



Article Efficient Link Scheduling Based on Estimated Number of Packets in Queue on Industrial Wireless Sensor Networks

Myung-Kyun Kim



Citation: Kim, M.-K. Efficient Link Scheduling Based on Estimated Number of Packets in Queue on Industrial Wireless Sensor Networks. *Energies* 2021, *14*, 6370. https:// doi.org/10.3390/en14196370

Academic Editor: Alon Kuperman

Received: 25 August 2021 Accepted: 28 September 2021 Published: 5 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Daehak-ro 93, Nam-gu, Ulsan 44610, Korea; mkkim@ulsan.ac.kr

Abstract: The links of low power wireless sensor networks are error prone and the transmission on a wireless link is determined probabilistically by the packet reception rate (PRR) of the link. On the other hand, there is a very strict requirement in the end-to-end reliability and delay of sensor data in industrial wireless sensor networks (IWSNs). The existing approaches to provide the end-to-end reliability in IWSNs is retransmitting the packet when failure occurs. These approaches transmit a packet multiple times in successive time slots to provide the required reliability. These approaches, however, can increase the average delay of packets and the number of packets buffered in a queue. This paper proposes a new scheme to estimate the probabilistic amount of packets, called queue level (QL), in the buffer of each node based on the PRRs of the wireless links. This paper also proposes a QL-based centralized scheduling algorithm to assign time slots efficiently in TDMA-based IWSNs. The proposed scheduling algorithm gives higher priority to the nodes with higher QL. By assigning time slots first to the node with the highest QL, we can reduce the average end-to-end delay of packets and reduce the amount of buffered packets in the queue while satisfying the required end-to-end reliability. The performance of the proposed scheduling algorithm have been evaluated through a simulation using the Cooja simulator and compared with the existing approach. In the simulation on an sample network with the target end-to-end reliability of 99%, all of the flows were shown to guarantee the target reliability in both algorithms: on average, 99.76% in the proposed algorithm and 99.85% in the existing approach. On the other hand, the proposed algorithm showed much better performance than the existing approach in terms of the average end-to-end delay of packets (about 47% less) and the number of maximally buffered packets in the queue of each node (maximally, more than 90% less).

Keywords: IEEE 802.15.4e; industrial wireless sensor networks (IWSNs); link scheduling; queue estimation; reliability in IWSNs

1. Introduction

The wireless sensor networks (WSNs) are receiving more and more attention for deployment in the industrial environment for monitoring and control applications [1]. Compared with wired networks, the challenge of wireless networks is that wireless signals are broadcasted through the air, which are vulnerable to interference with other signals and can be affected due to multipath fading [1]. Recently, to promote the development of IWSN applications, several industrial wireless sensor network standards such as WirelessHART [2], ISA 100.11a [3], and IEEE 802.15.4e TSCH [4] have been proposed. To satisfy the reliable and real-time requirement of industrial applications, these networks use the time-division multiple access (TDMA) MAC protocol to avoid collisions effectively among multiple simultaneous transmissions and channel hopping techniques to reduce packet loss due to multipath fading. Those IWSN standards define the TDMA-based transmission structure called slot frame, frequency hopping method, and protocols to provide IPv6 connectivity to IWSNs. However, the scheduling algorithm to assign time-slots to the wireless links is not defined.

There have been many studies in the scheduling algorithms on IWSNs, which are classified into centralized and decentralized scheduling schemes [5]. Decentralized schemes rely on nodes with partial knowledge of their neighborhood and traffic information. Such knowledge is utilized for distributed decision making processes. Decentralized schemes can better react to the dynamics of the network due to link status changes or the nodes' mobility [6]. However, centralized schemes are more suitable for IWSNs that have strict reliability and real-time requirements and utilize a centralized architecture and network management [7,8].

In industrial environments, sensor nodes are located in a harsh condition with many obstacles, high humidity levels, vibrations, dirt and dust, or other conditions that challenge performance. These harsh environmental conditions lower the performance of the wireless links and makes reliable message delivery even more difficult. Mostly, sensor nodes have a scarce resource in terms of processing power, memory, and energy. Constrained by the limited physical size, sensor nodes have limited battery energy supply and their memories are limited and have restricted computational capabilities [7].

