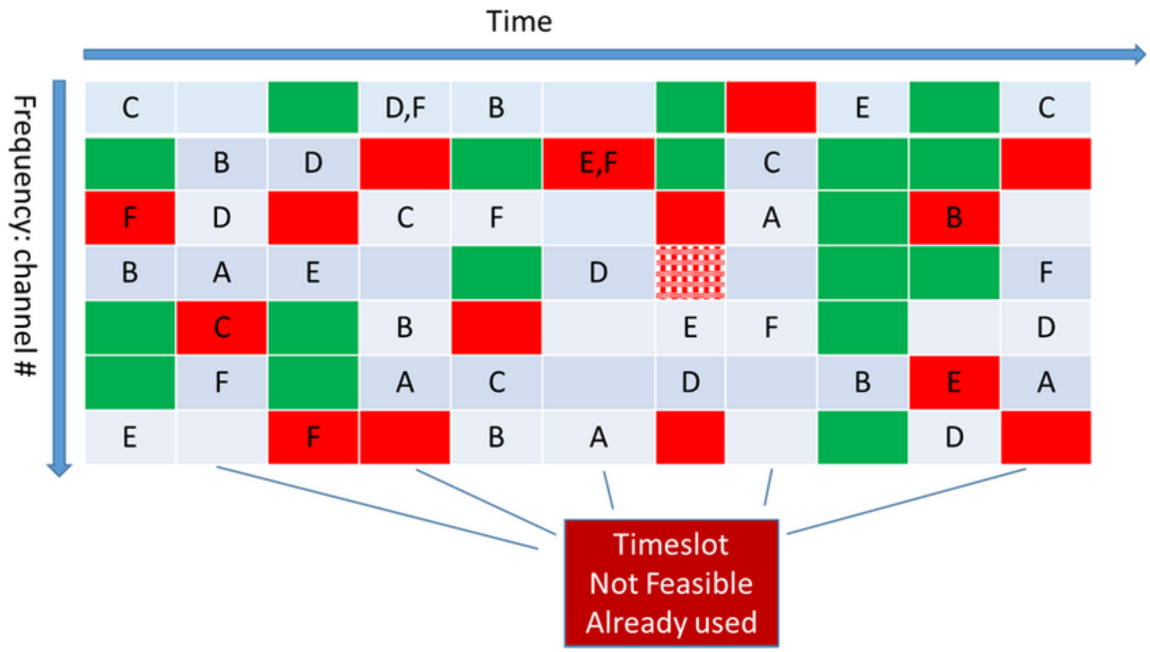

Figure 7

Figure 8 below illustrates the possibilities obtained by intersecting with the depth matrix.

