A Big Slot Scheduling Algorithm for the Reliable Delivery of Real-Time Data Packets in Wireless Sensor Networks

Hoon Oh and Md Abul Kalam Azad

Abstract In wireless sensor networks (WSNs), guaranteeing a reliable data transmission over a time-varying wireless channel is a challenging task. The existing TDMA-based MAC protocols assign time slots to the nodes individually for the transmission of data packets safely. However, these protocols may not be suitable for the industrial WSNs in which the stability of wireless links is threatened by various obstacles. In this paper, we propose a new slot allocation and utilization method that allocates one big slot for all nodes at each depth of a tree, and allows the nodes to share it through contention. The big slot constrains the packet transmission delay of all nodes at the same depth, thereby limiting the packet transmission delay to a sink. We show by simulation that the proposed approach is very dependable against the time-varying channel in WSNs.

Keywords Slot scheduling · Real-time · Safety-critical · TDMA · CSMA

1 Introduction

A safety-critical application [1] requires a timely and reliable data transmission over a communication channel in industrial WSNs. Thus, many TDMA-based MAC protocols [1–3] were proposed to tackle this problem. However, the harsh industrial environment incurs the frequent failures of wireless links due to the ambient noises and interferences, thus obsoleting part of the scheduled slots. Thus, it is highly required but challengeable to design a real-time MAC protocol that is robust against the time-varying wireless links.

H. Oh (✉) · M.A.K. Azad
Ubicom Lab, School of Computer Engineering and Information Technology, University of Ulsan, P.O. Box 18, Ulsan 680-749, South Korea
e-mail: hoonoh@ulsan.ac.kr

M.A.K. Azad
e-mail: akazad.ju@gmail.com

Q.-A. Zeng (ed.), *Wireless Communications, Networking and Applications*, Lecture Notes in Electrical Engineering 348,
DOI 10.1007/978-81-322-2580-5_2

TreeMAC [2] assigns nonoverlapping frames to all nodes where each frame consists of three slots. This protocol is advantageous in terms of channel efficiency by allowing slots to be reused by the nodes every three depths in a tree. However, the slot reuse can incur the irregular interference because the interference range is always farther than the transmission range. Furthermore, it does not suggest any measure against the link failures. I-MAC [1] that targets small control and monitoring networks excludes the spatial reuse of slots by allocating slots to each node in a distinct manner. Moreover, it tries to enhance reliable data transmission in industrial unstable WSNs such that data transmission within a slot is secured by control packets such as RTS, CTS, and ACK. However, it does not respond effectively to the time-varying failures of wireless links even though it suggests a spare time utilization scheme to salvage packets.

Z-MAC [3] combines the advantages of TDMA-based and CSMA-based protocols. Z-MAC allocates time slots to every node such that no two nodes within two-hop neighbors are assigned the same time slot in order to prevent interference. However, due to slot scheduling overhead, they recommend the execution of DRAND [4] only at network initialization time. This protocol does not address the irregular interference problem and does not take any measures against link failures, either.

Since the time-varying noise and interference in industrial fields is dependent on frequency, WirelessHART [5] employs a frequency (i.e., channel) hopping technique in which a sensor node switches randomly to one out of a predefined list of wireless channels every slot in order to reduce the effect of noise and interference. This definitely improves the data transmission reliability. However, WirelessHART reschedules slots network-wide to fix any link failure, similar to the other MAC protocols. On the other hand, GinMAC [6] calculates a number of additional slots deterministically that is reserved for improving reliability against the worst case channel characteristics. However, GinMAC loses its reliability control when the depth of a tree extends beyond the static topology envelope.

Some other protocols to improve reliability by reducing data collision have been proposed. The link activity scheduling approach [7] builds and examines a conflict graph while S-Web-based MAC [8] uses checkerboard and dartboard-based slot scheduling to remove data collisions by reducing the number of nodes contending for the media at the same time. Note that both protocols consider the link failures by the collisions only, but not by the external interference. RNP [9] employs overhearing and piggybacking techniques to forward data packets cooperatively that are missing due to the external interference. However, all data packets are broadcasted to allow for overhearing and piggybacking, which incurs high collisions.

