Article

A Scheduling Method Based on Packet Combination to Improve End-to-End Delay in TSCH Networks with Constrained Latency

Dinh Loc Mai and Myung Kyun Kim *

IT Convergence Department, University of Ulsan, Ulsan 44610, Korea; maindhloc@mail.ulsan.ac.kr

* Correspondence: mkkim@ulsan.ac.kr

Received: 17 May 2020; Accepted: 9 June 2020; Published: 12 June 2020

Abstract: Wireless sensor networks (WSN) are networks for gathering data from sensor nodes that have been applied in industry for a long time. In real-time industrial applications with tight latencies, schedulability is one of the most critical issues. Some authors have proposed centralized scheduling algorithms for time-slotted channel hopping (TSCH) networks for real-time applications in industry, however, they have some disadvantages such as schedulability and high data traffic. In this paper, we improve the schedulability, latency, data traffic by dynamically prioritizing packets which are based on number duplex-conflicts and dynamically combining the packets. As a result, we show that the packet-combining algorithm improves schedulability and minimizes the amount of traffic in a network when compared with existing approaches.

Keywords: TSCH networks; IEEE 802.15.4e; centralized scheduling; schedulability

1. Introduction

Nowadays, a lot of applications are deployed with wireless sensor networks (WSNs). In these applications, sensor data is transmitted from sensor nodes to a network manager, and especially for real-time applications, sensor nodes have latency requirements. Latency on the order of tens of milliseconds for manufacturing, seconds for process control, and minutes for monitoring is very important for real-time applications [1,2]. However, it is not easy to achieve the above requirements because of collisions and the sheer number of transmissions. We propose herein a scheduling algorithm to reduce latency, eliminate collisions and guarantee latency requirements.

There are some standards for low-power wireless sensor networks which uses IEEE 802.15.4e for the media access control (MAC) protocol and applied in industry such as timeslot channel hopping (TSCH), WirelessHART [3] and ISA100.11 [4]. TSCH, WirelessHART [3] and ISA100.11a [4] use a time division multiple access (TDMA) mechanism. However, TSCH and WirelessHART use channel hopping to be highly resistant to interference and multipath fading which are some of the critical issues in wireless sensor networks [5]. ISA100.11a uses two types of frequency hopping called slotted and slow hopping [6]. In slotted hopping, each timeslot uses a new channel which is predefined by a hopping pattern for communication. In slow hopping, the same channel is used in several consecutive timeslots before it is changed. In our paper, we study real-time periodical transmission scheduling from sensor nodes to a central station in a TSCH network.

There are studies focused on reducing the latency, which can be classified into two categories: scheduling methods [1,7–9] and random access methods [2,10]. Scheduling methods can reduce the latency based on eliminating the collisions among the nodes in the network, using parallel transmissions to reduce slot frame size, and using data aggregation to reduce number of transmissions. Random access methods can reduce the latency by using multiple access based on using multiple channels.

Our paper proposes a scheduling algorithm that is based on combining packets dynamically and dynamically prioritizing each packet transmission based on its laxity (i.e., the remaining time before the end-to-end deadline) applied to a TSCH network with constrained latency to increase schedulability and reduce the number of data transmissions and end-to-end (E2E) delay. As our set of scenarios show, our proposal improves schedulability and reduces the number of transmissions and E2E delay when compared with existing approaches in PCLLF [1] and CLLF [11].

The remainder of this paper is organized as follows: Section 2 describes related works. Section 3 introduces our system and application models. Section 4 describes our proposed approach in detail. Section 5 presents the evaluation results, with conclusions following in Section 6.

2. Related Work

A data-aggregation technique is used in gathering data which is collected from sensor nodes in WSNs to reduce energy consumption, reduce the number of transmissions, latency, and energy consumption.

With constrained latency, some studies have attempted to improve latency and E2E delay with methods such as PCLLF [1] and CLLF [11], which propose heuristic algorithms based on the amount of duplex conflict transmissions to prioritize all transmissions to choose the transmission which has the highest priority to schedule in each timeslot and each channel which does not collide with neighbors. However, there are many transmissions, so traffic is high and it has low schedulability with low periodicity.

Some other studies work with data-aggregation to reduce data traffic with centralized scheduling like TEACSA [12], DAHDA [13], OCCN [14] and with distributed scheduling, like DAS [15], PPSDA [16].

