Delayed Cell Relocation in 6TiSCH Networks

Gihwan Kim, Sanghwa Chung, Minjae Kim
Department of Electrical, Electronic and Computer Engineering
Pusan National University
Busan, South Korea
gihwan@pusan.ac.kr, schung@pusan.ac.kr, innersme@gmail.com

***Abstract*— The schedule collision handle scheme based on the housekeeping has overhead because two pair of colliding nodes relocate the cells. The Delayed Cell Relocation proposed in this paper aims to relocate only one pair of nodes by setting the delay time differently by the cell allocation time, so that reduce the overhead. Simulation with various parameters shows that there is significant performance improvement about 25%.**

Keywords—6TiSCH, 6top, Scheduling Function, Scheduling Collision, Collision Resolution

I. INTRODUCTION

The emergence of the Internet Protocol Version 6 enables IP address to be allocated to all things by eliminating the IP address exhaustion and opens up the era of the Internet of Things (IoT). There have been many attempts to make things interconnected in wireless environments as well as wired. The Wireless Sensor Networks, one of the Industrial Internet of Things for industry must provide a reliable connection. But, there are problems such as multipath fading and external interference due to facilities.

In order to overcome these limitations, the IEEE802.15.4e was extended with the proposal of the Time Slotted Channel Hopping (TSCH) mode in 2012. The TSCH is based on time synchronization and channel hopping so able to provide a reliable connection. It is suitable for industrial environments where external interference occurs frequently because of steel structures. The Time Division Multiple Access (TDMA) enables the network to synchronize time so that the TSCH network can divide time into a fixed timeslot (e.g. 15 ms) and assigns it to nodes in the network. Each node in the network turns on the radio for a WAKE-UP state at its assigned timeslot and waits for a SLEEP state until the next turn, so reduces energy consumption. A communication cell is identified by [slotOffset, channelOffset] tuple and a repeated sequence of the cells is called a schedule. A slotframe is consist of the schedule and repeats periodically to form a network. The Absolute Slot Number (ASN) is incremented by 1 for each timeslot from the start of the network and is synchronized to the entire network so that all nodes in the network can refer to the same time.

An equation for determining the communication frequency by the ASN is as follows:

$$channel = F[(channelOffset + ASN)\ mod\ NbCannels] \quad (1)$$

F is a vector containing a predetermined channel hopping list. NbChannels is the number of usable channels which is defined 16 in 2.4GHz band. Channel hopping is implemented through a mechanism in which the communication frequency changes as the ASN changes for every timeslot.

The TSCH is suitable for industrial environments because robust to multipath fading and external interference through channel hopping.

The TSCH describes operations of the cells synchronized by time but does not describe how to schedule them. So, the IETF 6TiSCH Working Group proposed the IPv6 over the TSCH mode of IEEE802.15.4e (6TiSCH). Main purpose of the 6TiSCH is the Low Power Lossy Networks (LLNs) standard based on the IEEE802.15.4e. It focuses on development of standard technology for IPv6 application in industrial IoT environments. The 6TiSCH defines 6TiSCH Operation Sublayer (6top) to combine IEEE802.15.4e TSCH PHY and MAC layer and IPv6. The 6top enables distributed scheduling of 6TiSCH networks to add/remove TSCH cell between two different nodes. The 6top consists of one or more Scheduling Function (SF) and 6top Protocol (6P) [1]. SF determines when to add/remove cells dynamically according to network requirements and triggers 6P Transaction. The 6P Transaction refers to the negotiation process of exchanging 6P requests and responses between two nodes for adding/removing cells.