To transmit sensor data reliably in wireless sensor networks with error prone links, retransmission after packet loss is the most widely used method [9]. Yan M, et al. [10] proposed two scheduling methods using a hypergraph-based approach to maximize the end-to-end reliability of sensor data with a delay constraint in IWSN: dedicated and shared scheduling methods. Their algorithm consists of two phases: the schedule construction phase and schedule improvement phase. In the construction phase, one time slot is allocated to a link group that consists of links that can transmit simultaneously on the path from the source to destination. Then, in the improvement phase, the retransmission time slots are added to link groups as long as they satisfy the delay constraint. In the shared time slot scheduling method, retransmission time slots are shared among multiple packets to reduce the number of slots while satisfying the required reliability. Dobslaw F. et al. [11] proposed a method called SchdEx to extend a scheduling algorithm to guarantee the end-toend reliability, which determines the number of time slots to each link based on packet loss rates of the entire network. The SchedEx method decomposes the end-to-end reliability into node-to-node reliability to make the calculation easy. Hashimoto M. et al. [12] proposed a flow-based assignment method to reduce the required number of time-slots by prioritizing the flows sharing links and assigning shared time-slots efficiently among the flows. The above methods in [10-12] focused on the reliability of the sensor data transmission of flows and reducing the number of time-slots to be assigned to guarantee the required end-to-end reliability. These approaches transmit a packet multiple times in successive time slots to provide the required reliability. These approaches, however, can increase the average end-to-end delay of packets and the number of packets buffered in the queue of the node where multiple flows are multiplexed.

This paper proposes a new scheme to estimate the probabilistic amount of packets, called queue level (QL), in the buffer of each node based on the packet reception rates (PRRs) of the wireless links. A node with higher QL means that it can have more packets probabilistically in the queue than the node with lower QL. This paper also proposes a QL-based centralized scheduling algorithm to assign time slots efficiently in TDMA-based IWSNs. The proposed algorithm first calculates the number of time slots that need to be assigned to each node based on the target end-to-end reliability and PRR values of the wireless links. To determine (time-slot, channel) for the time slots assigned to the nodes, the proposed scheduling algorithm uses a prioritization scheme, which gives higher priority to the node with higher QL. By assigning time slots first to the node with the highest QL, we can reduce the average end-to-end delay of packets and reduce the number of packets buffered in the queue while satisfying the required end-to-end reliability.

We evaluated the performance of the proposed scheduling algorithm through a simulation using a Cooja simulator in Contiki [13] and compared it with the flow-based prioritization approach in [12]. In the simulation on an example network in [12] with the target end-to-end reliability of 99%, each node transmitted one packet for each slot frame during 5000 slot frames following the schedules of the proposed algorithm and the flow-based approach, respectively. The simulation results showed that all of the flows guaranteed the target reliability in both of the algorithms: on average, 99.76% in the proposed algorithm and 99.85% in the existing approach. On the other hand, the proposed algorithm showed much better performance than the existing approach in terms of the average end-to-end delay and the number of buffered packets in the queue. The average end-to-end delay of packets of the flow-based approach was 184.4 ms and that of the proposed algorithm was 87.8 ms, which was more than 50 less than the flow-based approach. In the case of packet buffering, the average of maximally buffered packets of the proposed algorithm was lower than or equal to that of the flow-based approach in all nodes. In the case of the flow-based approach, the buffering requirement increases linearly, depending on the number of flows sharing a link. On the other hand, the number of buffered packets in all nodes were maintained at a fixed value (about 2 in the simulation).

The rest of this paper is organized as follows. In Section 2, we describe the network model and the notation used in the paper. In Section 3, we describe the method to estimate the probabilistic amount of packets in each node and the QL-based scheduling algorithm proposed in this paper. Section 4 describes the performance evaluation of the proposed algorithm based on the simulation using Cooja and Section 5 concludes the paper.

2. Network Model

In this paper, we considered a WSN consisting of a sink node (or a gateway) and sensor nodes with a half-duplex radio interface. A WSN is denoted by a directed graph G = (V, E) where V is a set of sensor nodes including the sink, and E is a set of the links between nodes in the network, respectively. Wireless links between two nodes i and j consist of two directed edges: edge from node i to j denoted by <i,j> and edge from j to i, <j,i>. Link quality estimation in WSNs is a fundamental building block in several mechanisms such as routing and topology control, but is a challenging problem due to the lossy and dynamic behavior of the wireless links. So far, a large number of link quality estimators have been proposed [13]. We used the PRR (Packet Reception Rate) estimator in this paper, which is a receiver side estimator that is simple to measure and has been widely used in WSNs. The PRR of link $\langle i,j \rangle$ is denoted by R_i in this paper. In WSNs, each sensor node generates a packet periodically and needs to transmit through a path from the node to the sink. The packet transmitted by source node k is denoted by p_k , and the hop count from the node k to the sink by h_k . We assumed that the routing paths are given by a directed convergecast tree constructed by using routing protocols such as RPL [14] or other reliable routing protocols [7]. A single path tree is assumed in this paper, thus, each node k has a parent node P_k to transmit its data toward the sink.