According to the discussion so far, any of the existing TDMA-based MAC protocols except for I-MAC directly address a method to ensure the reliability of data delivery and also a method to survive over the link failures in industrial WSNs. I-MAC also responds weakly to the link failures. Thus, the existing MAC protocols may not be suitable for industrial WSNs with the inherent link instability due to various obstacles or high signal noises.

We propose a new slot allocation and utilization method that improves the reliability of data transmission over the time-varying unstable links in industrial WSNs. In our approach, one big slot is allocated for all nodes at each depth of a tree and shared by those nodes for data transmission to their respective parents. The proposed approach reduces the channel competition because only the nodes at the same depth contend for channel within the same big slot. Furthermore, the transmission delay at lower depths will increase since the nodes at the lower depths have higher probability of collision due to their higher number of packet processing than that of nodes at higher depths. Considering these, we develop a formula to generate the variable-sized big slots according to tree depths. We show by simulation that the proposed approach is very dependable against the time-varying link failures in WSNs.

The rest of the paper is organized as follows. In Sect. 2, we discuss the background of the proposed approach with some necessary definitions and notations. In Sect. 3, we formally describe the proposed approach with the analysis of its properties. In Sect. 4, performance evaluation is given by resorting to simulation. Finally, we make concluding remarks in Sect. 5.

2 Background

2.1 Network Model

Industry fields are often harsh and non-friendly to the wireless communication activity due to the obstacles, changing structures, and ambient interferences, resulting in unstable wireless links. An industrial monitoring and control WSN usually consists of one data collection and control server (hereafter, referred to as simply node or sink node) and a number of sensor devices (hereafter, referred to as simply node or sensor node) that include at least one sensor module for sensing the environment. Each sensor node generates one data packet that is required to send to the sink node within a specified time bound, which is defined by an application.

A sink node is wall-powered and a sensor node is battery-powered. The transmission range of a sensor node is limited for spectrum efficiency and battery efficiency. A sink node collects data from the network at regular intervals, thus naturally bounding data delivery time. The sensor nodes form a tree originating from a server in which each node except for a sink node has a parent and may have multiple children. A node is said to be a tree-node if it belongs to a tree. Otherwise, it is an orphan node. Two nodes that can directly and mutually communicate with each other are said to have a link. A link between a node and its parent is specially called a tree link. A link can be broken because of node failure, battery depletion, interferences, or the intervention of some obstacles.

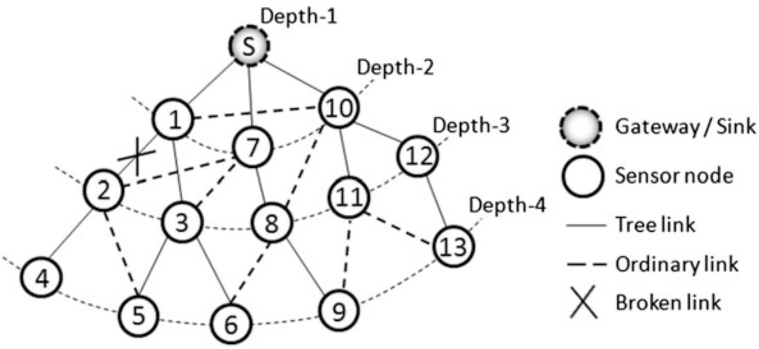

Fig. 1 A network model

Figure 1 shows an example network in which one sink node and 13 sensor nodes form a tree originating from sink node *S*. The solid lines and the dashed lines indicate *tree-links* and *ordinary-links*, respectively.

2.2 Motivations

A TDMA-based protocol eliminates contention among the nodes by assigning time slots to them for data transmission and gives them a guaranteed chance to deliver data packet to a sink. However, if a node loses one of its upstream links, it cannot deliver packets to a sink. There are two approaches for resolving this. One is to reconstruct a tree and then reschedule slots over the whole tree. Tree reconstruction and slot rescheduling can be performed either after every tree-link failure or after some percentage of nodes fails to deliver data packets to a sink. The former one suffers from a lot of overhead while the latter one causes some amount of slots to be wasted. Another is to repair the broken tree and reschedule slots locally for the broken part. However, the local rescheduling of slots is very difficult since it affects the amount of slots required by and the slot start times of the related nodes. In consequence, tree maintenance by TDMA scheme is radically unfavorable since it is accompanied by the costly slot rescheduling.

