JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 697-708 (2021)
DOI: 10.6688/JISE.202105_37(3).0012

Dynamic Slot Allocation in Restricted Access Window for IEEE 802.11ah Networks

YAN-JING WU[1], TSANG-LING SHEU[2,+] AND PEI-HSUN CHAN[3]
[1]*Department of Information Technology and Communication*
Shih Chien University
Kaohsiung Campus, Kaohsiung, 845 Taiwan
[2,3]*Department of Electrical Engineering*
National Sun Yat-sen University
Kaohsiung, 804 Taiwan

The IEEE 802.11ah standard has been released to support machine-to-machine (M2M) communications among many sensor devices, which is rapidly increasing with the Internet of Things (IoT). The medium access control of 802.11ah introduces a restricted access window (RAW) into each beacon interval (BI). The RAW is further divided into many slots that allow only sensor devices to contend for transmission. This paper proposes a dynamic slot allocation scheme (DSAS) in the RAW of 802.11ah networks. To alleviate the contention in crowded M2M environments, the DSAS defines two sets of thresholds, high and low, to determine the sleeping-time level of a machine and the contention level for a slot, respectively. The innovative aspect of the DSAS is that machines with short sleeping time are allocated to less congested slots whereas machines with long sleeping time are allocated to more congested slots. To avoid unnecessary slot reallocation, machines with sleeping-time levels or contention levels between the high and low thresholds continue to use their original transmission contention slots. Simulations performed on ns-3 show our proposed DSAS has significantly improved average backoff time and packet delay and increased the overall system throughput.

Keywords: 802.11ah, RAW, M2M communication, slot allocation, sleeping time

1. INTRODUCTION

Recently, the advancement in wireless communication technology has led to a tremendous growth in the number of wireless devices. Coincident with the automation of machines and the advent of various sensor devices, machine-to-machine (M2M) communications, also known as machine-type communications (MTC), have emerged as the next-generation communication system [1], in which many machines can intelligently communicate with one another with or without human intervention. Nowadays, M2M applications, such as smart meters, wearable devices, temperature control, remote medical care, environmental monitoring, smart farms, home care, and smart cities, are extensively deployed [2, 3].

Various protocols, such as 3GPP's LTE-M [4] and IEEE 802.11ah [5], were developed and released as standards to support M2M communications. IEEE 802.11ah is a wireless networking protocol, named as Wi-Fi HaLow [6], published in 2016 as an amendment of the IEEE 802.11-2007 wireless networking standards. It uses 900 MHz license exempt bands to provide extended range of Wi-Fi networks, as compared to conventional Wi-Fi

Received January 21, 2019; accepted April 3, 2019.
Communicated by Ren-Hung Hwang.
[+] Corresponding author: sheu@ee.nsysu.edu.tw

networks operating in the 2.4 GHz and 5 GHz bands. Further, it benefits from lower energy consumption, allows the creation of large groups of stations or sensors that cooperate to share a wireless medium, and supports the concept of IoT. To alleviate the severe contention caused by a large number of sensor devices in M2M environments, 802.11ah introduces a restricted access window (RAW) into each beacon interval (BI). As illustrated in Fig. 1, 802.11ah divides a BI into a contention period (CP) and a contention-free period (CFP) [5]. The CFP is just the RAW. The RAW is further divided into many time slots, each of which comprises an RTS/CTS period (RCP) and a data transmission period (DTP). During the RCP, sensor devices are allowed to contend for transmission opportunities in the DTP by following the distributed coordination function (DCF). For the other fragment in a BI, *i.e.*, the CP, any devices are allowed to contend for access to the medium. More detailed, Fig. 2 shows the frame format of a beacon. Compared to 802.11, 80.2.11ah adds three fields (in yellow color) into a beacon, namely Change Sequence (CS), Time of Next Beacon, and Time of Next Beacon Flag.

(1) Change Sequence (CS): The length is one byte. Each time the AP sends a new beacon message, the value of CS is incremented by one.
(2) Time of Next Beacon: The length is three bytes for recording the time when to send the next beacon.
(3) Time of Next Beacon Flag: The length is only one bit, used to indicate whether the field of Time of Next Beacon is added into the Beacon Header.