In industrial WSNs, there is a strict requirement in the end-to-end reliability of the sensor data transmitted by the nodes depending on the application. In order to ensure the reliability requirement, we assumed a TDMA-based MAC with channel hopping such as IEEE 802.15.4e TSCH [4] as a MAC protocol. In a TSCH network, there are multiple channels up to 16 and a slot frame with a fixed number of time slots. The number of available channels is denoted by N_{ch} and the number of time slots in a slot frame by N_{sf} . Thus, the scheduling problem is to decide which node transmits its packet, on which channel offset, and which time slot in a slot frame [8]. In TSCH networks, a frame transmission and an acknowledgment (ACK) of indicating successful reception of the frame is finished within a time slot. Two different types of time slots are defined in TSCH: dedicated and shared time slots. We assumed that the time slots are dedicated to guarantee the reliability of frame transmission in a time slot. As in TASA [7], we defined a set of duplex-conflict nodes of node k, DC_k , as a set of nodes that cannot transmit in the same slot due to conflict. The nodes in DC_k consists of the parent node P_k , the children of node k, and the children of P_k . If a node is not in DC_k , it can transmit its frame in the same time slot with node k using different channels to each other. In the example graph in Figure 1, DC_1 is {2, 4} and DC_4 is {1, 2}, thus, node 1 and node 3 (also node 3 and 4) can transmit in the same time slot using different channels.



Figure 1. An example network graph.

3. Proposed Scheduling Algorithm

We assumed in this paper that the convergecast tree for each node to transmit its sensor date to the sink is given and each node has a fixed number of packets to transmit in each slot frame. The proposed scheduling algorithm is a centralized algorithm and after the convergecast tree is constructed, the scheduling server receives the following information for scheduling from each node k.

- *T_k*: the number of packets to be transmitted by the node in each slot frame;
- *P_k*: parent node of node *k*; and
- R_k : PRR value of link $\langle k, P_k \rangle$.

3.1. End-to-End Reliability in WSNs

The most widely used method to provide the end-to-end reliability in IWSNs is retransmitting packets when packet loss happens. There have been many studies to determine the required number of time slots to satisfy a given end-to-end reliability requirement and schedule the time slots [10–12]. Two types of time slots are used to provide the reliability: dedicated time slot and shared time slot. In a dedicated time slot scheme, a fixed number of time slots are dedicated to transmit a packet, which needs a larger number of time slots compared with the shared time slot scheme. In the shared time slot scheme, we can reduce the number of time slots by taking the advantage of statistical multiplexing of independently transmitted packets when multiple packets share a link. We considered the shared time slot scheme in this paper.

When the target end-to-end PRR (e2ePRR) is given to a IWSN, the number of time slots that needs to be assigned to a packet transmitted by a node depends on e2ePRR, the number of links in the routing path, and the PRRs of the links in the path [10–12]. To determine the number of time slots for a packet transmitted by a node with k hops away from the sink, we first determine the required per-hop link PRR (PH_k) to satisfy the given e2ePRR as follows:

$$PH_k = e2ePRR^{1/k} \tag{1}$$

When multiple packets are sharing a link, we took the average of the per-hop link PRRs of the packets as the required per-hop link PRR (PA_k). Let S_k be a set of (p_i , h_i) sharing the transmission link of node k, $\langle k, P_k \rangle$. Then, PA_k of the link $\langle k, P_k \rangle$ is calculated as:

$$PA_k = \left(\sum_{(p_i, h_i) \in S_k} PH_{h_i}\right) / |S_k| \tag{2}$$

The number of time slots to be assigned to node *k* transmitting its packet through link $\langle k, P_k \rangle$, TS_k , is calculated as:

$$TS_{k} = \min_{n} \left\{ PA_{k} \le 1 - \sum_{j=0}^{|S_{k}|-1} \binom{n-|S_{k}|+j}{j} R_{k}^{j} (1-R_{k})^{n-|S_{k}|+1} \right\}$$
(3)

In the example network of Figure 1, let us assume that all nodes have one packet to transmit in each slot frame and e2ePRR = 0.99 for all packets. Then,

- $PH_1 = 0.99, PH_2 = 0.99499, PH_3 = 0.99666;$
- $S_1 = \{(p_1, 1), (p_2, 2), (p_4, 2), (p_3, 3)\};$ and

Thus, $TS_1 = 7$ by Equation (3), and the number of time slots that needs to be assigned to node 1 is 7 to satisfy the target end-to-end PRR. The PA and TS value for all nodes in Figure 1 is shown in Figure 2a.

| node | PA | TS | | | | | |
|------|-------------|------------|--|--|--|--|--|
| 1 | 0.99416 | 7 | | | | | |
| 2 | 0.99583 | 6 | | | | | |
| 3 | 0.99666 | 6 | | | | | |
| 4 | 0.99499 | 3 | | | | | |
| (a) | | | | | | | |
| node | QL (before) | QL (after) | | | | | |
| 3 | 100 | 30 | | | | | |
| 2 | 0 | 70 | | | | | |
| (b) | | | | | | | |

Figure 2. PA and TS values for each node in the example network of Figure 1 (**a**), and QL change after transmission on link <3,2> (**b**).