With centralized scheduling:

- ✓ In [12], the authors proposed a scheduling approach (TEACSA) which is based on the concept of an aggregation tree. A time slot scheduling method is used so this approach minimizes the time required for data gathering by aggregation convergecast. The value of the transmission is determined by the sum of number of children and receiver's depth. The tree is constructed by appending to it the path which has least overhead in each iteration. To supplement the scheduling algorithm a neighbor degree ranking scheme is implemented to assign time slots to the SNs in an effective manner [17]. It reduces data traffic, however, data transmissions are not periodical and the latency of each node is not constrained.
- ✓ In [13], the authors proposed a new algorithm named Dynamic Adaptive Hierarchical Data Aggregation (DAHDA) which extended the functionality of LEACH-C. In DAHDA, all of clusters are assigned weights which are based on their residual energy and density. Based on the weights, the algorithms decide which node should be a cluster head (CH) where data packets are combined. It improves energy consumption because it reduces data traffic, however, data transmissions are not periodical and the latency of each node is not constrained.
- ✓ In [14], the authors proposed a new method named Optimal Clustering in Circular Networks (OCCN) which aims to reduce energy consumption, data traffic and increase the lifetime of wireless sensor networks [18]. In that method, which was proposed for a circular area surrounding a sink, one hop communication between the cluster heads and the sink was replaced by an optimal multi-hop communication. Data packets are combined at cluster head nodes. It reduces the amount of traffic, however, data transmission is not periodical and the latency of each node is not constrained.

With distributed scheduling:

- ✓ In [15], the authors proposed a distributed energy-balance algorithm (DAS) which aims to balance the energy consumption for aggregators. In this algorithm, first, trees rooted at nodes are formed which are termed virtual sinks, to balance the number of children at a given level to level the energy consumption and then try to assign timeslots for all nodes to satisfy the minimum latency

and energy consumption demand. Data packets are combined at parent nodes. This algorithm reduces latency and data traffic, but data transmission is not periodical and the latency of each node is not constrained.

- ✓ In [16], the authors proposed a privacy-preserving in-network aggregation protocol (PPSDA) for wireless sensor networks which is based on the concept of data slicing, mixing and merging. Sensor nodes slice the data into some pieces then send these pieces to neighbor nodes which have the same key as this node. After receiving pieces of data from neighbor nodes, this node combines its own data then sends in to a sink node. This algorithm doesn't reduce the amount traffic and is low reliability.

We propose scheduling which is based on dynamic packet combination and dynamically prioritizes the packets with constrained latency.

3. System Model

3.1. System Model

We consider a TSCH network in which there is number of sensor nodes, one sink node and one network manager. Each sensor node has a latency requirement that is the period of nodes and each node has a fixed data size that is distance the data sensor node has to transmit to the network manager.

In the TSCH, we define periodic cycles which are called slotframes which have a fixed number of timeslots which are defined as a length of slotframe as shown in Figure 1: N_S . A single time slot length used in the IEEE.802.154e standard is 10 ms that is sufficient time for the transmitter to transmit packets which have a maximum length of 127 bytes, and the receiver transmits an ACK packet if required, as shown in Figure 2. In the TSCH, an absolute slot number (ASN) is defined to determine the number of timeslots which has already elapsed since the network started. There are 16 available channels which are used to communicate between nodes which value is from channel 11 to 26. In each timeslot, the transmitter transmits a packet after a TXOffset value and the receiver starts a listening packet after an RXOffset value from the beginning of this timeslot as shown in Figure 2. We define an assignment cell (timeslot, channel) where a timeslot is when the node wakes up and the channel which the node uses to transmit or receive packets. To eliminate collisions and interference, we assign only one transmission for each cell, so the maximum number of parallel transmissions equals the number of channels which is used in each timeslot.

Figure 1. A TSCH superframe consisting of N_S time slots and the transmission structure in one slotframe.

Figure 2. Timing in one timeslot in TSCH network.

3.2. Application Model

Figure 3 shows a network example, in which the period value of each node is the number of timeslot nodes generated by one packet which has to arrive at a sink node in this period and the data size is the data length (bytes) that this node has to transmit to the network manager. For example, node 5 has a period of 8 which means that every 8 timeslots, it generates one packet which has to arrive at a sink node (node 1) in this 8 timeslots, and it has a data size of 20 that means it has to send 20 bytes of data to the sink node.

Figure 3. Network example.

In our paper, the TSCH network is formed as tree network topology which is a directed acyclic graph (DAG) $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of all sensor nodes and arcs in E are set as tree links in the network. The link in E is identified by a pair of nodes that are parent and child nodes. For example, link $v_i v_j$ where v_i is a child node and v_j is a parent node and data packets are transmitted from v_i to v_j . In the network, packet collision can occur when duplex conflicting links are scheduled in the same timeslot. Two links are duplex conflicting links when they use the same sender or receiver node. The set of all duplex conflicting links with $v_i v_j$ is $CLNF(v_i v_j)$, for example $CLNF(v_7 v_3) = \{v_9 v_7, v_3 v_1, v_8 v_3\}$ in Figure 3. Packet collision can occur when interfering links is scheduled in the same timeslot and channel. Two links are interfering links when two receiver nodes are neighboring nodes.