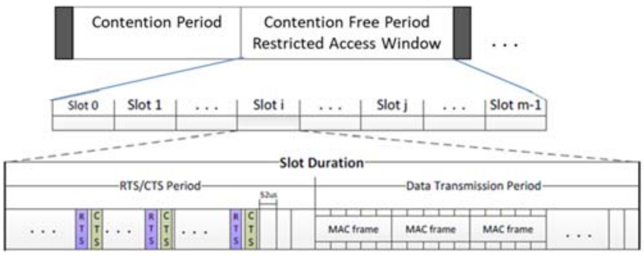


Fig. 1. BI resolution of 802.11ah networks.

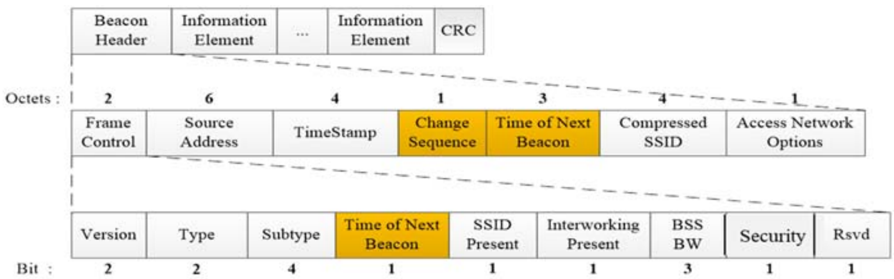


Fig. 2. Beacon format in 802.11ah networks.

Without considering the different sleeping times of various sensor devices, 802.11ah simply adopts a residual method to allocate a group of machines to each slot of the RAW. As a result, when devices wake up, they are often allocated to the same slot, leading to congestion. Conversely, devices that seldom wake up may be allocated to the same slot so that the channel remains idle for most of the slot duration. For instance, a hypermarket in which multiple types of machines are simultaneously present. In a hypermarket, there are monitors, large refrigerator temperature controllers, radio frequency ID (RFID) readers, and tags. These machines transmit large amounts of data, ranging from a few Mbps to a few Kbps. However, the same machines may have different sleeping times; for example, if a machine is used in a reservoir, the same machine can also be used to monitor water quality and quantity. Machines monitoring water quality usually have longer sleeping time since water quality changes intermittently. In contrast, machines monitoring the amount of water during downpours need to frequently report their recorded data so that they can send a warning when the amount of water exceeds a certain safe value, and will therefore have shorter sleeping time.

Recently, several works have been presented to improve the utilization of the scarce radio resource by modifying the slot allocation mechanism of 802.11ah in M2M environments [7-17]. As elaborated later in Section 2, the previously-proposed works include grouping strategies [7-9], adjusting the size of RAW in every BI [10], shifting the wake-up time of each machine [11, 12], dividing all machines into two classes, alarm-purposed and periodic [13, 14], with high priority given to the alarm-purposed, and focusing on extending the network lifetime by building an analytical model [15-17] to estimate the power consumption of each machine. Aiming to alleviate the contention among numerous machines in each slot of RAW, this paper proposes a dynamic slot allocation scheme (DSAS) that measures each machine's backoff time and sleeping time to solve the congestion problem encountered in M2M environments. The key design of the proposed DSAS is that machines with short sleeping time are allocated to less congested slots and those with long sleeping time are allocated to more congested slots.

The remainder of this paper is organized as follows. The related works of our scheme are discussed in Section 2, the proposed DSAS algorithm is explained in Section 3, the simulation results and discussions are presented in Section 4, and the concluding remarks are given in Section 5.

2. RELATED WORKS

With 802.11ah standard, every waking sensor device is assigned a slot j, which is obtained from Eq. (1), to contend for medium access by transmitting an RTS packet during the RCP. If no collision occurs, the associated access point (AP) will reply with a CTS packet piggybacking with an indication of when to start data transmission during the DTP. If a sensor device fails to receive its corresponding CTS packet before the end of the RCP, it will wake up again in the next BI to contend for medium access. In Eq. (1), x is the association ID (AID) of a sensor device, M is the total number of slots in the RAW, and N_{offset} is a shifting integer, used to avoid having the same slot allocated to a sensor device in every BI.