3.2. Estimating Amount of Packets in Queue

The transmission on a wireless link is determined probabilistically based on the PRR of the link. When a packet is transmitted on link $\langle k, P_k \rangle$ with PRR R_k , it will be transmitted to node P_k with probability R_k and stay in node k with probability $(1-R_k)$. We expressed the amount of packets in the queue of a node in percentage, called QL (queue level), by considering the probabilistic transmission of the wireless link. For example, when node 3 has one packet to transmit in the queue and there is no packet in node 2, as shown in Figure 2b, $QL_2 = 0$ and $QL_3 = 100$. If node 3 transmits a packet to node 2 in a time slot, the QLs of nodes 2 and 3 change so that $QL_2 = 70$ and $QL_3 = 30$ after the time slot because the PRR of link $\langle 3, 2 \rangle$ is 70%. We can estimate the amount of packets in the queue of a node by its QL. Higher QL means a higher number of packets in the queue.

The proposed scheduling algorithm utilizes the QL information to determine a node to transmit in a time slot. We assumed that all nodes had a fixed number of packets at the beginning of each slot frame. After the convergecast tree is constructed, the scheduling server receives $[T_k, P_k, R_k]$ from each sensor node k. After that, the scheduling server calculates PA_k and TS_k for each node k based on Equations (1)–(3). The proposed scheduling algorithm utilizes multiple channels and assigns multiple nodes to a time slot if they are not in duplex-conflicting relationship [8]. For scheduling, the server sets the following information to each node:

- QL_k (QL of node *k*) is initialized to $(100 \times T_k)$;
- MQL_k (Minimum QL of node k) is set to $100 \times (1-PA_k)$, which specifies the minimum QL of node k for a time slot to be assigned; and
- DC_k is a set of nodes that are in duplex-conflicting relationship with node k [7].

The proposed scheduling algorithm starts from the first time slot to decide the nodes to transmit among the nodes that have remaining time slots and their queue level is above their minimum level. The proposed algorithm uses a QL-based prioritization scheme to assign the current time slot to the nodes, which gives higher priority to the nodes with higher QL. The scheduling algorithm performs the following steps to determine the nodes that can transmit their packets in the current time slot.

- 1. Determine a set of nodes that need to be scheduled, *N*, in the current time slot *t* (initially 0);
- 2. Prioritize the nodes in *N* based on QL_k , TS_k , and tree level of node *k*;

- 3. Choose the node with the highest priority, *n*, and assigns channel 0 of time slot *t* to node *n*;
- 4. Delete *n* and the nodes in DC_n from *N*;
- 5. Choose the next highest priority node, *n*, in *N* and assigns the next available channel to node *n*;
- 6. Repeat steps 4 and 5 until *N* becomes empty or there is no available channel; and
- 7. Adjust the QLs of nodes that are transmitting or receiving in time slot t considering the PRR values of the links and go to 1 for the next time slot (t + 1).

Algorithm 1 shows the details of the proposed scheduling algorithm. After initializing the QL and MQL values of each node, we obtain a set of nodes, N, to which time slots can be assigned in step 3. All the nodes in N have to satisfy that $TS_k \ge 1$ and $QL_k \ge MQL_k$ for each node k in N. In the main loop of (4)~(18), the nodes in N are prioritized in the PRIORITIZE(N) function. We prioritized the nodes based on the order of the highest QL, the highest TS, and the highest tree level. In the loop of (7)~(14), we selected the nodes that can transmit in the current time slot ts at the same time and assign to the available channels. At first, we assigned (ts,0) (channel 0 of time slot ts) to N(0), which is the highest priority node in N. After that, we deleted N(0) and the nodes that are duplex-conflicting with N(0) from N. Then, we assigned the next available channel of the current time slot to node N(0), and deleted N(0) and the nodes that are duplex-conflicting with the selected node again. We repeated this step while the channels were available and N was not empty. After selecting all the nodes that can transmit in ts at the same time, we updated the TS and QL values of the nodes in *M* (the selected links in ts) in the ADJUST_TS_QL(*M*) function. We increased the time slot ts and set the channel number ch to 0, and updated N to a set of nodes to which channels can be assigned in the next time slot (ts + 1). We repeated the main loop of steps (4)~(18) while N was not empty. After the main loop, we checked the QL of each node and assigned additional time slots to the nodes if their QLs were greater than or equal to MQL in the ASSIGN_ADDITIONAL_TS() function.