Let us consider some tree-based MACs such as TreeMAC and I-MAC for a simpleWSN as shown in Fig. 1. If link (1, 2) is broken, the slots allocated to nodes 2 and 4 will be wasted until tree reconstruction and slot rescheduling are made. However, if node 2 can change its parent to node 7 without affecting the scheduled slots, it would make seamless data transmission via the new parent while resolving the problems such as control overhead and the waste of slots. Nonetheless, slot reusing is difficult to instantiate because it is not easy to identify whether or not any two nodes can use the same slot without interference since interference range is always farther than the physical transmission range [10]. However, the slot reuse increases channel efficiency, but impairs the freedom from interference in data transmission.

To achieve seamless data transmission, reduce control overhead caused by tree maintenance, and improve channel efficiency, it would be desirable to assign one unique big slot to all the nodes at the same depth of a tree so that they can share the big slot for data transmission. Even though a node changes its parent, it does not have to change the big slot as long as it maintains the same tree depth. If its depth is changed, it can utilize another big slot allocated to the changed depth. One problem is to incur collision and delay due to the competition of data transmission by employing the CSMA scheme within a big slot. One favorable aspect regarding this is that the competition of data transmission is confined to the nodes at the same tree depth. Another is that the opportunistic parallel transmission is made possible. If two nodes at same depth do not interfere with each other in terms of the CSMA operation, they can transmit packets to their respective parents simultaneously. Nodes 4 and 13 in Fig. 1 are such a case.

In consequence, tree construction and slot scheduling are of great importance to realize the proposed concept. In this paper, the function to generate big slots in a distributed manner will be developed, and a slot scheduling algorithm will be briefly described for space constraints.

2.3 Notations and Definitions

For convenience, we use some notations and definitions as follows.

- depth(i): The depth of node i
- $N(i)$: A set of neighbors of node i
- $C(i)$: A set of children of node i
- $P(i)$: The parent of node i

Definition 1 A bigslot (BS) is a time span that all nodes at the same depth share to receive data packets from their children and transmit their data packets to their respective parents using CSMA scheme.

A big slot allocated to the nodes at depth i is denoted by BS (i). BS(i) is divided into two parts, BSRx(i) and BSTx(i) that are used for the nodes at depth i to receive data packets from their children and to transmit data packets to their parents, respectively.

Definition 2 A superframe (SF) is given the sum of the transmission portions of all the big slots allocated to the nodes at different depths as follows.

$$\mathrm{SF} = \sum_{i=2}^{H} \mathrm{BS}^{\mathrm{TX}}(i)$$

where BSTx(i) and BSTx(j) when $i \neq j$ do not overlap.

3 Slot Scheduling

3.1 Tree Construction and Maintenance

During tree construction, every node i constructs its neighbor information table, $\mathrm{NIT}(i) = \{(x,\ \mathrm{status},\ \mathrm{depth}(x))|x \in \mathrm{N}(i)\}$, where status $\in$ {primary parent, secondary parent, child}. We assume that every link is bidirectional and time is synchronized over the entire network.

A tree construction process is as follows. At initialization, a sink is the only tree node and initiates tree construction by issuing a tree construction request, TCR = (node ID) message. Upon receiving TCR, an orphan node joins the sink by sending a join request, JREQ = (sender, receiver, depth) message. Upon receiving JREQ, a tree node sends a join response, JRES = (sender, receiver, depth) message and takes the orphan node as its child. When the orphan node receives JRES, it takes the tree-node as its parent. Another orphan node who has overheard JREQ can take the same procedure to become a tree node. If an orphan node overhears multiple JREQs from different tree nodes, it takes a tree node that provides the shortest distance (depth) to the sink as a primary parent, and other tree nodes with the same distance as the secondary parents.

3.2 Wait Time Generation Function

A slot scheduling in this paper is based on the wait time distribution function which was proposed in [11]. The wait time distribution function is given below.

$$\mathrm{WTime}(d) = W_1 \times a^{d-1} \tag{1}$$

where, WTime (d) is the time that a node at depth d has to wait to transmit data packets to its parent and the range of the base a is in (0, 1]. Since a sink does not have to send any packets, WTime (d) is in (0, W_1]. Therefore, the wait time for a sink, WTime(1) is equal to SF(=W_1).