The 6TiSCH Minimal Scheduling Function (MSF) [2] is the basic SF standardized by the IETF 6TiSCH Working Group. The MSF defines how to schedule communications in distributed manner. Each node implementing the MSF negotiates with the parent node by selecting candidate cells randomly from unscheduled cells for cell addition. But, scheduling algorithm of the MSF is completely distributed. Therefore, when two adjacent pairs of nodes allocate cells having the same [slotOffset, channelOffset], they may cause an interference of each other's communication. That is, if physical distance between two nodes is sufficiently close to interfere with communication, schedule collision occurs if the same cell is allocated. When the MSF detects the schedule collision, it issues the 6P RELOCATE command to relocated collided cell to another cell to handle the schedule collision.

The Delayed Cell Relocation (DCR) proposed in this paper complements the relocation mechanism of the MSF. The DCR reduces overhead required for relocation by inducing to relocate only one pair of collided nodes. Also, it helps the network stable by relocating not the cell which has been allocated for a long time and normally operated, but the cell newly allocated which cause the collision. Simulation results show that the DCR improves performance 25% over the MSF.

This paper contributes as follows:

- When the schedule collision occurs, we induce to relocate only one pair of nodes.
- We make a node can detect status of other colliding pair of nodes' schedule collision handling.
- The proposed scheme was applied to the MSF and evaluated comparatively based on simulation.

This paper is organized as follows. Section 2 briefly describes the existing scheduling algorithm of the 6TiSCH. Section 3 introduces the proposed scheme. Section 4 explains the results of simulated comparative analysis. Finally, we conclude in Section 5.

978-1-7281-1340-1/19/$31.00 ©2019 IEEE

II. BACKGROUND

First, we describes the background knowledge.

A. Related Works

Previously, there were some attempts to block the schedule collision itself. TASA [3] or Wave [4] is a centralized algorithm which establishes communication schedule using central scheduler entity. Each node in the network sends communication requirements and topology information to the central scheduler. Based on this information, the central scheduler establishes communication schedule in which no collision occurs. However, these centralized algorithms have overhead for transmission to the central scheduler. And it is difficult to handle the topology change efficiently.

Thus, almost modern scheduling algorithms follow distributed manner for the benefits of the overhead and complexity for scheduling. MSF is the SF standardized by the IETF 6TiSCH Working Group. The MSF's bandwidth estimation algorithm periodically evaluates the number of communication cells required for each neighbor. If the difference between the number of required cells and the number of allocated cells is greater/less than the threshold, the MSF triggers the 6P Transaction to add/remove cells. That is, if the number of packets to be transmitted is greater than the number of packet which can be transmitted by currently allocated cells, a new cell is added by triggering the 6P Transaction. Similarly, if the number of packets to be transmitted is smaller, a cell is removed. The node triggering the 6P Transaction selects the candidate cell list among the unscheduled cells randomly and issues the 6P ADD command to the parent node to start the cell addition negotiation. The parent node that received the 6P ADD command compares with its own unscheduled cells and selects a cell to be added randomly then completes the cell addition negotiation. Hysteresis is added so as not to add/remove cells continuously, but the schedule collision is possible because of the random selection of the cells to be added. Orchestra [5] is a distributed solution that schedules the cells without central entity. Each node derives slotOffset and channelOffset by applying the hash function to the MAC address of the neighbor node. It is an efficient network bootstrap mechanism which establishes a basic connection between neighbor nodes without negotiation overhead so is currently used in mechanism related to the autonomous cells in the MSF.

B. Schedule Collisions in Distributed Scheduling

Distributed scheduling algorithms have advantage of low overhead and scalability for reserving cells, but the schedule collision can occur because they do not know the information of adjacent nodes. That is, because the MSF is completely distributed and randomly selects the candidate cells, the schedule collision which the MSF allocates cells having same [slotOffset, channelOffset] to two adjacent pair of nodes, thereby causing interference to each other's communication can occur. For example, Fig. 1 shows a simple 6TiSCH network with the routing parents of node C and node E are node B and node D respectively. It assumes that the link between the node C and node B can interfere to the link between the node E and node D. Because the MSF is completely distributed, the node E and node D cannot know allocated cell information between the node C and node B, and vice versa. Therefore, if the node E assigns to the node D a cell with [slotOffset, channelOffset] equal to the cell assigned to the node B and node C, the schedule collision occurs because it interferes with each other's communication. There are two cases when the schedule collision can be occurred. First one is when two pair of nodes allocate the same cell at the same time by chance. The other is when one pair of nodes allocate a cell while another pair of nodes already have allocated the same cell a long time ago. In both cases, handling the schedule collision is essential because the schedule collision can cause serious problems such as network reliability decline and increased latency.

