Speed Improvement of Centralized Scheduling Algorithm on IEEE 802.15.4e TSCH Netwok Using Heuristic Method

Iman H. Santoso[1,2] and Kalamullah Ramli[1]
[1] Dept. of Electrical Engineering, Universitas Indonesia, Depok 16424, Indonesia
[2] School of Electrical Engineering, Telkom University, Bandung 40257, Indonesia
Email: iman.hedi@ui.ac.id; k.ramli@eng.ui.ac.id

Abstract —Two of the IoT challenges are many nodes that active simultaneously and limited link bandwidth. The challenges can be overcome by using a scheduling algorithm for channel allocation. A known centralized scheduling algorithms that mentioned in standard is Traffic Aware Scheduling Algorithm (TASA). TASA is a scheduling algorithm devoted to the IEEE 802.1.4e TSCH network. With reference to TASA, this research proposes a scheduling algorithm, named as IR-TASA, which can improve the performance of TASA in terms of speed to create time slot schedule for IEEE802.15.4e network. The results show that IRTASA increases the speed up to 5 times as compared to TASA, with a relatively equal duty cycle.

Index Terms—IEEE802.15.4e, IoT, Scheduling Algorithm, Matching Procedure, Time slot

I. Introduction

Over the last decade, IoT has attracted academic and industry attention in connection with its new and exciting concept of communication, i.e. it will allow things to intercommunicate with minimum human intervention. Some challenges that are key to the implementation of IoT are: better battery life, network resource allocation, heterogeneous terminal devices, scability, and security [2]. A wireless technology that can be a candidate for IoT is IEEE 802.15.4, which is also the basis for ZigBee technology [3]. Fig. 1 shows a comparison between various wireless technology. It shows that the superiority of IEEE802.15.4 over IEEE802.11 is its much lower power consumption, whereas for IEEE802.15.4 over IEEE802.15.1 (Bluetooth) is its higher range.

In relation to the problem of multiple nodes that active simultaneously and competing for gaining a limited channel resources, one way to solve that problem is by using a scheduling mechanism at datalink layer. As for the problems associated with battery life, the standard body has provided solutions with the launch of Time Synchronized Channel Hopping (TSCH) protocol which is part of the IEEE802.15.4e standard. TSCH protocol is the latest generation of MAC layer protocol that is highly reliable and low power [4]. Therefore, this research focuses on scheduling algorithm in datalink layer for IEEE802.15.4e network, as shown in Fig. 2.

Technology	Theoretical bit rate	Range	Power Consumption
IEEE 802.11b	1, 2, 5.5, and 11 Mbit/s	25-100 m (indoor) 100-500 m (outdoor)	~30 mW
IEEE 802.11g	Up to 54 Mbit/s	25-50 m (indoor)	~79 mW
Bluetooth (IEEE 802.15.1)	1 Mbit/s (v1.1)	10 m (up to 100 m)	1 mW (up to 100 mW)
UWB (IEEE 802.15.3)	110-480 Mbit/s	~10 m	100 mW, 250 mW
IEEE 802.15.4 (e.g. ZigBee)	20, 40 or 250 kbit/s	10-100 m	1 mW
HomeRF	1 Mbit/s (v1.0) 10 Mbit/s (v2.0)	~50 m	100 mW
IEEE 802.16e	up to 15 Mbit/s	2-5 km	Complex power control

Fig. 1. A comparison between various wireless technologies [5]

This research proposes a centralized scheduling algorithm which has a higher speed than TASA in generating time slot schedule for IEEE802.15.4e nodes network. The proposed scheduling algorithm is named as Iman Ramli-TASA (IR-TASA). IR-TASA uses a modified vesion of TASA matching procedure, so this paper discusses more about matching algorithm. In this research, modifications to the TASA matching procedure is performed heuristically.

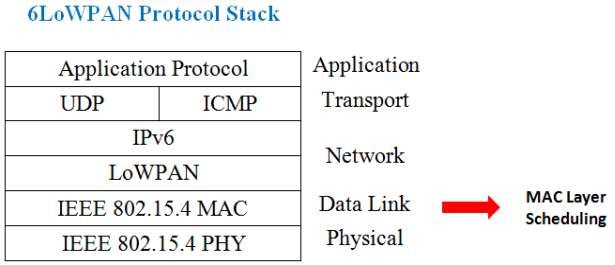


Fig. 2. Protocol stack of Internet of Things [6]

This paper is arranged as follows. Section II reviews some papers related to IoT network scheduling that refer to TASA. In Section III, the TASA algorithm is described in summarized form to make it easy to compare with IR-

Manuscript received August 5, 2017; revised December 12, 2017.
This work was supported by Universitas Indonesia under International Publication Grant for Student Final Project.
Corresponding author email: iman.hedi@ui.ac.id.
doi:10.12720/jcm.12.12.661-667

TASA. Section IV explains the proposed IR-TASA algorithm. The similarities and differences between IR-TASA and TASA is exposed in this Section. Results are discussed in Section V. Section VI concludes the paper

II. Related Works

There are three works [7]-[9] which are related to the TASA scheduling algorithm [4], [10], [11]. Those three researchers are trying to improve the performance of TASA by making improvements based on the findings in TASA that are considered a deficiency.