In each period, each node v_p generates a flow F_p and $F = \{F_1, \dots, F_n\}$ is set of all of flows in this period. Because the periods of nodes can be different, the number of packets a node has to transmit to the sink node can be different. When each period starts, all nodes generate data packets. Each node has a fixed data length to send to the sink node and data length can have different values between nodes. Each flow F_i is initially released periodically with period T_i and starts from the beginning of a slotframe as shown in Figure 4. The j -th instance of F_i in a slotframe transmits the j -th packet which is denoted by p_{ij} (L) where L is the data length which is contained in a packet. The released timeslot of F_i for the j -th packet is at timeslot $(j - 1) \times T_i$ in a slotframe as shown in Figure 3. To guarantee the latency requirement, the j -th packet which is generated by F_i has to reach to sink node before the $(j + 1)$ -th packet is generated. The transmission of p_{ij} is from node v_s to v_r where node v_r is k

hops away from a sink node and has a data length which is L bytes, named $T_{i,k}^j(L)$. The length of a slotframe is determined by the least common multiple (LCM) of the periods of all nodes in the network. The number of packets which is generated by F_i in one slotframe is determined by LCM/T_i .

Figure 4. Transmissions of the j -th packet generated by flow F_i .

The combined packets are denoted $TC_{i,k}^j(L)$ where v_j is a node has data in this packet which has the highest hop-count, and there are k hops between the receiver and the sink node as explained in next section.

4. Our Approach

In this section, we describe our approach. Our major contribution is as follows: first, in Section 4.1, we describe grouping nodes and how to dynamically combine packets. In Section 4.2, we describe the packet combining mechanism. In Section 4.3, we define the priority value for the transmission of each packet. In Section 4.3, we present our algorithm. The notations which are used in this section are summarized in Notations Table.

4.1. Grouping Node Mechanism and Dynamically Combining Packet

We propose a node grouping mechanism that is a dynamic grouping node which based on the data length of each packet, the earliest combination timeslot and the latest timeslot to determine which packet can combine with other packets and which node will combine packets to reduce the number of transmissions and increase schedulability. This progress is repeated each timeslot. The conditions for determining which node will combine one packet with another packet are as follows:

- Total data size in a subtree at this time slot where a root node is the node we consider is less than the maximum payload size (MPS).
- $LT(i) - ECT(i) \geq NC_i$ where NC_i is the number of child nodes which has packets in queue in the timeslot of node i .

We propose $ECT(i)$ is the earliest combined timeslot when all of the packets in subtree node i can arrive at node i

- $ECT(i) = \text{MAX}(ETR(j)) + NC_i$ where j is the child of node i and NC_i is the number of children of node i .
- $ECT(i) = \text{current timeslot}$ if the node has the highest hop-count to a sink node when node i has packets in a queue in the path that contains node i .

We propose $LT(i)$ that is latest timeslot node i that must to transmit a packet to a parent of node i to guarantee schedulability for all packets in this subtree.

- $LT(i)$ depends on the deadline of packets in the subtree.

In timeslot 0, all nodes have packets in the queue. Figure 5 shows ECT value of all nodes in the network at timeslot 0 with an example as shown in Figure 1.

Figure 5. Earliest combining timeslot for all nodes at timeslot 0.

In Figure 6, we show LT value of all nodes in the subtree at timeslot 0 with the example as shown in Figure 1.

Figure 6. Latest timeslot for all nodes at timeslot 0.

In the 802.15.4.e standard, we have a maximum packet size is 127 bytes and the MAC header length is 21 bytes so $MPS = 127 - 21 = 106$ bytes. We choose $MPS = 100$ bytes. In Figure 1, at timeslot 0, the example is node 5, based on the condition which we proposed above. We have a total data size in subtree 5 of 45 bytes $< MPS = 100$ bytes and $LT(5) - ECT(5) = 6 - 1 \geq 1$, so data node 5 can combine with packet node 6. With node 2, we have a total data size in the subtree of 65 bytes $< MPS = 100$ bytes and $LT(2) - ECT(2) = 7 - 2 = 5 \geq 1$ so data from node 2 can combine with data from nodes 5,6. We repeat this action with all of the parent nodes which children nodes have packets in the queue and we obtain results about which data packet can combine with another and which nodes will combine packets.