$$j = (x + N_{offset}) \bmod M \tag{1}$$

However, an extremely large number of machines contend for access to the radio channel, compared with traditional wireless networks. The medium access control (MAC) design of 802.11ah still encounters serious contention issues during RAW. To address this issue, several works have been presented to improve the utilization of the scarce radio resource by modifying slot allocation mechanism of 802.11ah in M2M environments. Some previously proposed approaches have grouped all sensor devices into two modes, contending and sleeping [7-9]. While aiming to save power and spread the contending nodes among all slots in the RAW, some of the slots may be idle since many machines have long sleeping time during a single slot. Similarly, the grouping strategies with Quality of Service (QoS) presented in some works [13, 14] have divided all sensor devices into two classes, alarm-purposed and periodic, and have reserved a set of slots for the periodic traffic. Moreover, the alarm-purposed traffic has a higher priority over the periodic traffic in case of an emergency. Park *et al.* [10] intended to solve severe contention among machines by dynamically adjusting the length of a slot during the RAW based on the estimated number of contending machines. However, increasing the length of a slot may result in idle slots because a small number of machines are assigned to them.

Instead of modifying slot assignment algorithm of 802.11ah, Liu *et al.* [11, 12] have developed an analytical model to obtain the Offset ListenInterval (OLi) of each machine such that the wake-up times of the associated machines are varied and then, all the M2M traffic is spread evenly to prolong the network lifetime. Note that the calculation load for a massive number of machines is considerable for an AP. Furthermore, some authors have built an analytical model to estimate the power consumption of each sensor device and have found a cluster size to maximize the network lifetime for 802.11ah [15-17].

Unlike the previously-proposed approaches, this paper presents a DSAS algorithm to alleviate the contention that occurs when many sensor devices are present in the same slot of the RAW by considering the average backoff time of each slot and the sleeping time of each machine. The proposed DSAS defines two sets of thresholds, high and low, to determine the sleeping-time level of a machine and the contention level for each slot, respectively. One of the innovations in the DSAS is that the shorter sleeping-time level a machine has, the lower contention level of a slot is allocated to it. Besides, machines with sleeping-time levels or contention levels falling between the high and low thresholds continue to use their original transmission contention slots in order to reduce slot re-allocation overhead.

3. THE DYNAMIC SLOT ALLOCATION SCHEME

3.1 Symbols and Notations

Table 1 summarizes the symbols and notations proposed to define the DSAS for 802.11ah networks. Let N be the total number of machines associated to an AP. M is defined as the total number of slots in RAW. Let M_H and M_L be the set of the slots whose contention levels are high and low, respectively. The symbol, c_i, denotes the number of consecutive BI's that sensor device i missed during its sleeping mode and b_j stands for the average backoff time in the jth slot. The symbols, Ts_h and Ts_l, represent the high and low thresholds for deciding the sleeping-time level of each device, respectively. Similarly, Tc_h and Tc_l denote the high and low thresholds for deciding the contention level of a slot, respectively. Let $Vs\_h(j)$ be the virtual index of slot j whose average backoff time, denoted as b_j, is larger than Tc_h. $Vs\_l(j)$ is the virtual index of slot j whose average backoff time is lower than *Tc_l*. *Vd_h*(*i*) represents the virtual ID of device *i* whose sleeping-time level is larger than *Ts_h*, and *Vd_l*(*i*) represents the virtual ID of device *i* whose sleeping-time level is lower than *Ts_l*.

Table 1. The symbols and notations.

Symbol	Notation
N	the number of the machines associated to an AP
M	the total number of slots in RAW
M_H	the set of the slots whose contention levels are high
M_L	the set of the slots whose contention levels are low
c_i	the number of consecutive BIs device *i* missed in its sleeping mode
b_j	the average backoff time of slot *j*
Ts_h	the high threshold of sleeping-time level
Ts_l	the low threshold of sleeping-time level
Tc_h	the high threshold of contention level
Tc_l	the low threshold of contention level
Vs_h(*j*)	the virtual index of slot *j* whose b_j > *Tc_h*
Vs_l(*j*)	the virtual index of slot *j* whose b_j < *Tc_l*
Vd_h(*i*)	the virtual ID of device *i* whose c_i > *Ts_h*
Vd_l(*i*)	the virtual ID of device *i* whose c_i < *Ts_l*

3.2 Model Description

Two sets of thresholds, high and low, to determine the sleeping-time level of a machine and the contention level for a slot, respectively, are obtained in the following. As elaborated in the Introduction, the field of CS contained in the Beacon Header is increased by one every broadcast. Each machine *i* can record two values of CS from the received beacons just before it enters sleeping mode and after it wakes up. As defined in Section 3.1, c_i is just equal to the difference between the two recorded values of CS. Let *C* be a sorted set of c_i's in a descending order, formulated as below.