Farias and Dujovne [7] developed PCE-enabled scheduling algorithm. The differences between that with TASA are: scheduling decision is based on traffic request from nodes, and it distributes time slots evenly between nodes. Even so, how the scheduling algorithm work is not explained.

Min *et al*. [8] focused on handling heterogeneous traffic in TSCH network. To handle such heterogeneous traffic, this research proposes a scheduling algorithm that uses multiple slotframes at the communication sessions of its nodes. The scheduling algorithms uses different slotframe lengths depending on the type of traffic being served. That algorithm was tested in experimental simulation.

Gaillard *et al*. [9] added TASA a mechanism that takes into account Packet Delivery Ratio (PDR) when making schedule decisions. The way to increase the PDR is to allocate additional cells for frame retransmission purposes.

To the best of our knowledge, this study is the first that explores the possibility to improve the speed of generating scheduling decision on IEEE802.15.4e network. The term scheduling decision is a term in the scheduling algorithm that desribed in [12], which means the decision taken by scheduling algorithm for each node, at which time slot a node should transmit, receive, or sleep.

The rate of speed in deciding the schedule is an important parameter to be considered based on the following aspects:

- The faster the decision is made, the faster the nodes in a network can transmit the data, thus minimizing the delay.
- The more quickly schedule for a network decided by master node the more networks can be served.
- The more networks that can be served, the greater the area that can be reached by master node so it increases the efficiency and effectiveness of master node.

III. Traffic Aware Scheduling Algorithm (TASA)

IEEE802.15.4e standard specified how MAC layer executes a schedule, but does not specify how to create the schedule. Therefore, Palatella *et al*. [4], [10], [11] has proposed TASA, a centralized scheduling algorithm for Internet of Things. TASA is one of several alternatives of scheduling algorithms that could be used on IoT according to RFC 7554 [13]. The following is a summary of TASA that has been proposed in [4], [5], and [8]:

A. Network Model

In TASA, the network is modeled in a tree topology represented by a directed G = (V,E), where V = {n_0, n_1, …, n_{N-1}} is a set of device, and |V| = N, number of nodes in the network. n_0 is master node, or PAN coordinator, and n_i, with $1 \le i \le N - 1$, is the i-th generic in the network.

Fig. 3 shows graph G for a network. Each n_i node has: i) parent node, p_i ; ii) a set of child node, ch(n_i); and iii) sub-tree ST(n_i). Each n_i node is connected to its parent p_i through a dedicated link, $(n_i, p_i) \in E \subset V \times V$.

B. Network Data Traffic

In this research each n_i node in graph G, except the PAN coordinator (n_0), generates a number of packet contantly, $\tilde{q}_i$, in a similar slotframe of size S slot. Each n_i node forwards the data packets to its parent (p_i), and finally, all packets move toward PAN coordinator.

Related to $\tilde{q}_i$, $\tilde{Q}$ parameter is defined, where $\tilde{Q}$ is the total number of packets sent to the PAN coordinator (n_0) in one slotframe. The formula of $\tilde{Q}$ stated in [4], [10], [11], is showed in equation (1) below:

$$\tilde{Q} = \sum_{i=1}^{N-1} \tilde{q}_i \tag{1}$$

Taking into account ST(n_i) sub-tree, Palatella [4], [10], [11] defines the global queue level of a node in slot k, $Q_i(k)$, as the total number of packets that are queueing in nodes of that sub-tree in slot k. The formula contained in [4], [10], [11] is showed in (2):

$$Q_i(k) = \sum_{j|n_j \in ST(n_i)} q_j(k) \tag{2}$$

C. Definition of Conflict

It is important to note that each node cannot transmit and receive at the same time. And also, each node cannot receive data from multiple nodes simultaneously. To illustrate this concept, a collection of links that are duplex-conflict, DCi, for each node n_i is defined.