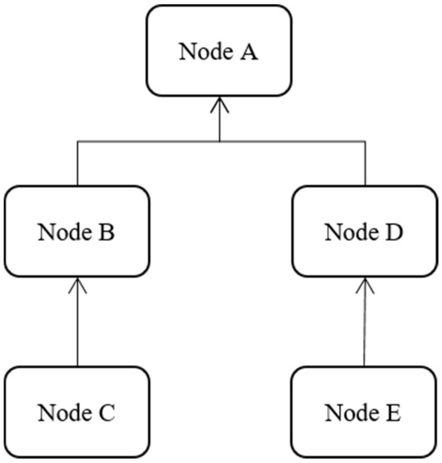

Fig. 1. The Simple 6TiSCH Network

Muraoka et al [6] proposed the housekeeping scheme that detects occurrence of the schedule collision. It is based on the fact that bundles of the cells for the same neighbor node have the same PDR. That is, the PDR of several cells for the same neighbor node almost equal. So, the schedule collision can be detected when the PDR of a cell is significantly lower than the PDR of other cells of the bundle. The schedule collision is handled by relocating the colliding cells to another cell of the schedule.

Chang et al [7] showed the concept of the Cell Equivalence through experiment evaluation. The Cell Equivalence means that when multiple cells are scheduled between two nodes, the corresponding cells must have the same PDR. In other words, cells are equal in terms of the PDR when the PDR is same regardless of the slotOffset at same distance and frequency. And the cells are also equal when collision is free because channels are hopped. Based on Cell Equivalence, Chang proposed the CCR that detects the schedule collision if a cell has the significantly lower PDR than the average PDR and relocates the cell only when relocation cost is lower.

Fahs et al [8] proposed the MECH that applies the local mutual exclusion so that neighbors cannot access the common resource at the same time. It collects the previously allocated cell information by overhearing the 6P Transactions on the shared cell. However, it does not provide a clear way to handle the schedule collision after the network bootstrapping phase.

C. Relocation Overhead

The schedule collision handling currently adopted in the MSF is based on the housekeeping. Both colliding nodes relocate the cells even if it is enough that only node relocates to clear the schedule collision. It is obviously overhead.

Moreover, because of the distributed scheduling algorithm, the cell selection mechanism to be relocated is also random. This implies that a relocated cell may not be the collision-free cell, so it would be better to reduce the relocation.

We introduced the delayed cell relocation scheme in this paper to induce only one pair of the colliding nodes relocate the cells, thereby reducing the number of relocation and reducing the overhead required for relocation. The simulation results for various parameters show that the number of relocations is reduced by about 25%.

III. DELAYED CELL RELOCATION

The purpose of this paper is to reduce the schedule collision handle overhead. The Delayed Cell Relocation (DCR) allows to prevent all two pairs of nodes relocate after the schedule collision detected. When the schedule collision occurs in a node implementing the DCR, it starts to detect whether another pair of colliding nodes relocate during the delay time. If the another node relocates during the delay time, the schedule collision is resolved and that corresponding cell can be reused. Conversely, if the schedule collision is not resolved until the end of the delay time, the node relocates the colliding cell directly. By differentiating the delay time based on the cell allocation time, the cell which has been allocated for a long time and normally operated is maintained as much as possible, and newly allocated cell will be relocated.

