Article

Link Quality Estimation from Burstiness Distribution Metric in Industrial Wireless Sensor Networks

Ngoc Huy Nguyen and Myung Kyun Kim *

IT Convergence Department, University of Ulsan, Daehak-ro 93, Nam-gu Ulsan 44610, Korea;
huynguyen.bau1412@gmail.com
* Correspondence: mkkim@ulsan.ac.kr

Received: 22 October 2020; Accepted: 3 December 2020; Published: 4 December 2020

Abstract: Although mature industrial wireless sensor network applications increasingly require low-power operations, deterministic communications, and end-to-end reliability, it is very difficult to achieve these goals because of link burstiness and interference. In this paper, we propose a novel link quality estimation mechanism named the burstiness distribution metric, which uses the distribution of burstiness in the links to deal with variations in wireless link quality. First, we estimated the quality of the link at the receiver node by counting the number of consecutive packets lost in each link. Based on that, we created a burstiness distribution list and estimated the number of transmissions. Our simulation in the Cooja simulator from Contiki-NG showed that our proposal can be used in scheduling as an input metric to calculate the number of transmissions in order to achieve a reliability target in industrial wireless sensor networks.

Keywords: IEEE 802.15.4e; industrial wireless sensor networks; link metric measurements; link burstiness; reliability

1. Introduction

Currently, industrial wireless sensor networks deploy many applications in fields such as environment monitoring, smart factories, healthcare, radiation checks, leakage detection, and process control. In these types of applications, many problems have to be considered, such as solving delays in data delivery, prolonging network lifetime [1,2], safety, and security [3–5]. However, the reliability of data transmission is the most important factor in designing a protocol for industrial wireless sensor networks. Many studies have tried to increase the reliability of data transmission by improving the physical layer [6], the MAC protocol [7], and the routing protocol [8–11]. However, the previous studies did not consider the effect of link quality; they assumed the link quality is always stable. In real applications, the link quality is not stable, and it can vary depending on the environment and time. Therefore, for industrial wireless sensor networks, it is necessary to study link quality measurements that have low computational complexity and that require low-energy consumption.

The link quality estimator can be divided into two groups: hardware-based estimator and software-based estimator. Some studies that used hardware-based metrics such as Noda et al. [12] proposed a new channel measurement based on the channel availability over time. This significantly quantified the utilization of the spectrum. However, in the presence of multipath fading, it may have problems estimating the channel quality. Audéoud and Heusse [13] studied the correlation between the Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI), and Packet Delivery Ratio (PDR). RSSI is a weak indication, loosely correlated with PDR. The work was therefore based on the use of LQI since it gives more valuable information. However, the LQI may overestimate the quality of the channel in impulsive noise scenarios, since the LQI does not account for the packet losses. Eskola and Heikkilä [14] proposed a method to classify wireless channel disturbances related to line-of-sight...
changes and radio interference. However, the proposed method did not give any metrics that apply to protocols for improving network performance. Gomes et al. [15] proposed a Link Quality Estimator (LQE) node, dedicated to real-time link quality estimation using the obtained information and RSSI from received packets. They used RSSI and LQI values to infer the Packet Reception Rate (PRR) of the given links. Unfortunately, RSSI and PDR have proven to be only loosely correlated in many situations. The hardware-based estimators do not require computation resources because they use built-in hardware metrics. However, they do not have precise measurements, as found and recorded in previous studies.

The PRR and the Required Number of Packets (RNP) are some examples of software-based estimators based on the upper layer calculated information. RNP is calculated based on the transmitted packets from senders and is more reactive when compared with PRR. Thus, as long as the traffic is created by the sender, RNP can assess the quality of the link. However, if packets are successfully received after being retransmitted many times, RNP can underestimate the quality of the link. Some RNP-based estimators are Window Mean with Exponentially Weighted Moving Average (WMEWMA), Expected Transmission Count (ETX), and Four-Bit (FB). WMEWMA uses the EWMA filter as the key estimation technique; the PRR is calculated and then smoothed to the previously calculated PRR, which gives a more reliable yet sufficiently agile approximation compared to the PRR [16]. The average-based LQEs show poor reactivity. In order to overcome that, the authors [17] proposed the Kalman Filter-based Link Quality Estimator (KLE). ETX [18] considers link asymmetry by calculating the PRR of the backward link and the forward link. However, Koksal and Balakrishnan [19] found that in congested networks, passive monitoring ETX overloaded, since a large number of nodes did not receive packets to calculate ETX. FB [20] link estimation is designed with four bits of information. FB integrates RNP and WMEWMA with an EWMA filter to approximate the number of retransmissions. The authors in [21] performed a simulation to compare five LQEs: PRR, RNP, WMEWMA, ETX, and FB in smart-grid environments. The simulation result showed that ETX and FB showed better performance since ETX and FB considered the link asymmetry. However, ETX and FB did not consider burstiness in the links during their measurements.