After the inner loop (step (7)~step (14)) is finished, M contains the links to which channels are assigned to transmit in the current time slot ts at the same time. The AD-JUST_TS_QL(M) function updates the TS and QL values of the sender and receiver nodes of the links in M to determine the channel assignment in the next time slot. The remaining number of time slots (TS values) are decreased by 1. The QL values are updated based on the number of packets to transmit in time slot ts and the PRR value of the link. The number of packets (tx_amount) to transmit is 100 (one packet) if QL_k is greater than 100, otherwise, tx_amount is QL_k . Thus, for each link <k, P_k > in M, we updated the QLs of node k and node P_k based on the PRR value, R_k of the link in steps (7) and (8) of the ADJUST_TS_QL(M) function.

To illustrate the operation of the proposed scheduling algorithm with an example, we performed the algorithm in a 2-channel network based on the example network in Figure 1. We assumed all nodes had one packet to transmit in each slot frame. Based on the PA value of each node in Figure 2, the MQL (minimum QL) of each node was set as $MQL_k = 100 \times$ $(1 - PA_k)$ for node k. We assigned time slots only to the nodes that satisfied $QL_k \geq MQL_k$ and $TS_k \ge 1$. Figure 3 shows the TS, QL, and MQL values of the nodes at each time slot and the (time-slot, channel) assignment result. In each step, the red color denotes the node with the highest priority and the blue color the non-duplex conflicting node. The highest node is assigned to channel 0 and the non-duplex conflicting node to channel 1 in the current time slot. During the assignment step, the QL of a node can become less than its MQL even though its TS is greater than 0. In this case, the remaining time slots are not assigned, which can reduce the number of assigned time slots, but can affect the end-to-end delivery ratio slightly. In Figure 3, node 3 at time slot 8 and node 2 at time slot 15 showed these cases. At time slot 16, there was no node that satisfied $QL_k \ge MQL_k$ and $TS_k \ge 1$, thus, the main loop of the scheduling algorithm was completed. After the main loop was completed at time slot 16, the ASSIGN_ADDITIONAL_TS() function checks the QL of each node and assigns one additional time slot to node 1 because $QL_1 > MQL_1$.

Algorithm 1: QL-Based Scheduling Algorithm

// NCH: the number of available channels // T_k : the number of packets that node k transmits in each slot frame // TS_k : the number of time slots that needs to be assigned to node k // PA_k : the per-hop link PRR of link <k, P_k > // DC_k : a set of nodes that are duplex-conflicting with node k (1) Initialization: $QL_k = 100 \times T_k$; $MQL_k = 100 \times (1-PA_k)$; (2) ts = 0; ch = 0;

(3) $N \leftarrow a \text{ set of nodes which satisfies } TS_k \ge 1 \text{ and } QL_k \ge MQL_k;$

- (4) while (N is not empty) {
- (5) *PRIORITIZE(N); // prioritize the nodes in N*
- (6) $M \leftarrow \varphi$;
- (7) while $(ch < N_{ch})$ {
- (8) $n \leftarrow N(0);$
- (9) $assign (ts, ch) to link < n, P_n >;$
- $(10) \qquad M \leftarrow M \cup \{< n, P_n >\};$
- (11) $DLETE_DC(N, n)$; // delete n and nodes in DC_n from N
- (12) *if* (*N is empty*) *break;*
- $(13) ch \leftarrow ch + 1;$
- (14)

}

- (15) ADJUST_TS_QL(M); // adjust TS, QL of nodes in M
- (16) $ts \leftarrow ts + 1; ch = 0;$
- (17) $N \leftarrow a \text{ set of nodes which satisfies } TS_k \ge 1 \text{ and } QL_k \ge MQL_k;$
- (18) }
- (19) ASSIGN_ADDITIONAL_TS(); // check the QL of nodes and assign additional time slots when $QL_k \ge MQL_k$

Function: PRIORITIZE(N)

- // N: a set of nodes to which a slot can be assigned
- (1) Sort the nodes in N in descending order of QL values;
- (2) In the case of a tie, sort the nodes in N in descending order of TS values;
- (3) In the case of a tie, sort the nodes in N in descending order of tree level;

Function: DLETE_DC(N, n)

// N: a set of nodes to which a slot can be assigned
// n: selected node
// DC_n: a set of duplex-conflicting nodes with node n
(1) Delete n and all nodes in DC_n;