A. Delay Calculation

Each node implementing the DCR allocates a new cell with the ASN at that point. If the schedule collision occurs, the delay time is calculated by applying (2) based on the difference between the ASN assigned to the cell and the current ASN.

$$y = \frac{256}{1 + e^{8(0.1x-1)}} - randint(0,8) \qquad (2)$$

x is the relative ASN of the cell assuming the current ASN is 20. The longer allocation time, the smaller x. When the schedule collision is detected, the node relocates the cell not immediately but after waiting for the calculated delay time. The unit of the delay time is slotframe. The maximum of the delay time is set to 256, which is smaller than the minimum threshold with statistical significance for determining the schedule collision recommended by the MSF. As the cell allocation time becomes longer, the delay time is set to be longer so that the cell normally operating can be maintained for a long time. By subtracting arbitrary value between 0 and 8, two pairs of nodes which allocate the same cell at the approximately same time are also induced to relocate only one pair.

Fig. 2 visualizes the (2). The steep curve formulas maximizes the difference of the delay time between the long and short allocated time. If the same cell is allocated at almost the same time, the delay time of the two pairs of colliding nodes will be short, so that it works like the MSF. The case where the DCR has large gain is that a node assigns a cell where another node has already assigned and communicated normally. It is because when the schedule collision occurs, the difference of the allocation time of the colliding cell becomes higher. Therefore, the longer delay time due to the difference of the allocation time, the higher probability that the newly allocated cell is relocated while the existing cell is waiting. As the network converges, the delay time of the cell which allocated long ago will be close to 256 and newly allocated will be close to zero. So, likelihood of the old cell's relocation become low, thus the network convergence state is continuously maintained.

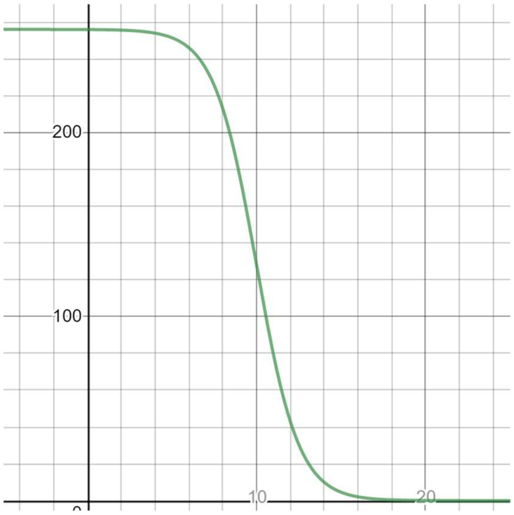

Fig. 2. The graph of the Delay Time

B. Keep-Alive Packet Transmission

Because the MSF is fully distributed, it is hard to know whether other nodes handle the schedule collision during the delay time. In other words, each node must wait until the delay time is over because cannot know that other nodes relocate the colliding cell or not. Therefore, Algorithm 1 is applied to detect whether the schedule collision is handled. To do this, each node transmits the Keep-Alive (KA) packet in the colliding cell during the delay time. If both colliding node pairs don't relocate the cell yet, the PDR of the KA packets is low because of the interference. When a pair of nodes relocates the cell and stops transmitting the KA packet, the schedule collision is resolved so that the PDR of the KA packets of the another pair would be increased. So, if the PDR of the KA packets increases, it is judge that the cell relocation is performed in another node pair. The relocation of the delayed cell will be stopped and the corresponding cell reused if the schedule collision is resolved during the delay time.

Algorithm 2 updates the PDR of the KA packets for the colliding cells. Each node maintains a bitmask for the KA packets to manage the PDR information with minimized memory usage. If the schedule collision is detected, the bitmask of the colliding cell is initialized to zero. Each time a KA packet is transmitted, the bitmask performs RIGHT-SHIFT to create space in which store ACK information of the last KA packet. If the ACK of the KA packet is received, a first bit of the bitmask is set to 1; otherwise, set to 0. Based on the bitmask value, it is determined that the PDR is improved when the transmission and reception of the last 4 KA packets are completed well. Since the bitmask performs RIGHT SHIFT, it is possible to judge by a simple operation which only necessary to check if decimal value of the bitmask is larger than 0xF0 which is 1111000b.