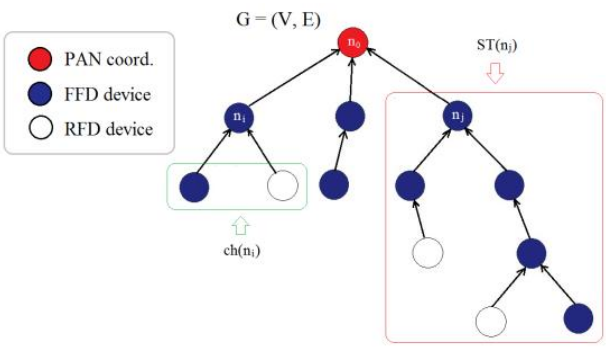


Fig. 3. Graf G = (V, E) modeled the Network [4]

DCi consists of a set of edges / links between pairs of nodes in G that cannot transmit at the same time slot because these links are connected to same active node and only one link can be used for transmit/receive. Only duplex-conflict free links can be scheduled in the same slot in one slotframe. A set of duplex-conflict free links for k slot is termed as DCFL(k).

This reseach defines links as interference-conflict links if they interfere each other when placed on the same offset channel. In this regard ICFLc(k) parameter is defined as a set of interference-conflict-free links that can use the same offset channel, i.e., offset channel c, at time slot k.

D. Scheduling Algorithm in Brief

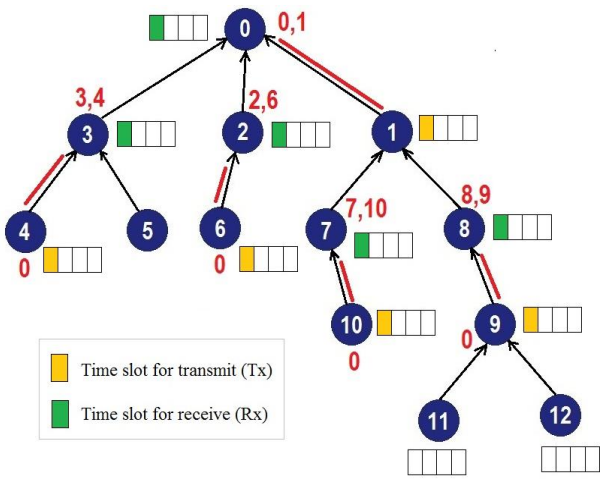
PAN coordinator is assumed to know the graph G, graph of physical connectivity P, and the traffic load generated by each node, i. e., $\tilde{q}_i, \forall n_i \in V$, with i≠0. From the information provided by graphs, master node, n_0, builds the schedule by running TASA procedure. The schedule information is then forwarded to all nodes so that each node knows the transmit/receive timetable, and on which channel it may transmit/receive.

TASA consists of two main procedures: i) matching and ii) coloring; which is implemented iteratively on graph G. A complete explanation of matching and coloring procedures is in the graph theory which can be seen in [14]. Matching procedure selects, step by step, a collection of DCFL(k) links that can be scheduled for each time slot. DCFL(k) is obtained by selecting n_i between the children of a parent node (p_i), based on following formula [4], [10], [11]:

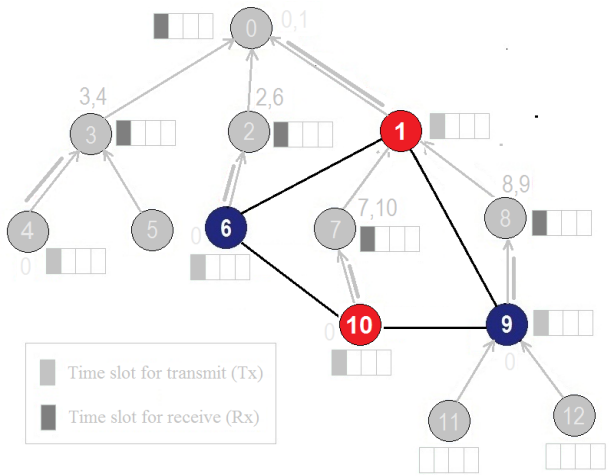
$$Q_i(k) = max\left\{Q_j(k) \mid n_j \in ch(p_i) \wedge q_j(k) \neq 0\right\} \quad (3)$$

Equation (3) states that slot k is only provided to a link (n_i, p_i) when child node, n_i, has at least one packet to be transmitted to its parent, p_i, during that slot. In this way, node power could be saved because a node sleeps when it has no schedule to transmit or receive. While in coloring procedure, TASA applies coloring process on interference graph, I(k). The adjacent nodes in I(k) are the nodes that will mutually interfere if active, so those nodes should use different offset channel. Fig. 4 shows the visualization if the matching and coloring procedure are applied to a network. The matching procedure will be more discussed in the next section because in this procedure there lies the difference between IR-TASA and TASA and makes IR-TASA faster in terms of generating time slot schedules for nodes in the network.