Figure 7, at timeslot 0, shows that after node 5 receives a packet from 6, this packet will be combined with packets which are generated at node 5. Then data from node 2 will combine with the combined packet from nodes 6, 5. Data from node 7 will combine with that of node 9 at node 7. Data from node 8 will combine with that of node 10 at node 8. Data from node 11 will combine with that from nodes 12,13 at node 11 and data from node 4 will combine packets nodes 11,12,13 at node 4.

Figure 7. Group node results at timeslot 0.

In each timeslot, with each node which is not in any group, if in this node's queue has two packets, we combine packets if the total data size in the two packets less than MPS. This action is referenced as combining transmissions.

In Figure 8, we present the combining packet action at node 4. In the node 4 queue, there are two packets from node 7 which are referenced by $T_{7,1}^1(20)$ and a packet from node 4 which is referenced by $T_{4,1}^1(30)$, the combination packet from node 7 with node 4 is referenced combining transmissions:

- $T_{4,1}^1(30) + T_{7,1}^1(20) \rightarrow TC_{4,1}^1(50)$ we remove $T_{4,1}^1(30)$, $T_{7,1}^1(20)$ and add $TC_{4,1}^1(50)$ to the transmission list.
- $T_{4,0}^1(30) + T_{7,0}^1(20) \rightarrow TC_{4,0}^1(50)$ we remove $T_{4,0}^1(30)$, $T_{7,0}^1(20)$ and add $TC_{4,0}^1(50)$ to the transmission list.

Figure 8. Combine transmissions example at node 4 in the path 7,4,2. (a) Transmissions in the path 7,4,2 before combining transmissions. (b) Transmissions in the path 7,4,2 after combining transmissions.

4.2. Packet Combining Mechanism

Figure 9 shows the packet format which contains a MAC header and payload.

Figure 9. Packet format.

When a node receives a packet, the received packet is combined with packet(s) in queue when the total data size less is than MPS, as shown in Figure 10.

Figure 10. Packet combination example.

Figure 10 shows an example of packet combination. When Packet 1 and packet 2 combine together, we combine the payloads of the two packets together.

4.3. Priority Assignment

In this section, we use a priority value for each transmission which is based on the remaining time available in a timeslot in which nodes can be scheduled to increase the schedulability. A higher remaining amount of an available timeslot means the transmission is easier to schedule within the limited laxity to the deadline. It indicates a lower priority and has a larger space for transmission which can be scheduled.

To determine the priority value of each transmission, we use a timeslot window for each transmission $T_{i,k}^j$ or $TC_{i,k}^j$ at timeslot, where t is defined as $TW(T_{i,k}^j, S)$ which is updated every timeslot. The timeslot window shows when a transmission can be released and the latest timeslot in which this transmission must be scheduled to guarantee the latency of packets. The earliest timeslot and latest timeslot of transmission $T_{i,k}^j$ is when the transmission can be scheduled, defined by $ET(T_{i,k}^j)$ and $LST(T_{i,k}^j)$ at timeslot S .

- $ET(T_{i,k}^j) = \text{MAX}(N_i - k + T_i^j, S)$
- $LST(T_{i,k}^j) = T_i^j - k - 1$

where transmission $T_{i,k}^j$ is k hops away from the destination node (sink node) and N_i is hops away from the receiver node.

$ET(T_{i,k}^j)$ of transmission $T_{i,k}^j$ depends on the current timeslot S , and it computed in each timeslot. $LST(T_{i,k}^j)$ is dependent on the period value and it has a fixed value.

For example packet $p_{m,n}$ will be combined with packet $p_{i,j}$; the transmission which transmits packet $p_{m,n}$ is $T_{a,b}^k [x:y]$ and the transmission which transmits packet $p_{i,j}$ is $T_{c,d}^z [t:q]$. Combining packets $p_{i,j}$ and $p_{m,n}$ is referenced by the combined transmission $T_{a,b}^k [x:y], T_{c,d}^z [t:q]$:

- $T_{a,b}^k [x:y] + T_{c,d}^z [t:q] \rightarrow TC_{a,b}^k [\text{MAX}(x,t):\text{MIN}(y,q)]$: adds transmission $TC_{a,b}^k$ and removes $T_{a,b}^k, T_{c,d}^z$.

Table 1 shows time window values for all transmissions before grouping at timeslot 0. After grouping nodes, the transmission list is updated. Table 2 shows the time window values for all transmissions before grouping at timeslot 0.

Table 1. Time window value for all transmissions at timeslot 0 before grouping nodes.