Function: ADJUST_TS_QL(M)

(8) $QL_{P_k} = QL_{P_k} + (tx_amount \times R_k);$

```
(9)}
```

| | tim slo | e- ot | noc MQ | le: L: | 4 (0.5 | 01) | | 3 (0.3 | 34) | | 2 (0.4 | 17) | (| 1 (0.58 | 34) | | |
|----|------------|----------|-----------|-----------|-----------|-----|----|-----------|---------|-----|-----------|------|-----|------------|-------|-------------|----|
| | | Г | TS | | | | 3 | | | 6 | | | 6 | | | 7 | |
| | 1 | t | QL | | | 10 | 00 | | 1(| 00 | | 1 | 00 | | 1 | 00 | |
| | _ | T | TS | | | | 3 | | | 5 | | | 6 | | | 6 | |
| | 2 | Ī | QL | | | 10 | 0 | | 1 | 30 | | 1 | 70 | | | 10 | |
| | 2 | Γ | TS | | | | 3 | | | 5 | | | 5 | | | 6 | |
| | 3 | _[| QL | | | 10 | 00 | | 1 | 30 | | | 90 | | | 90 | |
| | | -[| TS | | | | 2 | | | 4 | | | 5 | | | 6 | |
| | 4 | · [| QL | | | 1 | 10 | | | 9 | | 1 | 11 | | 1 | 80 | |
| | | . [| TS | | | | 2 | | | 3 | | | 5 | | | 5 | |
| | 5 | ' [| QL | | | 1 | .0 | | 2 | .7 | | 117 | 7.3 | | | 90 | |
| | 6 | | TS | | | | 2 | | | 3 | | | 4 | | | 5 | |
| | 0 | ΄[| QL | | | 1 | .0 | | 2 | .7 | | 37 | 7.3 | | 1 | L 70 | |
| | | -[| TS | | | | 2 | | | 2 | | | 4 | | | 4 | |
| | 7 | Ε | QL | | | 1 | .0 | | 0.8 | 31 | | 39. | 19 | | | 80 | |
| | ~ | Ι | TS | | | | 2 | | | 1 | | | 4 | | | 3 | |
| | 8 | | QL | | | 1 | .0 | | 0.24 | 13 | | 39.7 | 57 | | | 8 | |
| | 0 | Ι | TS | | | | 2 | | | 1 | | | 3 | | | 3 | |
| | 9 | | QL | | | 1 | .0 | | 0.24 | 13 | | 7.95 | 14 | | 39.80 |)56 | |
| | 10 | , [| TS | | | | 2 | | | 1 | | | 3 | | | 2 | |
| | 10 | '[| QL | | | 1 | 0 | | 0.24 | 13 | | 7.95 | 14 | | 3.980 | 056 | |
| | 11 | | TS | | | | 1 | | | 1 | | | 3 | | | 2 | |
| | | · [| QL | | | | 1 | | 0.24 | 13 | | 7.95 | 14 | 1 | 2.980 |)56 | |
| | 12 | , [| TS | | | | 1 | | | 1 | | | 3 | | | 1 | |
| | | - [| QL | | | | 1 | | 0.24 | 13 | | 7.95 | 14 | 1 | .2980 | 056 | |
| | 13 | . [| TS | | | | 1 | | | 1 | | | 2 | | | 1 | |
| | | ۶Ľ | QL | | | | 1 | | 0.24 | 13 | 1 | .590 | 28 | 7. | .6591 | .76 | |
| | 14 | , [| TS | | | | 1 | | | 1 | | | 2 | | | 0 | |
| | | * [| QL | | | | 1 | | 0.24 | 13 | 1 | .590 | 28 | 0.7 | 6591 | .76 | |
| | 15 | ; [| TS | | | | 1 | | | 1 | | | 1 | | | 0 | |
| | | Ĺ | QL | | | | 1 | | 0.24 | 13 | 0. | 3180 | 56 | 2.0 | 3814 | 16 | |
| | 16 | 5 L | TS | | | | 0 | | | 1 | | | 1 | | 0(- | +1) | |
| | | Ĺ | QL | | | 0 | .1 | | 0.24 | 13 | 0. | 3180 | 56 | 2.9 | 3814 | 16 | |
| | 17 | 7 L | TS | | | | 0 | | | 1 | | | 1 | | | 0 | |
| | | | QL | | | 0 | .1 | | 0.24 | 13 | 0. | 3180 | 56 | 0.29 | 93814 | 116 | |
| | | | | | | | | | (| a) | | | | | | | |
| ch | юΓ | 1 | 2 | 4 | 1 | 2 | 1 | 1 | 2 | 1 | 4 | 1 | 2 | 1 | 2 | 4 | 1 |
| ch | | 3 | | 3 | 3 | | 3 | 3 | | 3 | | | | | | | |
| | L | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| | | | | | | | | | · · · · | - / | | | | | | | |

Figure 3. Scheduling steps (**a**) and the assignment result (**b**) for the sample graph in Figure 1 with 2-channels.