To meet stringent requirements of packet delay and power consumption, TASA scheduling algorithm seeks minimum duty cycle for each node. To send network traffic of $\tilde{Q}$ to PAN coordinator TASA schedules all transmissions to be performed on the first slot λ in the slotframe, where λ ≤ S, so that the rest of slots as S-λ are left empty. Thus, the duty cycle obtained is λ/S. Palattella et al. state that the minimum number of active slots (λ) in one slotframe is $\tilde{Q}$ [11].



(a) DCFL(k)



(b) I(k)

Fig. 4. Matching and Coloring in TASA [9]

IV. IR-TASA Scheduling Algorithm

IR-TASA scheduling algorithm uses the principle of network model as used in TASA. Meanwhile, in the process of establishing a schedule, IR-TASA algorithm does not account for queue condition and children number of each node. IR-TASA arranges for each child to be given the same turn to send its data to its parent node. However, IR-TASA same as TASA, does not give schedule on empty nodes, so in that way packet delay will decrease, througput will increase, and node energy will not be wasted.

The IR-TASA algorithm proposed in this research can be seen in Fig. 5. The flow chart in that figure, summarizes the main steps of IR-TASA, running on the master node. It is shown in the picture that IR-TASA concentrates on matching procedure, so the coloring does not appear in the flowchart.

In Fig. 5 presented 4 new variable, namely: ND, N_i, s, and TS, here is an explanation for those varible:

- ND : Total number of nodes in the network.
- Ni : Number of children for node i.
- s : Number of iterations

- TS(s): Maximum number of active time slots required in each round (s).

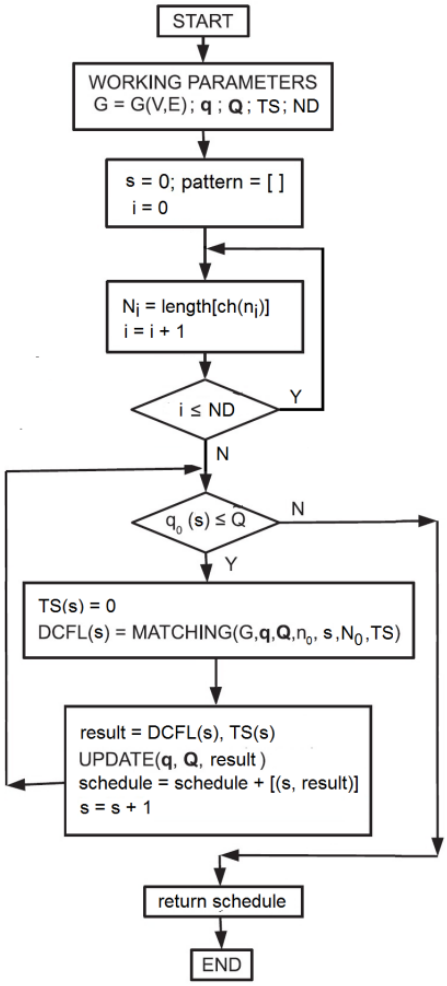


Fig. 5. The IR-TASA algorithm.

As described in section III.C, TASA known two types of conflicts, namely: Duplex Conflict (DC) and Interference Conflict (IC). DC can be solved by matching method while the IC by coloring. Since IR-TASA concentrates on time slot scheduling, so only the DC problem will be explored. Thus, IR-TASA assumes that all nodes scheduled to transmit/ receive are interference-free, it is because there is no interference or the interference issues have been settled. In this research, if there is a speed comparison between TASA and IR-TASA then the speed is viewed from how quickly DCFL can be formed on both algorithms.

In IR-TASA, to mark the iteration process is used variable s. This variable has no relation to time slot number, this is what distinguishes IR-TASA from TASA. TS (s) is the maximum number of time slots required by a node for transmitting/receiving in each round (s). TS (s) at IR-TASA has a relation to λ at TASA, that relationship expressed by (4) below:

$$\lambda = \sum_{s=1}^{Max} TS(s) \tag{4}$$

In this paper, λ is called by the name of active time slot. The Max-th round of TS(s) is the last round that indicates the entire data queuing up at all nodes has reached the master node. The variable **result** is used to hold the output of each round, this variable contains the DCFL (s) and TS (s). The entire data in **result** are stored in a variable named **schedule**. So when the iteration ends, **schedule** will contain two variables for all round, ie: the nodes that are given turn and maximum number of time slots.