$TW(T_{i,k}^j, 0)$		$TW(T_{i,k}^j, 0)$		$TW(T_{i,k}^j, 0)$		$TW(T_{i,k}^j, 0)$		$TW(T_{i,k}^j, 0)$	
$T_{2,0}^1$	[0:7]	$T_{6,2}^1$	[0:13]	$T_{7,1}^1$	[0:6]	$T_{8,1}^1$	[0:14]	$T_{13,2}^1$	[0:13]
$T_{2,0}^2$	[8:15]	$T_{6,1}^1$	[1:14]	$T_{7,0}^1$	[1:7]	$T_{8,0}^1$	[1:15]	$T_{13,1}^1$	[1:14]
$T_{5,1}^1$	[0:6]	$T_{6,0}^1$	[2:15]	$T_{7,0}^2$	[8:14]	$T_{11,1}^1$	[0:14]	$T_{13,0}^1$	[2:15]
$T_{5,0}^1$	[1:7]	$T_{3,0}^1$	[0:7]	$T_{7,1}^2$	[9:15]	$T_{11,0}^1$	[1:15]	$T_{10,2}^1$	[0:13]
$T_{5,1}^2$	[8:14]	$T_{3,0}^2$	[8:15]	$T_{9,2}^1$	[0:13]	$T_{12,2}^1$	[0:13]	$T_{10,1}^1$	[1:14]
$T_{5,0}^2$	[9:15]	$T_{4,0}^1$	[0:7]	$T_{9,1}^1$	[1:14]	$T_{12,1}^1$	[1:14]	$T_{10,0}^1$	[2:15]
		$T_{4,0}^2$	[8:15]	$T_{9,0}^1$	[2:15]	$T_{12,0}^1$	[2:15]		

Table 2. Time window value for all transmissions at timeslot 0 after grouping.

TW($T_{i,k}^j, 0$)		TW($T_{i,k}^j, 0$)		TW($T_{i,k}^j, 0$)		TW($T_{i,k}^j, 0$)		TW($T_{i,k}^j, 0$)	
$T_{2,0}^2$	[8:15]	$T_{6,2}^1$	[0:13]	$T_{3,0}^1$	[0:7]	$T_{10,2}^1$	[0:13]	$T_{9,2}^1$	[0:13]
$T_{5,1}^2$	[8:14]	$TC_{6,1}^1$	[1:6]	$T_{3,0}^2$	[8:15]	$TC_{10,1}^1$	[1:14]	$TC_{9,1}^1$	[1:6]
$T_{5,0}^2$	[9:15]	$TC_{6,0}^1$	[2:7]	$T_{13,2}^1$	[0:13]	$TC_{10,0}^1$	[2:15]	$TC_{9,0}^1$	[2:7]
$T_{4,0}^2$	[8:15]	$T_{7,0}^2$	[8:14]	$TC_{13,1}^1$	[1:14]				
$T_{12,2}^1$	[0:13]	$T_{7,1}^2$	[9:15]	$TC_{13,0}^1$	[2:15]				

The second parameter that accounts for prioritizing $T_{i,k}^j$ in the current time slot S is the average number of collision posed by the collisions to each remaining transmission (i.e., $T_{i,k}^j, T_{i,k-1}^j, \dots, T_{i,0}^j$). To do so, we consider the number of duplex conflicting collisions. A duplex conflicting collision is imposed by a transmission $T_{i,k}^j$ existing on a conflicting link in $CNF(v_p v_q)$ where its lifetime ($TW(T_{i,k}^j, S)$) overlaps with $TW(T_{i,k}^j, S)$. We define $N_{cnf}(T_{i,k}^j, S)$ that is the number of duplex conflict transmissions with transmission $T_{i,k}^j$ in timeslot S and $N_{cnf}^{avg}(T_{m,n}^k, S)$ that is the average number of duplex conflicting transmissions with $T_{m,n}^k$ in timeslot S :

$$N_{cnf}^{avg}(T_{i,k}^j, S) = \sum_{i \leq k} N_{cnf}(T_{i,k}^j, S) / (k + 1)$$

The priority of transmission $T_{i,k}^j$ or $TC_{i,k}^j$ in the timeslot S is defined by:

- $PR(T_{i,k}^j, S) = |TW(T_{i,k}^j, S)| - N_{cnf}^{avg}(T_{i,k}^j, S)$ [1]
- $PR(TC_{i,k}^j, S) = |TW(TC_{i,k}^j, S)| - N_{cnf}^{avg}(TC_{i,k}^j, S)$.

where $|TW(T_{i,k}^j, S)|$ or $|TW(TC_{i,k}^j, S)|$ is the number of available timeslots in which transmission $T_{i,k}^j$ or $TC_{i,k}^j$ can be scheduled at time slot S . For example, in Table 3 we present $N_{cnf}^{avg}(T_{i,k}^j, S)$, $PR(T_{i,k}^j, S)$ for all transmissions after grouping in timeslot 0, in the network given in Figure 1.