Algorithm 1. IsColliisionResolved

```
Output:
  isCollisionFree, TRUE if collision free or FALSE
Function: IsCollisionFree
ackReceived ← Send(KA)
collisionPDR ← UpdateCollisionPDR(collisionPDR,
                                  ackReceived)
if IsImprovedCollisionPDR(collisionPDR) then
    return TRUE
else
    return FALSE
```

Algorithm 2. UpdateCollisionPDR

```
Input:
  collisionPDR, bitmasked PDR before the last KA packet
  ackReceived, ACK of the last KA packet received or not
Output:
  collisionPDR: bitmasked PDR with the lask KA packet
Function: UpdateCollisionPDR
collisionPDR ← ( collisionPDR >> 1 )
if ackReceived is TRUE then
    collisionPDR ← ( collisionPDR OR 0x80 )
else
    collisionPDR ← ( collisionPDR AND 0x7F )
endif
return collisionPDR
```

IV. SIMULATION

In order to compare performance of the DCR proposed by this paper with that of the MSF, we developed a simulator that implements the mechanism of the SF. We assume that the nodes are arranged as shown in Fig. 1, and that all nodes can affect communication of the other nodes. A node generates a packet with probability of 75% for each cell, thereby increasing the number of packets in the entire network so that assuming a situation where the schedule collisions occur more frequently due to cell addition negotiations. If there is no collision, the probability of the communication success between two nodes is 100%. In a cell with the same [slotOffset, channelOffset], interference occurs when both nodes transmit packets. We used the MSF as the SF and compared and analyzed the DCR implementation and non-implementation. The length of the slotframe is 101, and the number of usable channel is 16 equal to the standard. The detailed parameters are shown in Table 1. The simulation was performed 8 times for the same parameters. The simulation results were compared and analyzed using the mean value.

TABLE 1. The Simulation Parameter

RUNNING TIME	60 min
NUMBER OF NODES	5
NUMBER OF FREQUENCIES	16
SLOTFRAME LENGTH	101
TIMESLOT LENGTH	10 ms
RELOCATE PDRTHRES	0.5
LIM NUMCELLUSED HIGH	0.75
LIM NUMCELLUSED LOW	0.25
HOUSEKEEPING PERIOD	60 s

A. Analysis about the Number of Relocation

Fig. 3 shows that the implementation of the DCR reduces the cumulative number of cell relocation by about 25% compared to the MSF. When the slotframe length is 101 and the number of channels is 16, the probability that two adjacent neighbor nodes select the same slotOffset and channelOffset is 1 / 1616. So the schedule collision rarely occurs in the early stage of the simulation. However, as more and more cells are allocated, the probability increases so that the cell relocation becomes frequent. This can be confirmed by the steep slope in the middle of the simulation. Since the allocation time for both colliding cells is similar in early phase, there are many cell relocation and so the performance is improved just a little bit. And then the simulation result graph becomes flattened as the network progressively converges and the cell relocation hardly occur. That is, as the simulation progresses, the performance increases.

B. Analysis about the Allocation Time

The DCR induces to relocate a newly allocated cell rather than a cell that has been allocated long ago and operated normally. Fig. 4 shows the boxplot about the allocation time of the relocated cells during the simulation. Because the MSF relocates the colliding cells as soon as the schedule collision is detected, the DCR's allocation time is smaller. As a result, the old cells are continually maintained and the newly allocated cells are relocated, which leads the network to remain converged.