The wait time distribution function has two basic principles. The first one is that it generates a skewed wait time according to tree depths. This implies that a node waits for all of its children completing their data transmission for the favor of data aggregation. The second one is that the wait time gap between any two nodes of two consecutive depths increases exponentially as depth decreases. This way of gap distribution is necessary since the nodes at lower depths have higher contention to acquire channel since they have to process the more number of data packets in proportion to the number of their descendants, and have the reduced possibility of parallel transmission due to the reduced distance among the nodes.

3.3 Estimation of Key Parameters

One big issue is how a and W_1 can be determined. In [11], the approximate range of a for a small tree-based network is determined as $0.63 \leq a \leq 0.82$. One good thing is that the values of a within this range are less sensitive to the number of nodes and the dimension of the network. Thus, a will be chosen 0.7 for our experiment in the later part of this paper.

The size of a superframe, W_1 should be greater than the summation of transmission times of all packets from each node to a sink, assuming that each node generates only one packet within one superframe. Thus, the bound of W_1 can be given as follows.

$$\sum_{d=2}^{H}(d-1) * n_d * T \leq W_1 \leq \sum_{d=2}^{H}(d-1) * n_d * E[D] \tag{2}$$

where H is the depth of a tree, n_d is the number of nodes at depth d, T is the one hop transmission time of a packet and $E[D]$ indicates an expected delay of a packet when a node sends the packet to its parent using CSMA.

The initial values of the parameters H, n_d, T, and $E[D]$ were studied in other papers [11, 12]. Especially, n_d is dependent on the number of nodes in the considered network. Thus, this value can be increased adaptively according to the increase in the number of participating nodes. The values of W_1 and a can be included in the TCR, JREQ, and JRES messages so that every node can know them.

3.4 Big Slot Calculation and Slot Scheduling

We can calculate the length of a big slot (BS) using Eq. (1). The receiving slot length, $\mathrm{BS}^{\mathrm{Rx}}(i)$ of a node at depth i is given as follows.

$$\mathrm{BS}^{\mathrm{Rx}}(i) = \begin{cases} \mathrm{WTime}(i) - \mathrm{WTime}(i+1) & \text{if a node at depth } i \text{ has a child} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The sending slot length, $\mathrm{BS}^{\mathrm{Tx}}(i)$ of the same node at depth i,

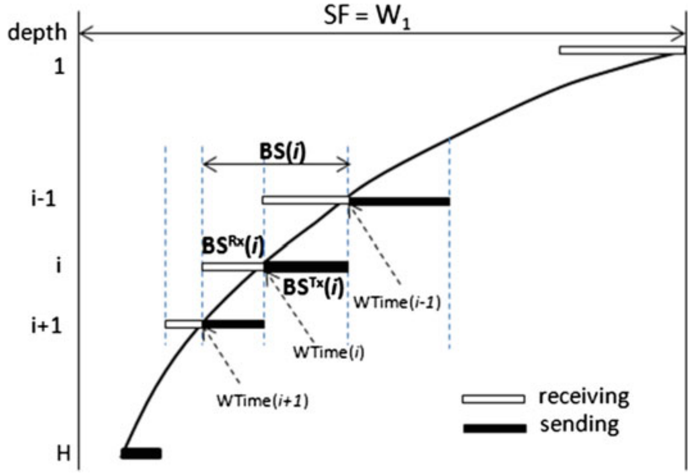

Fig. 2 The variable size of big slots at different depths and their relationship

$$\mathrm{BS}^{\mathrm{Tx}}(i) = \begin{cases} \mathrm{WTime}(i-1) - \mathrm{WTime}(i) & \text{if } i > 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

To find the size of the BS of a node at depth i, we need to sum up Eqs. (3) and (4) as follows.

$$\mathrm{BS}(i) = \mathrm{BS}^{\mathrm{Rx}}(i) + \mathrm{BS}^{\mathrm{Tx}}(i) = \begin{cases} W_1(a^{-2} - 1) * a^i \text{ if } a \text{ node is an intermediate node} \\ W_1(a^{-2} - a^{-1}) * a^i \text{ if } a \text{ node is a leaf node} \\ W_1(a^{-1} - 1) * a^i \text{ if } a \text{ node is a sink node} \end{cases} \tag{5}$$

According to Eq. (5), the BS size varies depending on the depth of a node in a tree such that nodes at lower depths require a BS larger than those at higher depths. It is worth mentioning that the summation of all BS's is larger than the SF size as the BS's at adjacent depths overlap with each other.