Table 3. $N_{cnf}^{avg}(T_{i,k}^j, S)$, $PR(T_{i,k}^j, S)$ for each transmission after grouping in timeslot 0.

TW($T_{i,k}^j, 0$)		$N_{cnf}^{avg}(T_{i,k}^j, 0)$	Priority Value	TW($T_{i,k}^j, 0$)		$N_{cnf}^{avg}(T_{i,k}^j, 0)$	Priority Value
$T_{12,2}^1$	[0:13]	3.33	9.67	$T_{3,0}^1$	[0:7]	6	1
$T_{9,2}^1$	[0:13]	2.33	11.67	$T_{6,2}^1$	[0:13]	2.67	11.23
$T_{10,2}^1$	[0:13]	5	8	$T_{13,2}^1$	[0:13]	3.33	9.67

4.4. PC-PCLLF: Packet Combining with Path Conflict Aware Least Laxity First Algorithm

Our proposed: PC-PCLLF scheduling algorithm is defined as follows (Algorithm 1). The input parameters of the PC-PCLLF algorithm are G, F, D, P, n_{ch}, S , and the output is a feasible scheduling solution. In the first step, the algorithm defines a transmission list for all of packets. In line 2.1, for each timeslot S , we detect nodes and which packets can be combined as explained above to combine transmissions and consider the node's queue which is not in any group in the grouping step. If in this node's queue there are two packets, we combine these two packets if the total data size is less than MPS. Then, we update the transmission list. After that, we calculate released (S) based on the transmission list and the schedule list. We calculate a priority value of all transmissions as described above. Our algorithm orders all released transmission, based on the priority value as explained in the section above then we choose the highest priority (the transmission that has the lowest priority value). Otherwise, our algorithm continues to schedule the highest priority transmissions in the current time slot in line 2.5.2. We repeat this progress until all transmissions are scheduled or there is a transmission which misses the deadline.

Algorithm 1: PC-PCLLF (G, F, D, P, n_{ch})

```

1: Initiate the current time slot  $S = 0$ , calculate transmission list, and define  $\Pr(T_{i,k}^j, 0)$  for each  $T_{i,k}^j$ 
2: While (not all the transmissions are scheduled) do
2.1: Grouping node, combine packet, and update transmission list
2.2: Calculate the set Released( $S$ )
2.3: Calculate  $\Pr(T_{i,k}^j, S)$  for all transmissions in Released( $S$ )
2.4: If there is a transmission miss deadline ( $LST(T_{i,k}^j, S) > S$ )
    2.4.1 Return “unschedulable”, Exit.
2.5: Else
    2.5.1 Initiate the channel counter,  $n = 0$ 
    2.5.2 While ( $n \leq n_{ch}$ ) do
        2.5.2.1 Select a transmission in released transmission list with the highest priority, transmission  $T^*$ 
        When several transmissions have the same priority value, assign a higher priority to a transmission
        with the smallest length for its time window.
        2.5.2.2 Assign transmission  $T^*$  to current slot  $S$  on current channel  $m$ 
        2.5.2.3 Remove transmission  $T^*$  and all transmissions conflicting with transmission  $T^*$  from
        Released( $S$ )
        2.5.2.4 Go to the next channel offset,  $n = n + 1$ 
    2.5.3 End while
    2.5.4 Go to next time slot  $S = S + 1$ 
2.6 End if
3: End While
4: Return the Schedule table

```

5. Performance Evaluation

In this section, we evaluate the performance of PC-PCLLF using the schedulability ratio, E2E average delay and number of transmissions as metrics. We evaluate the metrics of our algorithm by changing the network scale, period and data size of each node. We compare the performance of PC-PCLLF with other approaches like PCLLF and CLLF which are applied to networks with constrained latency. The main difference between our approach and PCLLF or CLLF is that our approach uses dynamically combined packets.

In our experiments, all networks are generated with a random tree network topology with the same or different numbers of nodes. With one topology, we assign a number of child nodes randomly selected in the range [1–3].