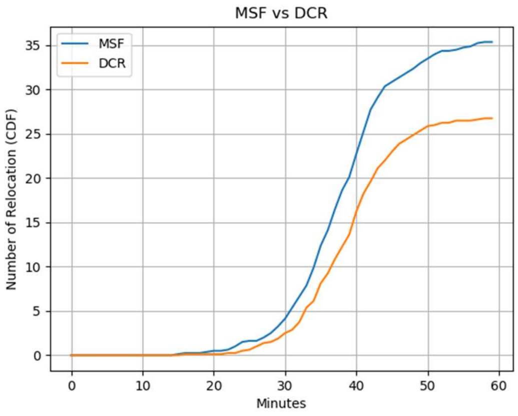

Fig. 3. The Number of Relocation (CDF) using DCR or not

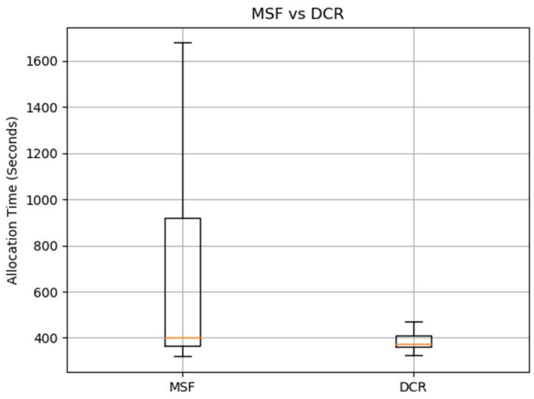

Fig. 4. The Allocation Time in Seconds using DCR or not

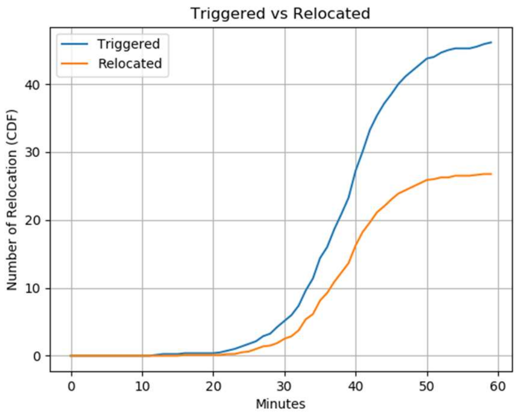

Fig. 5. The Number of Cells Triggered and Relocated

C. *Triggered vs Relocated*

Fig. 5 shows the number of actually relocated cells against the number of triggered cells during the simulation. As the simulation progresses, the allocation time between two colliding cells gradually increases, so that the difference of the delay time becomes larger. So, about 43% fewer cells are relocated because of the difference between the delay time. This means that 43% of the colliding cells are reused.

D. *Analysis about the Maximum Delay Time*

Fig. 6 and Fig. 7 show the performance analysis according to the maximum delay time. With the maximum delay time 128, 256 and 512 slotframe, the number of cell relocation decrease about 18%, 25% and 32%, respectively. The simulation results show that the larger maximum delay time, the fewer cell relocation. This is because, as the maximum delay time is larger, the difference between long-allocated cell and newly allocated cell is higher, and the probability that the newly allocated cell would be relocated increases. That is, as the network progresses, the polarization between the old cell and the new cell grows. Because the network is gradually converged, it is more frequent that a new cell is allocated with the same [slotOffset, channelOffset] as that of normally operating old cell. In this case, the difference of the delay time between the two colliding node pairs would be almost the maximum delay time. The larger the delay time is, the more opportunity is given to relocate the cell by only one node pair, so the total number of cell relocation decreases. However, if the maximum delay time becomes too large, the node would have to wait longer, so it is necessary to set the tradeoff between the maximum delay time and the performance improvement. In this paper, we set the maximum delay time to 256, which is the minimum value until a cell becomes statistically meaningful when using the MSF.