$$C = \{c_1, c_2, c_3, \ldots, c_N | c_k \geq c_{k+1}, k \in Z^+, k \leq N\} \tag{2}$$

Then, the high threshold of sleeping-time level (*Ts_h*) is obtained by averaging the sum of the first $\lfloor N/2 \rfloor$ terms in *C*. Conversely, the low threshold of sleeping-time level (*Ts_l*) is an average of the sum of the last $(N - \lfloor N/2 \rfloor)$ terms in *C*. On the other hand, the machines with transmission opportunities during slot *j* have to send their respective backoff integers piggybacked in the transmitted frames. Just after the end of slot *j*, the average backoff time of slot *j*, denoted as b_j, is calculated by averaging the backoff times of those machines that have successful transmissions during slot *j*. Let *B* be a sorted set of b_j's in a descending order, formulated as below.

$$B = \{b_1, b_2, b_3, \ldots, b_M | b_k \geq b_{k+1}, k \in Z^+, k \leq M\} \tag{3}$$

We consider the high threshold of contention level (*Tc_h*) of each slot as the average on the sum of the first $\lfloor M/2 \rfloor$ terms in *B*. Conversely, the low threshold of contention level (*Tc_l*) is obtained by averaging the sum of the last $(M - \lfloor M/2 \rfloor)$ terms in *B*.

Table 2. Values of *Lc* and *Ls*.

Symbol	Value	Condition
Lc	high	$b_j > Tc\_h$
	moderate	$Tc\_l <= b_j <= Tc\_h$
	low	$b_j < Tc\_l$
Ls	high	$c_i > Ts\_h$
	moderate	$Ts\_l <= c_i <= Ts\_h$
	low	$c_i < Ts\_l$

As summarized in Table 2, the contention level (denoted as *Lc*) of each slot during the RAW can be classified as high, moderate, or low, based on the two thresholds, *Tc_h* and *Tc_l*. Similarly, the sleeping-time level (denoted as *Ls*) of each machine can be classified as high, moderate, or low, based on the two thresholds, *Ts_h* and *Ts_l*. The DSAS can dynamically allocate slots for sensor devices during every BI depending on the sleeping-time level of each machine and the contention level of each slot at the previous BI.

Let each machine *i* whose sleeping-time level is high be given a virtual ID, denoted as *Vd_h*(*i*), other than its original AID. Each machine *i* whose sleeping-time level is low is given a virtual ID, denoted as *Vd_l*(*i*). *Vd_h*(*i*) and *Vd_l*(*i*) are expressed in terms of their individual indexes in *C* by Eqs. (4) and (5), respectively.

$$Vd\_h(i) = k - 1, \tag{4}$$

$$Vd\_l(i) = N - k, \tag{5}$$

where *k* is the index of machine *i* in *C*. Similarly, all slots during the RAW are originally indexed from zero to ($M - 1$). Further, we give each slot *j* having high or low contention level a virtual index, denoted as *Vs_h*(*j*) or *Vs_l*(*j*), respectively, which are correspondingly expressed in terms of their individual indexes in *B* by Eqs. (6) and (7).

$$Vs\_h(j) = k - 1, \tag{6}$$

$$Vs\_l(j) = M - k, \tag{7}$$

where *k* is the index of slot *j* in *B*. Just after finishing the computation of Eqs. (4)-(7), machine *i* will be assigned a new slot with virtual index *s* in the next BI depending on that the sleeping-time level of machine *i* is high or low based on Eqs. (8) or (9), respectively. However, the machines whose sleeping-time levels are moderate retain their original slots.

$$s = Vd\_h(i) \bmod m_H, \tag{8}$$

$$s = Vd\_l(i) \bmod m_L, \tag{9}$$

where $m_H(m_L)$ is the size of $M_H(M_L)$. Fig. 3 illustrates an example of slot allocation. It shows that the machine with the longest sleeping time is assigned to the most congested slot in M_H, that's $s = 0$, according to Eq. (8). Conversely, the machine with the shortest sleeping time is assigned to the least congested slot in M_L, that's, *s* is zero, according to Eq. (9).