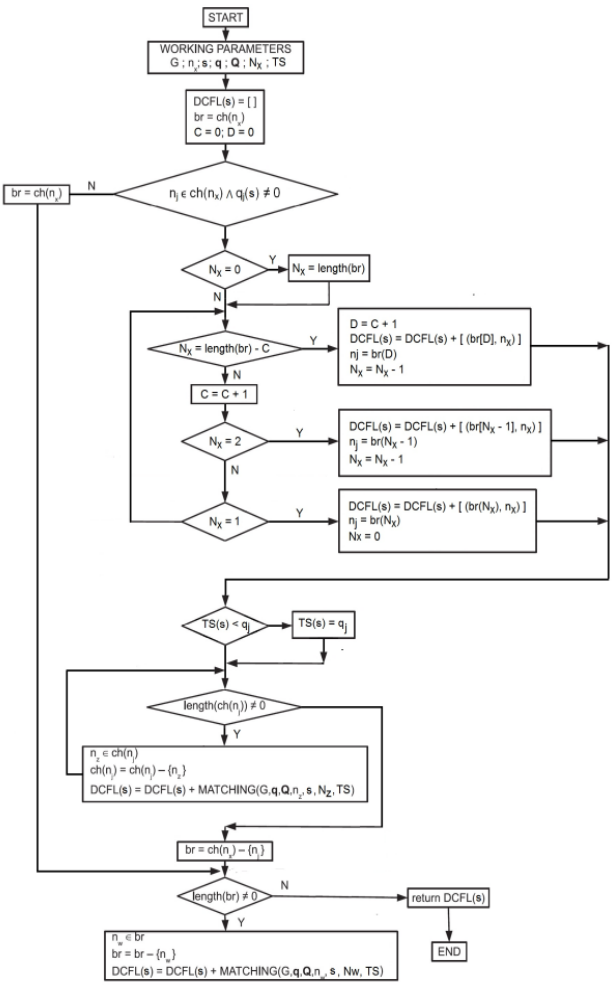


Fig. 6. Matching procedure in IR-TASA.

Fig. 6 shows the matching procedure algorithm that is used in IR-TASA. The most visible difference between IR-TASA's and TASA's matching procedure is on the way of each algorithm in deciding which node will be selected to be active. In IR-TASA the decision is only made based on turn between nodes provided it contains at least 1 packet data. While in TASA, the scheduling decision will be given based on node's queue size and number of children. In both matching procedure there are variable **br** (brother) that represent the set of child nodes of a given node. So, the equation , br = ch (n_x), as shown in Fig. 6 means br is a set of children of node n_x. The output of matching procedure on IR-TASA is DCFL (s) and TS (s).

Here is Tabel I that show comparison between TASA and IR-TASA:

TABLE I. COMPARISON BETWEEN TASA AND IR-TASA

Item	TASA	IR-TASA
Scheduling scheme	Performed for each slot	Performed for block of time slots
Relation between number of iterations and number of active time slots	Number of iterations = number of timeslots	Number of iterations < number of timeslots
Relation between number of iterations and number of nodes	Number of iterations = number of nodes	Number of iterations < number of nodes
Relation between number of active time slots and number of nodes	Number of time slots = number of nodes	Number of time slots ~ number of nodes
Sequence of channel allocation	From a node with a larger ranking	Idem
Considering node's queue size	Yes (prioritizing node with large queue size/ more children)	No
Strategies for saving energy	Does not provide schedule to nodes whose queue is empty	Idem

V. RESULTS AND DISCUSSION

To explain how IR-TASA work, thus making it differ with TASA, the following is an example of applying IR-TASA scheduling algorithm to a small network of 13 nodes (1 master node with 12 children), as shown ini Fig. 7. This experiment deliberately choose a small network, because large networks will need more space for itsexplanation. The assumptions used are as follows:

- In each slotframe period, every node has only 1 packet in queue. The appearance of packets is periodic, similar to slotframe period.
- All packets that are queued in 12 child nodes must be sent to the master node in a slotframe period.
- No packet loss.

Based on the TASA algorithm, as well as IR-TASA, the master node will inform the schedule to all of child nodes about when each node can transmit or receive the data.

In TASA, for delivering all data to master node, n_0, takes minimum of 12 scheduling step, while at each step the node may transmit, receive, or sleep. When a node gets a transmit schedule then 1 packet at that node moves to its parent, the packet one step closer to the master.

Based on the assumptions above, that is, at each node queuing 1 packet, then the total packet waiting to be sent to master node is 12 packets. Meaning that minimum 12 rounds is required to transmit all the packet to master node. The more packet queueing, the more round required. Thus, if the number of packets in nodes increases then the number of iterations in scheduling will increases linearly.