In this study, we examined the effect of the link quality on designing a MAC protocol for industrial wireless sensor networks. Then, we proposed a link quality estimation mechanism that is based on counting the number of consecutively lost packets on each link at the receiving end, as calculated by the receiver node. The number of consecutively lost packets then gave the distribution of burstiness for each link in the networks. In our proposal, link quality measurement was executed during network formation; however, its result could be utilized during network working. Furthermore, based on the new link quality estimator, we proposed a metric called the burstiness distribution metric, which is used for estimating the number of transmissions in order to obtain target PRR that can be used to find the shortest transmission path in routing protocol. The metric can also be used for designing a MAC protocol for the industrial wireless sensor network or applying soft-sensor techniques [22,23] to give real-time estimation for the link quality measurement.

The rest of the paper is organized as follows. In Section 2, we described the system model and gave a link burstiness research overview. The proposed LQE is described in Section 3. In Section 4, we showed the performance evaluation from our method, and the conclusions are given in Section 5.

2. System Model

2.1. System and Network Model

In this study, we focused on convergecast multi-hop networks that consist of one coordinator (sink) node and some sensor nodes in a tree topology. All nodes are deployed randomly during network formation and they have identification numbers. The communication between two nodes is called a tree-link if they have a parent–child relationship and exchange data through this link. The data of the sensor are generated by sensor nodes and transmitted periodically through the tree-link to the sink.
2.2. Link Burstiness Research Overview

Physical property in wireless communications exhibits consecutive packet losses in a burst called a link burstiness. The authors in [24,25] proposed mechanisms to solve the link burstiness problem. Srinivasan et al. [24] proposed a new metric, denoted as $\beta$, showing that providing an inter-packet delay of 500 ms can avoid link burstiness. Even in the industrial fields, a 500 ms delay is a disadvantage for the network; one should not have to wait for an extended period of time to prevent packet failure when a link burst occurs. Munir et al. [25] proposed a maximum burst length metric (denoted $B_{\text{max}}$), estimated by long experiment data. The authors connected the scheduling algorithm and the routing algorithm called the least-burst-route to reduce the total number of burst lengths on the route. However, the algorithm to calculate $B_{\text{max}}$ was based on a long experiment data-trace before constructing the network. After processing the data-trace by computing the $B_{\text{max}}$ value, they applied it to the sensor network. They did not calculate the metric during network construction. That makes this algorithm hard to apply to an ad hoc network.

3. Link Quality Measurement and the Metric for Evaluating

3.1. Measure Link Quality Principle

To measure the link quality between two sensor nodes, probe packets are used in which the number of probes is predefined. Each node generates a probe packet, $P(\text{node}_\text{id}, \text{sequence})$, in which sequence is a sequence of probe packets generated, and it increases from 1 up to the number of probes. Upon receiving $P(\text{node}_\text{id}, \text{sequence})$, the receiver calculates the burstiness value by using the sequence number in $P$ for the sender node and then saving it to the burstiness distribution list (BDL). If a burstiness value duplicates others in the BDL, the burstiness time count for this burstiness value increases by 1. For example, let us consider the measurement link quality for nodes 1 and 4 in Figure 1, where node 1 received a probe packet from node 4 with the sequence number 1, denoted by $P(4, 1)$. The next received probe packet from node 4 is $P(4, 4)$, which means probe packets $P(4, 2)$ and $P(4, 3)$ were lost due to burstiness, and thus the burstiness value is 2. At this time, the current received probe packet is $P(4, 4)$, and the receiver will receive the next probe packet and will again calculate the burstiness value. Assume that the probe packet is $P(4, 7)$ the next time, so the burstiness value is again 2, but the burstiness time count will increase by 1 to indicate that burstiness value 2 was duplicated. Let the result of a transmission per link at the ith sequence be “S” if it was successful and “F” if it failed. With the above example of node 1 and node 4, we have the result of the transmissions after seven probe packets as “SFFSFFS”. After finishing the measure link quality (MLQ) period, we have the BDL of a link with 1000 probe packets transmitted, as shown in Table 1.