3.3 Slot Allocation Mechanism

At the end of each BI, the two sets of thresholds, {*Tc_h*, *Tc_l*} and {*Ts_h*, *Ts_l*}, are updated by an AP. Subsequently, the contention level of each slot is classified as high, moderate, or low according to *Tc_h* and *Tc_l*. Similarly, the sleeping-time level of each machine is classified as high, moderate, or low according to *Ts_h* and *Ts_l*. After determining the sleeping-time level of each machine and the contention level of each slot, a virtual ID is given to a machine with its sleeping-time level defined as high or low by referring to Eqs. (4) or (5). Additionally, a virtual index is given to a slot belonging to M_H or M_L based on Eqs. (6) or (7). Slot allocation for the next BI is performed by an AP by referring to Eqs. (8) and (9). Principally, machines with low sleeping-time levels are allocated to high contention-level slots and in contrast, machines with high sleeping-time levels are allocated to low contention-level slots. Furthermore, to avoid unnecessary slot reallocation, machines whose sleeping-time levels or contention levels of using slots are moderate retain their original slots.

During the RAW in the next BI, each machine contends for a data transmission opportunity following DCF in the assigned slot by trying to send RTS packets during RCP. A machine can transmit its data packet during DTP if it receives a corresponding CTS packet during. However, the packet will be dropped if the number of retransmission retried exceeds the limit, denoted as *Rlimit*.

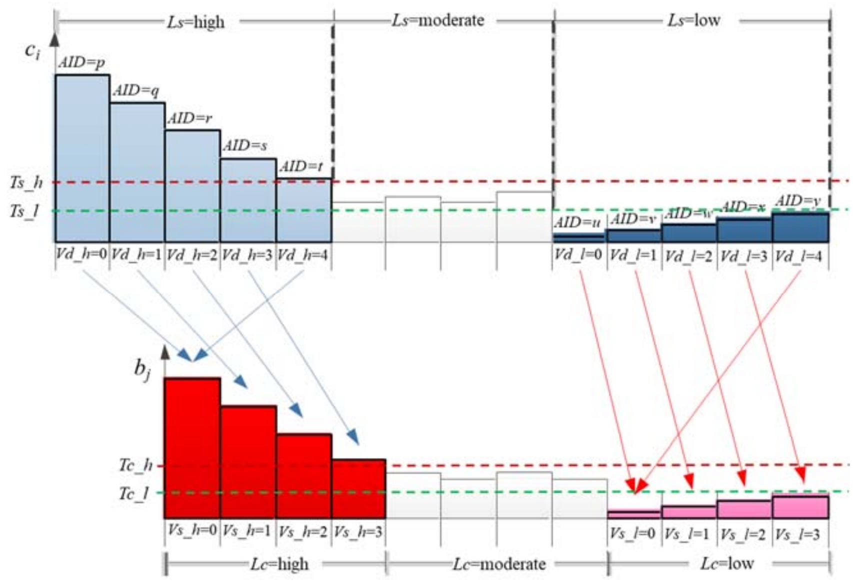


Fig. 3. An example of slot allocation with the DSAS.

4. SIMULATION RESULTS

We evaluated the performance of our proposed DSAS via simulations on ns-3 [18]. The 802.11ah network we considered is based on an AP with a number of associated machines, denoted as N. The MAC parameters and their values used in the simulation are summarized in Table 3. N varies from 500 to 3000. The length of a BI is 500 milliseconds, including a RAW of 200 milliseconds. Each slot during the RAW is 20 milliseconds, composed of an RTS/CTS period of 2 milliseconds and a data transmission period of 18 milliseconds. Thus, there are ten slots in total during RAW, dedicated for the medium access of machines every BI. A unit of backoff time implies a delay in a machine's access for 52 microseconds. We assumed that the data generation rate of each machine is 600 Kbps and the sleeping time of each machine randomly spreads from 5 to 30 BI's. Compared with 802.11ah standard, the performance metrics for MAC layer include the average backoff time, packet delay, and overall system throughput, as plotted in Figs. 4 to 8. Unless explicitly specified, all simulation results were obtained by averaging ten random samples. Each sample considered only the last 70 seconds in a running duration.

Table 3. The parameters and values.