Figure 2 shows the relationship of the variably sized big slots at different depths. $\mathrm{BS}(i)$ is divided into $\mathrm{BS}^{\mathrm{Rx}}(i)$ and $\mathrm{BS}^{\mathrm{Tx}}(i)$. $\mathrm{BS}^{\mathrm{Rx}}(i)$ and $\mathrm{BS}^{\mathrm{Tx}}(i)$ overlaps with $\mathrm{BS}^{\mathrm{Tx}}(i + 1)$ and $\mathrm{BS}^{\mathrm{Rx}}(i - 1)$, respectively, so that every node can receive and transmit data packets from its children and to its parent, respectively.

3.5 Big Slot Assignment Algorithm

Every node can compute its BS in a distributed manner according to Eq. (5) since it knows its depth. Now, a node should be able to determine the start time of its BS within the superframe, W_1. Suppose the start time of a superframe is sTime. Then, a node at depth i can determine the start times of $BS^{Rx}(i)$ and $BS^{Tx}(i)$ denoted as RxTime(i) and TxTime(i), respectively.

$$\text{RxTime}(i) = \text{sTime} + \text{WTime}(i + 1) \tag{6}$$

$$\text{TxTime}(i) = \text{sTime} + \text{WTime}(i) \tag{7}$$

$$\text{Sleep Time}(i) = \text{sTime} + \text{WTime}(i - 1) \tag{8}$$

According to Eqs. (6), (7), and (8), every node at depth i wakes up at RxTime (i) to receive data packets from its children. As soon as it finishes receiving data packets, it gets into sleep mode and wakes up at TxTime(i) to forward data packets to its parent. Then, it gets into sleep mode at SleepTime(i) until the next RxTime (i) being RxTime(i) + W_1, regardless of the success of data forwarding.

4 Performance Evaluation

We evaluated the proposed big slot scheduling, named *BSSA* for convenience, using the QualNet simulator version 5.0.2. We compare our approach with I-MAC [1], which has exhibited better performance than other contemporary real-time MAC protocols.

4.1 Simulation Model

To ease topology dimensioning, we enlarge both simulation area and transmission range to the same extent. Therefore, the experimental results by these modified parameters are in congruence with the network model in Sect. 2.1. Using mathematical formulas in [13, 14], we get the average value of H as 7. Substituting $H = 7$, $T = 3.125$ ms [12], $E[D] = 30$ ms [11] in Eq. (2), we get a theoretical range of W_1 as 0.25 s ≤ W_1 ≤ 2.4 s. However, we set an optimum value of W_1 to 1.6 within this range according to our simulation study. Table 1 shows the key simulation parameters and values.

Table 1 Simulation parameters and values

Parameter	Value
Number of node, *n*	1 sink and 25 sensor nodes
Dimension, *d*	100 m × 100 m
Simulation time, *T*	600 s
W_1 (for BSSA)	1.6
a (for BSSA)	0.7
Slot size (for I-MAC)	20 ms
Transmission range, *R*	20 m (−25 dBm)
Channel frequency, *Fr*	2.4 GHz
Path loss model	2-ray ground
Sensor energy model	MicaZ
Battery model	Linear
Maximum Tx times (MAX_TIMES in I-MAC)	2
Data packet length	100 bytes

4.2 Simulation Scenarios

All sensor nodes are static. Twenty-five sensor nodes are uniformly distributed within the boundary of a simulation area of 100 m by 100 m, and a sink is placed at the middle of the top of the area. Each sensor node transmits only one packet of 100 bytes every superframe. The following two scenarios are used to evaluate the performance.

- Scenario I: Stable link scenario
  In this model, all the links remain connected till the simulation ends. In this case, we want to examine the impact of using BS in network operations. We are especially interested in knowing how much deviation in performance happens due to the use of CSMA within BS instead of the totally slotted approach.
- Scenario II: Randomly disconnected link scenario
  In this model, a statistical generator is used to break the links in a purely random manner. The location and the number of link breaks within a cycle cannot be predicted or controlled from a user interface. Every SF, a node picks up a value *k* from a subset of all nodes participating in the simulation. If *k* is equal to its ID, the node is considered to be disconnected to its parent. So, the node remains disconnected to its primary parent for the current SF. The number of link breaks in an SF is controlled by the Link Break Index (LBI), α. We can set α to a low value for a high rate link failure, and vice versa.