4. Performance Evaluation

We evaluated the performance of the proposed scheduling algorithm and compared it with the approach in [12] using the Cooja simulator on the Contiki platform [15]. The existing approaches in [10–12] for guaranteeing the end-to-end reliability used two types of time slots: dedicated and shared time slots. The shared time slot scheme can reduce the number of time slots by utilizing the advantage of statistical multiplexing of independently transmitted packets. We also used the shared time slot approach in this paper. After determining the number of time slots for each node, the scheduling algorithm decides the time-slot number and channel number for each time slot assigned to the nodes. The scheduling of the existing approaches are similar, so we only compared it with the approach in [12] (we call this a flow-based approach).

For fair comparison, we used the same example in [12], which is shown in Figure 4a. The scheduling result of the flow-based approach for the example on a 3-channel network is shown in Figure 4b. We assumed that all sensor nodes had one packet to transmit at the beginning of each slot frame and the target end-to-end reliability of packets was 99%. Figure 4c shows the MQL and TS values for scheduling in the proposed scheduling

algorithm and the scheduling result of the proposed algorithm is shown in Figure 4d. The number of assigned time slots of the proposed algorithm is 23, which is one time slot less than that of the flow-based approach.



Figure 4. Scheduling results of the proposed algorithm and flow-based algorithm on the example network with 3-channels.

Based on the scheduling results in Figure 4, we ran the simulation using the Cooja simulator and analyzed the performance of the proposed algorithm in terms of the end-toend reliability, the average end-to-end delay of packets, and the number of packets queued in each node. The slot frame length was 50 and each time slot was 10 ms. We set the target end-to-end reliability to 99%. During the simulation, each sensor node transmitted its data 5000 times, one for each slot frame using the time-slot and channel assigned to the node. The flow-based approach uses prioritized transmission. Thus, if there are multiple packets in the queue at its assigned time slot, it transmits the highest priority packet based on the defined priority of flows, as shown in Figure 4b [12]. In the case of the proposed algorithm, each node transmits packets in the queue in FIFO order.

Figure 5 shows the end-to-end reliability of each node in the proposed algorithm and the flow-based approach. In both of the algorithms, every node satisfies the target end-to-end reliability of 99%. The average end-to-end reliability of the nodes in the proposed algorithm was 99.76%, which was a little less than that of the flow-based approach (99.85%).

Figure 6 shows the average end-to-end delay of packets transmitted by each node in both of the approaches. We assumed that all packets to be transmitted in each slot frame were ready at the beginning of the slot frame. We measured the end-to-end delay of a packet as the time between the slot frame starting time and the packet reception time at the sink node. As shown in Figure 6, the average end-to-end delay of packets of the proposed algorithm was lower than that of the flow-based approach in all nodes, except node 6. In the flow-based approach, node 6 had the shortest end-to-end delay because it had the highest priority. After the packets of all flows were buffered in node 1, the packet of node 6 was transmitted from the first time slot assigned to node 1 (time-slot 14). The average end-to-end delay of packets transmitted by all nodes in the flow-based approach was 184.4 ms, and that of the proposed algorithm was 98.1 ms, which was about 47% less than

the flow-based approach. In the case of node 1, the average end-to-end delay of packets of the proposed algorithm was 13.36 ms, which was only about 7% of that of the flow-based approach (206.58 ms).



Figure 5. End-to-end reliability of the proposed algorithm and the flow-based approach.



Figure 6. Average end-to-end delay of packets in the proposed algorithm and the flow-based approach.

Figure 7 shows the average of maximally buffered packets in the queue of each node in both of the approaches. As we can see in the figure, in the case of the flow-based approach, as the number of multiplexed packets increased, the number of buffered packets also increased. The average values of maximally buffered packets in nodes 1, 2, and 3 were 6.994, 1.998, and 2.998 in the flow-based approach. In the case of the proposed algorithm, the average of the maximally buffered packets in nodes 1, 2, and 3 were 2, 1.992, and 2.027, respectively. From this simulation, we can see that the proposed algorithm keeps the queue level (the number of buffered packets) of the nodes to about a fixed value, which was 2 in this example.



Figure 7. Average number of maximally buffered packets of each node in the proposed algorithm and the flow-based approach.