Parameter	Value
The number of machines (N)	500 – 3000
BI duration	500 ms
RAW duration	200 ms
Slot duration	20 ms
The number of slots per BI	10
Data transmission period (DTP)	18 ms
RTS/CTS period (RCP)	2 ms
Unit of backoff timer	52 us
Data rate of a machine	600 Kbps

Fixing N = 2000, a large number of sensor devices as encountered in many M2M environments, Fig. 4 shows the average backoff time of each slot for only 30 consecutive BI's, *i.e.*, 300 slots, since the observations repeat. Fig. 4 shows that the average backoff time for 300 slots with DSAS is between 3 and 7 units, and that with 802.11ah is between 3 and 12 units. This is because the proposed DSAS considers the sleeping time of each machine and the congestion condition of each slot when performing slot assignment, the number of very idle or very busy slots is alleviated. However, contention failure may occur even after dynamic slot allocation because machines assigned to the same slot fail to receive the corresponding CTS during RCP. Moreover, many machines may successfully get transmission opportunities during RCP but a DTP following the current RCP is overloaded. As observed in Fig. 5, the number of machines contention fail during RCP in each slot with the DSAS is much smaller than that with 802.11ah. Also, Fig. 6 shows ~20% improvement in the number of overloaded machines during DTP with the DSAS. The superiority of the DSAS over 802.11ah in Figs. 5 and 6 results from contending machines are more evenly distributed in a BI with the DSAS.

Figs. 7 and 8 show the dependence of the average packet delay and the overall system throughput with an increase in the number of machines from 500 to 3000, respectively. Fig. 7 shows that the average packet delay (APD) with either DSAS or 802.11ah makes no difference when the number of machines is smaller than 1500. However, the APD with 802.11ah drastically increases when the number of machines approaches 2000, while the

APD with the DSAS increases much more slowly than 802.11ah since the DSAS can dynamically assign a less (more) congestive slot to a machine with shorter (longer) sleeping time by considering both the sleeping time of a machine and the contention level of each slot. Further, Fig. 8 shows that the overall system throughputs with both DSAS and 802.11ah are reduced when the number of machines exceeds 1000 because of traffic saturation. Nevertheless, the overall system throughput with the DSAS decreases more slowly than it would with 802.11ah.

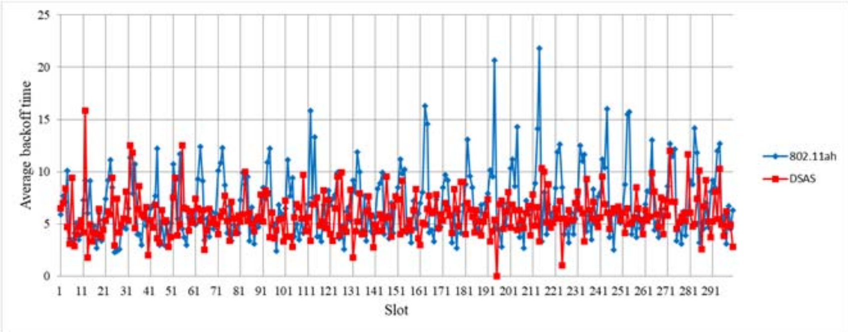


Fig. 4. The average backoff time for 30 consecutive BI's.

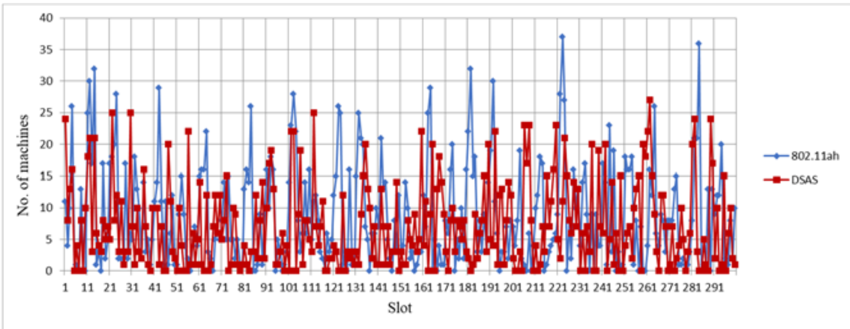


Fig. 5. The number of failed machines during RCP for 30 consecutive BI's.