4.2.1 Simulation with Stable Link Scenario

Packet Delivery Ratio (PDR). In Fig. 3, it is shown that PDR with BSSA is slightly decreasing with an increasing depth by about 1 % at depth 7. This implies that about 1 % of the packets generated at depth 7 are lost while they move along the upstream tree paths. Since packets travel through multiple hops by resorting to CSMA, some of them becomes lost due to collision. However, this is not noticeable considering the properties of wireless networks; thereby supporting that data collision made among the nodes at same depth can be almost disregarded.

4.2.2 Simulation with Randomly Disconnected Link Scenario

Packet Delivery Ratio. In this scenario, a random number of links is broken down every superframe (SF). Since BSSA allows nodes to manage multiple parents and to utilize the same big slot, it can quickly respond to the broken links by changing parents with no overhead.

Referring to Fig. 4, it is shown that BSSA achieves higher PDR than I-MAC and remains stable against the link breakage. However, I-MAC waits link recovery until a tree is reconstructed in the start of next superframe. A more number of link failures demands for more frequent tree reconstructions, which causes a high decrease in PDR. However, I-MAC PDR reaches an equilibrium point for a specific value of LBI, α, in which tree re-construction rate does not lower down PDR any further.

Energy Consumption. BSSA is low in energy consumption due to distributed slot scheduling and the flexibility of topology change compared with I-MAC.

We measured energy consumption for each node as simulation time progresses. Then, average energy consumption for all participating nodes is depicted in Fig. 5. It is natural that both curves show a linearly increasing pattern with simulation time. According to the figure, we can see that BSSA consume much less energy than

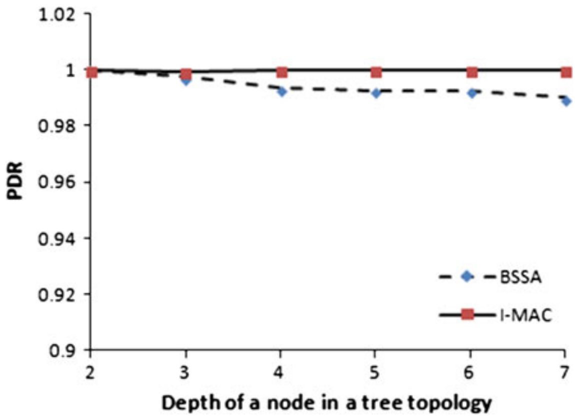

Fig. 3 Packet delivery ratio with stable links

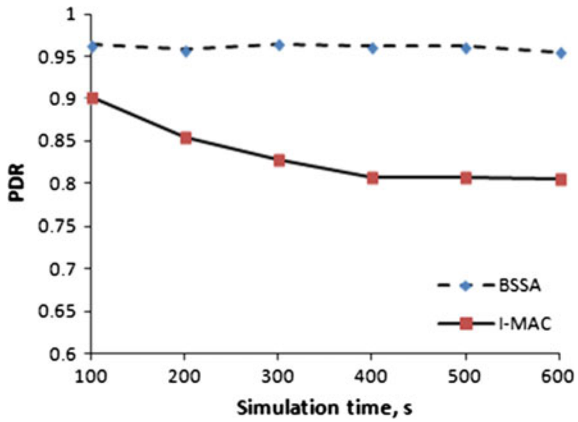

Fig. 4 Packet delivery ratio in random link failure

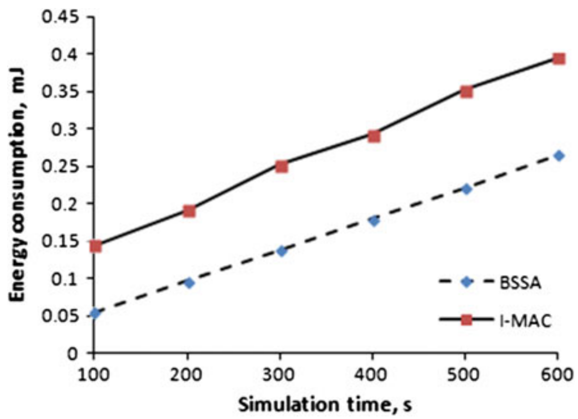

Fig. 5 Average power consumption in random link failure

I-MAC and shows a slightly increasing pattern. This is because in BSSA, every node determines a big slot independently of other nodes and even though a link is broken, tree reconstruction is not necessary.