It can be concluded that, in TASA, the more nodes in the network, the more time it takes to make scheduling decisions. If maximum number of timeslots in 1 slotframe is 1000 slots, with maximum active slot of 50% and 1ts = 10ms, as mentioned in the standard [13], then the maximum number of nodes that can be served is 500 nodes. The time it takes to decide the scheduling for one slotframe is 500 * (time needed for one scheduling iterations). Certainly on high-speed networks, or networks with strict delay requirements, faster scheduling algorithms is preferred.

This research proposes an IR-TASA scheduling algorithm that can reduce the time required by the master node to create a "scheduling decision", even though there is little trade-off between the time to generate the schedule and the number of time slots in a slotframe. Compared to TASA, the IR-TASA algorithm can increase the speed to create schedule from 300% to 500% with number of timeslot increment only form -2% to 5%.

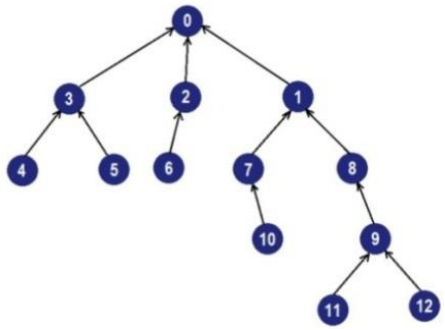


Fig. 7. Network topology of 13 nodes

Based on the network topology in Fig. 7, Table II show the schedule information that sent by the master node to all nodes based on IR-TASA scheduling scheme:

TABLE II. SCHEDULING INFORMATION IN IR-TASA

Node0: Tx,ts: - Rx,ts: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14	**Node5**: Tx,ts: 2 Rx,ts: -	**Node10**: Tx,ts: 1 Rx,ts: -
Node1: Tx,ts: 1, 7, 8, 9, 10, 13, 14 Rx,ts: 2, 3, 4, 5, 11, 12	**Node6**: Tx,ts: 1 Rx,ts: -	**Node11**: Tx,ts: 2 Rx,ts: -
Node2: Tx,ts: 2,3 Rx,ts: 1	**Node7**: Tx,ts: 2,3 Rx,ts: 1	**Node12**: Tx,ts: 4 Rx,ts: -
Node3: Tx,ts: 4,5,6 Rx,ts: 1,2	**Node8**: Tx,ts: 4, 5, 11, 12 Rx,ts: 1, 7, 8	
Node4: Tx,ts: 1 Rx,ts: -	**Node9**: Tx,ts: 1, 7, 8 Rx,ts: 2, 4	

Fig. 8 shows that the IR-TASA scheduling algorithm can make scheduling decisions faster than TASA, from which initially minimum of 12 steps to just 6 steps. In Fig. 8 it appears that to transmit data from all nodes to the master only requires 6 rounds with the required timelot is as many as 34, only increase 1 time slot compared to TASA. Thus, it has been shown that the speed of release of the scheduling decision has increased by 550% with an increase in the number of time slot by 3% only.

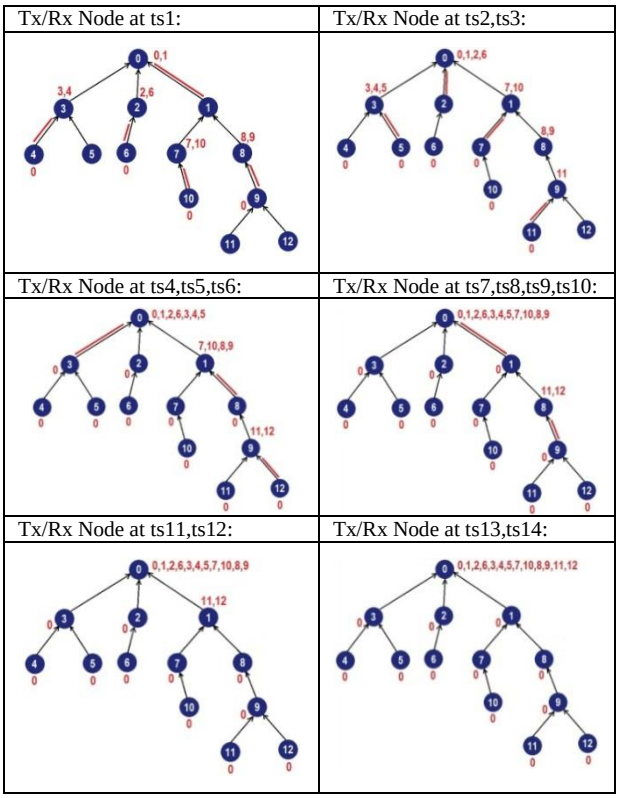


Fig. 8. Number of iterations in IR-TASA

As explained in previous sections, TASA algorithm performs link scheduling at the same frequency as the time slot repeat. Tabel I shows that the total number of active time slots in a slot frame is equal to or greater than the total amount of data on all nodes in the network. Therefore, a scheduling decision for one slotframe can only be generated after the iteration process as much as the number of active timeslots. Thereby, if number of nodes in the network increases so the time required to generate scheduling decision becomes longer.