![Figure 1. An example of a network topology.](image)

Consider the relationship between burstiness value and burstiness time count in the BDL. The burstiness value is defined as the number of consecutive losses during probe packet transmission, and the burstiness time count is defined as the number of times the burstiness value appeared.
For example, in Table 1, after transmitting 1000 probe packets, a burstiness value of 0 occurred 634 times, a burstiness value of 1 occurred 129 times, and so on. It means that in the 1000 probe packets transmitted, 634 times the result of transmission was “SS” and 129 times the result of transmission was “SFS”, and so on. With a link of good quality, the burstiness value will be small, whereas the burstiness time count will be large for the minimum burstiness value. Conversely, a bad link has a higher burstiness value and a low burstiness time count for the minimum burstiness value.

| Burstiness Value | 0  | 1  | 2  | 3  | 4  |
|------------------|----|----|----|----|----|
| Burstiness Time Count | 634 | 129 | 31 | 2  | 1  |

For example, Figure 2 shows the simulation result from one pair of nodes with the different configurations of the PRR value. In Figure 2, the distribution of the burstiness values is compared with a link PRR of 70%, 80%, and 90%, when the number of probes is 1000 packets. Figure 2 describes how with a good link (PRR 90%) we have a burstiness time count with a minimum burstiness value at 816 and a maximum burstiness value of 2. On the other hand, with bad link quality (PRR 70%), we have a burstiness time count with a minimum burstiness value of 480 and a maximum burstiness value of 4.

![Figure 2. The distribution of Burstiness compared with link Packet Reception Rate (PRR).](image)

### 3.2. Burstiness Distribution Metric

In this section, we present a scheme to calculate the number of transmissions by using the BDL with a named burstiness distribution (Bdist) metric.

#### 3.2.1. Calculate Burstiness Distribution List

First, the neighbor list (nbrlist) and the BDL are set to empty. When sensor node x receives a probe packet from node y, it saves node y’s ID and sequence to its nbrlist if node y is not already on node x’s nbrlist. If node y is on node x’s nbrlist, node x will calculate the burstiness value for node y as follows:

\[
\text{burstiness\_value}(y) = \text{seqno}(y) - \text{nbrlist}[y].\text{seqno}(y) - 1
\]

In which, seqno(y) is the sequence number in the new incoming probe packet from node y, and nbrlist[y].seqno(y) is the previous sequence number of node y saved in the neighbor list.

The parameter burstiness\_time\_count(y) refers to a burstiness\_value(y) set to 1, and the pair \{burstiness\_value(y), burstiness\_time\_count(y)\} will be saved in node y’s BDL if this value is not already on node y’s BDL. If burstiness\_value(y) is on node y’s BDL, node x will increase the burstiness\_time\_count(y) referring to burstiness\_value(y) by 1. Then, it updates the seqno(y) value in the neighbor list.
This process will repeat during the measure link quality period. The detailed algorithm is in Algorithm 1, and the flowchart is in Figure 3.

**Algorithm 1: Calculate burstiness distribution list**

```
Input : Incoming probe P
Output : BDL
1:   Initialize nbrlist is empty, BDL is empty
2:   When node y send probe packet:
3:       P = (y, seqno(y));
4:   When node x received probe packet from node y:
5:       If (y not belong to x’s nbrlist)
6:           nbrlist ← y, y’s seqno;
7:       Else
8:           burstiness_value(y) = seqno(y) − nbrlist[y].seqno(y) − 1;
9:           If(burstiness_value(y) not in y’s BDL)
10:              burstiness_time_count(y) = 1;
11:              (burstiness_value(y), burstiness_time_count(y)) → y’s BDL;
12:       Else
13:          BDL[burstiness_value(y)].burstiness_time_count(y)++;
14:       Endif
15:   Endif
16:   nbrlist[y].seqno(y) ← seqno(y);
```

### 3.2.2. Calculate Burstiness Distribution Metric

To calculate the Bdist metric, we consider the possible loss ratio for each link, and determine the Bdist value from the BDL. We assume that a target PRR is considered. For example, to reach a PRR target of 99% for the data transmission period, the loss ratio should be lower than 1% for each link. If the route has \( h \) hops, then every link that the route goes through has a loss ratio lower than \( \sqrt[\sqrt{0.01}]{h} \).