First, we conduct experiments to evaluate the performance according to the network scale, period and data size of each node of our approach, CLLF and PCLLF. The experiments are classified as with loose and tight periods. In the experiments with loose periods, we evaluate the performance of algorithms on case studies with periods randomly chosen among the range of $[2^5-2^8]$ time slots. In the hard experiments, we use periods randomly chosen among the range of $[2^4-2^6]$ time slots, respectively. In each experiment, data size is chosen randomly (15,20,25,30) bytes. To evaluate exactly the performance of our approach when compared with PCLLF and CLLF, we choose a small period in this case, where it is more difficult to achieve schedulability. In these experiments, with 10 nodes, we generate 100 random trees, with 20 nodes we generate 150 random trees and with 30 or 40 nodes, we generate 200 random trees. The schedulability ratio value is calculated by the number of trees we can get schedule tables for divided by the total number of random trees.

Figure 11 shows the schedulability ratio per number of nodes in the network. The horizontal axis is the number of nodes in the network, which is in the range [10,20,30,40]. When the number of nodes increases, the schedulability ratio decreases because in each cell, we schedule only one transmission to eliminate collisions, so the number of cells is limited. In all experiments, our approach has schedulability that is not less than that of other approaches. Especially, in experiments with 40 nodes with loose periods and 30,40 nodes with tight periods our approach is much better than other

approaches because our approach uses dynamically combined packets so it minimizes the number of transmissions.

Figure 11. Schedulability ratios obtained by changing the number of nodes in the network. (a) Schedulability with loose period. (b) Schedulability with tight period.

In the next experiments, the E2E delay value for each packet equals the timeslot when the packet arrives at a sink node minus the timeslot when the packet was generated, so the average E2E delay is the average of all packets which are generated in the data slotframe. Figure 12 shows the average E2E delay per number of nodes in the network. To evaluate the average E2E delay, with experiments which are schedulable, we calculate the average E2E delay. By dynamically combining packets, our approach reduces the number of transmissions, so the average E2E delay in our approach is less than with other approaches in all experiments.

Figure 12. Average E2E delay obtained by changing the number of nodes in the network. (a) Average E2E delay with loose period. (b) Average E2E delay with tight period.

In the next experiments, Figure 13 shows total number of transmission per number of nodes in the network. Because, our approach optimizes packet length by dynamically combining packets, it optimizes the number of transmissions. When compared with PCLLF and CLLF, the number of transmissions is reduced by up to 34% with 20 nodes with tight period. Because of this, our approach reduces energy consumption.

Figure 13. Total number of transmissions by changing number of nodes in the network. (a) Total number of transmissions with loose period. (b) Total number of transmissions with tight period.

6. Conclusions

In this study, we propose a heuristic scheduling algorithm that's based on dynamically combining packets and dynamically prioritizing each packet's transmission dynamically based on its laxity (i.e., the remaining time before the end-to-end deadline) when each node has a fixed data length and constrained latency to apply to a TSCH network to increase schedulability, reduce the number of data transmissions, E2E delay, and ensure high reliability.

We propose a dynamic packet combination mechanism to reduce the number of packet transmissions. This mechanism is based on the period of nodes and the total data size in the path. Because in a TSCH network the maximum packet size is 127 bytes and the MAC header size is 21 bytes the maximum payload size is 106 bytes. The periods of all nodes can be different which is POW of two, so we choose nodes in the paths in which the period of a parent node is not less than that of the child nodes and packets from parents will be combined with other nodes to increase the schedulability.

According to the evaluation results, the proposed mechanisms show high schedulability and reduce the number of transmissions that is meant to reduce data traffic when compared with some approaches like PCLLF and CLLF. Therefore, we conclude that our proposal is highly suitable for monitoring and control systems in industrial wireless sensor networks, especially ones with low latency in real-time applications. In the future we will apply our centralized scheduling to real-devices and make a testbed in an industrial environment.

Author Contributions: Conceptualization, D.L.M. and M.K.K.; Methodology, D.L.M. and M.K.K.; Software, D.L.M.; Validation, M.K.K.; Formal Analysis, D.L.M.; Investigation, D.L.M.; Resources, M.K.K.; Data Curation, D.L.M.; Writing-Original Draft Preparation, D.L.M. and M.K.K.; Writing-Review & Editing, D.L.M.; Supervision, M.K.K.; Project Administration, M.K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work (Grants No. S2665996) was supported by project for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Ministry of SMEs and Startups in 2018.

Conflicts of Interest: The authors declare no conflict of interest.