Fig. 9 shows the comparison of the number of iterations required by each algorithm to convey the data of all nodes to its master (n_0) in one slotframe. In TASA, number of iterations increases linearly with the increasing number of network nodes. While in IR-TASA, for the number of nodes ranging from 10 to 100, the increase in the number of iterations is relatively small. As can be seen in Fig. 10, which is a graph of speed ratio (V), that graph V continues to increase along with the increasing number of nodes in the network. It shows that I_{TASA} increases linearly to the number of nodes, whereas in $I_{IR-TASA}$ the increment is relatively small. The speed ratio (V) to be meant by Fig. 10 is:

$$V = \frac{I_{TASA}}{I_{IR\text{-}TASA}} \tag{5}$$

where I_{TASA} dan $I_{IR\text{-}TASA}$ is the number of iterations in TASA and IR-TASA . With the considerable increase of speed in generating scheduling decision, the resource of master node can be used to serve 4 to 5 times more networks.

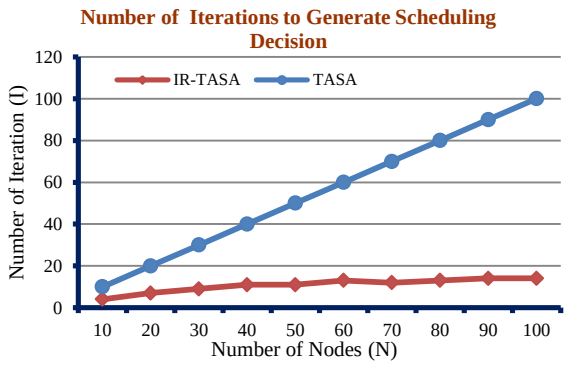


Figure 9. Number of iterations for IR-TASA and TASA

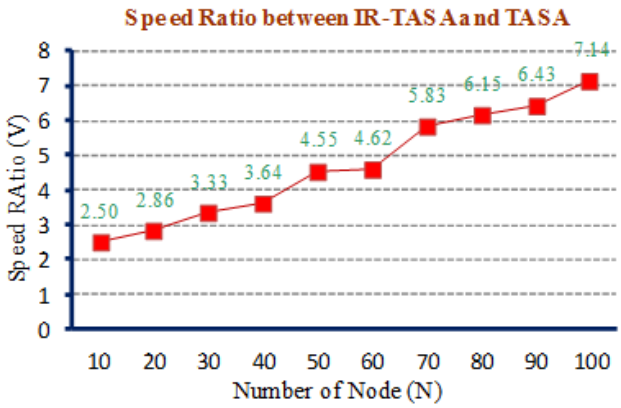


Fig. 10. Increased speed of scheduling decision

Fig. 11 shows the number of active time slot (T) in a slotframe for TASA and IR-TASA. It can be seen that T value for TASA and IR-TASA is almost the same.

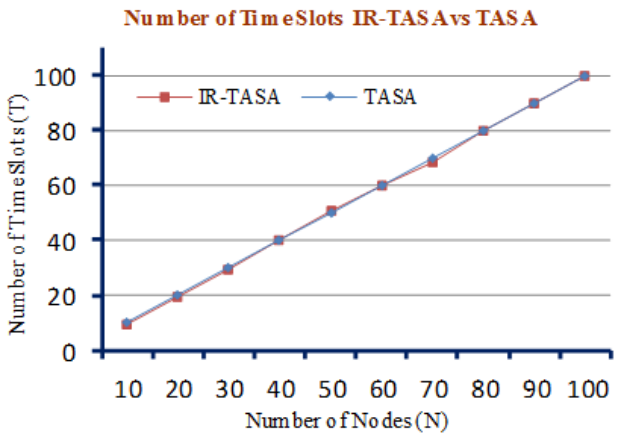


Fig. 11. Number of active timeslot in one slotframe for IR-TASA and TASA

Fig. 12 shows more clearly the difference values between T_{TASA} and $T_{IR\text{-}TASA}$, where the difference is expressed in the form of subtraction between T_{TASA} and $T_{IR\text{-}TASA}$ (D). The graph in Fig. 12 shows that the D value ranges from -2 to 2. It shows that IR-TASA algorithm may also generate smaller number of active time slot than TASA. It is something very expected if the number of active time slots that are almost the same, even less, can be obtained with a much faster scheduling mechanism