First, based on the hop count of the current node and target PRR, the end-to-end loss ratio (\( e_{loss\ ratio} \)) that we accepted to reach the PRR target is calculated by Equation (1):

\[
e_{loss\ ratio} = 1 - \sqrt[\sqrt{PRR_{target}}]{h} = 1 - h^{\frac{1}{h\sqrt{PRR_{target}}}}
\]  

(1)

Then, we can calculate the number of packet loss threshold (denoted \( N_{loss\_threshold} \)) corresponding to the number of probes (denoted \( N_{probes} \)) with Equation (2):

\[
N_{loss\_threshold} = e_{loss\ ratio} \times N_{probes}
\]  

(2)

The number of current packet loss (denoted \( N_{current\_loss} \)) is calculated by multiply pair of \( \{burstiness\_value, burstiness\_time\_count\} \) at \( i \)th in BDL as follows:

\[
N_{current\_loss} = i \times BDL[i];
\]

in which, \( i \) is burstiness value, \( BDL[i] \) is burstiness_time_count of \( i \).
The Bdist metric is set equal to the burstiness value, such that the total of $N_{current\_loss}$ lower than, or equal to, the $N_{loss\_threshold}$. The detailed calculation of Bdist is in Algorithm 2 and the flowchart for Algorithm 2 is in Figure 4.

Algorithm 2: Calculate Burstiness Distribution Metric

| Input     | Ascending sorted burstiness distribution list (BDL), number of probes ($N_{probes}$), target PRR (PRRtarget), hop count (h) |
|-----------|-------------------------------------------------------------------------------------------------------------------------------|
| Output    | Burstiness distribution metric (Bdist)                                                                                           |
| 1         | Calculate end-to-end loss ratio by using Equation (1)                                                                             |
| 2         | Calculate $N_{loss\_threshold}$ by using Equation (2)                                                                             |
| 3         | Initialize the $N_{current\_loss} = \text{Bmax} \times \text{BDL[Bmax]}$                                                        |
| 4         | For loop in BDL from $i = \text{Bmax}$ to 0                                                                                      |
| 5         | If($N_{current\_loss} \leq N_{loss\_threshold}$)                                                                                |
| 6         | $N_{current\_loss} += i \times \text{BDL}[i]$                                                                                  |
| 7         | Else                                                                                                                           |
| 8         | Bdist = $i + 1$                                                                                                                |
| 9         | Break;                                                                                                                         |
| 10        | End if                                                                                                                         |
| 11        | End for                                                                                                                        |
| 12        | Return Bdist                                                                                                                  |
For example, a one-hop link has a target PRR of 99%, and the number of probes is 1000 in order to measure the link quality, which means that 1% of 1000 packets can be lost. The number of packets that can be lost during data transmission is 10 packets. In the BDL in Figure 2, we can calculate the Bdist value for successful packet transmission as 2 since the consecutive loss of two times transmission in the link at a PRR of 90% was 10 packets, equal to the allowed packet losses.

![Figure 4. The flowchart of the main procedures in Algorithm 2.](image)

### 4. Evaluation

By using the Cooja simulator, the Bdist metric was compared with schemes such as ETX and PRR. The key parameters are presented in Table 2.

| Table 2. Summary of the simulation set-up parameters. |
|-----------------------------------------------------|
| Number of nodes                                    | 10     |
| Number of probes                                   | 1000   |
| Target PRR                                         | 99%    |
| Link quality of the channel (set)                  | 70–90% |
| Number of the data packets                         | 1000   |
| Network area                                       | 100 m × 100 m |
| Payload size                                       | 26 bytes |
| Simulation time for each scenario                  | About 55 min |
The link quality of the channel for each link indicates the percentage of successful packets at the receiver, compared with the number of packets from senders. For example, a channel link quality at 70% indicates that the receiver will successfully receive 70 packets if sent 100 packets. The number of probes to measure the link quality and the number of data packets to evaluate the measured link quality metric are both 1000.