Notations

$ET(T_{i,k}^j, S)$	Earliest start time for $T_{i,k}^j$ in time slot S
$LT(i)$	Latest timeslot node i must to transmit packet to parent of node i to guarantee schedulability.
$N_{cnf}^{avg}(T_{i,k}^j, S)$	Number of transmissions duplex conflicting with $T_{i,k}^j$ in time slot S
$PR(T_{i,k}^j, S)$	Priority value of $T_{i,k}^j$ in time slot S
$LST(T_{i,k}^j, S)$	Latest start time for $T_{i,k}^j$ in time slot S
$ECT(i)$	Earliest combined timeslot when all of packet in subtree node i can arrive to node i .
$TW(T_{i,k}^j, S)$	Time window with $T_{i,k}^j$ in time slot S

References

1. Darbandi, A.; Kim, M.K. Path collision-aware real-time link scheduling for TSCH wireless networks. *KSII Trans. Internet Inf. Syst.* **2019**, *13*, 4429–4445.
2. Nguyen, T.T.; Oh, H. SCSMA: A Smart CSMA/CA Using Blind Learning for Wireless Sensor Networks. *IEEE Trans. Ind. Electron.* **2019**, *1*. [\[CrossRef\]](#)
3. Gutierrez, J.A.; Callaway, E.H.; Barrett, R.L. The WirelessHART Physical Layer and Data Link Layer. In *Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15.4*; IEEE: Piscataway, NJ, USA, 2007; pp. 207–217, ISBN 9781118098882.
4. Ishii, Y. Exploiting Backbone Routing Redundancy in Industrial Wireless Systems. *IEEE Trans. Ind. Electron.* **2009**, *56*, 4288–4295. [\[CrossRef\]](#)
5. Nguyen, T.T.; Oh, H. A Receiver for Resource-Constrained Wireless Sensor Devices to Remove the Effect of Multipath Fading. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6009–6016. [\[CrossRef\]](#)
6. Nguyen, T.T. Design and Implementation of Smart Multi-Channel MAC Protocol for Industrial Wireless Sensor Networks. Ph.D. Thesis, University of Ulsan, Ulsan, Korea, February 2020.
7. Palattella, M.R.; Accettura, N.; Dohler, M.; Grieco, L.A.; Boggia, G. Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks. In Proceedings of the 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC), Sydney, Australia, 9–12 September 2012; pp. 327–332.
8. Aijaz, A.; Raza, U. DeAMON: A Decentralized Adaptive Multi-Hop Scheduling Protocol for 6TiSCH Wireless Networks. *IEEE Sens. J.* **2017**, *17*, 6825–6836. [\[CrossRef\]](#)
9. Kang, B.; Nguyen, P.K.H.; Zalyubovskiy, V.; Choo, H. A Distributed Delay-Efficient Data Aggregation Scheduling for Duty-Cycled WSNs. *IEEE Sens. J.* **2017**, *17*, 3422–3437. [\[CrossRef\]](#)
10. Nguyen, T.T.; Oh, H. Constructing an optimally balanced tree to maximize data throughput with multiple channels. *Wirel. Netw.* **2018**, *24*, 993–1005. [\[CrossRef\]](#)
11. Saifullah, A.; Xu, Y.; Lu, C.; Chen, Y. Real-time scheduling for WirelessHART networks. In Proceedings of the 2010 31st IEEE Real-Time Systems Symposium, San Diego, CA, USA, 30 November–3 December 2010; pp. 150–159.
12. Pan, C.; Zhang, H. A time efficient aggregation convergecast scheduling algorithm for wireless sensor networks. *Wirel. Netw.* **2016**, *22*, 2469–2483. [\[CrossRef\]](#)
13. Randhawa, S.; Jain, S. DAHDA: Dynamic Adaptive Hierarchical Data Aggregation for Clustered Wireless Sensor Networks. *Wirel. Pers. Commun.* **2017**, *97*, 6369–6399. [\[CrossRef\]](#)
14. Arghavani, M.; Esmaeili, M.; Esmaeili, M.; Mohseni, F.; Arghavani, A. Optimal energy aware clustering in circular wireless sensor networks. *Ad Hoc Netw.* **2017**, *65*, 91–98. [\[CrossRef\]](#)
15. Saginbekov, S.; Jhumka, A. Many-to-many data aggregation scheduling in wireless sensor networks with two sinks. *Comput. Netw.* **2017**, *123*, 184–199. [\[CrossRef\]](#)
16. Singh, V.K.; Verma, S.; Kumar, M. Privacy Preserving In-network Aggregation in Wireless Sensor Networks. In Proceedings of the FNC/MobiSPC, Montreal, QC, Canada, 15–18 August 2016; pp. 216–223.
17. Mehak; Khandnor, P. Structure and structure-free data aggregation protocols in wireless sensor networks-a review. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2017; pp. 2136–2140.
18. Li, C.; Bai, J.; Gu, J.; Yan, X.; Luo, Y. Clustering routing based on mixed integer programming for heterogeneous wireless sensor networks. *Ad Hoc Netw.* **2018**, *72*, 81–90. [\[CrossRef\]](#)