5. Conclusions

In this paper, we proposed a novel method to estimate the number of packets, called the queue level (QL), in the queue of the nodes in TDMA-based IWSNs. We can estimate the QL of each node based on the probabilistic transmission of wireless links. We also proposed a centralized scheduling algorithm on IWSNs by utilizing the QL information of the nodes. After determining the number of time slots for each node to satisfy the target end-to-end reliability, the proposed scheduling algorithm decides the time-slot and channel to each of the assigned time slot. The proposed scheduling algorithm assigns time slots based on the priority, which gives the highest priority to the node with the highest QL. By assigning time slots based on the QL-based priority scheme, we can transmit packets to the sink node fast and reduce the number of packets buffered in each node while satisfying the end-to-end reliability requirement of IWSNs. Based on the simulation using Cooja, we have shown that the proposed algorithm guarantees the target end-to-end reliability required in IWSNs. The simulation results also showed that the proposed algorithm provides a much lower end-to-end delay of packets and lower queue level (number of buffered packets in the queue) compared with the flow-based approach [12].

Funding: This work was supported by the 2020 Research Fund of University of Ulsan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

- Gungor, V.C.; Hancke, G. Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. *IEEE Trans. Ind. Electron.* 2009, 56, 4258–4265. [CrossRef]
- Song, J.; Han, S.; Mok, A.; Chen, D.; Lucas, M.; Nixon, M.; Pratt, W. WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. In Proceedings of the Real-Time and Embedded Technology and Applications Symposium (RTAS'08), St. Louis, MO, USA, 22–24 April 2008; pp. 377–386. [CrossRef]
- 3. ISA. Wireless Systems for Industrial Automation: Process Control and Related Applications. *Int. Soc. Autom.* 2011, 1–792, ANSI/ISA-100.11a-2011.
- Gutierrez, J.A.; Callaway, E.H.; Barrett, R.L. IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Standard 802.15.4–2015 (Revision of IEEE Std 802.15.4–2011); IEEE Standards Association: Piscataway, NJ, USA, 2016. [CrossRef]
- Sgora, A.; Vergados, D.D. A Survey of TDMA Scheduling Schemes in Wireless Multihop Networks. ACM Comput. Surv. 2015, 47, 1–39. [CrossRef]
- Saifullah, A.; Xu, Y.; Lu, C.; Chen, Y. Distributed Channel Allocation Protocols for Wireless Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.* 2013, 25, 2264–2274. [CrossRef]
- Nobre, M.; Silva, I.; Guedes, L.A. Routing and Scheduling Algorithms for WirelessHARTNetworks: A Survey. Sensors 2015, 15, 9703–9740. [CrossRef] [PubMed]

- 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 15.4e networks. In Proceedings of the IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications. (PIMRC'12), Sydney, Australia, 9–12 September 2012; pp. 327–332. [CrossRef]
- 9. Mahmood, M.A.; Seah, W.K.; Welch, I. Reliability in wireless sensor networks: A survey and challenges ahead. *Comput. Netw.* **2015**, *79*, 166–187. [CrossRef]
- Yan, M.; Lam, K.-Y.; Han, S.; Chan, E.; Chen, Q.; Fan, P.; Chen, D.; Nixon, M. Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints. *Inf. Sci.* 2014, 278, 34–55. [CrossRef]
- 11. Dobslaw, F.; Zhang, T.; Gidlund, M. End-to-End Reliability-Aware Scheduling for Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2014**, 12, 758–767. [CrossRef]
- Hashimoto, M.; Wakamiya, N.; Murata, M.; Kawamoto, Y.; Fukui, K. End-to-end reliability- and delay-aware scheduling with slot sharing for wireless sensor networks. In Proceedings of the 8th International Conference on Communication Systems and Networks (COMSNETS'16), Bangalore, India, 12–16 September 2016; pp. 1–8. [CrossRef]
- 13. Baccour, N.; Koubaa, A.; Mottola, L.; Zúñiga, M.A.; Youssef, H.; Boano, C.A.; Alves, M. Radio link quality estimation in wireless sensor networks. *ACM Trans. Sens. Netw.* **2012**, *8*, 1–33. [CrossRef]
- Winter, T.; Thubert, P.; Brandt, A.; Hui, J.; Kelsey, R.; Levis, P.; Pister, K.; Struik, R.; Vasseur, J.; Alexander, R. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. IETF RFC 6550. 2012. Available online: https://tools.ietf.org/html/rfc6550 (accessed on 15 September 2020).
- 15. Contiki-NG: The OS for Next Generation IoT Devices. Available online: http://www.contiki-ng.org/ (accessed on 15 March 2020).