4.1. Relationship between the Number of Retransmissions and Network Performance

To evaluate the effect of the number of retransmissions on the performance of the network, a simple network with one sink (gateway) and nine sensor nodes was considered. The link quality of the channel was fixed, while the number of retransmissions varied from 0 to 3, and we examined the packet reception rate at the sink with each value for the number of retransmissions.

The results in Figure 5 show that since the sensor nodes were not allowed retransmissions, the PRR at the sink achieved a low PRR: 66.3%. When the sensor nodes were allowed retransmissions, the PRR improved since the number of retransmissions increased. It is shown that when the link quality is not very good, the number of retransmissions will decide the network performance. Therefore, to estimate the number of transmissions is very important.

![Figure 5. Relationship between the number of retransmissions and the packet reception rate.](image)

4.2. Effect of the Hop Count on Network Performance

We evaluated the effect of the hop count on PRR. A simple linear network with the hop count varying from 1 to 4 was considered. We compared our proposed approach with schemes like ETX and PRR. Each scheme estimates the number of transmissions by itself. We examined the packet reception rate at the sink for each value of the hop count.

The results are presented in Figure 6, where it is indicated that our proposed method achieved a very high PRR, even at the highest hop count. That is because our proposed method provided a good estimation of the number of transmissions by using the target PRR to estimate link quality. Under other schemes, the PRR decreased since the hop count increased. This is because with the two other schemes, the estimate of the number of transmissions was not very good.
In this examination, nine sensor nodes were deployed randomly, and the link quality between every two nodes was set randomly from 70% to 90%. Each sensor node generated 1000 packets periodically. The Time Slotted Channel Hopping (TSCH) MAC protocol was used with the Path Collision-aware Least Laxity First (PCLLF) [26] scheduling algorithm to guarantee that no collisions occurred in the network. We collected the packet reception rate of each sensor node at the sink.

The resulting PRR data for each sensor node at the sink are presented in Figure 7, where the PRR of the proposed scheme almost achieved the target PRR with the estimated number of transmissions for each sensor node. It shows that the algorithm for calculating the number of transmissions worked well. Furthermore, the other schemes achieved a much lower PRR since the estimation of the number of transmissions by ETX and PRR was not very good. This is because ETX and PRR did not consider the burstiness that happens in the links.

In this examination, we ran nine sensor nodes deployed with a link quality between two nodes set randomly between 70% and 90%. We compared network schemes using minimal scheduling, orchestra scheduling, and PCLLF scheduling, with Bdist as the metric value and PCLLF with no retransmission configuration. Each sensor node generated 1000 packets and transmitted using the time slot assigned by the scheduling algorithm. We collected the data packet reception rate at the sink node.

Figure 8 shows the data PRR of each sensor node at the sink. From Figure 8, the network performance of the proposed Bdist metric had the highest reliability compared with the other network schemes. Almost all sensor nodes achieved the target PRR by using the number of transmissions calculated with our method. Other network schemes showed lower performance since they did not consider the retransmissions.
5. Conclusions

In this study, we proposed an LQE to apply to industrial wireless sensor networks with high reliability. Based on the burstiness property of wireless links, we estimated the number of transmissions required to reach the PRR target by using the burstiness distribution list proposed in Section 3. We proved by simulation in Cooja that our approach estimated the number of transmissions that can reach the target PRR, as we expected. Based on the simulation in which we compared our approach with some RNP-based methods (ETX and PRR), our proposal can be used as the input metric to calculate the number of transmissions in scheduling for industrial wireless sensor networks. We conclude that our approach is highly suitable for industrial wireless sensor networks that require high reliability for data transmissions. In future work, we will evaluate our approach in regards to the real devices for monitoring and controlling systems in industrial environments.

Probe packets distribution makes increasing the overhead of the network. However, it can be helpful in guaranteeing the reliability of data transmission when the link quality is not very good. Especially in real applications, we cannot guarantee that the link quality can be stable every time. Every link in the network may have a very different link quality. Therefore, estimating the link quality to guarantee the reliability of data transmission is more important than energy consumption in the industry application.