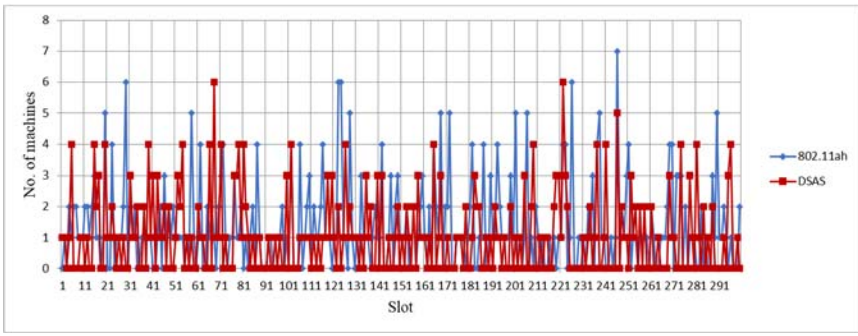


Fig. 6. Overloaded machines during DTP for 30 consecutive BI's.

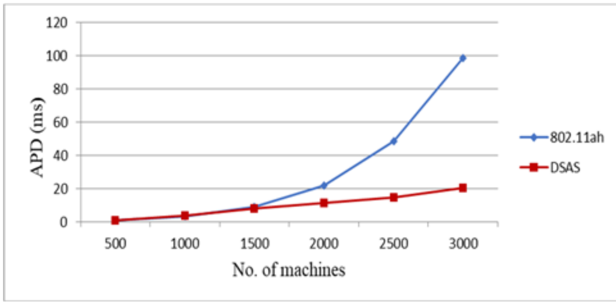


Fig. 7. Average packet delay vs. the number of machines.

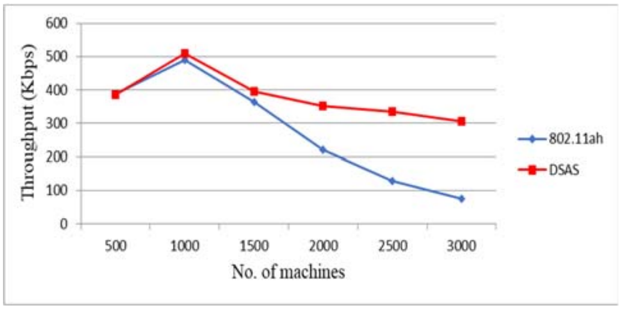


Fig. 8. Overall system throughput vs. the number of machines.

5. CONCLUSIONS

We have presented a dynamic slot allocation scheme (DSAS) for 802.11ah networks. The DSAS defines two sets of thresholds, high and low, to determine the sleeping-time level of a machine and the contention level for a slot, respectively. Slots in a BI are dynamically assigned to the contending machines by taking into account both the sleeping time of each machine and the congestion condition of each slot. The effectiveness and superiority of the proposed DSAS are demonstrated by performing simulations on ns-3. Performance measures include averages of the backoff time, packet delay, and overall system throughput. Although the proposed DSAS significantly improves these performance metrics with the MAC parameters and valued defined in Section 4, there could be smaller amount of data and longer sleeping cycle, *e.g.*, payload of 1 Kbps with sleeping time of 300 or more BI's, for a node in M2M environments. However, it is very straightforward that similar phenomenon can be observed by increasing the number of machines from 3000 to 30000 or more. We are now working on dynamically adjusting the duration of the RAW in every BI to accommodate machines that have successfully received their corresponding CTS during RCP with enough DTP duration to finish data transmission.

ACKNOWLEDGEMENTS

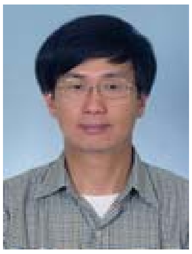
The authors would like to thank the anonymous reviewers for their constructive com ments, which helped in improving the quality of this paper. This research was supported in part by the Ministry of Science and Technology under the Grant No: MOST-107-2221-E-110-015 and the USC intramural project with the Grant No: 107-05-05002-01.