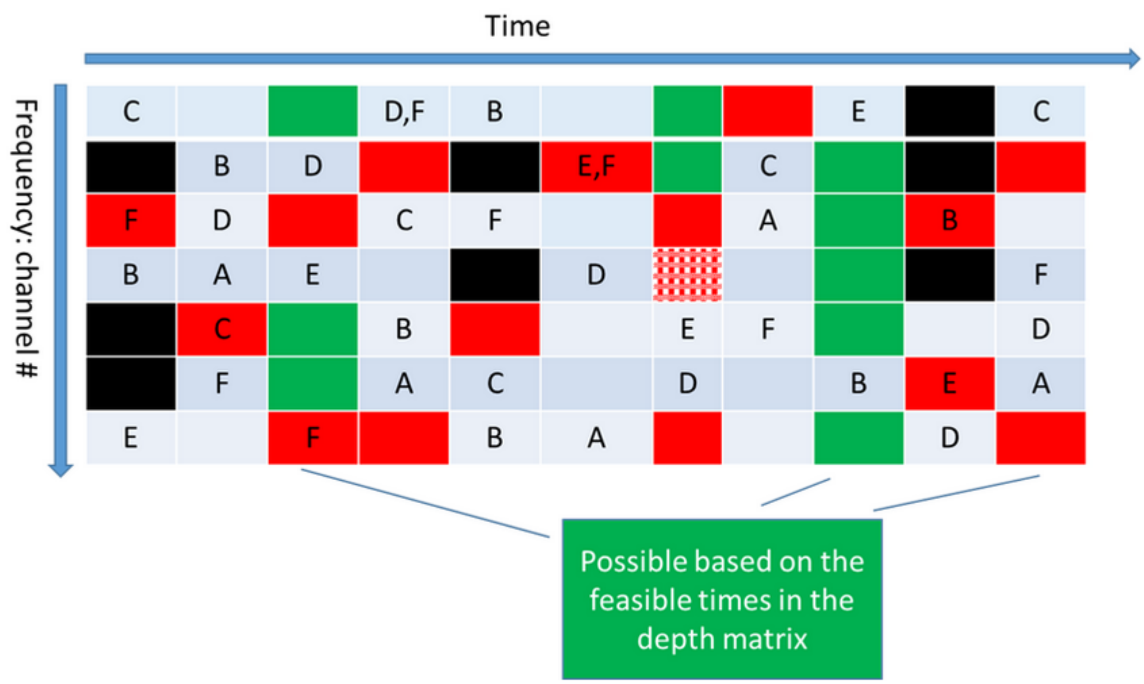

Figure 8

The selected alternate time and frequency depends on the distance expressed as d = aT + bF and on whether ICE slots are preferably avoided, as illustrated in Figure 9 below.

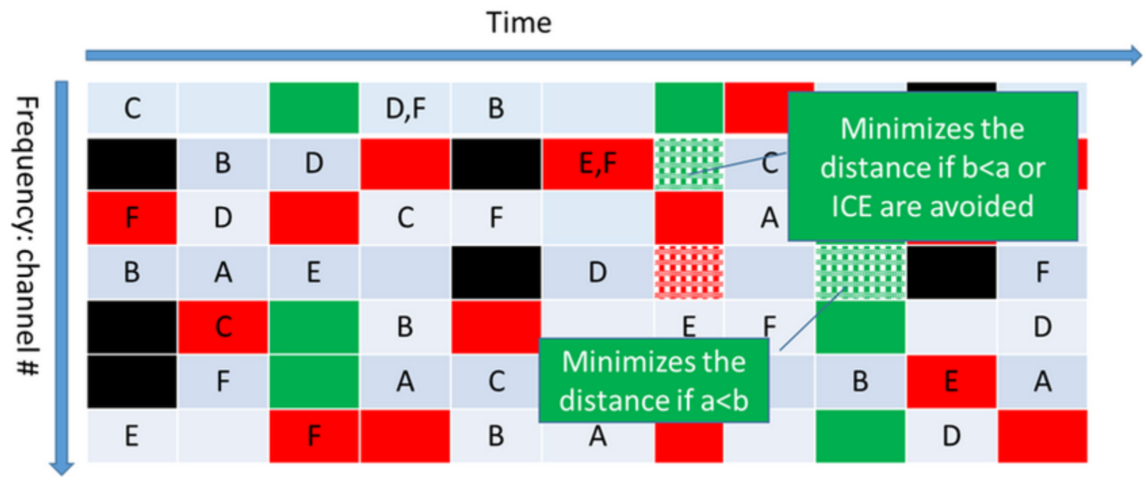

Figure 9

Either way, if all nodes are programmed with the same logic for the distance computation, then any node that wishes to transmit to node A at timeslot six can discover the collision and find the alternate time and frequency using the same distance-based logic.

These two approaches can be used together, with the first method being used to select an alternate channel at the same time slot, and the second method only at other times, and then using the distance to select one or the other.

In summary, techniques are described herein to share a network between flows that follow a deterministic hopping sequence (same sequence for all) and flows that follows a best effort hopping sequence (depends on the receiver MAC address for unicast). This is accomplished by moving the best effort channel in case of a collision in a fashion that can be predicted by the sender and that maintains the pseudo-randomness of the selection.