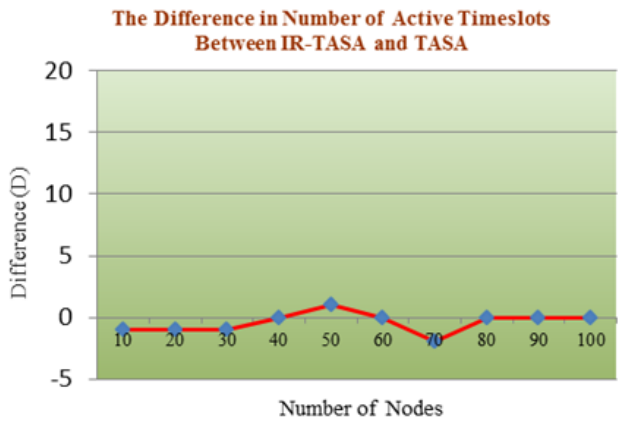


Fig. 12. The difference in number of active timeslots

VI. Conclusion

In this research it has been shown that the proposed scheduling algorithm, IR-TASA, has a higher speed in generating the schedule. The results show that IR-TASA increase the speed to generate scheduling decision up to 5 times compared to TASA, with the number of active time slots are almost the same. The increase in speed is obtained by modifying the TASA through heuristic approach. TASA performs a scheduling mechanism for and at time slot period, whereas in IR-TASA the period of scheduling algorithm and timeslot running independently. In further research, theoretical foundations of IR-TASA will be developed. The performance of IR-TASA algorithm will also be tested in a simulated network with various patterns.

Acknowledgment

The authors wish to thank Universitas Indonesia. This work was supported by a grant from Universitas Indonesia, in the framework of International Publication Grant for Student Final Project.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, 2013.

[2] Y. Chen, *et al*., "Time-reversal wireless paradigm for green internet of things: An overview," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 81-98, 2014.

[3] M. R. Palattella, *et al*., "Standardized protocol stack for the internet of (important) things," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 1389-1406, 2013.

[4] M. R. Palattella, *et al*., "Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15. 4e networks," in *Proc. 23rd International Symposium on Personal Indoor and Mobile Radio Communications*, 2012.

[5] J. Hoebeke, *et al*., "An overview of mobile ad hoc networks: applications and challenges," *Journal-Communications Network*, vol. 3, no. 3, pp. 60-66, 2004.

[6] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*, vol. 43, John Wiley & Sons, 2011.

[7] A. A. Far ás and D. Dujovne, "A queue-based scheduling algorithm for PCE-enabled industrial internet of things networks," in *Proc. Sixth Argentine Conference on Embedded Systems*, 2015.

[8] M. Min, *et al*., "Traffic aware multiple slotframes scheduling algorithm in industrial IoT applications using IEEE802. 15.4 e TSCH," in *Proc. 16th International Conference on Communication Technology*, 2015.

[9] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois. "Enabling flow-level reliability on FTDMA schedules with efficient hop-by-hop over-provisioning," PhD dissertation, INRIA Grenoble-Rh ône-Alpes, 2016.

[10] M. P. Palattella, *et al*., "Traffic-aware time-critical scheduling in heavily duty-cycled IEEE 802.15. 4e for an industrial IoT," *Proc. of IEEE Sensors*, pp. 1-4, 2012.

[11] M. P. Palattella, *et al*., "On optimal scheduling in duty-cycled industrial IoT applications using IEEE802. 15.4 e TSCH," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3655-3666, 2013.

[12] A. Sayenko, "Adaptive scheduling for the QoS supported networks," PhD thesis, University of Jyv äskyl ä, Finland, 2005.

[13] T. Watteyne, M. R. Palattella, and L. A. Grieco, "Using ieee 802.15. 4e time-slotted channel hopping (TSCH) in the Internet of Things (IoT): Problem statement," No. RFC 7554. 2015.

[14] R. Diestel, "Graph Theory, Electronic," Graduate Texts in Mathematics 173, 2005.

Iman H. Santoso was born in West Java Province, Indonesia. He received the B.Eng. degree from Institut Teknologi Bandung (ITB), Bandung, in 1998 and the M.Eng. degree from the same institution, in 2007, both in electrical engineering. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Universitas Indonesia. His research interests include data communication network, teletraffic engineering, and IoT network protocol and traffic.

Kalamullah Ramli is a Professor in Computer Engineering at the Department of Electrical Engineering, Universitas Indonesia. Obtained his Ph.D at Universitaet Duisburg-Essen, Germany, he currently supervises 10 (ten) Ph.D students at the Center For Advanced and Integrated Research on Networked Embedded System and Security (Center FAIRNESS). His research interests include embedded system, data and network security, big data analysis, and IoT network management and protocol.