REFERENCES

1. N. Xia, H.-H. Chen, and C.-S. Yang, “Radio resource management in machine-to-machine communications-a survey,” *IEEE Communications Surveys & Tutorials*, Vol. 20, 2018, pp. 791-828.
2. F. J. Lin, Y. Ren, and E. Cerritos, “A feasibility study on developing IoT/M2M applications over ETSI M2M architecture,” in *Proceedings of International Conference on Parallel and Distributed System*, 2013, pp. 558-563.
3. F. Ghavimi and H.-H. Chen, “M2M communications in 3GPP LTE/LTE-A networks: architectures, service requirements, challenges, and applications,” *IEEE Communications Surveys & Tutorials*, Vol. 17, 2015, pp. 525-549.
4. 3GPP TR 36.888, “Study on provision of low-cost machine-type communications (MTC) user equipment (UE) based on LTE,” v.12.0.0, June 2013.
5. 802.11ah-2016, IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 2: sub 1GHz license exempt operation.
6. Wi-Fi Alliance introduces low power, long range Wi-Fi HaLow, https://wi-fi.org, 2016.
7. K. Ogawa, Y. Sangenya, M. Morikura, K. Yamamoto, and T. Sugihara, “IEEE 802.11 ah based M2M networks employing virtual grouping and power saving methods,” in *Proceedings of IEEE 78th Vehicular Technology Conference*, 2013, pp. 1-5.
8. L. Zheng, L. Cai, J. Pan, and M. Ni, “Performance analysis of grouping strategy for dense IEEE 802.11 networks,” in *Proceedings of IEEE Global Communications Conference*, 2013, pp. 219-224.
9. L. Zheng, M. Ni, L. Cai, J. Pan, C. Ghosh, and K. Doppler, “Performance analysis of group-synchronized DCF for dense IEEE 802.11 networks,” *IEEE Transactions on Wireless Communications*, Vol. 13, 2014, pp. 6180-6192.
10. C. W. Park, D. Hwang, and T. J. Lee, “Enhancement of IEEE 802.11ah MAC for M2M communications,” *IEEE Communications Letters*, Vol. 18, 2014, pp. 1151-1154.
11. R. P. Liu, G. J. Sutton, and I. B. Collings, “Power save with offset listen interval for IEEE 802.11ah smart grid communications,” in *Proceedings of IEEE International Conference on Communications*, 2013, pp. 4488-4492.
12. R. P. Liu, G. J. Sutton, and I. B. Collings, “WLAN power save with offset listen interval for machine-to-machine communications,” *IEEE Transactions on Wireless Communications*, Vol. 13, 2014, pp. 2552-2562.
13. G. C. Madueño, Č. Stefanović, and P. Popovski, “Reliable and efficient access for alarm-initiated and regular M2M traffic in IEEE 802.11ah systems,” *IEEE Internet of Things Journal*, Vol. 3, 2016, pp. 673-682.
14. X. Zhang and K. L. Yeung, “LLE: a timer extension mechanism for alarm-triggered traffic in IEEE 802.11ah WLANs,” in *Proceedings of IEEE International Conference on Communications*, 2017, pp. 130-135.

15. O. Raeesi, J. Pirskanen, A. Hazmi, T. Levanen, and M. Valkama1, "Performance evaluation of IEEE 802.11ah and its restricted access window mechanism," in *Proceedings of IEEE International Conference on Communications Workshops*, 2014, pp. 460-466.
16. A. Argyriou, "Power-efficient estimation in IEEE 802.11ah wireless sensor networks with a cooperative relay," in *Proceedings of IEEE International Conference on Communications*, 2015, pp. 6755-6760.
17. G. Miao, A. Azari, and T. Hwang, "E^2-MAC: energy efficient medium access for massive M2M communications," *IEEE Transactions on Communications*, Vol. 64, 2016, pp. 4720-4735.
18. The ns-3 simulator, https://www.nsnam.org/developers/.

Yan-Jing Wu received the Ph.D. degree in Computer Engineering from Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. She became an Assistant Professor in 2007 and an Associate Professor in 2013 at Department of Information Technology and Communication, Shih Chien University, Kaohsiung Campus, Taiwan. Her research interests are in the area of wireless communication networks, with emphasis on resource allocation and mobility management for multimedia traffic.



Tsang-Ling Sheu received the Ph.D. degree in Computer Engineering from the Department of Electrical and Computer Engineering, Penn State University, University Park, Pennsylvania, USA, in 1989. From September 1989 to July 1995, he worked with IBM Corporation at Research Triangle Park, North Carolina, USA. In August 1995, he became an Associate Professor, and was promoted to a Full Professor in January 2006 at Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. His research interests include wireless mobile networks and multimedia networking. He was the recipient of the 1990 IBM Outstanding Paper Award. Dr. Sheu is a member of the IEEE, and the IEEE Communications Society.

Pei-Hsun Chan received the MS degree from the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, in November 2016. From February 2017, he serves as a network programmer for a high-tech company located in Hsinchu Scientific Park, Taiwan. His research interests include wireless networks, mobile communication networks, M2M/D2D, and IoT.