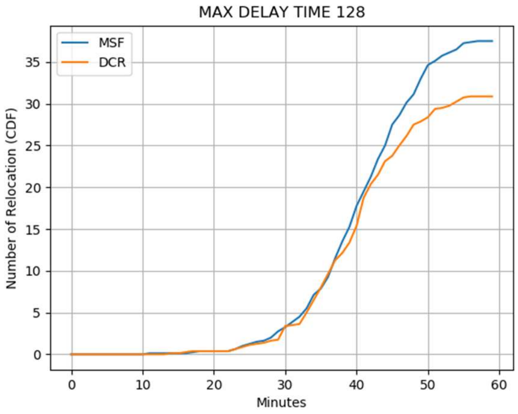

Fig. 6. The Number of Relocation (CDF) with Maximum Delay Time 128

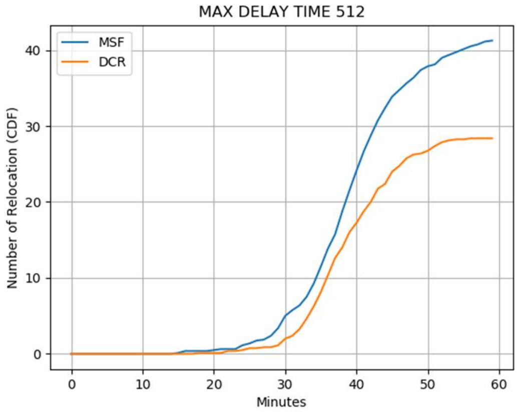

Fig. 7 The Number of Relocation (CDF) with Maximum Delay Time 512

V. CONCLUSION

Each sensor node of the TSCH based Wireless Sensor Network implementing the 6TiSCH standard and operating in a distributed environment allocated cells independently without any knowledge about each other. So it is possible to collide the schedule. Handling the schedule collision is essential because the schedule collision can cause serious problems such as degraded network reliability and increase latency. The housekeeping based schedule collision handle process applied to the MSF has overhead because both colliding nodes relocate the colliding cell even if it is to free the schedule collision with only one node's cell relocation. Therefore, in this paper, we made node relocate the cell after waiting for different delay time so that only one pair of nodes relocate the cell. With the simulation on various parameters, it was showed that there is an effect of reducing the number of cell relocation about 25%.

VI. ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2018R1D1A1B07049355).

REFERENCES

[1] Q. Wang, X. Vilajosana, T. Watteyne, "6TiSCH Operation Sublayer (6top) Protocol (6P)", RFC 8480, Internet Engineering Task Force, November 2018

[2] T. Chang, M. Vucinic, X. Vilajosana, S. Duquennoy, D. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)", Internet-Draft draft-ietf-6tisch-msf-01, April 2019, work in Progress. [Online]. Available: https://datatracker/ietf.org/doc/draft-ietf-6tisch-msf

[3] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco and G. Boggia, "Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks," 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC), Sydney, NSW, 2012, pp. 327-332.

[4] Soua, R., Minet, P., and Livolant, E. (2016) Wave: a distributed scheduling algorithm for convergecast in IEEE 802.15.4e TSCH networks. Trans. Emerging Tel. Tech., 27: 557–575.

[5] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. 2015. Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15). ACM, New York, NY, USA, 337-350.

[6] K. Muraoka, T. Watteyne, N. Accettura, X. Vilajosana and K. S. J. Pister, "Simple Distributed Scheduling With Collision Detection in TSCH Networks," in IEEE Sensors Journal, vol. 16, no. 15, pp. 5848-5849, Aug.1, 2016.

[7] Chang T, Watteyne T, Vilajosana X, Wang Q. CCR: Cost‐aware cell relocation in 6TiSCH networks. Trans Emerging Tel Tech. 2018;29:e3211.

[8] A. J. Fahs, R. Bertolini, O. Alphand, F. Rousseau, K. Altisen and S. Devismes, "Collision prevention in distributed 6TiSCH networks," 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Rome, 2017, pp. 1-6.