5 Conclusions

We have proposed a new slot scheduling algorithm, BSSA that exhibits a highly robust behavior in an interference prone environment. BSSA provides a run-time defense against unpredictable link failures by delivering packets through secondary parents. The proposed approach achieves this capability by instantiating a local and controlled CSMA operation within a big slot.

The simulation result shows that BSSA provides a comparable level of performance to I-MAC in a stable link scenario. However, it clearly outperfroms I-MAC in the unpredictable link breaking scenario. Moreover, the proposed protocol consumes significantly lower amount of energy compared to I-MAC. Therefore, we conclude that BSSA should be a promising approach for the noisy and interfere-prone industrial applications.

Acknowledgments This research is supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2013396).

References

1. Oh, H., Vinh, P.V.: Design and implementation of a MAC protocol for timely and reliable delivery of command and data in dynamic wireless sensor networks. Sensors **13**, 13228–13257 (2013)
2. Song, W.-Z., Huang, R., Shirazi, B., LaHusen, R.: TreeMAC: localized TDMA MAC protocol for real-time high-data-rate sensor networks. In: IEEE International Conference on Pervasive Computing and Communications, pp. 1–10. IEEE, Texas (2009)
3. Rhee, I., Warrier, A., Aia, M., Min, J., Sichitiu, M.L.: Z-MAC: a hybrid MAC for wireless sensor networks. IEEE/ACM Trans. Netw. **16**, 511–524 (2008)
4. Rhee, I.,Warrier, A., Min, J., Xu, L.: DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks. In: 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 190–201. ACM New York, Florence (2006)
5. Petersen, S., Carlsen, S.: WirelessHARTversus ISA100.11a: the format war hits the factory floor. IEEE Ind. Electron. Mag. **5**, 23–34 (2011)
6. Suriyachai, P., Brown, J., Roedig, U.: Time-critical data delivery in wireless sensor networks. In: 6th IEEE International Conference on Distributed Computing in Sensor Systems, pp. 216–229. Springer, Berlin, Santa Barbara (2010)
7. Cheng, M.X., Gong, X., Xu, Y., Cai, L.: Link activity scheduling for minimum end-to-end latency in multihop wireless sensor network. In: IEEE Global Telecommunications Conference, pp. 1–5, IEEE, Houston (2011)
8. Le, H., Eck, J.V., Takizawa, M.: An efficient hybrid medium access control technique for digital ecosystems. IEEE Trans. Industr. Electron. **60**, 1070–1076 (2013)
9. Silvo, J., Eriksson, L.M., Bjorkbom, M., Nethi, S.: Ultra-reliable and real-time communication in local wireless applications. In: 39th Annual Conference of the IEEE Industrial Electronics Society, pp. 5611–5616. IEEE, Vienna (2013)
10. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: 2nd International Conference on Embedded Networked Sensor Systems, pp. 95–107. ACM New York, Baltimore (2004)
11. Ngo, C.T., Oh, H.: A tree-based mobility management using message aggregation based on a skewed wait time assignment in infrastructure based MANETs. Wireless Netw. **20**, 537–552 (2014)
12. IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-rate wireless personal area networks (LR-WPANs). IEEE Std. 802.15.4-2011 (2011). http://standards.ieee.org/findstds/standard/802.15.4-2011.html
13. Bettstetter, C., Hartenstein, H., Perez-Costa, X.: Stocasticproprties of the random waypoint moility model. Wireless Netw. **10**, 555–567 (2004)
14. Dung, L.T., An, B.: A modeling framework for supporting and evaluating performance of multi-hop paths in mobile Ad-Hoc wireless networks. Comput. Math Appl. **64**, 1197–1205 (2012)

http://www.springer.com/978-81-322-2579-9

Wireless Communications, Networking and Applications
Proceedings of WCNA 2014
Zeng, Q.-A. (Ed.)
2016, XXI, 1310 p. 656 illus. In 2 volumes, not available separately., Hardcover
ISBN: 978-81-322-2579-9