Author Contributions: Conceptualization, N.H.N. and M.K.K.; methodology, N.H.N.; software, N.H.N.; validation, N.H.N. and M.K.K.; formal analysis, N.H.N.; investigation, N.H.N.; resources, M.K.K.; data curation, N.H.N.; writing—original draft preparation, N.H.N.; writing—review and editing, N.H.N. and M.K.K.; visualization, N.H.N.; supervision, M.K.K.; project administration, M.K.K.; funding acquisition, 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 2020.

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

References

1. Shah, R.C.; Roy, S.; Jain, S.; Brunette, W. Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Netw.* 2003, *1*, 215–233. [CrossRef]
2. He, X.; Fu, X.; Yang, Y. Energy-Efficient Trajectory Planning Algorithm Based on Multi-Objective PSO for the Mobile Sink in Wireless Sensor Networks. *IEEE Access* 2019, *7*, 176204–176217. [CrossRef]
3. Fu, X.; Yao, H.; Yang, Y. Cascading failures in wireless sensor networks with load redistribution of links and nodes. *Ad Hoc Netw.* 2019, *93*, 101900. [CrossRef]
4. Fu, X.; Yang, Y.; Postolache, O. Invulnerability of Clustering Wireless Sensor Networks Against Cascading Failures. *IEEE Syst. J.* 2019, *13*, 1431–1442. [CrossRef]
5. Mittal, M.; Iwendi, C.; Khan, S.; Rehman Javed, A. Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using Levenberg-Marquardt neural network and gated recurrent unit for intrusion detection system. *Trans. Emerg. Telecommun. Technol.* 2020, 13, 6430, e3997. [CrossRef]

6. 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]

7. Nguyen, T.T.; Oh, H. SCSMA: A Smart CSMA/CA Using Blind Learning for Wireless Sensor Networks. *IEEE Trans. Ind. Electron.* 2019, 67, 10981–10988. [CrossRef]

8. Winter, T.; Thubert, P.; Brandt, A.; Hui, J.W.; Kelsey, R.; Levis, P.; Pister, K.; Struik, R.; Vasseur, J.-P.; Alexander, R.K. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *RFC 2012*, 6550, 1–157.

9. Mittal, M.; Iwendi, C. A survey on energy-aware wireless sensor routing protocols. *EAI Endorsed Trans. Energy Web* 2019, 6. [CrossRef]

10. Clausen, T.; Yi, J.; Herberg, U. Lightweight on-demand ad hoc distance-vector routing-next generation (LOADng): Protocol, extension, and applicability. *Comput. Netw.* 2017, 126, 125–140. [CrossRef]

11. Nguyen, T.T.; Oh, H. Constructing an optimally balanced tree to maximize data throughput with multiple channels. *Wirel. Netw.* 2018, 24, 993–1005. [CrossRef]

12. Noda, C.; Prabh, S.; Alves, M.; Boano, C.A.; Voigt, T. Quantifying the channel quality for interference-aware wireless sensor networks. *ACM SIGBED Rev.* 2011, 8, 43–48. [CrossRef]

13. Audéoud, H.-J.; Heusse, M. Quick and efficient link quality estimation in wireless sensors networks. In Proceedings of the 2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Isola, France, 6–8 February 2018; pp. 87–90.

14. Eskola, M.; Heikkilä, T. Classification of radio channel disturbances for industrial wireless sensor networks. *Ad Hoc Netw.* 2016, 42, 19–33. [CrossRef]

15. Gomes, R.D.; Queiroz, D.V.; Lima Filho, A.C.; Fonseca, I.E.; Alencar, M.S. Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes. *Ad Hoc Netw.* 2017, 59, 116–133.

16. Gungor, V.C.; Korkmaz, M.K. Wireless link-quality estimation in smart grid environments. *Int. J. Distrib. Sens. Netw.* 2012, 8, 214068. [CrossRef]

17. Kaneko, H.; Funatsu, K. Application of online support vector regression for soft sensors. *AIChE J.* 2014, 60, 600–612. [CrossRef]
26. 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.

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

© 2020 by the authors. 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 (http://creativecommons.org/licenses/by/4.0/).